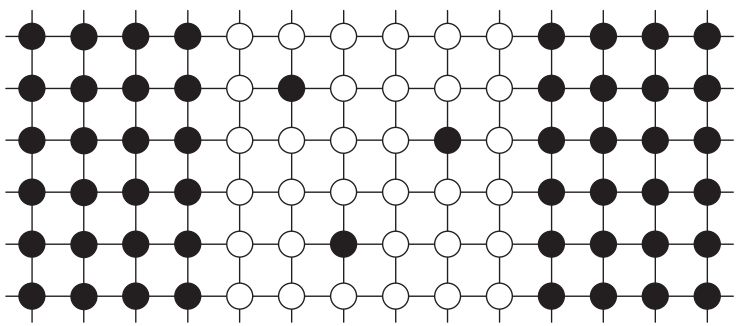
Atomistic Simulation of Quantum Transport in Nanoelectronic Devices

www.ebook3000.com

This page intentionally left blank

Atomistic Simulation of Quantum Transport in Nanoelectronic Devices



Yu Zhu and Lei Liu
NanoAcademic Technologies Inc., Canada

Foreword by
Professor Hong Guo

World Scientific
NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI • TOKYO
www.ebook3000.com

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

Library of Congress Cataloging-in-Publication Data

Names: Zhu, Yu (Physicist), author. | Liu, Lei (Physicist), author. |
Guo, Hong (Professor of Physics), writer of foreword.

Title: Atomistic simulation of quantum transport in nanoelectronic devices /
Yu Zhu (NanoAcademic Technologies Inc., Canada),
Lei Liu (NanoAcademic Technologies Inc., Canada).

Description: Singapore ; Hackensack, NJ : World Scientific Publishing Co. Pte. Ltd., [2016] |
Includes bibliographical references.

Identifiers: LCCN 2016010331| ISBN 9789813141414 (hardcover ; alk. paper) |
ISBN 9813141417 (hardcover ; alk. paper) | ISBN 9789813141421 (pbk. ; alk. paper) |
ISBN 9813141425 (pbk. ; alk. paper)

Subjects: LCSH: Nanoelectronics--Computer simulation. | Nanoelectromechanical systems. |
Transport theory--Data processing. | Electron transport. |
Quantum electronics--Mathematical models.

Classification: LCC TK7874.84 .Z48 2016 | DDC 621.381--dc23

LC record available at https://lccn.loc.gov/2016010331

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

Copyright © 2016 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

Printed in Singapore

To our families

www.ebook3000.com

This page intentionally left blank

Foreword

I am pleased to be asked by the authors to write a foreword for their monograph. *Atomistic Simulation of Quantum Transport in Nanoelectronic Devices* provides an in-depth discussion on how to carry out quantum mechanical simulations of charge transport from the bottoms-up. Advances in technology have made it possible to fabricate semiconductor transistors whose channel length is only a few nanometers. It is only natural to investigate such atomistic devices using atomistic methods. This monograph summarizes the theoretical background and mathematical formulation of quantum transport, and presents carefully the procedure that leads to a powerful atomistic simulator.

Exactly solving the quantum transport problem for realistic nanotransistors is not possible — at least in the near future, due to the many complicated microscopic details of the physics. The authors have chosen to present a "first order simulator" which contains adequate microscopic physics and at the same time, can be implemented in a software to solve realistic problems of device physics. Many examples are given. It is particularly helpful that the simulator software is published together with this monograph, a reader can therefore easily learn how each physical quantity is calculated from a single atom all the way to a device. The monograph will tremendously benefit researchers who are solving quantum transport problems in the ever expanding field of nanoelectronics. It is also the starting point for developing higher level simulators that includes more physical processes.

The authors are extremely experienced researchers in quantum transport and materials physics who worked in academic and are now in industrial settings. I had great pleasure to work with them at McGill University and in Nanoacademic Technologies Inc., and witnessed their original R&D

www.ebook3000.com

that have led to the science and software presented here. To some extent the presentation in this monograph follows their development curve as inventors of atomistic quantum transport simulators.

There are many excellent texts on the topics of quantum transport, Green's function formalism, density functional theory and device simulation. This monograph is unique in presenting a practically very powerful method and associated software readily applicable to solving contemporary research topics of nonequilibrium quantum transport from atomistic principles. Anyone who is interested in first principles modeling of nanoelectronics will benefit from reading this monograph.

Hong Guo
James McGill Professor of Physics
McGill University
Montreal, QC Canada

March 30, 2016

Preface

In 2015, Intel, Samsung and TSMC began to mass-market the 14nm node field effect transistor technology called FinFETs. In the same year, IBM, working with GlobalFoundries, Samsung, SUNY, and various equipment suppliers, announced their success in fabricating 7nm devices [1]. A 7nm silicon channel is a few tens of atoms long and these devices are truly atomic! It is clear that one cannot shrink the channel length much further from these atomic dimensions. The device engineers and industrial leaders are facing some tantalizing and extremely difficult questions: what is the new switching technology beyond the field effect transistor? What is the next "silicon"? Is it economically viable to pursue the Moore's law any further? If not, what should one do?

As theoretical physicists we are fortunate not to be forced to answer these hard questions. We are however forced to develop practical and reasonably accurate theories and modeling methods to meet the challenges of device physics at the atomic scale, namely theories and methods that may assist device engineers to address the above questions and to help industrial leaders to evaluate future technologies. This is no easy task and many scientists devoted careers to it. The difficulty comes from the fact that at the atomic scale, how a device works is dominated by nonequilibrium quantum transport phenomena that is critically influenced by the microscopic details of the device material. The problem becomes more complicated if one is to consider practical realities such as the inevitable defects, impurities, roughness, disorder, phonons, correlations, temperature, etc. Indeed, even if one can develop a perfect theoretical formalism, it is only qualitatively useful unless it can be reduced to a practical device simulator.

To first order, such a device simulator should contain all of the following ingredients: quantum mechanics, nonequilibrium statistical mechanics,

www.ebook3000.com

material property prediction, disorder effects and computational efficiency that can handle thousands of atoms. Aspects of these ingredients were developed in the past twenty years which form what we shall call "zeroth order simulators". Here, quantum mechanics is needed to deal with transport properties at the atomic scale; nonequilibrium statistical mechanics is needed for predicting nonequilibrium device operation and current flow. The combination of them gives rise to quantum transport equations that are reminiscent of the classical Boltzmann transport equation which essentially combines Newton's equation with nonequilibrium statistical mechanics. For the purpose of analyzing atomic scale devices, a capability of predicting materials properties is necessary — and the best is from atomic first principles parameter-free; such a capability should also be powerful enough to handle disorder effects. The latter is nontrivial since with disorder, the calculated properties must be averaged over disorder configurations. Finally, this "first order simulator" should be computationally very efficient such that practical device structures can be simulated.

The purpose of this monograph is to present such a "first order simulator" of atomic scale devices: its theoretical formulation, approximation, practical implementation, subtlety, application domain and many examples. The readers should be researchers who are interested in predicting quantum transport properties of emerging atomic scale devices and structures from first principles. For those readers who are mostly interested in quantum transport phenomenology, we refer to the wonderful book by S. Datta [2]; for those who want to read about device physics, a great text is the book by M. Lundstrom [3]; if one wishes to push further into topics related to both atoms and transport, the relatively recent text of S. Datta [4] is a very good start; for more theoretically oriented researchers who are interested in the nonequilibrium Green's function formalism of quantum transport, there are many excellent reviews, e.g., Ref. [5]; for advanced researchers and developers, the original paper of the NEGF-DFT theory [6] is strongly recommended. On the side of first principles simulation of materials, the excellent book of R. Martin [7] should be studied. Furthermore, handling disorder effects was done by the coherent potential approximation theory (CPA) of P. Soven [8] and D. W. Taylor [9], and expertly formulated for transport calculations by Velický [10]. The monograph of I. Turek *et al.* [11] has presented theoretical and implementation details of equilibrium quantum transport including applications of CPA.

While the vast literature has provided basic theoretical backgrounds for the development of the "first order simulator" of atomic scale device physics,

it is somewhat surprising that such a simulator was only realized recently — hence this monograph. The missing link has been the nonequilibrium statistical mechanics needed for evaluating the nonequilibrium density matrix of device materials having some disorder when there is a current flow. We shall carefully discuss technical details of the theoretical formulation of the "first order simulator" and present the software. For researchers who are mostly interested in using the simulator to solve device problems, many examples are given and this monograph can be read as a user manual. We however strongly suggest students and postdoctoral fellows go through the theoretical formulation so that they may contribute to the development of "higher order simulators" which will be definitely demanded by device physics of the near future.

Bibliography

[1] https://www-03.ibm.com/press/us/en/pressrelease/47301.wss.
[2] S. Datta, *Electronic Transport in Messoscopic Systems* (Cambridge University Press, Cambridge, UK, 1995).
[3] M. Lundstrom, *Fundamentals of carrier transport* (Cambridge University Press, Cambridge, UK, 2000).
[4] S. Datta, *Quantum Transport: Atom to Transistor* (Cambridge University Press, Cambridge, UK, 2005).
[5] A.-P. Jauho, N. S. Wingreen, and Y. Meir, Phys. Rev. B **50**, 5528 (1994).
[6] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B **63**, 245407 (2001); **63**, R121104 (2001).
[7] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods* (Cambridge University Press, Cambridge, UK, 2004).
[8] P. Soven, Phys. Rev. **156**, 809 (1967).
[9] D. W. Taylor, Phys. Rev. **156**, 1017 (1967).
[10] B. Velický, Phys. Rev. **184**, 614 (1969).
[11] I. Turek, V. Drchal, J. Kudrnovský, M. Šob, and P. Weinberger, *Electronic Structure of Disordered Alloys, Surfaces and Interfaces* (Kluwer Academic, Dordrecht, 1997).

www.ebook3000.com

This page intentionally left blank

Acknowledgments

This monograph is based on the work of many others. First of all, we are greatly indebted to our mentor and years-long friend Prof. Hong Guo of McGill University. We benefited so much from the collaboration with him and his research group members at McGill University. The NECPA-DFT theory presented in this monograph can be viewed as an extension of the NEGF-DFT theory and CPA-NVC theory invented by him and his collaborators. The affiliated NanoDsim package borrowed many numerical techniques from the research code MatDcal developed in his group. We also wish to thank Prof. Hong Guo for his direct contribution to this monograph: He read the manuscript critically, wrote the foreword, and gave us advice on further improvement. Of course, he is not responsible for any errors or deficiencies that remain in the monograph.

We wish to thank the following collaborators who contributed in essential ways to the theoretical formulation and numerical implementation of the NanoDsim package: Prof. Youqi Ke (Shanghai-Tech University) for inventing the original CPA-NVC theory and its LMTO based implementation; Prof. Ke Xia (Beijing Normal University) for introducing and teaching us the LMTO method of DFT; Prof. Yibin Hu (Institute of Technical Physics, Chinese Academy of Sciences) for solving many computational and algorithm problems in XC-functionals, parallelization, and compilation; Prof. Jian Wang (The University of Hong Kong) for his many helps in the quantum transport theory which has been critical for us; Dr. Jeremy Taylor (now at ViaScience) for inventing the original NEGF-DFT theory and the first implementation in the world; Dr. Waldron Derek (now at McKinsey & Company) for his introducing and teaching us MATLAB, object oriented programming, and parallelization.

We wish to thank the following researchers who carried out atomistic

www.ebook3000.com

simulation using NanoDsim or research code and contributed to the quantum transport physics presented in this monograph: Prof. Ferdows Zahid (Independent University, Bangladesh), Mathieu César (McGill University), Zi Wang (McGill University), Dr. Dongping Liu (now at Think Focus Group), Prof. Jesse Maassen (Dalhousie University), Dr. Lining Zhang (Hong Kong University of Science and Technology), Dr. Yin Wang (The University of Hong Kong), Prof. Haitao Yin (Harbin Normal University), Dr. Ronggen Cao (Fudan University), Qing Shi (McGill University), Zhen Zhu (Fudan University), Prof. Jun Zhuang (Fudan University), Saeed Bohloul (McGill University), Prof. Lei Zhang (Shanxi University and University of Hong Kong), Dr. Jianing Zhuang (The Hong Kong Polytechnic University), Prof. Yonghong Zhao (Sichuan Normal University), Prof. Wengang Lu (Institute of Physics, Chinese Academy of Sciences), and many others.

Yu Zhu wish to thank Prof. Tsung-han Lin (Peking University) for guiding him to the field of quantum transport; Prof. Qing-feng Sun (Peking University) for enormous help in the Green's function technique; Prof. Wei Li (Institute of Theoretical PhysicsChinese Academy of Science) and Prof. Yi-Feng Yang (Institute of Physics, Chinese Academy of Science) for stimulating discussions on quantum physics; Prof. Zhong-Shui Ma (Peking University) and Prof. Guang-Shan Tian (Peking University) for introducing him to a numerical simulation group; Prof. Jian Wang (The University of Hong Kong) and Prof. Fu-Chun Zhang (Zhejiang University) for invaluable help in the early stage of his academic career; and many others.

There are many people who directly or indirectly have contributed to Lei Liu's personal understanding of the subject. He particularly wishes to thank Prof. Jia-Ming Li (Tsinghua University and Shanghai Jiao Tong University) for being an extraordinary supervisor; Prof. Li introduced him to the field of first-principle simulations of physical properties of materials, with critical training in numerical techniques and quantum theories of multi-scattering self-consistent-field, multi-channel quantum defect theory, and density functional theory. His gratitude also goes to Prof. Yufen Li and Prof. Fuming Li (Fudan University) for invaluable help in the early stage of his academic career. He wishes to thank Prof. S. Y. Wu and Prof. C. S. Jayanthi (University of Louisville) for bringing an opportunity to enter the field of quantum transport simulation with tight-binding method, fruitful cooperation, and personal help during his time at University of Louisville. He also thanks Prof. Jun Zhuang (Fudan University) for being a twenty-year-long collaborator and friend. He wishes to express his

sincere thanks to Prof. Hongjie Dai (Stanford University) and Prof. G. Y. Guo (National Taiwan University) for their encouragement and fruitful collaboration. Prof. Xin-Gao Gong (Fudan University), Prof. Jinlong Yang (University of Science and Technology of China), Prof. Wenqing Zhang (Shanghai University), and many others are also deserving of thanks for stimulating discussions and personal help and friendship.

Last but not least, we are grateful to our parents and family members for bearing with us.

www.ebook3000.com

This page intentionally left blank

Contents

www.ebook3000.com

www.ebook3000.com

www.ebook3000.com

Chapter 1

Introduction

Approaching the end of Moore's law, we are facing a revolution from traditional microelectronics to a new field called nanoelectronics where all the characteristic lengths are of the order of nanometers. According to the *International Technology Roadmap for Semiconductors* [1], the device size is expected to shrink continuously from 22 nm (2012) to 14 nm (2014), 10 nm (2016), 7 nm (2018), and 5 nm (2020). The continuous shrinking of the device size results in discontinuous changes in the device physics: At the nanometer scale, electrons behave more like waves than particles, and the well-established semi-classical transport theory needs to be updated to a quantum version; At the nanometer scale, material is no longer continuous, and atomic details may have great impact on the transport properties; At the nanometer scale, the impurities and defects generate considerable randomness, such that disorder effects on the device performance and reliability deserve careful analysis. This chapter aims to provide a physics background and to elaborate some features of nanoelectronic devices.

1.1 What is quantum transport?

Before discussing quantum transport, let us first examine the Ohm's law of classical transport. Ohm's law says that the current through a resistor is proportional to the applied voltage. At the first glance, the statement is quite natural: Without any voltage, the current must be zero; Applying a small voltage, the current should respond linearly to the driving force. However, there is a loophole in the argument. By definition, the current is proportional to the velocity of electrons, and the voltage is proportional to the electric field or the force exerted on the electrons. According to the Newton's law, it is the acceleration instead of the velocity that is proportional to the force. Something goes wrong?

www.ebook3000.com

The key to solving the puzzle lies in the fact that Ohmic resistor has a large number of scatterers. In the transport, electrons collide with the scatterers frequently. Imagine an electron has zero velocity at the beginning and is accelerated by the electric field. After flying for a period of time, the electron collides with a scatterer and loses the velocity. After that the electron is accelerated again and collides with another scatterer, and so on and so forth. Therefore the electron moves at an average speed [2]

$$\overline{v} = \frac{1}{2} a\tau, \tag{1.1}$$

where a is the acceleration and τ is the average time between two collisions. Eq. (1.1) indicates that the average speed is proportional to the acceleration due to random scattering and hence the puzzle is solved.

The random scattering can be classified into two categories: elastic scattering and inelastic scattering. Elastic scattering means that the electron collides with a "hard" scatterer such that the electron changes its direction but maintains its speed. Inelastic scattering means that the electron collides with a "soft" scatterer such that the electron changes both its direction and speed. In elastic scattering, the energy of the electron is conserved while the momentum is not conserved. In inelastic scattering, both the energy and the momentum of the electron are not conserved. On average an electron may encounter an elastic scatterer after traveling a distance l_m or an inelastic scatter after a distance l_ϕ. The lengths l_m and l_ϕ are referred to as the *mean free path* and the *coherence length* respectively. In some cases, l_ϕ can be much larger than l_m. For example, in Si:P δ-doped devices, l_ϕ is about 80 nm while l_m is about 10 nm at 4.2 K [3]. A good discussion of these length scales can be found in Ref. [4].

The coherence length l_ϕ is a border between the classical world and the quantum world. Notice that the energy E appears in the phase factor $e^{-\mathrm{i}Et}$ of a wave function. If the system size is much larger than l_ϕ, an electron may run into many inelastic collisions which adds a random phase shift to the phase factor. As a result, the phase coherence of the wave function is destroyed and the electron will behave like a particle. If the system size is comparable to or smaller than l_ϕ, an electron is allowed to conserve its energy and stay in a quantum state described by the wave function $\psi(t) = \psi(0)\, e^{-\mathrm{i}Et}$. Hence the electron will propagate according to $\psi(t)$ and behave like a wave. In reality, the border between the classical world and the quantum world is not so distinct. The transition from wave-like behavior to particle-like behavior is called dephasing. The two

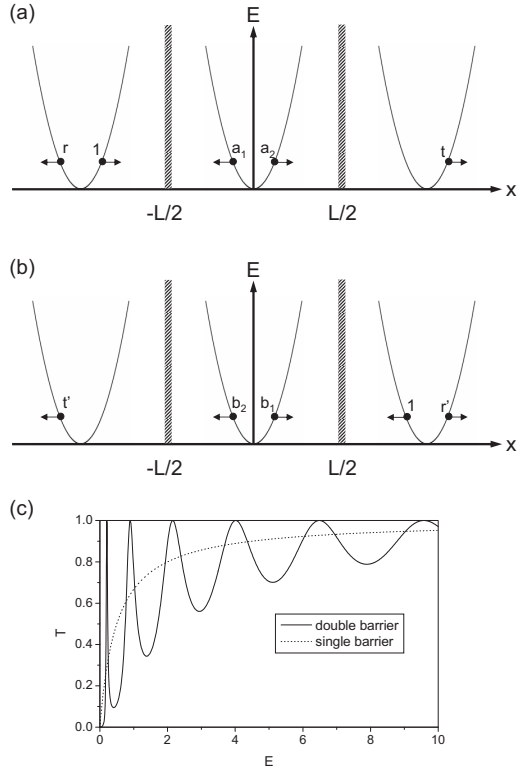


Fig. 1.1 The scattering states and the transmission coefficient of the double δ-barrier model. (a) The incoming wave from the left region is scattered by the double δ-barrier into the reflected wave r and the transmitted wave t. (b) The incoming wave from the right region is scattered by the double δ-barrier into the reflected wave r' and the transmitted wave t'. (c) Transmission coefficient as a function of energy for the double δ-barrier ($\gamma_1 = \gamma_2 = \gamma$) and the single δ-barrier ($\gamma_1 = \gamma$, $\gamma_2 = 0$). Other parameters are $L = 4$ and $\gamma = 1$.

main dephasing mechanisms in semiconductor devices are electron-phonon scattering and disorder scattering. We shall investigate disorder scattering in details in the following chapters.

In nanoelectronic devices, due to small system size, electrons are more like waves than particles. The lack of inelastic scattering has two important consequences: quantum coherence and nonequilibrium statistics. These two features distinguish quantum transport from classical transport, and the difference will be elaborated by an analytically solvable model below.

www.ebook3000.com

Let us consider a free electron gas in a one-dimensional (1d) double δ-barrier potential

$$V(x) = \gamma_1 \delta\left(x + \frac{L}{2}\right) + \gamma_2 \delta\left(x - \frac{L}{2}\right), \tag{1.2}$$

where γ_1 and γ_2 are the barrier heights, and L is the distance between the two δ-barriers. The scattering wave function $\psi(x)$ satisfies the Schrödinger equation (atomic units are used throughout this monograph unless otherwise stated)

$$\left[-\frac{1}{2}\partial_x^2 + V(x)\right]\psi(x) = E\psi(x).$$

Since $V(x)$ is nonzero only at $x = \pm\frac{L}{2}$, $\psi(x)$ can be constructed in terms of plane waves

$$\psi(x) = \begin{cases} e^{\mathrm{i}kx} + re^{-\mathrm{i}kx} & x \in \left(-\infty, -\frac{L}{2}\right) \\ a_1 e^{\mathrm{i}kx} + a_2 e^{-\mathrm{i}kx} & x \in \left(-\frac{L}{2}, +\frac{L}{2}\right) \\ te^{\mathrm{i}kx} & x \in \left(+\frac{L}{2}, +\infty\right) \end{cases}, \tag{1.3}$$

where $k = \sqrt{2E} > 0$. In Eq. (1.3), $e^{\mathrm{i}kx}$ is the incoming wave from the left region, $re^{-\mathrm{i}kx}$ is the reflected wave to the left region, $te^{\mathrm{i}kx}$ is the transmitted wave to the right region, and $a_1 e^{\mathrm{i}kx}$ and $a_2 e^{-\mathrm{i}kx}$ are the scattering waves in the central region (see Fig. 1.1a). One can verify that Eq. (1.3) satisfies Eq. (1.2) in the three regions respectively. The unknown coefficients r, t, a_1, a_2 are determined by connecting the piecewise solution at $x = \pm\frac{L}{2}$,

$$\begin{aligned} \psi\left(-\frac{L}{2} + 0^+\right) - \psi\left(-\frac{L}{2} - 0^+\right) &= 0, \\ \psi\left(+\frac{L}{2} + 0^+\right) - \psi\left(+\frac{L}{2} - 0^+\right) &= 0, \\ \psi'\left(-\frac{L}{2} + 0^+\right) - \psi'\left(-\frac{L}{2} - 0^+\right) &= 2\gamma_1 \psi\left(-\frac{L}{2}\right), \\ \psi'\left(+\frac{L}{2} + 0^+\right) - \psi'\left(+\frac{L}{2} - 0^+\right) &= 2\gamma_2 \psi\left(+\frac{L}{2}\right). \end{aligned}$$

After some algebra [5], the solution is obtained as

$$\begin{aligned} t &= \frac{1}{d}k^2, \\ r &= \frac{1}{d}\left(-e^{-\mathrm{i}kL}\right)\left[\mathrm{i}k(\gamma_1 + \gamma_2) + \gamma_1\gamma_2\left(e^{\mathrm{i}k2L} - 1\right) + \mathrm{i}k\gamma_2\left(e^{\mathrm{i}k2L} - 1\right)\right], \\ a_1 &= \frac{1}{d}\left(k^2 + \mathrm{i}k\gamma_2\right), \\ a_2 &= \frac{1}{d}\left(-e^{\mathrm{i}kL}\right)\mathrm{i}k\gamma_2, \end{aligned}$$

where $d \equiv k^2 + \mathrm{i}k(\gamma_1 + \gamma_2) + \gamma_1\gamma_2\left(e^{\mathrm{i}k2L} - 1\right)$. One can verify that $|t|^2 + |r|^2 = 1$ as required by probability conservation.

An equally good scattering wave function for electrons incoming from the right region can be constructed similarly (see Fig. 1.1b)

$$\tilde{\psi}(x) = \begin{cases} t'e^{-\mathrm{i}kx} & x \in \left(-\infty, -\frac{L}{2}\right) \\ b_1 e^{-\mathrm{i}kx} + b_2 e^{\mathrm{i}kx} & x \in \left(-\frac{L}{2}, +\frac{L}{2}\right) \\ e^{-\mathrm{i}kx} + r'e^{\mathrm{i}kx} & x \in \left(+\frac{L}{2}, +\infty\right) \end{cases}. \tag{1.4}$$

By the geometric symmetry, it is easy to see that $t' = (t)_{\gamma_1 \longleftrightarrow \gamma_2}$, $r' = (r)_{\gamma_1 \longleftrightarrow \gamma_2}$, $b_1 = (a_1)_{\gamma_1 \longleftrightarrow \gamma_2}$, $b_2 = (a_2)_{\gamma_1 \longleftrightarrow \gamma_2}$. One can verify that $|t|^2 = |t'|^2$ and $|r|^2 = |r'|^2$ as required by the time-reversal symmetry.

The transmission coefficient is defined by $T \equiv |t|^2$ which is to measure the probability of an incoming wave being scattered into transmitted wave. Fig. 1.1c shows T as a function of E for the double δ-barrier ($\gamma_1 = \gamma_2 = \gamma$) and the single δ-barrier ($\gamma_1 = \gamma$, $\gamma_2 = 0$). One can see that $T(E)$ of the double δ-barrier exhibits oscillations while $T(E)$ of the single δ-barrier increases monotonically. It is interesting to observe that the double δ-barrier can be even more transparent than the single δ-barrier in certain energy range. This is a typical quantum interference effect which is beyond the classical picture. The two barriers can be viewed as the two mirrors of a Fabry-Pérot interferometer, and the transmission maximum 1 is reached at the energies where resonances are formed by the reflected waves between the two barriers.

The scattering wave function is only part of the story. To calculate physical quantities, we also need to know how these scattering states are populated. Suppose $L \ll l_\phi$ and inelastic scattering only occurs in the left and right regions far from the double δ-barrier. The left and right regions are infinitely large and act as electron reservoirs, they are assumed to have reached local equilibrium with chemical potentials μ_L and μ_R, respectively. As a result, a scattering wave ψ incoming from the left region has a statistical weight $f_L(E) = f(E - \mu_L)$ where $f(E)$ is the Fermi function. The scattering wave ψ generates a current $T(E) V_g(E) D(E) f_L(E)$ from the left to the right, and leaves a charge density $\left|a_1(E) e^{\mathrm{i}kx} + a_2(E) e^{-\mathrm{i}kx}\right|^2 D(E) f_L(E)$ in the central region. Here $V_g(E) = \frac{dE}{dk}$ is the group velocity and $D(E) = \frac{1}{2\pi}\frac{dk}{dE}$ is the density of the states with positive group velocity. Similarly, a scattering wave $\tilde{\psi}$ incoming from the right region has a statistical weight $f_R(E) = f(E - \mu_R)$, generates a current $T(E) V_g(E) D(E) f_R(E)$ from the right to the left,

www.ebook3000.com

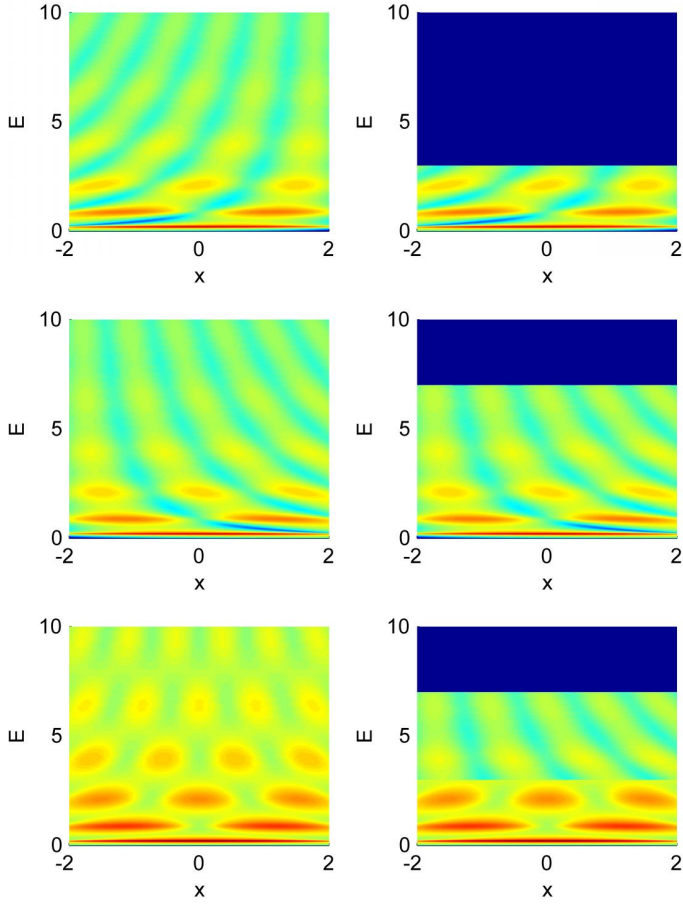


Fig. 1.2 The color maps of local density of states (left column) and nonequilibrium occupation (right column) at zero temperature in logarithmic scale. The left column plots $D_L(x,E)$, $D_R(x,E)$, and $D(x,E) \equiv D_L(x,E) + D_R(x,E)$ from top to bottom. The right column plots $N_L(x,E) \equiv D_L(x,E) f_L(E)$, $N_R(x,E) \equiv D_R(x,E) f_R(E)$, and $N(x,E) \equiv N_L(x,E) + N_R(x,E)$ from top to bottom. The local chemical potentials are $\mu_L = 3$ and $\mu_R = 7$. Other parameters are $L = 4$ and $\gamma_1 = \gamma_2 = 1$.

and leaves a charge density $\left|b_1(E) e^{-\mathrm{i}kx} + b_2(E) e^{\mathrm{i}kx}\right|^2 D(E) f_R(E)$ in the central region. Therefore the net current is

$$I = \int \frac{dE}{2\pi} T(E) \left[f_L(E) - f_R(E)\right], \tag{1.5}$$

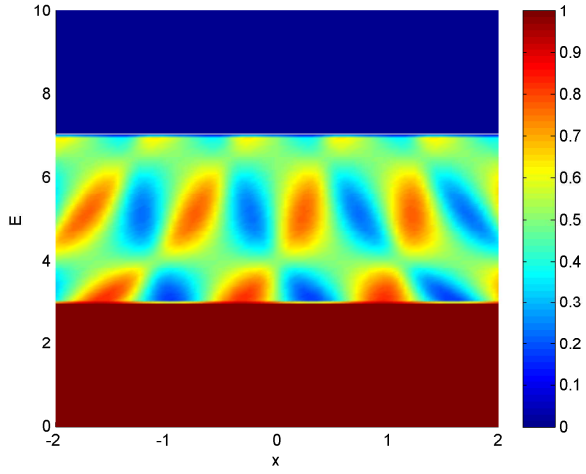


Fig. 1.3 The color map of the local statistical function $F(x,E)$ at zero temperature. The local chemical potentials are $\mu_L = 3$ and $\mu_R = 7$. Other parameters are $L = 4$ and $\gamma_1 = \gamma_2 = 1$.

and the charge density in the central region is

$$\begin{aligned}\rho(x) &= \int dE \left[\left| a_1(E) e^{ikx} + a_2(E) e^{-ikx} \right|^2 D(E) f_L(E) + \right. \\ &\qquad \left. \left| b_1(E) e^{-ikx} + b_2(E) e^{ikx} \right|^2 D(E) f_R(E) \right] \\ &= \int dE \left[D_L(x,E) f_L(E) + D_R(x,E) f_R(E) \right], \end{aligned} \tag{1.6}$$

where $D_L(x,E) \equiv \left| a_1(E) e^{ikx} + a_2(E) e^{-ikx} \right|^2 D(E)$ and $D_R(x,E) \equiv \left| b_1(E) e^{-ikx} + b_2(E) e^{ikx} \right|^2 D(E)$. The nonequilibrium charge density has two contributions, one from the left incoming waves and the other from the right incoming waves which are proportional to $f_L(E)$ and $f_R(E)$ respectively. Fig. (1.2) shows the local density of states $D_L(x,E)$, $D_R(x,E)$, and $D(x,E) \equiv D_L(x,E) + D_R(x,E)$, as well as the occupations of these states. One can observe the resonances with one node, two nodes, three nodes, $\cdots$, by increasing energy. These resonances correspond to the transmission peaks in Fig. 1.1c. The resonant states are populated by step-like Fermi functions and hence the occupations are discontinuous at $E = \mu_L$ and $E = \mu_R$.

The nonequilibrium distribution in Eq. (1.6) is a complicated combination of $f_L(E)$ and $f_R(E)$, which is qualitatively different from a local

www.ebook3000.com

equilibrium one. If we attempt to construct the nonequilibrium distribution by local density of states and local statistical function, the latter will be highly nontrivial. As an illustration, we rewrite Eq. (1.6) to

$$\rho(x) = \int dE D(x,E) F(x,E), \tag{1.7}$$

where $D(x,E)$ is the local density of states defined by

$$D(x,E) \equiv D_L(x,E) + D_R(x,E),$$

and $F(x,E)$ is the local statistical function defined by

$$F(x,E) \equiv \frac{f_L(E) D_L(x,E) + f_L(E) D_R(x,E)}{D_L(x,E) + D_R(x,E)}.$$

We are interested in how the local statistical function looks like. In a special case, $x = 0$ and $\gamma_1 = \gamma_2$, $F(x,E)$ can be obtained analytically

$$F(0,E) = \frac{1}{2} f_L(E) + \frac{1}{2} f_R(E),$$

which is a two-step function. In general, $F(x,E)$ strongly depends on x, and the color map of $F(x,E)$ is shown in Fig. 1.3. One can see that $F(x,E) = 1$ for $E < \min(\mu_L, \mu_R)$ and $F(x,E) = 0$ for $E > \max(\mu_L, \mu_R)$. In the intermediate regime, $F(x,E)$ has a very complicated pattern, indicating the nontrivial characteristics of the nonequilibrium statistics. It is simply impossible to approximate $F(x,E)$ by a local Fermi function $f(E - \mu(x))$.

Despite its simplicity, the 1d model captures some essential features, namely quantum transport can be described by a nonequilibrium occupation of the scattering states. A more realistic example is provided in Appendix A.3, where the potential and the current of a p-n junction are calculated by the classical drift-diffusion model and the quantum effective-mass model. The two models further demonstrate the conceptual difference between classical transport and quantum transport. Finally this section is summarized by the "formula"

quantum transport = quantum coherence + nonequilibrium statistics.

1.2 Every atom counts

In nanoelectronic devices, due to small system size, the atomic details are no longer details. The microscopic structure has enormous influence on the material properties as well as the transport properties. Fig. 1.4 is a

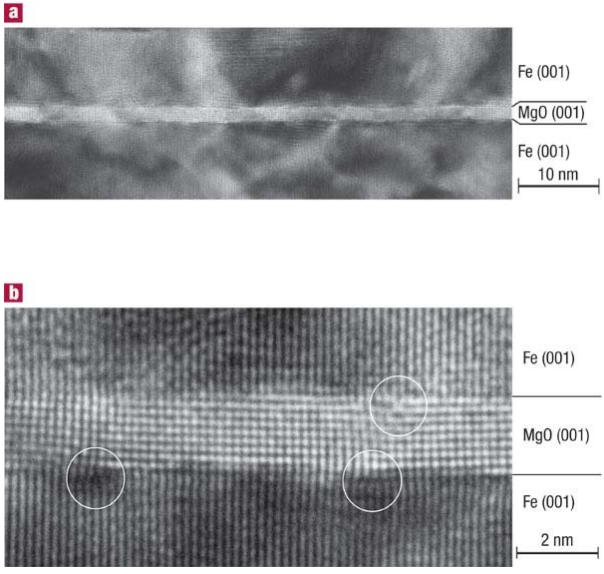


Fig. 1.4 (Reproduced from Ref. [6] with permission) TEM images of a single-crystal MTJ with the Fe(001)/MgO(001)(1.8 nm)/Fe(001) structure. (b) is a magnification of (a). The vertical and horizontal directions respectively correspond to the MgO[001] (Fe[001]) axis and MgO[100] (Fe[110]) axis. Lattice dislocations are circled. The lattice spacing of MgO is 0.221 nm along the [001] axis and 0.208 nm along the [100] axis. The lattice of the top Fe electrode is slightly expanded along the [110] axis.

TEM image of a Fe/MgO/Fe magnetic tunnel junction where three lattice dislocations are indicated by the circles [6]. Since the junction is composed of only a few atomic layers, such atomic defects have considerable impact on quantum transport. Simulations indicate that a 1% change in the bond length result in remarkable changes in the magnetoresistance (see Fig. 2d of Ref. [7]).

A similar problem appears in silicon devices. The number of dopants decreases drastically as device size shrinks: The number is about 10^5 in 10 μm technology nodes and 10^1 in 10 nm technology nodes [9]. Fig. 1.5 shows that a small number of dopants makes the electrostatic potential very rough in nanoelectronic devices. One can expect that the position of each dopant has non-negligible impact on the transport current. Fig. 1.6 shows the transmission coefficient of a nanowire simulated with a tight-binding $spds^*$ model [10]. One can see that adding a *single* dopant results in noticeable changes in the transmission coefficient. It is also interesting to observe

www.ebook3000.com

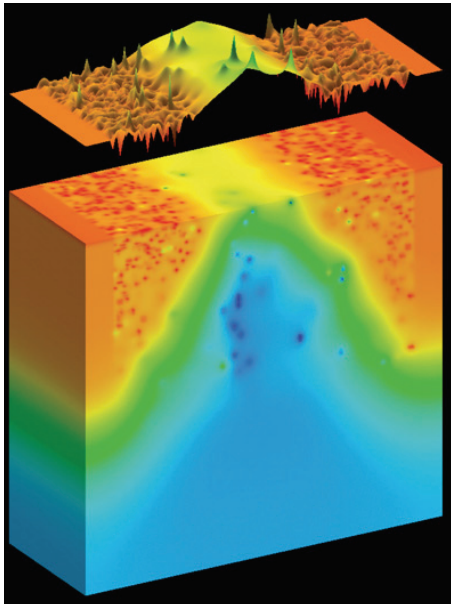


Fig. 1.5 (Reproduced from Ref. [8] with permission from the image owner) In nanoelectronic devices, the electrostatic potential has a spiky surface due to discrete dopant atoms.

that different dopant atoms have their own "fingerprints": An aluminium atom has major impact on the transmission of the valence band while a phosphorus atom has major impact on the transmission of the conduction band. Every atom counts in the transport of nanoelectronic devices!

So nano-materials can no longer be viewed as continuous medium. It is necessary to think in terms of discrete atoms. As a starting point, let us first investigate how to solve a single atom. In a hydrogen atom, there is only one electron moving around the nucleus. The analytical solution of the hydrogen atom can be found in any quantum mechanics textbook. In other species of atoms, there are many electrons and the electron-electron interaction makes the problem much harder to solve. Even in classical mechanics, no general analytical solution exists for the three-body problem, not to mention the many-body problem in the quantum mechanics. Surprisingly, with increasing electron number, the problem becomes less difficult. When the electron number is sufficiently large, the electron-electron interaction can be treated approximately in a mean-field manner. For any one particular electron, the interaction from other electrons can be taken into account

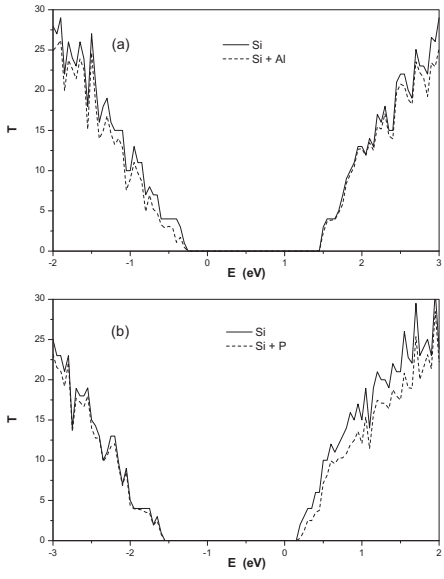


Fig. 1.6 Transmission coefficient of a Si nanowire with or without a single dopant atom in the center. The cross section of the nanowire is 2.2 nm × 2.2 nm. The dopant atom is Al in (a) and P in (b).

by the Hartree potential of their charge densities. The physics beyond this mean-field picture can be taken into account by adding a correction term called exchange-correlation potential within the framework of density functional theory (DFT) [11–15]. Therefore the unsolvable many-body problem is reduced to many single-body problems with a special form of potential in the DFT [16].

Thus the Schrödinger equation of a single atom can be written as

$$\left[-\frac{1}{2}\nabla^2 + V(r)\right]\psi(\mathbf{r}) = E\psi(\mathbf{r}), \tag{1.8}$$

where the potential $V(r)$ in DFT has three terms

$$\begin{aligned} V(r) &= V_Z(r) + V_H(r) + V_{XC}(r), \\ V_Z(r) &= \frac{-Z}{r}, \\ V_H(r) &= \int_0^r \frac{4\pi r'^2 \rho(r')\,dr'}{r} + \int_r^\infty \frac{4\pi r'^2 \rho(r')\,dr'}{r'}, \\ V_{XC}(r) &= V_{XC}[\rho(r)]. \end{aligned} \tag{1.9}$$

www.ebook3000.com

Here $V_Z(r)$ is the nuclear potential and Z is the atomic number; $V_H(r)$ is the Hartree potential and $\rho(r)$ is the spherically symmetric charge density; $V_{XC}(r)$ is the exchange-correlation potential whose expression can be found in Appendix A.9. In Eq. (1.8), $\psi(\mathbf{r})$ is a solution in a spherically symmetric potential $V(r)$. Due to the spherical symmetry, $\psi(\mathbf{r})$ can be decomposed into the radial part and the angular part $\psi(\mathbf{r}) = R_{nl}(r) Y_{lm}(\theta, \phi)$, in which $Y_{lm}(\theta, \phi)$ are the spherical harmonics and $R_{nl}(r)$ satisfies the radial equation

$$\left[-\frac{1}{2}\partial_r^2 + \frac{l(l+1)}{2r^2} + V(r)\right] \chi_{nl}(r) = E\chi_{nl}(r), \tag{1.10}$$

where $\chi_{nl}(r) \equiv rR_{nl}(r)$. Notice that the charge density $\rho(r)$ in turn depends on the radial wave function by

$$\rho(r) = \sum_{nl} N_{nl} R_{nl}^2(r), \tag{1.11}$$

where N_{nl} is the occupation number of the states $\psi(\mathbf{r}) = R_{nl}(r) Y_{lm}(\theta, \phi)$ with $m = -l, -l+1, \cdots, l$.

Since $V(r)$, $R(r)$, $\rho(r)$ are coupled together by Eqs. (1.9,1.10,1.11), they need to be solved self-consistently. As a demonstration, an atom solver is implemented in *ResearchCode/Chapter1/AtomSolver* for interested readers to play with [17]. For example, to solve a gold atom, issue the command *AtomSolver('Au')*, and the charge density $\rho(r)$ will converge in 18 iteration steps (see Fig. 1.7). Remember that a gold atom has 79 electrons arranged in a shell structure

$$1s^2 \quad 2s^2 2p^6 \quad 3s^2 3p^6 \quad 3d^{10} 4s^2 4p^6 \quad 4d^{10} 5s^2 5p^6 \quad 4f^{14} 5d^{10} 6s^1.$$

The first four core shells can be observed in the charge distribution $q(r) \equiv 4\pi r^2 \rho(r)$, while the fifth core shell and the valence electrons are hidden in the flat tail.

Once we know how to solve a single atom, in principle we are able to solve millions of atoms with sufficiently large computers. The idea coincides with the thought of the ancient Chinese philosopher, Laozi, who said "*Tao* derives the one, the one derives the two, the two derives the three, and the three derives the universe." In our context, *Tao* is the first principles of physics laws. The derivation from our *Tao* to a single atom has been demonstrated by the atom solver. The derivation from a single atom to nanoelectronic devices will be the main theme of the remaining chapters.

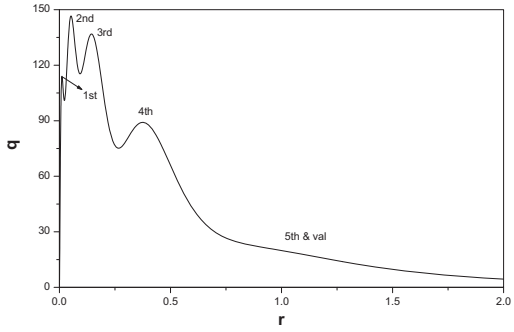


Fig. 1.7 The charge distribution $q(r)$ of a gold atom solved by the atom solver. The first four core shells are marked in the plot while the fifth core shell and the valence orbitals are hidden in the flat tail.

1.3 Disorder and coherent potential

Any realistic device has a certain degree of disorder. For instance, semiconductor transistors contain dopants, oxide layers might be amorphous, interconnect wires have surface roughness, etc. On the other hand, many interesting material properties rely on impurities and disorder, examples are the ferromagnetism in dilute magnetic semiconductors and the enhanced mobility in silicon-germanium alloy. So the knowledge about disorder effects is of crucial importance for understanding real materials and devices. Another motivation to study disorder comes from the device-to-device variability [19]. A modern chip contains billions of transistors, each of which has its own impurity configuration and transport characteristics (see Fig. 1.8). How do we characterize the behavior of billions of transistors? Obviously simulating a single device is inadequate; one needs a statistical analysis of the ensemble to predict the mean and the variation of the device parameters.

Impurities and defects are likely distributed randomly at unknown atomic sites. They may also reside near surfaces, interfaces, grain boundaries and material imperfections. Any calculated physical quantity must therefore be averaged over disorder configurations. In principle, one needs to generate many such configurations, calculate each of them and average over the ensemble (see Fig. 1.9). Such a brute-force calculation can be very difficult for several reasons. First, if the impurity concentration is very low, e.g. 0.1%, one would need a thousand host atoms to accommodate a

www.ebook3000.com

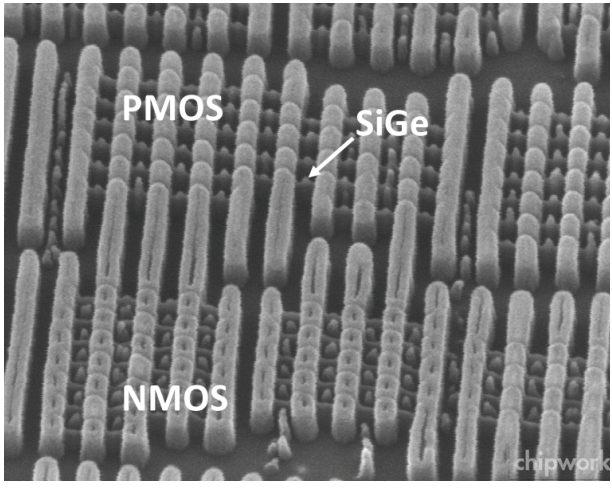


Fig. 1.8 (Reproduced from Ref. [18] with permission) Tilt SEM image of NMOS/PMOS transistors.

single impurity atom, and the system can be too large for a first principles calculation. Second, the impurity atom can assume many positions in the lattice, hence a large number of configurations must be calculated in order to average. Third, in many problems it is necessary to compute physical quantities by continuously varying the doping concentration, resulting in a prohibitively heavy computation.

An intriguing question is whether it is possible to do the disorder average analytically without using brute-force. Fortunately such techniques existed in the literature, and a particular one used in this monograph is called the coherent potential approximation (CPA). In 1967 Soven [20] and Taylor [21] independently developed the CPA technique to calculate the density of states in disordered alloys. The CPA technique is a Green's function-based theory for multiple scattering. To gain some insights into the CPA, let us consider a disordered system described by an $N \times N$ Hamiltonian matrix

$$H = T + \varepsilon, \tag{1.12}$$

where T is the off-diagonal part and ε is the diagonal part. To simulate the disorder effect, the diagonal part is composed of discrete random variables, $\varepsilon = diag\left([\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_N]\right)$, where ε_i may take the value ε_{iq} with probability x_{iq} and $q = 1, 2, \cdots$ is the index of impurity species. The probability normalization requires $\sum_q x_{iq} = 1$. The density of states can be expressed

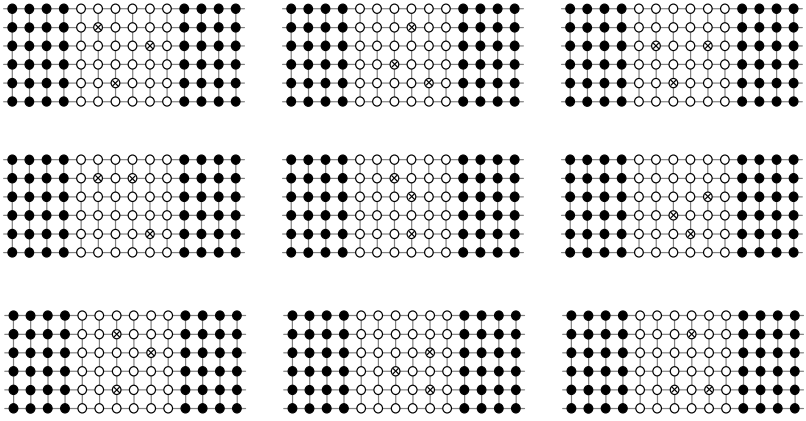


Fig. 1.9 Ensemble of devices with various disorder configurations. There are actually $C_{36}^{3} = 7140$ disorder configurations and only 9 of them are shown here.

in terms of the Green's function

$$D\left(E\right) = -\frac{1}{\pi}\text{Tr Im } G^{r}\left(E\right), \tag{1.13}$$

where $G^{r}\left(E\right)$ is defined by

$$G^{r}\left(E\right) = \left(E^{+} - H\right)^{-1}, \tag{1.14}$$

in which $E^{+} = E + \text{i}0^{+}$. Given a disorder configuration $\{\varepsilon_i\}$, one can calculate $G^{r}\left(E\right)$ and $D\left(E\right)$ with Eqs. (1.14,1.13). By averaging over disorder configurations, one obtains the disorder averaged quantities $\overline{G^{r}\left(E\right)}$ and $\overline{D\left(E\right)}$.

The CPA asserts that $\overline{G^{r}\left(E\right)}$ and $\overline{D\left(E\right)}$ can be calculated approximately by replacing the random variable ε_i with an energy-dependent effective medium $\tilde{\varepsilon}_i^{r}$. Consequently the Hamiltonian H can be rewritten as

$$H = \left(T + \tilde{\varepsilon}^{r}\right) + \left(\varepsilon - \tilde{\varepsilon}^{r}\right) \equiv H_0 + V, \tag{1.15}$$

where H_0 is a definite effective Hamiltonian and V is a random scattering potential. The effective medium $\tilde{\varepsilon}^{r}$ is chosen such that the averaged scattering vanishes and hence $\overline{G^{r}\left(E\right)}$ is approximated by

$$\overline{G^{r}\left(E\right)} = G_0^{r}\left(E\right) \equiv \left(E^{+} - H_0\right)^{-1}. \tag{1.16}$$

The scattering amplitude on site i by species q is obtained as [20]

$$\begin{aligned} t_{iq}^{r} &= V_{iq} + V_{iq}G_{0,ii}^{r}V_{iq} + V_{iq}G_{0,ii}^{r}V_{iq}G_{0,ii}^{r}V_{iq} + \cdots \\ &= V_{iq}\left(1 - G_{0,ii}^{r}V_{iq}\right)^{-1}, \end{aligned} \tag{1.17}$$

www.ebook3000.com

where $V_{iq} \equiv \varepsilon_{iq} - \tilde{\varepsilon}_i^r$ and $G_{0,ii}^r$ is the i^{th} diagonal element of G_0^r. The series in the RHS of Eq. (1.17) represents multiple scattering processes. As a result, the CPA condition amounts to

$$\overline{t_i^r} = \sum_q x_{iq} t_{iq}^r = 0, \tag{1.18}$$

which is the central result of this section.

Eq. (1.18) is applicable to general disordered systems. Specially, in disordered bulk systems, $N \to \infty$ and H_0 is periodic. One can carry out a Fourier transform to reduce the CPA calculation to a unit cell. As a illustration, we shall derive the CPA equation for a 1d disordered bulk system. The 1d bulk system is assumed to have nearest neighbor coupling 1 and random on-site energy ε_i. The random variable ε_i may take the value ε_A with probability x_A or the value ε_B with the probability x_B. The Hamiltonian of the 1d system is

$$H = \begin{pmatrix} \ddots & \ddots & & & \\ \ddots & \varepsilon_{i-1} & 1 & & \\ & 1 & \varepsilon_i & 1 & \\ & & 1 & \varepsilon_{i+1} & \ddots \\ & & & \ddots & \ddots \end{pmatrix}.$$

In the spirit of the CPA, H can be rewritten as

$$\begin{aligned} H = & \begin{pmatrix} \ddots & \ddots & & & \\ \ddots & \tilde{\varepsilon}^r(E) & 1 & & \\ & 1 & \tilde{\varepsilon}^r(E) & 1 & \\ & & 1 & \tilde{\varepsilon}^r(E) & \ddots \\ & & & \ddots & \ddots \end{pmatrix} \\ & + \begin{pmatrix} \ddots & & & & \\ & \varepsilon_{i-1} - \tilde{\varepsilon}^r(E) & & & \\ & & \varepsilon_i - \tilde{\varepsilon}^r(E) & & \\ & & & \varepsilon_{i+1} - \tilde{\varepsilon}^r(E) & \\ & & & & \ddots \end{pmatrix} \\ & \equiv H_0 + V, \end{aligned}$$

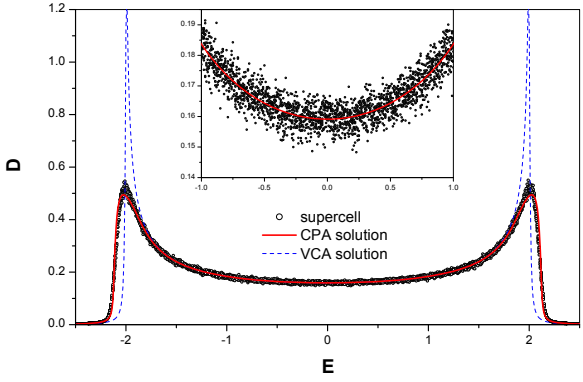


Fig. 1.10 Density of states in a disordered 1d bulk system calculated with the supercell method (black dots), the CPA method (red solid line), and the VCA method (blue dashed line). The inset is a zoom-in of the energy range $[-1, 1]$. The parameters of the 1d bulk system are: $\varepsilon_A = 0.2$, $\varepsilon_B = -0.2$, $x_A = x_B = 0.5$. The supercell has 20000 sites and is averaged over 10 disorder configurations.

where the first term is for the effective medium and the second term is for the random scattering. The CPA condition Eq. (1.18) implies that

$$x_A \left[\frac{1}{\varepsilon_A - \tilde{\varepsilon}^r(E)} - G^r_{00}(E)\right]^{-1} + x_B \left[\frac{1}{\varepsilon_B - \tilde{\varepsilon}^r(E)} - G^r_{00}(E)\right]^{-1} = 0, \tag{1.19}$$

where $G^r_{00}(E)$ is one of the diagonal elements of $G^r(E) = (E^+ - H_0)^{-1}$. Due to the periodicity of H_0, $G^r_{00}(E)$ can be calculated with the Fourier transform

$$G^r_{00}(E) = \int_0^{2\pi} \frac{dk}{2\pi} \frac{1}{E^+ - \tilde{\varepsilon}^r(E) - 2\cos k}. \tag{1.20}$$

Notice that $G^r_{00}(E)$ and $\tilde{\varepsilon}^r(E)$ are coupled in Eqs. (1.19,1.20) and need to solved self-consistently. Finally the disorder averaged density of states is obtained as

$$\overline{D(E)} = -\frac{1}{\pi} \mathrm{Im}\, G^r_{00}(E). \tag{1.21}$$

Fig. 1.10 plots the density of states calculated with three different methods: the supercell method, the CPA method, and the VCA method. Here the VCA refers to the virtual crystal approximation, meaning that each disorder site is replaced by a "virtual atom" whose on-site energy is an average of all impurity species. One can see that the CPA method is in

www.ebook3000.com

excellent agreement with the supercell method, while the VCA method deviates considerably for neglecting disorder scattering. So the CPA technique does capture the essential features of the disorder effects and is capable of calculating the disorder average without using brute-force.

The CPA technique was developed for analyzing disordered bulk systems which are always in equilibrium. The disorder average in quantum transport is more complicated because both the density matrix and the transmission coefficient involve the nonequilibrium Green's function. It is necessary to generalize CPA to the nonequilibrium situation. For device physics it is also desirable to calculate the variation of physical quantities in addition to the average. These topics will be discussed in Chapter 2 and Appendix A.5.

1.4 NECPA-DFT theory and NanoDsim package

In previous sections we introduced three theoretical components in nanoelectronic device simulation: quantum transport, atomistic modeling and disorder. Each component is supported by vast literatures. For quantum transport, two popular theoretical approaches are commonly used, the scattering matrix (SC) approach [4,23,24] and the nonequilibrium Green's function (NEGF) approach [25–32]. For atomistic modeling, there are different types of DFT [11–15] implementation with regard to the basis functions and the way to treat atomic cores. For disorder, the treatment includes using brute-force or sometimes called supercells, the CPA technique [20–22], and the VCA method [33, 34]. More recently, theoretical developments in the overlapping domain of any two of the three theoretical components were quite successful, techniques such as NEGF-DFT [7,35–53], SC-DFT [54–60], CPA-DFT [61–70], and NEGF-CPA [71–75] were made available.

The work presented in this monograph is to solve the challenge of developing a unified formalism that self-consistently includes the three theoretical components. Such a unified formalism was first reported in Ref. [76] within the vertex correction theory [77] to deal with binary systems in nonequilibrium. Later on, another unified formalism called NECPA-DFT [78] was developed based on a generalized Langreth theorem [79]. For binary systems, the NECPA-DFT formalism reduces to precisely the vertex correction theory.

The NECPA-DFT formalism (or the vertex correction theory) is in the intersection of the three theoretical components necessary for atomistic simulation of nonequilibrium quantum transport, see Fig. 1.11. Using

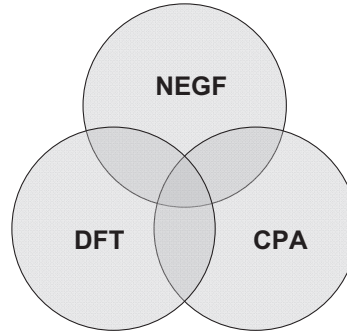


Fig. 1.11 NECPA-DFT theory is in the intersection of three groups of theoretical methods.

this unified formalism, first principles device simulation can be carried out parameter-free on reasonable computers. Quantum transport physics is naturally obtained from the very basic principles of quantum mechanics and nonequilibrium statistical mechanics. It is worth mentioning that quantum transport can also be studied with semi-empirical models such as effective-mass model, extended Hückel model and tight-binding model [80–90]. In general first principles device simulation is more expensive than that of semi-empirical models. But the former does not rely on material parameters which are usually unavailable for emerging materials and structures.

So far a wide range of theoretical studies of quantum transport has been carried out using the unified theoretical formalism, including surface scattering in cooper thin films [91, 92] to understand practical issue of interconnects; spin polarized charge conduction in magnetic tunnel junctions [93, 94] to reveal disorder effects of oxygen vacancies and cation impurities; disorder limited mobilities of two-dimensional materials such as graphene [95] and black phosphorous [96], and disorder effects in tunnel field-effect transistors [97]; discrete dopant effects in silicon nanotransistor channels [98, 99] and its associated device-to-device variability [100, 101]; band offsets of semiconductor heterojunctions [102]; resistance of copper grain boundaries [103]; composition-dependent band gaps and indirect-direct band gap transitions in group-IV semiconductor alloys [104]; transmission coefficient and density of states in iron-cobalt alloy layer embedded in copper [105]. We refer interested readers to Chapter 8 for further details of these calculations.

www.ebook3000.com

The unified theoretical formalism has been implemented in the NanoDsim package. **NanoDsim**, short for **nano**electronic **d**evice **sim**ulator, is a software for atomistic simulation of quantum transport and electronic properties in solid state nanostructures. NanoDsim implements NECPA-DFT with many other innovations in algorithms and computational methods. Some notable features of NanoDsim are: (1) It solves device Hamiltonian in nonequilibrium self-consistently in the presence of atomic disorder [78]; (2) It is capable of simulating nanoelectronic devices containing a few thousand atomic sites on a moderate computer cluster [106, 107]; (3) It has implemented a semilocal exchange-correlation potential which provides good band gap and effective-mass for common semiconductors [108, 109]; (4) It has implemented a post-analysis tool to predict device-to-device variability due to random discrete dopants [100]; (5) It has implemented an optimizer to tune the atomic sphere radiuses and vacuum sphere centers to fit a given band structure.

NanoDsim is designed to simulate two types of systems: bulk systems and two-probe systems. A bulk system is a periodic crystal structure with or without substitutional disorder sites. A two-probe system is an open structure comprised of a left lead, a central region, and a right lead [35]. The left and right lead are semi-infinite crystals with applied external voltages. The central region contains the nanostructure of interest which may have some substitutional disorder sites. To smoothly connect the nanostructure to both leads, some lead materials (called buffer layers) are also included in the central region. An example of a two-probe system is illustrated in Fig. 1.12.

Although NanoDsim is capable of simulating traditional silicon transistors, its real strength is in the development of disruptive technologies. For a long time, researchers and device engineers have been thinking about disruptive technologies beyond silicon and the current design of transistors: quantum principles, molecular devices, two-dimensional materials, quantum materials, spins, topological degrees of freedom, all the way to quantum computers. In all these, quantum transistor modeling is a dominant problem since the parameters and even knowledge gathered over the past five decades on how to model silicon transistors become more and more redundant. A change of device physics is expected for perhaps all the future disruptive technologies. NanoDsim provides a possibility to explore such disruptive technologies at a low cost. Using NanoDsim, one is allowed to "fabricate" nanoelectronic devices on a computer and simulate quantum transport from first principles.

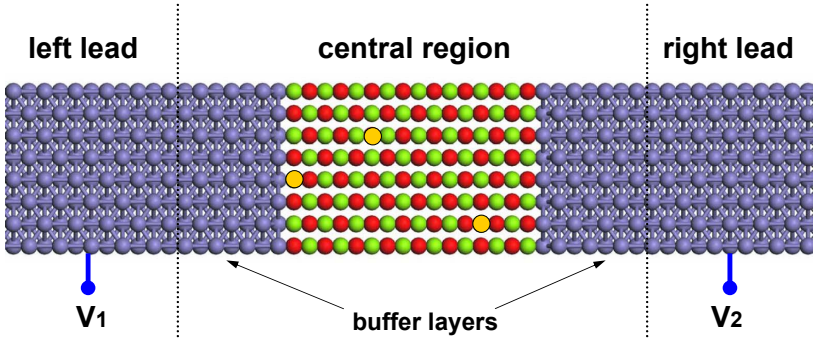


Fig. 1.12 A schematic plot of a two-probe system which is comprised of a left lead, a central region, and a right lead. Disorder sites are marked by the yellow dots in the central region.

1.5 A few words about this monograph

This monograph attempts to make a modest contribution to computational nanoelectronics by thoroughly analyzing the NECPA-DFT theory and presenting its implementation in the NanoDsim package. Computational nanoelectronics is an emerging multi-discipline research field which involves the traditional fields of condensed matter physics, computer science, applied mathematics, and electronic engineering. There have been many books in each of these fields. Reading all of them may cost a researcher several years before doing his or her own research. The purpose of this monograph is to extract a bare minimum from these books and combine with our own R&D experiences to create a shortcut for new researchers.

A unique feature of this monograph is that it contains not only text, equations, and figures but also a large volume of source code. The source code is available for all the numerical examples, which serves to illustrate the procedure from a derivation of equations to getting some publishable results. This should be helpful for those graduate students who are learning how to do research. More importantly, the complete source code of the NanoDsim package is also published with the monograph, which illustrates how to develop a professional software package from a theoretical formalism. This can be useful for advanced researchers who are doing R&D in this field. The readers are encouraged to reuse the source code in any non-commercial research activities with a proper citation.

www.ebook3000.com

As a byproduct, this monograph also illustrates the usage of MATLAB and mathematica through realistic examples. According to our research experiences, MATLAB and mathematica are extremely useful for numerical calculation and symbolic derivation respectively. The senior generation of researchers are familiar with FORTRAN, *Numerical Recipes*, and *Table of Integrals, Series, and Products*. To some extent, we would guess that MATLAB and mathematica can be substitutes of these techniques and knowledge for the younger generation.

Since we are doing atomistic device simulation, this monograph adopts atomic units in which length is measured in *Bohr* and energy is measured in *Hartree*. Those who are not familiar with atomic units are referred to Appendix A.1. In DFT calculations, it is more convenient to consider electrons to have a "positive" charge. The physics is exactly the same since this is just another convention of charge polarity. In physical quantities such as voltage and current, a factor $Q_e = -1$ is multiplied in order to be consistent with the usual convention that electrons have a negative charge.

The outline of the remaining chapters are as follows: Chapter 2 presents the NECPA theory which is the soul of the monograph; Chapter 3 presents the LMTO method and the NECPA-LMTO formalism; Chapter 4 presents the design of NanoDsim package based on the MATLAB platform; Chapter 5 discusses the numerical algorithms used in the implementation of bulk systems; Chapter 6 discusses the numerical algorithms used in the implementation of two-probe systems; Chapter 7 discusses the parallelization and optimization for large scale atomic simulations; and Chapter 8 demonstrates the functionality of the NanoDsim package by various applications in nanoelectronic device simulation.

Enjoy reading the rest of the book!

Bibliography

[1] International Technology Roadmap for Semiconductors: www.itrs2.net.
[2] Here the speed refers to the electron's drift velocity which is several orders of magnitude smaller than the eletron's thermal velocity.
[3] K. E. J. Goh and M. Y. Simmons, Appl. Phys. Lett. **95**, 142104 (2009).
[4] S. Datta, *Electronic Transport in Messoscopic Systems* (Cambridge University Press, Cambridge, UK, 1995).
[5] The algebra can be easily done with mathematica, see *ResearchCode/Chapter1/DoubleBarrier/derivation.nb*.
[6] S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, K. Ando, Nature Materials **3**, 868 (2004).

[7] D. Waldron, V. Timoshevskii, Y. Hu, K. Xia, and H. Guo, Phys. Rev. Lett. **97**, 226802 (2006).
[8] M. Miranda, *The Threat of Semiconductor Variability*, http://spectrum.ieee.org/semiconductors/design/the-threat-of-semiconductor-variability. Image: Gold Standard Simulations (www.goldstandardsimulations.com).
[9] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, S. Mudanai, IEEE Trans. Electron Devices **58**, 2197 (2011).
[10] L. Liu, Y. Zhu, H. Guo, unpublished (2014).
[11] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[12] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[13] R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules* (Oxford University Press, New York, 1989).
[14] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992).
[15] R. O. Jones and O. Gunnarsson, Rev. Mod. Phys. **61**, 689 (1989).
[16] Usually DFT is presented by constructing a universal functional of total energy. Here we describe a practical way of using it. Interested readers are referred to the literature for a systemmatic study of DFT.
[17] In the implementation of atom solver, the Schrödinger equation is replaced by the Dirac equation and relativistic correction is made to V_{XC} to take into account the relativistic effect in heavy atoms.
[18] D. James, http://chipworksrealchips.blogspot.ca/2012/04/intels-22-nm-trigate-transistors.html.
[19] *Special Issue on Characterization of Nano CMOS Variability by Simulation and Measurements*, IEEE Trans. Electron Devices **58**, 2190 (2011).
[20] P. Soven, Phys. Rev. **156**, 809 (1967).
[21] D. W. Taylor, Phys. Rev. **156**, 1017 (1967).
[22] R. J. Elliott, J. A. Krumhansl, and P. L. Leath, Rev. Mod. Phys. **46**, 465 (1974).
[23] R. Landauer, IBM J. Res. Dev. **32**, 306 (1988).
[24] M. Büttiker, IBM J. Res. Dev. **32**, 63 (1988).
[25] J. Schwinger, J. Math. Phys. **2**, 407 (1961).
[26] L. P. Kadanoff and G. Baym, *Quantum Statistical Mechanics* (Benjamin, New York, 1962).
[27] L. V. Keldysh, Zh. Eksp. Teor. Fiz. **47**, 1515 (1964) [Sov. Phys. JETP **20**, 1018 (1965)].
[28] C. Caroli, R. Combescot, P. Nozieres, and D. Saint-James, J. Phys. C: Solid St. Phys. **4**, 916 (1971).
[29] G. D. Mahan, *Many-Particle Physics* (Kluwer Academic, New York, 2000).
[30] A.-P. Jauho, N. S. Wingreen, and Y. Meir, Phys. Rev. B **50**, 5528 (1994).
[31] H. Haug and A. -P. Jauho, *Quantum Kinetics in Transport and Optics of Semiconductor* (Springer-Verlag, Berlin, 1996).
[32] G. Stefanucci and R. van Leeuwen, *Nonequilibrium Many-Body Theory of Quantum Systems* (Cambridge University Press, New York, 2013).
[33] L. Nordheim, Ann. Physik **9**, 607 (1931); **9**, 641 (1931).

www.ebook3000.com

[34] R. H. Parmenter, Phys. Rev. **97**, 587 (1955).
[35] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B **63**, 245407 (2001); **63**, R121104 (2001).
[36] M. Brandbyge, J.-L. Mozos, P. Ordejón, J. Taylor, and K. Stokbro, Phys. Rev. B **65**, 165401 (2002).
[37] Y. Xue, S. Datta, and M. A. Ratner, J. Chem. Phys. **115**, 4292 (2001); Chem. Phys. **281**, 151 (2002).
[38] S.-H. Ke, H. U. Baranger, and W. Yang, Phys. Rev. B **70**, 085410 (2004).
[39] A. Calzolari, N. Marzari, I. Souza, and M. B. Nardelli, Phys. Rev. B **69**, 035108 (2004).
[40] A. R. Rocha, V. M. García-suárez, S. W. Bailey, C. J. Lambert, J. Ferrer, and S. Sanvito, Nature Materials **4**, 335 (2005).
[41] W. Lu, V. Meunier, and J. Bernholc, Phys. Rev. Lett. **95**, 206805 (2005).
[42] K. Hirose, T. Ono, Y. Fujimoto, and S. Tsukamoto, *First-Principles Calculations in Real-Space Formalism: Electronic Configurations and Transport Properties of Nanostructures* (Imperial College Press, London, 2005).
[43] T. Ozaki, K. Nishio, and H. Kino, Phys. Rev. B **73**, 235323 (2006).
[44] D. Waldron, L. Liu, and H. Guo, Nanotechnology **18**, 424026 (2007).
[45] J. Enkovaara et al, J. Phys.: Condens. Matter **22**, 253202 (2010).
[46] J. A. Driscoll and K. Varga, Phys. Rev. B **81**, 115412 (2010).
[47] W. H. Butler, X.-G. Zhang, T. C. Schulthess, and J. M. MacLaren, Phys. Rev. B, **63**, 054416 (2001).
[48] J. J. Palacios, A. J. Pérez-Jiménez, E. Louis, E. SanFabián, and J. A. Vergés, Phys. Rev. B **66**, 035322 (2002).
[49] S. V. Faleev, F. Leonard, D. A. Stewart, and M. van Schilfgaarde, Phys. Rev. B **71**, 195422 (2005).
[50] J. E. Inglesfield, J. Phys. C: Solid State Phys. **14**, 3795 (1981).
[51] D. Wortmann, H. Ishida, and S. Blügel, Phys. Rev. B **66**, 075113 (2002).
[52] G. Stefanucci and C.-O. Almbladh, Europhys. Lett. **67**, 14 (2004).
[53] X. Zheng, F. Wang, and G. Chen quant-ph/0605104 (unpublished); quant-ph/0606169 (unpublished).
[54] N. D. Lang, Phys. Rev. B **52**, 5335 (1995).
[55] H. J. Choi, J. Ihm, Y.-G. Yoon, S. G. Louie, Phys. Rev. B **60**, 14009 (1999).
[56] J. Cerdá, M. A. Van Hove, P. Sautet, and M. Salmeron, Phys. Rev. B **56** 15885 (1997).
[57] C. C. Wan, J.-L. Mozos, G. Taraschi, J. Wang, and H. Guo, Appl. Phys. Lett. **71** 419 (1997).
[58] M. Di Ventra, S. T. Pantelides, and N. D. Lang, Phys. Rev. Lett. **84**, 979 (2000).
[59] K. Hirose, N. Kobayashi, and M. Tsukada, Phys. Rev. B **69**, 245412 (2004).
[60] K. Xia, M. Zwierzycki, M. Talanana, P. J. Kelly, G. E. W. Bauer, Phys. Rev. B **73**, 064420 (2006).
[61] G. M. Stocks, W. M. Temmerman, and B. L. Gyorffy, Phys. Rev. Lett. **41**, 339 (1978).
[62] W. H. Butler, Phys. Rev. B **31**, 3260 (1985).
[63] H. Akai, J. Phys.: Condens. Matter **1** 8045, (1989).

[64] J. Kudrnovský and V. Drchal, Phys. Rev. B **41**, 7515 (1990).
[65] K. Koepernik, B. Velický, R. Hayn, and H. Eschrig **55**, 5717 (1997).
[66] I. Turek, V. Drchal, J. Kudrnovský, M. Šob, and P. Weinberger, *Electronic Structure of Disordered Alloys, Surfaces and Interfaces* (Kluwer Academic, Dordrecht, 1997).
[67] P. R. Tulip, J. B. Staunton, S. Lowitzer, D. Ködderitzsch, and H. Ebert, Phys. Rev. B **77**, 165116 (2008).
[68] S. Lowitzer, D. Ködderitzsch, and H. Ebert, Phys. Rev. Lett. **105**, 266604 (2010).
[69] H. Ebert, S. Mankovsky, D. Ködderitzsch, and P. J. Kelly, Phys. Rev. Lett. **107**, 066603 (2011).
[70] D. D. Johnson, D. M. Nicholson, F. J. Pinski, B. L. Gyorffy, G. M. Stocks, Phys. Rev. Lett. **56**, 2088 (1986).
[71] K. Carva, I. Turek, J. Kudrnovský, O. Bengone, Phys. Rev. B **73**, 144421 (2006).
[72] A. V. Kalitsov, M. G. Chshiev, J. P. Velev, Phys. Rev. B **85**, 235111 (2012).
[73] M. Ye. Zhuravlev, A. V. Vedyayev, K. D. Belashchenko, and E. Y. Tsymbal1, Phys. Rev. B **85**, 115134 (2012).
[74] J. N. Zhuang and J. Wang, J. Appl. Phys. **114**, 063708 (2013).
[75] J. Yan and Y. Ke, arXiv:1511.09182v1 [cond-mat.mes-hall] 30 Nov 2015.
[76] Y. Ke, K. Xia, H. Guo, Phys. Rev. Lett. **100**, 166805 (2008).
[77] B. Velický, Phys. Rev. **184**, 614 (1969).
[78] Y. Zhu, L. Liu, H. Guo, Phys. Rev. B **88**, 205415 (2013).
[79] D. C. Langreth, *in Linear and Nonlinear Electron Transport in Solids*, NATO Advanced Study Institute, Series B: Physics, Vol. 17, edited by J. T. Devreese and V. E. van Doren (Plenum Press, New York, 1976).
[80] Z. Ren, R. Venugopal, S. Goasguen, S. Datta, and M. S. Lundstrom, IEEE Trans. Electron Dev. **50**, 1914 (2003).
[81] M. Luisier, A. Schenk, W. Fichtner, and G. Klimeck, Phys. Rev. B **74**, 205323 (2006).
[82] M. Lusier, G. Klimeck, *OMEN an Atomistic and Full-Band Quantum Transport Simulator for post-CMOS Nanodevices*, 8th IEEE Conference on Nanotechnology, 354-357 (2008).
[83] L. Liu, C. S. Jayanthi, M. Tang, S. Y. Wu, T. W. Tombler, C. Zhou, L. Alexseyev, J. Kong, and H. Dai, Phys. Rev. Lett. **84**, 4950 (2000).
[84] V. Mishra, S. Smith, L. Liu, F. Zahid, Y. Zhu, H. Guo, S. Salahuddin, IEEE Trans. Electron Dev. **62**, 2457 (2015).
[85] M. B. Nardelli, PRB **60** 7828 (1999).
[86] S. Sanvito, C. J. Lambert, J. H. Jefferson, and A. M. Bratkovsky, Phys. Rev. B **59**, 11936 (1999).
[87] A. Brown, A. Asenov and J. Watling, IEEE Transactions on Nanotechnology **1**, 195 (2002).
[88] T. Z. Raza, J. I. Cerdá, and H. Raza, J. Appl. Phys. **109**, 023705 (2011).
[89] A. M. Roy, D. E. Nikonov, and I. A. Young, J. Appl. Phys. **112**, 104510 (2012).
[90] Z. Stanojević, O. Baumgartner, F. Mitterbauer, H. Demel, C. Kernstock,

www.ebook3000.com

M. Karner, V. Eyert, A. France-Lanord, P. Saxe, C. Freeman, and E. Wimmer, *Physical Modeling C a New Paradigm in Device Simulation*, 2015 IEEE International Electron Device Meeting (IEDM), Dec. 15-17, 2015, Washington DC, USA.

[91] Y. Ke, F. Zahid, V. Timoshevskii, K. Xia, D. Gall, H. Guo, Phys. Rev. B **79**, 155406 (2009).

[92] F. Zahid, Y. Ke, D. Gall, H. Guo, Phys. Rev. B **81**, 045406 (2010).

[93] Y. Ke, K. Xia, H. Guo, Phys. Rev. Lett. **105**, 236801 (2010).

[94] D. Liu, X. Han, and H. Guo, Phys. Rev. B **85**, 245436 (2012).

[95] Z. Wang, Y. Ke, D. Liu, H. Guo, K. H. Bevan, Appl. Phys. Lett. **101**, 093102 (2012).

[96] S. Bohloul, L. Zhang, K. Gong, and H. Guo, Appl. Phys. Lett. **108**, 033508 (2016).

[97] Q. Shi, L. Zhang, Y. Zhu, L. Liu, M. Chan, H. Guo, *Atomic Disorder Scattering in Emerging Transistors by Parameter-Free First Principle Modeling*, 2014 IEEE International Electron Device Meeting (IEDM), Dec. 15-17, 2014, San Francisco, USA.

[98] J. Maassen, H. Guo, Phys. Rev. Lett. **109**, 266803 (2012).

[99] L. Zhang, F. Zahid, Y. Zhu, L. Liu, J. Wang, H. Guo, P. C. H. Chan, M. Chan, IEEE Transactions on Electron Devices, **60**, 3527(2013).

[100] Y. Zhu, L. Liu, and H. Guo, Phys. Rev. B **88**, 085420 (2013).

[101] Q. Shi, H. Guo, Y. Zhu, L. Liu, Phys. Rev. Appl. **3**, 064008 (2015).

[102] Y. Wang, F. Zahid, Y. Zhu, L. Liu, J. Wang, and H. Guo, Appl. Phys. Lett. **102**, 132109 (2013).

[103] M. César, D. Liu, D. Gall, and H. Guo, Phys. Rev. Appl. **2**, 044007 (2014).

[104] Z. Zhu, J. Xiao, H. Sun, Y. Hu, R. Cao, Y. Wang, L. Zhao, and J. Zhuang, Phys. Chem. Chem. Phys. **17**, 21605 (2015).

[105] C. Franz, M. Czerner, and C. Heiliger, J. Phys.: Condens. Matter **25**, 425301 (2013).

[106] Y. Zhu, L. Liu, and H. Guo, technical report of the industrial research assistance program *NAQEDA: a software tool for nanoelectronics modeling and design*, NRC-IRAP Project #700796 (2012).

[107] J. Maassen, M. Harb, V. Michaud-Rioux, Y. Zhu, and H. Guo, Proc. IEEE **101**, 518 (2013).

[108] F. Tran and P. Blaha, Phys. Rev. Lett. **102**, 226401 (2009).

[109] Y. Wang, H. Yin, R. Cao, F. Zahid, Y. Zhu, L. Liu, J. Wang, H. Guo, Phys. Rev. B **87**, 235203 (2013).

Chapter 2

The NECPA theory

This chapter aims to present the theory of quantum transport in disordered open systems. The theory is built upon the nonequilibrium coherent potential approximation (NECPA) which is a generalization of the coherent potential approximation (CPA) developed in disordered bulk systems. The generalization is carried out by applying the Langreth theorem to a contour ordered CPA equation in the formalism of nonequilibrium Green's function (NEGF). Section 2.2,2.3,2.4 introduce some background knowledge of the NEGF; Section 2.1,2.5,2.6,2.7 derive the NEGF formalism for two-probe systems; Section 2.8 presents the NECPA theory which is the heart of this chapter; Section 2.9 discusses the dephasing effect due to disorder scattering; and Section 2.10 illustrates the application of the NECPA theory with a toy model.

2.1 Two-probe Hamiltonian

An electronic device is neither like a molecule which has finite number of atoms nor like a crystal which has periodicity in three dimensions. Instead an electronic device is an open system which can be viewed as a black box connected to several leads extending to infinity. Each of the leads has its own local chemical potential and hence the whole system is in nonequilibrium. Below we shall investigate electronic devices with two leads (two-probe systems) which are the most common in realistic applications.

The model of a two-probe system is shown schematically in Fig. 2.1: A central scattering region is connected to the left and right semi-infinite leads. The leads extend to reservoirs at $z = \pm\infty$ where bias voltage is applied and electric current measured.

In the second quantization representation, the two-probe system can be

www.ebook3000.com

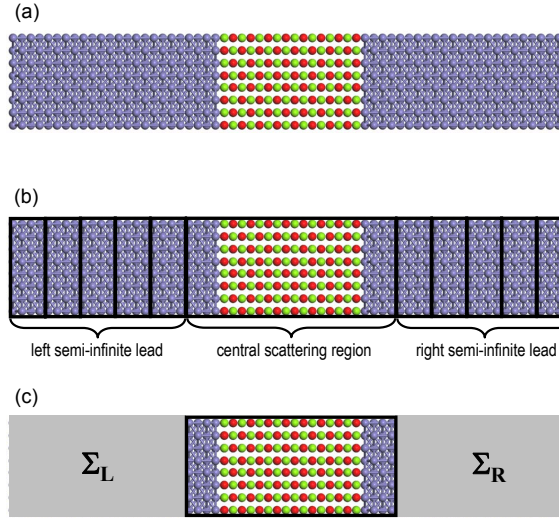


Fig. 2.1 (a) A two-probe system is composed of a left semi-infinite lead, a central scattering region, and a right semi-infinite lead. (b) The left and right leads can be partitioned into unit cells extending to $z = \pm\infty$. Some lead materials (buffer layers) are included in the central region to smoothly connect the central region to the leads. (c) The influence of semi-infinite lead can be taken into account by using the lead self-energy.

described by the following Hamiltonian

$$\hat{H} = \hat{H}_C + \hat{H}_L + \hat{H}_R + \hat{H}_{CL} + \hat{H}_{CR}, \tag{2.1}$$

where

$$\hat{H}_C = \sum_i \varepsilon_i c_i^\dagger c_i + \sum_{i \neq i'} t_{ii'} c_i^\dagger c_{i'}, \tag{2.2}$$

is the Hamiltonian of the central region;

$$\hat{H}_L = \sum_p \varepsilon_p c_p^\dagger c_p + \sum_{p \neq p'} t_{pp'} c_p^\dagger c_{p'}, \tag{2.3}$$

$$\hat{H}_R = \sum_q \varepsilon_q c_q^\dagger c_q + \sum_{q \neq q'} t_{qq'} c_q^\dagger c_{q'}, \tag{2.4}$$

are the Hamiltonians of the left and right leads; and

$$\hat{H}_{CL} = \sum_{ip} t_{ip} c_i^\dagger c_p + t_{pi} c_p^\dagger c_i, \tag{2.5}$$

$$\hat{H}_{CR} = \sum_{iq} t_{iq} c_i^\dagger c_q + t_{qi} c_q^\dagger c_i, \tag{2.6}$$

are the Hamiltonians of the interactions between the central region and the left and right leads. Notice that all the terms in the Hamiltonian are of quadratic form which is consistent with the Kohn–Sham Hamiltonian to be discussed in Chapter 3.

In the Hamiltonians, $c_i^\dagger$ and c_i are the creation and annihilation operators of state-i in the central region and satisfy the Fermion anti-commutator

$$\left\{c_i, c_{i'}^\dagger\right\} = \delta_{ii'};$$

$c_p^\dagger$ and c_p are the creation and annihilation operators of state-p in the left lead and satisfy the Fermion anti-commutator

$$\left\{c_p, c_{p'}^\dagger\right\} = \delta_{pp'};$$

$c_q^\dagger$ and c_q are the creation and annihilation operators of state-q in the right lead and satisfy the Fermion anti-commutator

$$\left\{c_q, c_{q'}^\dagger\right\} = \delta_{qq'}.$$

Notice that the Fermion operators of different regions satisfy the anti-commutator

$$\left\{c_i, c_p^\dagger\right\} = 0,$$
$$\left\{c_i, c_q^\dagger\right\} = 0.$$

The coefficients ε_i, ε_p, ε_q are the on-site energies of state-i, state-p, and state-q which are real numbers. The coefficients $t_{ii'}$, $t_{pp'}$, $t_{qq'}$, t_{ip}, t_{pi}, t_{iq}, t_{qi} are the off-diagonal hopping energies which satisfy $t_{ab} = t_{ba}^*$ due to the Hermitian property of Hamiltonian.

The Hamiltonian Eq. (2.1) uniquely determines all the quantum mechanical properties of the two-probe system. However quantum transport is determined not only by quantum mechanics but also by nonequilibrium statistics. Since the left and right leads extend to infinity, they are in local equilibrium and serve as reservoirs. Assume that the local chemical potentials of the left and right leads are μ_L and μ_R respectively. Without bias voltage, μ_L and μ_R line up with each other, and the whole two-probe system is in equilibrium. By applying a bias voltage $V = V_L - V_R$, μ_L and μ_R are shifted such that

$$\mu_L - \mu_R = Q_e V, \tag{2.7}$$

where $Q_e = -1$ due to the fact that electron has a negative charge. As a result, electric current will flow in the two-probe system and charge will redistribute in the central region. Since the left and right leads are infinitely large, the current flow has little effect on them. The central region, in contrast, is driven to a nonequilibrium steady-state, which is to be solved with the NECPA theory presented in the following sections.

www.ebook3000.com

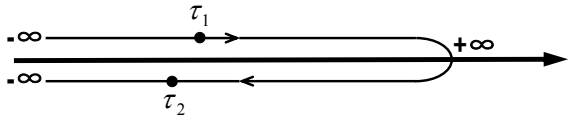


Fig. 2.2 Schematic plot of the complex-time contour that goes above the real time axis from $t = -\infty$ to $t = +\infty$ and returns below the real time axis to $t = -\infty$.

2.2 NEGF formalism

Nonequilibrium Green's function (NEGF) incorporates both quantum mechanics and nonequilibrium statistics and hence is the proper language to describe the quantum transport in nonequilibrium systems. Compared to the scattering states approach (see Section 1.1), the NEGF formalism has the following advantages: (1) Various physical effects (e.g., electron-phonon coupling, electron-electron interaction, impurity scattering) can be easily taken into account by including proper self-energies; (2) The computational cost of energy integral can be reduced significantly by using the analytical properties of the NEGF. Below we shall briefly review some basic knowledge of NEGF and the discussion will be limited to a bare minimum. Interested readers are referred to Ref. [1] for a systematic discussion of the NEGF formalism.

First of all, nonequilibrium statistics is very different from the equilibrium statistics: In equilibrium, by turning on ($t = -\infty$) and turning off ($t = +\infty$) a perturbation adiabatically, the system can return to its unperturbed ground state (up to a phase factor) after the time evolution from $t = -\infty$ to $t = +\infty$ as stated in the Gell-Mann and Low theorem [2]. In nonequilibrium, however, the theorem does not hold due to the loss of time-reversal symmetry. To see this point, let's do a thought experiment in a two-probe system. At $t = -\infty$, assume that the left lead, the central region, and the right lead do not interact with each other and they are in local equilibrium with the chemical potentials μ_L, μ_C, μ_R, respectively. Now let us turn on the interactions between the leads and the central region adiabatically. At $t = 0$, the two-probe system reaches its nonequilibrium steady-state where the distribution of the central region is uniquely determined by μ_L and μ_R. Notice that the information of μ_C has been lost completely at this moment. After that we turn off the interactions between the leads and the central region adiabatically. At $t = +\infty$, the central region can no longer return to its initial state since μ_C is unknown.

Therefore it is necessary to introduce a time contour which evolves from $t=-\infty$ to $t=+\infty$ in the upper branch and returns back to $t=-\infty$ in the lower branch, see Fig. 2.2. Thus the initial state and the final state are identical after the time evolution and the Gell-Mann and Low theorem is recovered on the time contour. The complex-time Green's function can be defined on the time contour as

$$G\left(\tau_1,\tau_2\right)\equiv-\mathrm{i}\langle T_c\hat{a}\left(\tau_1\right)\hat{b}\left(\tau_2\right)\rangle, \tag{2.8}$$

where $\hat{a}$ and $\hat{b}$ are Fermion creation or annihilation operators in the Heisenberg picture, and τ_1 and τ_2 are the complex time on the time contour. T_c is the contour order operator: If τ_1 is earlier than τ_2 on the time contour, $\hat{a}$ and $\hat{b}$ are interchanged with the addition of a minus sign (Fermion). The average $\langle\cdots\rangle$ is carried out for all quantum states with proper statistical weights. Sometimes $G\left(\tau_1,\tau_2\right)$ is also denoted as $\langle\langle\hat{a}\left(\tau_1\right)|\hat{b}\left(\tau_2\right)\rangle\rangle$ to indicate the operator $\hat{a}$ and $\hat{b}$ explicitly.

It should be emphasized that the complex-time Green's function has identical mathematical structure as that of retarded Green's function in equilibrium. One can formally develop the theory of nonequilibrium statistics in parallel to that of equilibrium except that the retarded Green's function is replaced by the complex-time Green's function. As an example, we shall show in Section 2.8 how to generalize CPA of equilibrium bulk systems to NECPA of nonequilibrium open systems.

The complex-time Green's function provides an elegant theoretical formalism. However it is the real-time Green's functions that are directly related to physical quantities. The six real-time Green's functions are defined by

$$G^{<}\left(t_1,t_2\right)\equiv+\mathrm{i}\langle\hat{b}\left(t_2\right)\hat{a}\left(t_1\right)\rangle=G\left(t_1^{+},t_2^{-}\right), \tag{2.9}$$

$$G^{>}\left(t_1,t_2\right)\equiv-\mathrm{i}\langle\hat{a}\left(t_1\right)\hat{b}\left(t_2\right)\rangle=G\left(t_1^{-},t_2^{+}\right), \tag{2.10}$$

$$G^{c}\left(t_1,t_2\right)\equiv-\mathrm{i}\langle T\hat{a}\left(t_1\right)\hat{b}\left(t_2\right)\rangle=G\left(t_1^{+},t_2^{+}\right), \tag{2.11}$$

$$G^{\overline{c}}\left(t_1,t_2\right)\equiv+\mathrm{i}\langle\overline{T}\hat{a}\left(t_1\right)\hat{b}\left(t_2\right)\rangle=G\left(t_1^{-},t_2^{-}\right), \tag{2.12}$$

$$G^{r}\left(t_1,t_2\right)\equiv-\mathrm{i}\theta\left(t_1-t_2\right)\langle\left\{\hat{a}\left(t_1\right),\hat{b}\left(t_2\right)\right\}\rangle, \tag{2.13}$$

$$G^{a}\left(t_1,t_2\right)\equiv+\mathrm{i}\theta\left(t_2-t_1\right)\langle\left\{\hat{a}\left(t_1\right),\hat{b}\left(t_2\right)\right\}\rangle, \tag{2.14}$$

where t^{+} and t^{-} are the complex time on the upper and lower time branch respectively, and $\theta\left(t\right)$ is the Heaviside step function. T is the time order operator: If $t_1<t_2$, $\hat{a}$ and $\hat{b}$ are interchanged with the addition of a minus sign (Fermion). $\overline{T}$ is the anti-time order operator: If $t_1>t_2$, $\hat{a}$ and $\hat{b}$ need to exchange their positions by adding a minus sign (Fermion).

www.ebook3000.com

The six real-time Green's functions are not independent. It is straightforward to verify that

$$G^r(t_1,t_2) = +\theta(t_1-t_2)\left[G^>(t_1,t_2) - G^<(t_1,t_2)\right], \tag{2.15}$$
$$G^a(t_1,t_2) = -\theta(t_2-t_1)\left[G^>(t_1,t_2) - G^<(t_1,t_2)\right], \tag{2.16}$$
$$G^c(t_1,t_2) = \theta(t_1-t_2)\,G^>(t_1,t_2) + \theta(t_2-t_1)\,G^<(t_1,t_2), \tag{2.17}$$
$$G^{\overline{c}}(t_1,t_2) = \theta(t_2-t_1)\,G^>(t_1,t_2) + \theta(t_1-t_2)\,G^<(t_1,t_2), \tag{2.18}$$

which leads to the following relations

$$G^c(t_1,t_2) = G^<(t_1,t_2) + G^r(t_1,t_2), \tag{2.19}$$
$$G^c(t_1,t_2) = G^>(t_1,t_2) + G^a(t_1,t_2), \tag{2.20}$$
$$G^{\overline{c}}(t_1,t_2) = G^<(t_1,t_2) - G^a(t_1,t_2), \tag{2.21}$$
$$G^{\overline{c}}(t_1,t_2) = G^>(t_1,t_2) - G^r(t_1,t_2), \tag{2.22}$$

$$G^r(t_1,t_2) - G^a(t_1,t_2) = G^>(t_1,t_2) - G^<(t_1,t_2), \tag{2.23}$$

$$G^c(t_1,t_2) + G^{\overline{c}}(t_1,t_2) = G^>(t_1,t_2) + G^<(t_1,t_2). \tag{2.24}$$

Among the six real-time Green's functions, the lesser Green's function ($G^<$), the larger Green's function ($G^>$), the retarded Green's function (G^r), and the advanced Green's function (G^a) are the most useful ones in the theory of quantum transport. Their physical meanings are as follows: $G^<$ corresponds to the density matrix of electrons; $G^>$ corresponds to the density matrix of holes; G^r is the outgoing wave propagator; and G^a is the incoming wave propagator. As an application of the NEGF, we shall derive the current formula in terms of these real-time Green's functions in Section 2.6.

2.3 Langreth theorem

So far we have defined two types of Green's functions: complex-time Green's function G_τ and real-time Green's functions G_t. The nonequilibrium statistics can be built upon G_τ formally, while physical quantities can be expressed in terms of G_t. We need a bridge to connect G_τ and G_t, which is achieved by the following Langreth theorem [3].

Langreth theorem: Assume that the complex time Green's functions A, B, and C satisfy

$$C(\tau_1,\tau_2) = \int_c d\tau A(\tau_1,\tau)\,B(\tau,\tau_2),$$

where $\int_c$ is the time contour of Fig. 2.2. The corresponding real-time Green's functions can be obtained as

$$C^{<}(t_1,t_2)=\int_{-\infty}^{+\infty}dt\left[A^{r}(t_1,t)B^{<}(t,t_2)+A^{<}(t_1,t)B^{a}(t,t_2)\right], \quad (2.25)$$

$$C^{>}(t_1,t_2)=\int_{-\infty}^{+\infty}dt\left[A^{r}(t_1,t)B^{>}(t,t_2)+A^{>}(t_1,t)B^{a}(t,t_2)\right], \quad (2.26)$$

$$C^{r}(t_1,t_2)=\int_{-\infty}^{+\infty}dtA^{r}(t_1,t)B^{r}(t,t_2), \quad (2.27)$$

$$C^{a}(t_1,t_2)=\int_{-\infty}^{+\infty}dtA^{a}(t_1,t)B^{a}(t,t_2). \quad (2.28)$$

Proof: (1) The lesser Green's function $C^{<}$ can be related to the complex-time Green's function C by $C^{<}(t_1,t_2)=C\left(t_1^{+},t_2^{-}\right)$, in which t_1^{+} is on upper time branch and t_2^{-} on the lower time branch. The time contour can be split into upper and lower branches c^{+} and c^{-}

$$\begin{aligned}
C^{<}(t_1,t_2)&=C\left(t_1^{+},t_2^{-}\right)\\
&=\int_c d\tau A\left(t_1^{+},\tau\right)B\left(\tau,t_2^{-}\right)\\
&=\int_{c^{+}}d\tau A\left(t_1^{+},\tau\right)B\left(\tau,t_2^{-}\right)+\int_{c^{-}}d\tau A\left(t_1^{+},\tau\right)B\left(\tau,t_2^{-}\right)\\
&=\int_{-\infty}^{+\infty}dt^{+}A\left(t_1^{+},t^{+}\right)B\left(t^{+},t_2^{-}\right)+\int_{+\infty}^{-\infty}dt^{-}A\left(t_1^{+},t^{-}\right)B\left(t^{-},t_2^{-}\right)\\
&=\int_{-\infty}^{+\infty}dt\left[A^{c}(t_1,t)B^{<}(t,t_2)-A^{<}(t_1,t)B^{\overline{c}}(t,t_2)\right].
\end{aligned}$$

The theorem Eq. (2.25) can be obtained by eliminating A^{c} and $B^{\overline{c}}$ in the above expression using Eq. (2.19) and Eq. (2.21). Analogously the theorem Eq. (2.26) can be proved along the same route.

(2) C^{r} can be expressed in terms of $C^{>}$ and $C^{<}$ using Eq. (2.15)

$$C^{r}(t_1,t_2)=\theta(t_1-t_2)\left[C^{>}(t_1,t_2)-C^{<}(t_1,t_2)\right].$$

Applying the proved theorem Eq. (2.25) and Eq. (2.26) to $C^{>}$ and $C^{<}$, one obtains

$$\begin{aligned}
C^{r}(t_1,t_2)&=\theta(t_1-t_2)\int_{-\infty}^{+\infty}dt\left\{\left[A^{r}(t_1,t)B^{>}(t,t_2)+A^{>}(t_1,t)B^{a}(t,t_2)\right]\right.\\
&\quad\left.-\left[A^{r}(t_1,t)B^{<}(t,t_2)+A^{<}(t_1,t)B^{a}(t,t_2)\right]\right\}\\
&=\theta(t_1-t_2)\int_{-\infty}^{+\infty}dt\left\{A^{r}(t_1,t)\left[B^{>}(t,t_2)-B^{<}(t,t_2)\right]\right.\\
&\quad\left.+\left[A^{>}(t_1,t)-A^{<}(t_1,t)\right]B^{a}(t,t_2)\right\}.
\end{aligned}$$

www.ebook3000.com

Eliminating A^r and B^a using Eq. (2.15) and Eq. (2.16), one obtains

$$\begin{aligned} C^r(t_1,t_2) &= \int_{-\infty}^{+\infty} dt\theta(t_1-t_2)[\theta(t_1-t)-\theta(t_2-t)]\left[A^>(t_1,t)-A^<(t_1,t)\right] \\ &\quad\times\left[B^>(t,t_2)-B^<(t,t_2)\right] \\ &= \int_{-\infty}^{+\infty} dt\theta(t_1-t)\,\theta(t-t_2)\left[A^>(t_1,t)-A^<(t_1,t)\right] \\ &\quad\times\left[B^>(t,t_2)-B^<(t,t_2)\right]. \end{aligned}$$

The theorem Eq. (2.27) can be obtained by eliminating $A^> - A^<$ and $B^> - B^<$ in the above expression using Eq. (2.15). Analogously the theorem Eq. (2.28) can be proved along the same route.

QED.

For simplicity, the Langreth theorem can be expressed in a compact notation

$$(AB)^< = A^rB^< + A^<B^a, \tag{2.29}$$
$$(AB)^> = A^rB^> + A^>B^a, \tag{2.30}$$
$$(AB)^r = A^rB^r, \tag{2.31}$$
$$(AB)^a = A^aB^a. \tag{2.32}$$

where the multiplication between A and B is understood as an integral over the inner time variable [4]. In the steady-state, the multiplication is simply the matrix multiplication of the Green's functions in the energy domain.

The Langreth theorem of multiplication can be further generalized to the inverse operation

$$\left(A^{-1}\right)^< = -\left(A^r\right)^{-1}A^<\left(A^a\right)^{-1}, \tag{2.33}$$
$$\left(A^{-1}\right)^> = -\left(A^r\right)^{-1}A^>\left(A^a\right)^{-1}, \tag{2.34}$$
$$\left(A^{-1}\right)^r = \left(A^r\right)^{-1}, \tag{2.35}$$
$$\left(A^{-1}\right)^a = \left(A^a\right)^{-1}, \tag{2.36}$$

which can be easily proved by applying Eqs. (2.29,2.30,2.31,2.32) to $A^{-1}A = 1$.

To sum up, Eqs. (2.29,2.30,2.31,2.32) are the original Langreth theorem, and Eqs. (2.33,2.34,2.35,2.36) are the extended Langreth theorem. Hereafter Eqs. (2.29–2.36) together are referred to as the generalized Langreth theorem and will be used in the derivation of the Dyson equation in Section 2.5 and the NECPA equations in Section 2.8.

2.4 NEGF in steady-state

In Section 2.2 and 2.3, we have briefly discussed some general properties and a useful theorem of NEGF. In this section, we shall focus on the NEGF of quadratic Hamiltonian in the steady states. On the one hand, the general formalism will be greatly simplified in this special situation. On the other hand, the simplified formalism is sufficient for the purpose of this monograph. We note that the goal of this monograph is to investigate the steady-state quantum transport in nanostructures described by the Kohn–Sham Hamiltonian of DFT (quadratic).

In Section 2.2, we have defined real-time Green's functions, which are functions of two time variables t_1 and t_2. In the steady-state, the Hamiltonian is time independent, and the real-time Green's functions only depend on the time difference $t \equiv t_1 - t_2$. Namely $G^\lambda(t) = G^\lambda(t_1 - t_2)$ where $\lambda = r, a, <, >$. One can carry out a Fourier transform to convert the Green's functions from the time domain to the energy domain

$$G^\lambda(E) = \int dt e^{\mathrm{i}Et} G^\lambda(t), \tag{2.37}$$

or from the energy domain back to the time domain

$$G^\lambda(t) = \int \frac{dE}{2\pi} e^{-\mathrm{i}Et} G^\lambda(E). \tag{2.38}$$

Below we shall work mainly in the energy domain and derive explicit expressions of NEGF in terms of eigenstates of the Hamiltonian.

Firstly, we rewrite the Hamiltonian operator $\hat{H}$ into matrix form

$$\hat{H} = \sum_{ij} H_{ij} c_i^\dagger c_j = \begin{pmatrix} c_1^\dagger & c_2^\dagger & \cdots \end{pmatrix} \begin{pmatrix} H_{11} & H_{12} & \cdots \\ H_{21} & H_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \end{pmatrix} \equiv C^\dagger H C, \tag{2.39}$$

which maps the quadratic operator $\hat{H}$ to a Hermitian matrix H. In a two-probe system, H is an infinitely large matrix

$$H_\infty = \begin{pmatrix} H_{LL} & H_{LC} & 0 \\ H_{CL} & H_{CC} & H_{CR} \\ 0 & H_{RC} & H_{RR} \end{pmatrix}, \tag{2.40}$$

where H_{CC} is the Hamiltonian matrix of the central region (finite), H_{LL} and H_{RR} are the Hamiltonian matrices of the left and right leads (semi-infinite), and H_{CL}, H_{LC}, H_{CR}, H_{RC} are the couplings between the central region and the leads. Notice that $H_{LR} = H_{RL} = 0$ since the left and right

www.ebook3000.com

leads do not interact with each other provided that the central region is sufficiently long.

Because H is Hermitian, it can be diagonalized with eigenstates

$$H = U\Lambda U^{\dagger}, \tag{2.41}$$

where

$$U = \left[\left|\phi_1\right\rangle, \left|\phi_2\right\rangle, \cdots\right], \tag{2.42}$$

$$\Lambda = diag\left[\lambda_1, \lambda_2, \cdots\right], \tag{2.43}$$

in which $H\left|\phi_k\right\rangle = \lambda_k \left|\phi_k\right\rangle$ and the eigenvectors $\{\left|\phi_k\right\rangle\}$ have been orthonormalized such that $UU^{\dagger} = 1$. Consequently $\hat{H}$ can be diagonalized as

$$\hat{H} = \sum_k \lambda_k \tilde{c}_k^{\dagger} \tilde{c}_k, \tag{2.44}$$

where $\{\tilde{c}_k\}$ is related to $\{c_i\}$ by a unitary transform

$$\begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \end{pmatrix} = U^{\dagger} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \end{pmatrix}. \tag{2.45}$$

Secondly, we derive the Green's functions for the Hamiltonian of a single particle $\hat{h} = \varepsilon c^{\dagger} c$. By using the equation of motion for an operator in the Heisenberg picture

$$i\partial_t c\left(t\right) = \left[c\left(t\right), \hat{h}\left(t\right)\right], \tag{2.46}$$

one obtains $c\left(t\right) = c\left(0\right) e^{-\mathrm{i}\varepsilon t}$. From the definition of the real-time Green's functions, $g^{\lambda}\left(t\right)$ $(\lambda = r, a, <, >)$ can be obtained as

$$g^r\left(t\right) = -\mathrm{i}\theta\left(t\right)\left\langle\left\{c\left(t\right), c^{\dagger}\left(0\right)\right\}\right\rangle = -\mathrm{i}\theta\left(t\right) e^{-\mathrm{i}\varepsilon t},$$

$$g^a\left(t\right) = \mathrm{i}\theta\left(-t\right)\left\langle\left\{c\left(t\right), c^{\dagger}\left(0\right)\right\}\right\rangle = \mathrm{i}\theta\left(-t\right) e^{-\mathrm{i}\varepsilon t},$$

$$g^{<}\left(t\right) = \mathrm{i}\langle c^{\dagger}\left(0\right) c\left(t\right)\rangle = \mathrm{i}e^{-\mathrm{i}\varepsilon t} n,$$

$$g^{>}\left(t\right) = -\mathrm{i}\langle c\left(t\right) c^{\dagger}\left(0\right)\rangle = -\mathrm{i}e^{-\mathrm{i}\varepsilon t}\overline{n},$$

where $n \equiv \langle c^{\dagger} c\rangle$ and $\overline{n} \equiv \langle c c^{\dagger}\rangle$ are the electron and hole occupation number of the single particle state respectively. Carrying out the Fourier transform, $g^{\lambda}\left(E\right)$ in the energy domain can be obtained as

$$g^r\left(E\right) = \int dt e^{\mathrm{i}Et} g^r\left(t\right) = \frac{1}{E - \varepsilon + \mathrm{i}0^+}, \tag{2.47}$$

$$g^a\left(E\right) = \int dt e^{\mathrm{i}Et} g^a\left(t\right) = \frac{1}{E - \varepsilon - \mathrm{i}0^+}, \tag{2.48}$$

$$g^{<}\left(E\right) = \int dt e^{\mathrm{i}Et} g^{<}\left(t\right) = +\mathrm{i}2\pi n\delta\left(E - \varepsilon\right), \tag{2.49}$$

$$g^{>}\left(E\right) = \int dt e^{\mathrm{i}Et} g^{>}\left(t\right) = -\mathrm{i}2\pi\overline{n}\delta\left(E - \varepsilon\right), \tag{2.50}$$

where the Fourier transforms

$$\int dte^{\mathrm{i}\omega t}\left[\pm\mathrm{i}\theta\left(\pm t\right)\right]=\frac{1}{\omega\pm\mathrm{i}0^{+}}, \tag{2.51}$$

$$\int dte^{\mathrm{i}\omega t}=2\pi\delta\left(\omega\right), \tag{2.52}$$

are used in the derivation.

In the denominator of $g^{r}\left(E\right)$ and $g^{a}\left(E\right)$, there is a tiny imaginary part $\mathrm{i}0^{+}$. Although this imaginary part is infinitesimal, it makes a real difference between $g^{r}\left(E\right)$ and $g^{a}\left(E\right)$. Notice that

$$\begin{aligned}\frac{1}{x\pm\mathrm{i}0^{+}} & =\lim_{\eta\rightarrow0^{+}}\frac{1}{x\pm\mathrm{i}\eta}\\ & =\lim_{\eta\rightarrow0^{+}}\left[\frac{x}{x^{2}+\eta^{2}}\mp\mathrm{i}\frac{\eta}{x^{2}+\eta^{2}}\right]\\ & =\frac{\left(P\right)}{x}\mp\mathrm{i}\pi\delta\left(x\right),\end{aligned} \tag{2.53}$$

where (P) means to take the principal part. Consequently the difference between $g^{r}\left(E\right)$ and $g^{a}\left(E\right)$ is obtained as

$$g^{r}\left(E\right)-g^{a}\left(E\right)=-2\pi\mathrm{i}\delta\left(E-\varepsilon\right).$$

Thirdly, we derive the Green's functions for the full Hamiltonian $\hat{H}$. Define Green's function matrix by

$$G^{\lambda}\equiv\begin{pmatrix}\left\langle\left\langle c_{1}^{\dagger}\left|c_{1}\right|\right\rangle\right\rangle^{\lambda} & \left\langle\left\langle c_{1}^{\dagger}\left|c_{2}\right|\right\rangle\right\rangle^{\lambda} & \cdots\\ \left\langle\left\langle c_{2}^{\dagger}\left|c_{1}\right|\right\rangle\right\rangle^{\lambda} & \left\langle\left\langle c_{2}^{\dagger}\left|c_{2}\right|\right\rangle\right\rangle^{\lambda} & \cdots\\ \vdots & \vdots & \ddots\end{pmatrix}. \tag{2.54}$$

Notice that the quadratic Hamiltonian $\hat{H}$ can be diagonalized as a summation of single particles (see Eq. (2.44)) and the Green's functions of a single particle have been derived in Eqs. (2.47,2.48,2.49,2.50). One can obtain the Green's function matrices G^{r}, G^{a}, $G^{<}$, $G^{>}$ by using the unitary transform Eq. (2.41):

$$G^{r}\left(E\right)=U\cdot diag\left[\frac{1}{E+\mathrm{i}0^{+}-\lambda_{i}}\right]\cdot U^{\dagger}=\left(E^{+}-H\right)^{-1}, \tag{2.55}$$

$$G^{a}\left(E\right)=U\cdot diag\left[\frac{1}{E-\mathrm{i}0^{+}-\lambda_{i}}\right]\cdot U^{\dagger}=\left(E^{-}-H\right)^{-1}, \tag{2.56}$$

$$G^{<}\left(E\right)=U\cdot diag\left[+\mathrm{i}2\pi n_{i}\delta\left(E-\lambda_{i}\right)\right]\cdot U^{\dagger}, \tag{2.57}$$

$$G^{>}\left(E\right)=U\cdot diag\left[-\mathrm{i}2\pi\overline{n_{i}}\delta\left(E-\lambda_{i}\right)\right]\cdot U^{\dagger}, \tag{2.58}$$

www.ebook3000.com

in which $E^{\pm} \equiv E \pm \mathrm{i}0^{+}$. Here $n_i \equiv \langle c_i^{\dagger} c_i \rangle$ and $\overline{n}_i \equiv \langle c_i c_i^{\dagger} \rangle$ are the electron and hole occupation number of the ith eigenstate respectively. Eqs. (2.55,2.56,2.57,2.58) provide explicit expressions of NEGF in the steady-state and are the central results of this section.

Finally, we would like to investigate some mathematical properties of NEGF as consequences of Eqs. (2.55,2.56,2.57,2.58). (1) G^r and G^a are mutual-conjugate, and $-\mathrm{i}G^{<}$ and $+\mathrm{i}G^{>}$ are self-conjugate, namely,

$$\left[G^r(E)\right]^{\dagger} = G^a(E), \tag{2.59}$$

$$\left[-\mathrm{i}G^{<}(E)\right]^{\dagger} = -\mathrm{i}G^{<}(E), \tag{2.60}$$

$$\left[+\mathrm{i}G^{>}(E)\right]^{\dagger} = +\mathrm{i}G^{>}(E). \tag{2.61}$$

(2) G^r (G^a) is analytic in the upper-half (lower-half) plane as a function of complex energy z. All the singularities of G^r (G^a) lie on the lower (upper) side of the real axis, each of which corresponds to an eigenstate of H. Due to the analytical property, one can change the integral path of G^r from the real axis to a complex contour in the calculations of equilibrium occupation number [5] as well as equilibrium supercurrent [6]. (3) In the equilibrium limit, the occupation numbers n_i and $\overline{n_i}$ are reduced to Fermi functions $f(\lambda_i)$ and $\bar{f}(\lambda_i)$ respectively. Here $f(E)$ is the electron's Fermi function

$$f(E) = \frac{1}{e^{(E-\mu)/k_B T} + 1}, \tag{2.62}$$

and $\overline{f}(E)$ is the hole's Fermi function

$$\overline{f}(E) = 1 - f(E) = \frac{1}{e^{-(E-\mu)/k_B T} + 1}, \tag{2.63}$$

where μ is the chemical potential, k_B is the Boltzmann constant, and T is the temperature. By using Eq. (2.53), $G^{<}$ and $G^{>}$ can be expressed in terms of G^r and G^a

$$G^{<}(E) = f(E)\left[G^a(E) - G^r(E)\right], \tag{2.64}$$

$$G^{>}(E) = \overline{f}(E)\left[G^r(E) - G^a(E)\right]. \tag{2.65}$$

Eqs. (2.64,2.65) are referred to as the fluctuation-dissipation theorem. The theorem will used in the calculation of lead self-energies in Section 2.5.

We would like to point out that the above mathematical properties for the NEGF of steady-state are valid beyond the assumption of quadratic Hamiltonian. In Appendix A.4, the Green's functions are derived in the Lehmann spectral representation which can be viewed as a generalization of Eqs. (2.55,2.56,2.57,2.58). With the Lehmann spectral representation the fluctuation-dissipation theorem can be proved for a general Hamiltonian.

2.5 Dyson equation

We have seen in Eq. (2.55) that the way to calculate $G^r(E)$ is to invert the matrix of $E^+ - H$. In a two-probe system, the Hamiltonian matrix H_∞ is infinitely large. Since no computer can handle infinitely large matrices, we need to reduce H_∞ to an effective Hamiltonian with finite size. This will be achieved by introducing an additional term called **self-energy**.

As shown in Fig. 2.1, a two-probe system can be divided into the left lead, the central region, and the right lead. Since the wave scattering occurs in the central region, we shall focus on this region and define the retarded Green's function by

$$G^r(E) \equiv \left[\left(E^+ - H_\infty\right)^{-1}\right]_{CC}, \tag{2.66}$$

where $[\cdots]_{CC}$ means to take the subspace corresponding to the central region. Notice that the inverse operation $[\cdots]^{-1}$ and the subspace operation $[\cdots]_{CC}$ are not exchangeable in the definition of $G^r(E)$. If the order of two operations is exchanged, we shall obtain a new retarded Green's function

$$g^r(E) \equiv \left(E^+ - [H_\infty]_{CC}\right)^{-1} = \left(E^+ - H_{CC}\right)^{-1}, \tag{2.67}$$

which is the Green's function of an isolated central region without coupling to the leads. Obviously $G^r(E) \neq g^r(E)$ because the effects of semi-infinite leads are completely ignored in the latter. Nevertheless $g^r(E)$ provides a good starting point to calculate $G^r(E)$. The lead effects can be taken into account by adding some additional terms to the expression of $g^r(E)$

$$G^r(E) = \left[E^+ - H_{CC} - \Sigma_L^r(E) - \Sigma_R^r(E)\right]^{-1}, \tag{2.68}$$

where $\Sigma_L^r(E)$ and $\Sigma_R^r(E)$ are the self-energies of the left and right leads. Here $H_{CC} + \Sigma_L^r(E) + \Sigma_R^r(E)$ can be viewed as an effective Hamiltonian $H_{eff}(E)$ such that $G^r(E)$ calculated by $[E^+ - H_{eff}(E)]^{-1}$ is identical to the original definition of Eq. (2.66). But how to construct $\Sigma_L^r(E)$ and $\Sigma_R^r(E)$?

To obtain the explicit form of $\Sigma_L^r(E)$ and $\Sigma_R^r(E)$, we first derive a useful mathematical lemma. Assume B is a 2×2 block matrix and A is the inverse of B. By definition

$$\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

www.ebook3000.com

One can solve the matrix blocks of A by Gauss elimination and obtain

$$A_{11} = \left(B_{11} - B_{12}B_{22}^{-1}B_{21}\right)^{-1}, \tag{2.69}$$
$$A_{22} = B_{22}^{-1} + B_{22}^{-1}B_{21}A_{11}B_{12}B_{22}^{-1}, \tag{2.70}$$
$$A_{12} = -A_{11}B_{12}B_{22}^{-1}, \tag{2.71}$$
$$A_{21} = -B_{22}^{-1}B_{21}A_{11}, \tag{2.72}$$

and

$$A_{11} = B_{11}^{-1} + B_{11}^{-1}B_{12}A_{22}B_{21}B_{11}^{-1}, \tag{2.73}$$
$$A_{22} = \left(B_{22} - B_{21}B_{11}^{-1}B_{12}\right)^{-1}, \tag{2.74}$$
$$A_{12} = -B_{11}^{-1}B_{12}A_{22}, \tag{2.75}$$
$$A_{21} = -A_{22}B_{21}B_{11}^{-1}. \tag{2.76}$$

Now we are ready to work on the lead self-energies. In the above lemma, denote 1 the central region and 2 the leads. Let $B = E^+ - H_\infty$ and consequently $G^r(E) = A_{11}$. By using Eq. (2.69), one obtains

$$G^r(E) = \left[E^+ - H_{CC} - \Sigma^r(E)\right], \tag{2.77}$$

where $\Sigma^r(E)$ is derived as

$$\Sigma^r(E) = \Sigma_L^r(E) + \Sigma_R^r(E), \tag{2.78}$$
$$\Sigma_L^r(E) = H_{CL}\left(E^+ - H_{LL}\right)^{-1}H_{LC}, \tag{2.79}$$
$$\Sigma_R^r(E) = H_{CR}\left(E^+ - H_{RR}\right)^{-1}H_{RC}. \tag{2.80}$$

In the above derivation, the total self-energy Σ^r is a sum of lead self-energies because all the leads are completely decoupled ($H_{LR} = H_{RL} = 0$). One can see that the mathematical nature of self-energy is nothing but Gauss elimination.

Having derived Eq. (2.77), it is straightforward to write down the Dyson equation of retarded Green's function

$$G^r(E) = \left[\left(g^r(E)\right)^{-1} - \Sigma^r(E)\right]^{-1}, \tag{2.81}$$

where $g^r(E)$ is defined by Eq. (2.67). Remember that the complex-time Green's function has the same mathematical structure as that of the retarded Green's function. One can simply remove the superscript r in Eq. (2.81) to obtain the Dyson equation of complex-time Green's function

$$G = \left[g^{-1} - \Sigma\right]^{-1}. \tag{2.82}$$

The Dyson equation may have other equivalent forms

$$G = g + g\Sigma g + g\Sigma g\Sigma g + \cdots, \tag{2.83}$$

$$G = g + g\Sigma G, \tag{2.84}$$

$$G = g + G\Sigma g, \tag{2.85}$$

which can be easily tested by recursion.

By applying the Langreth theorem Eqs. (2.31,2.35) to Eq. (2.82), one can recover the Dyson equation of G^r

$$G^r = \left[(g^r)^{-1} - \Sigma^r\right]^{-1}. \tag{2.86}$$

Similarly one can obtain the Dyson equation of G^a

$$G^a = \left[(g^a)^{-1} - \Sigma^a\right]^{-1}. \tag{2.87}$$

By applying the generalized Langreth theorem Eqs. (2.29–2.36) to Eq. (2.82)

$$\begin{aligned} G^< &= -\left(\left[g^{-1} - \Sigma\right]^{-1}\right)^r \left[g^{-1} - \Sigma\right]^< \left(\left[g^{-1} - \Sigma\right]^{-1}\right)^a \\ &= -G^r \left[\left(g^{-1}\right)^< - \Sigma^<\right] G^a \\ &= -G^r \left[-\left(g^r\right)^{-1} g^< \left(g^a\right)^{-1} - \Sigma^<\right] G^a, \end{aligned}$$

one can obtain the Dyson equation of $G^<$

$$G^< = G^r \left(g^r\right)^{-1} g^< \left(g^a\right)^{-1} G^a + G^r \Sigma^< G^a. \tag{2.88}$$

Similarly one can obtain the Dyson equation of $G^>$

$$G^> = G^r \left(g^r\right)^{-1} g^> \left(g^a\right)^{-1} G^a + G^r \Sigma^> G^a. \tag{2.89}$$

Due to historical reasons, Eq. (2.88) and Eq. (2.89) are referred to as the Keldysh equations although they are actually the Dyson equations of $G^<$ and $G^>$.

The first term in the RHS of Keldysh equation vanishes in two-probe systems. This is obvious from a physics argument: By connecting the central region to the leads (reservoirs), the initial statistical information of the central region is wiped off after reaching the steady state. Since the first term contains $g^<$ which is proportional to the density matrix of the isolated central region, it must vanish in two-probe systems. The statement can be

www.ebook3000.com

further justified by a quantitative analysis: In the vicinity of a resonance, G^r and G^a can be approximated by

$$G^{r,a} \sim \frac{1}{E - E_0 \pm \mathrm{i}\Gamma},$$

where E_0 is the resonance energy and Γ is the broadening. In the vicinity of an eigenvalue, g^r, g^a, $g^<$ can be approximated by

$$\begin{aligned} g^{r,a} &\sim \frac{1}{E - \varepsilon_0 \pm \mathrm{i}\delta}, \\ g^< &\sim 2\pi \mathrm{i} n_0 \frac{1}{2\pi} \frac{2\delta}{(E - \varepsilon_0)^2 + \delta^2}, \end{aligned}$$

where ε_0 is the eigenenergy, δ is the broadening, and n_0 is the occupation number of the state ε_0. The contribution of the first term to an energy integral can be estimated as

$$\begin{aligned} N &\sim -\mathrm{i} \int \frac{dE}{2\pi} G^r \left(g^r\right)^{-1} g^< \left(g^a\right)^{-1} G^a \\ &\sim \int \frac{dE}{2\pi} \frac{2\delta n_0}{(E - E_0)^2 + \Gamma^2} = \frac{\delta}{\Gamma} n_0. \end{aligned} \tag{2.90}$$

As long as $\delta \ll \Gamma$, the contribution of the first term is always negligible. Generally the resonant state of a two-probe system has a finite lifetime and the eigenstate of an isolated system has an infinite lifetime. Due to the uncertainty relation, the imaginary energy is inversely proportional to the lifetime, hence $\Gamma > 0$ and $\delta \to 0$. As a result, the first term vanishes in two-probe systems [7], and the Keldysh equations are reduced to

$$G^< = G^r \Sigma^< G^a, \tag{2.91}$$

$$G^> = G^r \Sigma^> G^a. \tag{2.92}$$

Two comments on the Keldysh equations are in order. First, a useful identity can be obtained by using the above Keldysh equations,

$$G^r - G^a = G^r \left(\Sigma^r - \Sigma^a\right) G^a. \tag{2.93}$$

The proof of the identity is as follows:

$$\begin{aligned} G^r - G^a &= G^> - G^< \\ &= G^r \left(\Sigma^> - \Sigma^<\right) G^a \\ &= G^r \left(\Sigma^r - \Sigma^a\right) G^a, \end{aligned}$$

where Eq. (2.23) and Eqs. (2.91,2.92) are used in the derivation. Second, $\Sigma^<$ in the Keldysh equation can be derived as

$$\Sigma^< = \sum_{\beta=L,R} \Sigma^<_\beta = \sum_{\beta=L,R} f_\beta \left(\Sigma^a_\beta - \Sigma^r_\beta\right), \tag{2.94}$$

where f_β is the Fermi function of the lead-β. Here the fluctuation-dissipation theorem (see Eq. (2.64)) is applied to each lead to obtain its lesser self-energy $\Sigma^<_\beta$. In particular if the two-probe system is in equilibrium, $f_L = f_R = f_0$, the lesser self-energy is reduced to

$$\Sigma^< = f_0 \left(\Sigma^a - \Sigma^r\right), \tag{2.95}$$

and the lesser Green's function is reduced to

$$G^< = G^r f_0 \left(\Sigma^a - \Sigma^r\right) G^a = f_0 \left(G^r - G^a\right), \tag{2.96}$$

which is nothing but the fluctuation-dissipation theorem applied to the central region.

To sum up, we have derived the Dyson equations for $G^{r,a,<,>}$ in two-probe systems, and Eqs. (2.86,2.87,2.91,2.92) are the central results of this section. In the derivation of the Dyson equations, we have developed an important concept called self-energy. The key idea is to identify a system of interest (the central region) and environmental degrees of freedom (the left and right leads). The environmental degrees of freedom can be eliminated by adding proper self-energies to the system. In addition to the lead self-energy discussed in this section, other physical processes such as disorder scattering and electron-phonon coupling can also be taken into account by various self-energies

$$\Sigma = \Sigma_{lead} + \Sigma_{disorder} + \Sigma_{e-ph} + \cdots . \tag{2.97}$$

In the following sections, we shall derive the explicit forms of lead self-energy (Section 2.7) and disorder scattering self-energy (Section 2.8).

2.6 Current formula

We have developed the NEGF formalism in the previous sections. In this section, we shall apply the technique to study quantum transport in two-probe systems. A general recipe of using Green's function technique to solve physics problems is as follows. Step 1, express the physical quantities in terms of Green's functions; Step 2, calculate Green's functions by using standard techniques; Step 3, interpret the obtained results in the context of physics problems. The procedure is analogous to driving from A to B

www.ebook3000.com

through a highway. In step 1, one needs to drive along some local road to the entrance of highway. In step 2, one simply drives along the highway; In step 3 one needs to exit from the highway and drive through another local road to the destination. The advantage of using the highway (Green's function) is that it is systematic and fast. As a price, one may miss some beautiful scenery on the way: The physical meaning is less transparent in the derivation of Green's functions.

The most important physical quantities in a two-probe system are the occupation number and the electric current. It is straightforward to obtain the occupation number of the site i by using lesser Green's function

$$N_i = \int \frac{dE}{2\pi} \left(-\mathrm{i}\right) G_{ii}^{<}\left(E\right). \tag{2.98}$$

The electric current is more complicated and the formula will be derived below step by step. The derivation can be viewed as a simplified version of Ref. [9].

Define $\hat{N}_L\left(t\right) \equiv \sum_p c_p^{\dagger}\left(t\right) c_p\left(t\right)$ as the operator (in the Heisenberg picture) of the total electron number in the left lead. Due to electron number conservation, the electric current flowing out of the left lead can be written as

$$I_L = -Q_e \langle \partial_t \hat{N}_L\left(t\right) \rangle, \tag{2.99}$$

where $Q_e = -1$ is to take into account that electron has a negative charge. Notice that the operator in the Heisenberg picture satisfies

$$\mathrm{i}\partial_t \hat{N}_L\left(t\right) = \left[\hat{N}_L\left(t\right), \hat{H}\left(t\right)\right], \tag{2.100}$$

where $\hat{H}$ is the Hamiltonian operator of the two-probe system defined in Eq. (2.1). It follows

$$\begin{aligned} I_L &= Q_e \mathrm{i} \langle \sum_{ip} t_{pi} c_p^{\dagger}\left(t\right) c_i\left(t\right) - t_{ip} c_i^{\dagger}\left(t\right) c_p\left(t\right) \rangle \\ &= Q_e 2\mathrm{Re}\left[\mathrm{i} \sum_{ip} t_{pi} \langle c_p^{\dagger}\left(t\right) c_i\left(t\right) \rangle\right] \\ &= Q_e 2\mathrm{Re}\left[\sum_{ip} t_{pi} \langle\langle c_i\left(t\right) | c_p^{\dagger}\left(t\right) \rangle\rangle^{<}\right] \\ &= Q_e \int \frac{dE}{2\pi} 2\mathrm{Re}\left[\sum_{ip} t_{pi} \langle\langle c_i | c_p^{\dagger} \rangle\rangle^{<}\right], \end{aligned}$$

where Eq. (2.38) is used to transform from the time domain to the energy domain. Notice that

$$\begin{aligned}\sum_{ip} t_{pi}\langle\langle c_i|c_p^\dagger\rangle\rangle^< &= \sum_{ip}\left[\langle\langle c_i|c_p^\dagger\rangle\rangle t_{pi}\right]^< \\ &= \mathrm{Tr}\left[G_{CL}\left(E\right)H_{LC}\right]^< \\ &= \mathrm{Tr}\left[G_{CC}\left(E\right)H_{CL}g_{LL}\left(E\right)H_{LC}\right]^< \\ &= \mathrm{Tr}\left[G\left(E\right)\Sigma_L\left(E\right)\right]^<,\end{aligned}$$

where Eq. (2.71) and Eq. (2.79) are used in the derivation. By using the Langreth theorem Eq. (2.29), one obtains $\left(G\Sigma_L\right)^< = G^r\Sigma_L^< + G^<\Sigma_L^a$, and I_L is finally derived as

$$I_L = Q_e\int\frac{dE}{2\pi}2\mathrm{Re}\ \mathrm{Tr}\left[G^r\left(E\right)\Sigma_L^<\left(E\right)+G^<\left(E\right)\Sigma_L^a\left(E\right)\right]. \tag{2.101}$$

Similarly one can derive the electric current flowing out of the right lead as

$$I_R = Q_e\int\frac{dE}{2\pi}2\mathrm{Re}\ \mathrm{Tr}\left[G^r\left(E\right)\Sigma_R^<\left(E\right)+G^<\left(E\right)\Sigma_R^a\left(E\right)\right]. \tag{2.102}$$

Although Eq. (2.101) and Eq. (2.102) can be applied to calculate the electric current, they are not so transparent in physics. As we have demonstrated from the scattering states approach the electric current can be expressed as an energy integral of transmission coefficient multiplied by the Fermi function difference (see Eq. (1.5)). This is the "scenery" that has been missed by driving along the "highway". Let us get it back by using some mathematical tricks. Define F as

$$F \equiv 2\mathrm{Re}\ \mathrm{Tr}\left[G^r\Sigma_L^< + G^<\Sigma_L^a\right]. \tag{2.103}$$

F can be simplified by using the conjugate relations Eqs. (2.59,2.60)

$$\begin{aligned}F &= \mathrm{Tr}\left[G^r\Sigma_L^< + G^<\Sigma_L^a + H.c.\right] \\ &= \mathrm{Tr}\left[G^r\Sigma_L^< + G^<\Sigma_L^a - \Sigma_L^<G^a - \Sigma_L^rG^<\right] \\ &= \mathrm{Tr}\left[G^r\Sigma_L^< + G^<\Sigma_L^a - G^a\Sigma_L^< - G^<\Sigma_L^r\right] \\ &= \mathrm{Tr}\left[\left(G^r-G^a\right)\Sigma_L^< - G^<\left(\Sigma_L^r-\Sigma_L^a\right)\right],\end{aligned}$$

where $H.c.$ means Hermitian conjugate. By inserting Eqs. (2.91,2.93,2.94) into the above expression, F can be reduced to a symmetric form with respect to the left and right leads

$$\begin{aligned}F &= -\mathrm{Tr}\left[G^r\left(\Sigma_R^a-\Sigma_R^r\right)G^a\left(\Sigma_L^a-\Sigma_L^r\right)\left(f_L-f_R\right)\right] \\ &= \mathrm{Tr}\left(G^r\Gamma_RG^a\Gamma_L\right)\left(f_L-f_R\right),\end{aligned}$$

www.ebook3000.com

where the linewidth function $\Gamma_\beta(E)$ is defined by

$$\Gamma_\beta(E) \equiv \mathrm{i}\left[\Sigma_\beta^r(E) - \Sigma_\beta^a(E)\right]. \tag{2.104}$$

Therefore the electric current Eq. (2.101) can be rewritten as

$$I_L = Q_e \int \frac{dE}{2\pi} T_{LR}(E)\left[f_L(E) - f_R(E)\right], \tag{2.105}$$

where the transmission coefficient $T_{LR}(E)$ is defined by

$$T_{LR}(E) \equiv \mathrm{Tr}\left[G^r(E)\Gamma_R(E)G^a(E)\Gamma_L(E)\right]. \tag{2.106}$$

Similarly the electric current Eq. (2.102) can be rewritten as

$$I_R = Q_e \int \frac{dE}{2\pi} T_{RL}(E)\left[f_R(E) - f_L(E)\right], \tag{2.107}$$

where the transmission coefficient $T_{RL}(E)$ is defined by

$$T_{RL}(E) \equiv \mathrm{Tr}\left[G^r(E)\Gamma_L(E)G^a(E)\Gamma_R(E)\right]. \tag{2.108}$$

Due to the current conservation, $I_L + I_R = 0$, and hence $T_{LR}(E)$ and $T_{RL}(E)$ must be equal (see Section 2.9 for a proof). Hereafter we won't distinguish between $T_{LR}(E)$ and $T_{RL}(E)$, and write the current formula of $I = I_L = -I_R$ as

$$I = Q_e \int \frac{dE}{2\pi} T(E)\left[f_L(E) - f_R(E)\right], \tag{2.109}$$

where the transmission coefficient is defined by

$$T(E) \equiv \mathrm{Tr}\left[G^r(E)\Gamma_L(E)G^a(E)\Gamma_R(E)\right]. \tag{2.110}$$

Notice that Eq. (2.109) formally recovers the result of the scattering state approach.

Below we shall demonstrate that the transmission coefficient defined by Eq. (2.110) is equivalent to the transmission coefficient defined in the scattering states approach. To proceed, we carry out a Γ-decomposition [8] to the linewidth function Γ_β. Notice that $\Gamma_\beta^\dagger = \Gamma_\beta$ due to its definition Eq. (2.104). Γ_β can be eigen-decomposed as $\Gamma_\beta = U_\beta D_\beta U_\beta^\dagger$, where U_β is a unitary matrix (eigenvector matrix) and D_β is a real diagonal matrix (eigenvalue matrix). Due to the physical meaning, the linewidth function Γ_β must be positive-definite and hence $D_\beta > 0$. Define $W_\beta \equiv U_\beta\sqrt{D_\beta}$, one obtains

$$\Gamma_\beta = W_\beta W_\beta^\dagger = \sum_k |\beta_k\rangle\langle\beta_k|, \tag{2.111}$$

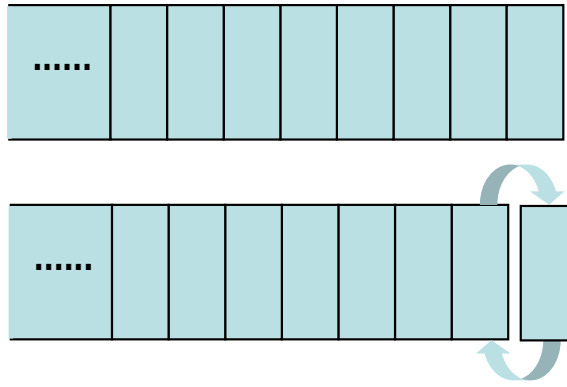


Fig. 2.3 (a) A semi-infinite lead can be partitioned into unit cells along the transport direction, each of which only interacts with its nearest neighbors. (b) The semi-infinite lead resembles itself after removing the unit cell on the surface.

where $|\beta_k\rangle$ is the k-th column of W_β. Therefore transmission defined in Eq. (2.110) can be rewritten as

$$\begin{aligned} T &\equiv \sum_{kk'} \mathrm{Tr}\left[G^r\left|L_k\right\rangle\left\langle L_k\right| G^a\left|R_{k'}\right\rangle\left\langle R_{k'}\right|\right] \\ &= \sum_{kk'} \left\langle R_{k'}\right| G^r\left|L_k\right\rangle\left\langle L_k\right| G^a\left|R_{k'}\right\rangle = \sum_{kk'} \left|t_{k'k}\right|^2, \end{aligned} \tag{2.112}$$

where $t_{k'k} \equiv \left\langle R_{k'}\right| G^r\left|L_k\right\rangle$ is the amplitude of the incoming wave $|L_k\rangle$ scattered to the outgoing wave $|R_{k'}\rangle$. So the result of scattering states approach is completely recovered [8].

To sum up, the current formula Eq. (2.109) and the transmission formula Eq. (2.110) are the central results of this section. Although the NEGF formula is equivalent to the scattering states approach for the quadratic Hamiltonian, the former can go much further than the latter. For example, the NEGF formula is still valid even in the presence of Coulomb interactions in the central region [9], the NEGF formula can be easily generalized to treat time-dependent Hamiltonian [9], and the NEGF formula can include superconducting leads by using the Nambu representation [10], etc. In Section 2.9, the current conservation and dephasing effect will be further investigated for general multi-probe systems.

www.ebook3000.com

2.7 Surface Green's function

In this section, we shall discuss the retarded self-energy of the semi-infinite leads. As shown in Eqs. (2.79,2.80), the lead self-energy can be expressed in terms of the lead Green's function

$$\Sigma_\beta^r = H_{C\beta} g_{\beta\beta}^r H_{\beta C}, \tag{2.113}$$

$$g_{\beta\beta}^r \equiv \left(E^+ - H_{\beta\beta}\right)^{-1}, \tag{2.114}$$

where $\beta = L, R$ is the lead index. Since the lead Hamiltonian $H_{\beta\beta}$ is still infinitely large, Eq. (2.114) is not applicable in a practical calculation.

Further study indicates that most of the matrix elements of $H_{\beta\beta}$, $H_{C\beta}$, $H_{\beta C}$ are zeros, and one can make use of those zeros to simplify the calculation of Eq. (2.113). Notice that in a localized atomic basis set (e.g., the LMTO discussed in Chapter 3) the matrix elements between two atoms are nonzero only if the atoms are within a cutoff length. Suppose the thickness of a lead unit cell is larger than the cutoff length. The semi-infinite lead can be partitioned into a series of unit cells along the transport direction so that each unit cell only interacts with its nearest neighbors (see Fig. 2.3). As a result, $H_{\beta\beta}$ is a block tridiagonal matrix, and $H_{C\beta}$ and $H_{\beta C}$ have nonzero matrix elements only between the central region and the surface lead unit cell S_β. Eq. (2.113) is reduced to

$$\Sigma_\beta^r = H_{CS_\beta} g_{S_\beta}^r H_{S_\beta C}\ , \tag{2.115}$$

$$g_{S_\beta}^r \equiv \left[\left(E^+ - H_{\beta\beta}\right)^{-1}\right]_{S_\beta S_\beta}\ , \tag{2.116}$$

where $g_{S_\beta}^r$ is called surface Green's function. Thus the calculation of lead self-energy is reduced to the calculation of surface Green's function.

Surface Green's function satisfies the equation

$$g_{S_\beta}^r = \left(E^+ - H_\beta^0 - H_\beta^+ g_{S_\beta}^r H_\beta^-\right)^{-1}, \tag{2.117}$$

where $H_\beta^0 \equiv H_\beta(1,1)$ is the Hamiltonian of a lead unit cell, and $H_\beta^+ \equiv H_\beta(1,2)$ ($H_\beta^- \equiv H_\beta(2,1)$) is the Hamiltonian between the two nearest neighbors unit cell-1 and unit cell-2. In the left (right) lead, unit cell-1 is located on the right (left) side of unit cell-2 (see Fig. 2.3). The derivation of Eq. (2.117) is based on the following observation: If the surface unit cell of a semi-infinite lead is removed, the remaining part of the lead is identical to the original one. By definition the surface Green's function is the Green's function of the surface unit cell, and the remaining part of the lead can be taken into account by a self-energy $\tilde{\Sigma}^r$

$$g_{S_\beta}^r = \left[\left(E^+ - H_\beta^0 - \tilde{\Sigma}^r\right)\right]^{-1}.$$

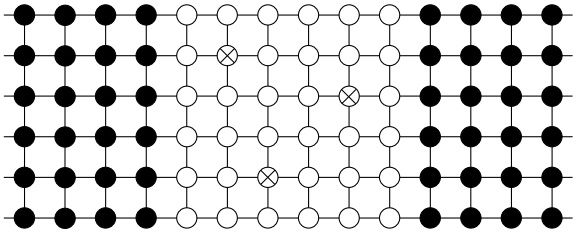


Fig. 2.4 Schematic plot of a two-probe system with disorder in the central scattering region. The black dots are clean sites in the left and right leads which extend to infinity. White circles and crossed circles are clean sites and disorder sites in the central region respectively.

The self-energy $\tilde{\Sigma}^r$ mimics the lead self-energy of Eq. (2.115) and can be obtained as $\tilde{\Sigma}^r = H_\beta^+ g_S^r H_\beta^-$. Thus Eq. (2.117) is derived for the surface Green's function.

To sum up, we have derived the self-energy due to coupling with semi-infinite lead, and Eqs. (2.115,2.117) are the central results of this section. Deriving Eq. (2.117) is straightforward, but solving it requires much more effort. In general the equation has to be solved numerically, and the algorithms will be discussed in Section 6.4.

2.8 NECPA equations

This section is the key of the chapter. We shall present the NECPA theory [11] which forms the theoretical foundation of quantum transport in disordered two-probe systems. The NECPA theory is simple and elegant in the theoretical derivation although it is mathematically connected to other previous works [12–15]. We shall demonstrate that the disorder effect can be taken into account by a special self-energy called **nonequilibrium coherent potential**.

Why are we interested in two-probe systems with disorder? The reason is that realistic devices usually have a certain degree of randomness and disorder. For example, a single computer chip contains billions of transistors. The dopant configuration varies from one transistor to another. It is more relevant to study the ensemble-averaged transport property than that of a particular transistor. For another example, in a magnetic tunnel junction, structural defects exist at the material interfaces due to lattice

www.ebook3000.com

mismatch. One does not know exactly where the defects are but can estimate the probability of the occurrence. The transport current is determined by an average over the defect distribution. In both examples, the two-probe system contains some randomness and needs to be averaged over disorder configurations.

To carry out the disorder average, a simple idea is to use brute-force average. In the first example, one can carry out simulations for a large number of transistors with different dopant configurations and make an ensemble average afterward. In the second example, one can construct a system with large cross section (supercell) and generate a few random defect configurations. As long as the size of ensemble or the area of cross section is large enough, the result should converge to the average. Such brute-force average, however, is very costly for atomistic simulations. Moreover the system size can be exceedingly large for a small disorder concentration. Fortunately the disorder average can be done analytically by using a Green's function based technique called coherent potential approximation (CPA) [16,17]. Originally CPA was developed for equilibrium bulk systems; in this section we shall generalize CPA to NECPA which is applicable to nonequilibrium two-probe systems [11].

The starting point is the Hamiltonian of a disordered two-probe system. We assume that the atoms in a disordered two-probe system are still located on a regular lattice. The randomness comes from the composition of atomic sites: An atomic site can be occupied by atom-A with the probability x_A or atom-B with the probability x_B, etc., see Fig. 2.4. The Hamiltonian of a disordered two-probe system is nearly the same as Eq. (2.1) except that the on-site energies of some atomic sites in the central region are random variables

$$\varepsilon_i = \begin{cases} \varepsilon_A & P_A = x_A \\ \varepsilon_B & P_B = x_B \\ \cdots & \cdots \end{cases}, \tag{2.118}$$

where $\sum_q x_q = 1$ due to the probability normalization. The atomic sites with random on-site energies are called disorder sites and other sites with definite on-site energies called clean sites. Notice that all the atomic sites in the leads are assumed to be clean sites. Otherwise one can always enlarge the central region to enclose all the disorder sites [18].

To calculate the disorder-averaged physical quantities from the Hamiltonian, we only need to worry about the disorder-averaged Green's functions. In fact all the physical quantities can be expressed in terms of Green's functions. To make the disorder average, one can simply replace the Green's

functions by the disorder-averaged Green's functions. For example, in the presence of disorder, the current formula Eqs. (2.101,2.102) are modified to

$$\overline{I_\beta} = Q_e \int \frac{dE}{2\pi} 2\mathrm{Re}\ \mathrm{Tr}\left[\overline{G^r(E)}\Sigma_\beta^<(E) + \overline{G^<(E)}\Sigma_\beta^a(E)\right], \tag{2.119}$$

where $\overline{\cdots}$ means to carry out the disorder average. Notice that there is no disorder average over $\Sigma_\beta^r(E)$ and $\Sigma_\beta^<(E)$ since the leads are assumed to be clean. Transmission coefficient Eq. (2.110) contains two Green's functions' product, and the disorder average is much more difficult. Due to the statistical correlation, $\overline{T(E)}$ cannot be calculated directly with $\overline{G^r(E)}$ and $\overline{G^a(E)}$,

$$\overline{T(E)} = \mathrm{Tr}\ \overline{G^r(E)\Gamma_L(E)G^a(E)\Gamma_R(E)} \neq \mathrm{Tr}\ \overline{G^r(E)}\Gamma_L(E)\overline{G^a(E)}\Gamma_R(E).$$

Instead $\overline{T(E)}$ can be reduced to the calculation of $\overline{G^<}$ by using a mathematical trick: $\Sigma^<(E)$ is reduced to $\mathrm{i}\Gamma_L(E)$ by setting $f_L(E) = 1$ and $f_R(E) = 0$ in Eq. (2.94). Consequently $\overline{T(E)}$ is obtained as

$$\overline{T(E)} = \mathrm{Tr}\ \overline{G^r(E)\Gamma_L(E)G^a(E)}\Gamma_R(E) = \mathrm{Tr}\ (-\mathrm{i})\,\overline{G_L^<(E)}\Gamma_R(E), \tag{2.120}$$

where $\overline{G_L^<(E)}$ is defined by

$$\overline{G_L^<(E)} \equiv \left[\overline{G^<(E)}\right]_{f_L(E)=1, f_R(E)=0}. \tag{2.121}$$

For physical quantities involving more Green's functions' product, one can use the diagrammatic technique to reduce the problem to the calculation of disorder-averaged Green's functions and vertex corrections [19]. Therefore all the calculations of disorder average are reduced to the calculations of $\overline{G^r}$ and $\overline{G^<}$ (Hereafter the argument E is omitted for simplicity of notation).

Now we derive the NECPA equations for calculating $\overline{G^r}$ and $\overline{G^<}$. The plan is as follows: We first derive the CPA equation for the retarded Green's function. Afterward we generalize the CPA equation to a contour ordered CPA equation and apply the Langreth theorem to obtain the NECPA equations. The elegance of the NECPA equations is that it treats $\overline{G^r}$ and $\overline{G^<}$ on an equal footing, analogous to the electric field and the magnetic field in the electromagnetism.

CPA equation was originally derived from the multiple scattering theory [16,17]. The idea is to construct an effective on-site energy on each disorder site so that the averaged scattering amplitude vanishes. Instead of following the historical development, we shall adopt a heuristic derivation where CPA

www.ebook3000.com

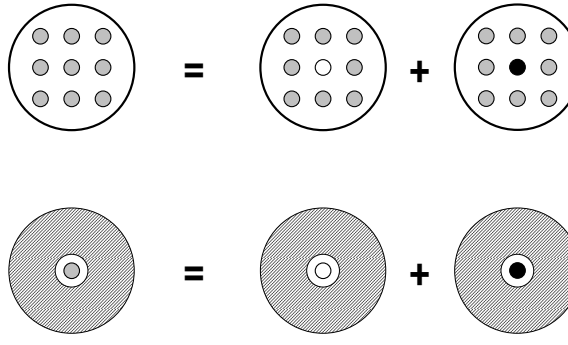


Fig. 2.5 (a) A schematic diagram of the derivation for the CPA equation (2.129). (b) The schematic diagram of the derivation for the coherent interactor Eqs. (2.130,2.131).

is regarded as an ansatz. Suppose the central region contains N disorder sites. In CPA the random on-site energy of a disorder site is replaced by an effective on-site energy called coherent potential. To determine the N coherent potentials, we need to set up N equations. Let us focus on one particular disorder site i. Define the on-site retarded Green's function $\overline{G_i^r}$ as

$$\overline{G_i^r} \equiv \left[\overline{G^r}\right]_{ii}. \tag{2.122}$$

We shall calculate $\overline{G_i^r}$ by two methods to establish the CPA equation. On the one hand, $\overline{G_i^r}$ can be calculated by applying the CPA ansatz to all the disorder sites. That is to replace all the random on-site energies by the coherent potentials and obtain

$$\overline{G^r} = \left[E^+ - H_{CC}^0 - \tilde{\varepsilon}^r - \Sigma^r\right]^{-1}, \tag{2.123}$$

where H_{CC}^0 is the off-diagonal (definite) part of H_{CC}, and $\tilde{\varepsilon}^r$ is a diagonal matrix of coherent potentials

$$\tilde{\varepsilon}^r \equiv diag\left([\cdots, \tilde{\varepsilon}_1^r, \cdots, \tilde{\varepsilon}_i^r, \cdots, \tilde{\varepsilon}_N^r, \cdots]\right). \tag{2.124}$$

Notice that the coherent potential of a clean site is just its definite on-site energy. Eq. (2.122) and Eq. (2.123) lead to

$$\overline{G_i^r} = \left\{\left[E^+ - H_{CC}^0 - \tilde{\varepsilon}^r - \Sigma^r\right]^{-1}\right\}_{ii}. \tag{2.125}$$

On the other hand, $\overline{G_i^r}$ can be calculated by applying the CPA ansatz to all the disorder sites other than site i. For site i, the on-site energy can take the value ε_{iq} with probability x_{iq}. So the $\overline{G_i^r}$ is an average of all possible occupation of site i

$$\overline{G_i^r} = \sum_q x_{iq} \overline{G_{iq}^r}, \tag{2.126}$$

where $\overline{G_{iq}^r}$ is called conditional Green's function meaning that the site i takes the definite on-site energy ε_{iq} and all other disorder sites remain random sites. By applying the CPA ansatz to the $N-1$ disorder sites, one obtains

$$\overline{G_{iq}^r} = \left\{ \left[E^+ - H_{CC}^0 - \tilde{\varepsilon}_{iq}^r - \Sigma^r \right]^{-1} \right\}_{ii}, \tag{2.127}$$

in which $\tilde{\varepsilon}_{iq}^r$ is defined by

$$\tilde{\varepsilon}_{iq}^r \equiv diag\left([\cdots, \tilde{\varepsilon}_1^r, \cdots, \varepsilon_{iq}, \cdots, \tilde{\varepsilon}_N^r, \cdots] \right). \tag{2.128}$$

Comparing Eq. (2.125) and Eq. (2.127), one obtains N equations for the N unknowns

$$\left\{ \left[E^+ - H_{CC}^0 - \tilde{\varepsilon}^r - \Sigma^r \right]^{-1} \right\}_{ii} = \sum_q x_{iq} \left\{ \left[E^+ - H_{CC}^0 - \tilde{\varepsilon}_{iq}^r - \Sigma^r \right]^{-1} \right\}_{ii}, \tag{2.129}$$

where the disorder site index i runs from 1 to N. Eq. (2.129) is the CPA equation for the retarded Green's function.

The CPA equation (2.126) can be further simplified by introducing a new quantity called coherent interactor (see Fig. 2.5). Notice that the diagonal matrices $\tilde{\varepsilon}^r$ and $\tilde{\varepsilon}_{iq}^r$ only differ in the ith diagonal element. If the ith site is regarded as the system and the remaining part as the environment, one can apply the lemma of Section 2.5 to Eq. (2.125) and Eq. (2.127). Define $B = E^+ - H_{CC}^0 - \tilde{\varepsilon}^r - \Sigma^r$ and $B' = E^+ - H_{CC}^0 - \tilde{\varepsilon}_{iq}^r - \Sigma^r$, and denote site i by 1 and all the other sites by 2. By applying Eq. (2.69) to B and B', one obtains

$$\overline{G_i^r} = A_{11} = \left(E^+ - \tilde{\varepsilon}_i^r - \Omega_i^r \right)^{-1}, \tag{2.130}$$

$$\overline{G_{iq}^r} = A'_{11} = \left(E^+ - \varepsilon_{iq} - \Omega_i^r \right)^{-1}, \tag{2.131}$$

where $\Omega_i^r \equiv B_{12} B_{22}^{-1} B_{21} + \Sigma_{ii}^r = B'_{12} B'^{-1}_{22} B'_{21} + \Sigma_{ii}^r$ is called coherent interactor. In fact the coherent interactor can be viewed a self-energy due to the coupling between the site and the remaining part of the two-probe system. Thus the CPA Eq. (2.129) is reduced to an equivalent form

www.ebook3000.com

Eqs. (2.123,2.126,2.130,2.131). The obtained CPA equations can be applied to study the equilibrium properties of disordered two-probe systems.

To study the quantum transport it is necessary to generalize the above CPA equations to the nonequilibrium situation. The generalization is amazingly simple by using the fact that the complex-time Green's function has exactly the same mathematical structure as that of the retarded Green's function (see Section 2.2). One can simply remove the superscript r in Eqs. (2.123,2.126,2.130,2.131) to obtain contour ordered CPA equations:

$$\begin{cases} \overline{G_i} = \sum_q x_{iq} \overline{G_{iq}}, \\ \overline{G} = \left[E - H_{CC}^0 - \tilde{\varepsilon} - \Sigma\right]^{-1}, \\ \overline{G_i} = \left[\overline{G}\right]_{ii}, \\ \overline{G_i} = \left[E - \tilde{\varepsilon}_i - \Omega_i\right]^{-1}, \\ \overline{G_{iq}} = \left[E - \varepsilon_{iq} - \Omega_i\right]^{-1}. \end{cases} \tag{2.132}$$

As pointed out in Section 2.3, Langreth theorem is the bridge between the complex-time Green's function and the real-time Green's function. By applying the generalized Langreth theorem Eqs. (2.29,2.30,2.31,2.32) and Eqs. (2.33,2.34,2.35,2.36) to Eq. (2.132), two sets of equations can be obtained for $\overline{G^r}$ and $\overline{G^<}$ respectively:

$$\begin{cases} \overline{G_i^r} = \sum_q x_{iq} \overline{G_{iq}^r}, \\ \overline{G^r} = \left[E - H_{CC}^0 - \tilde{\varepsilon}^r - \Sigma^r\right]^{-1}, \\ \overline{G_i^r} = \left[\overline{G^r}\right]_{ii}, \\ \overline{G_i^r} = \left[E - \tilde{\varepsilon}_i^r - \Omega_i^r\right]^{-1}, \\ \overline{G_{iq}^r} = \left[E - \varepsilon_{iq} - \Omega_i^r\right]^{-1}, \end{cases} \tag{2.133}$$

$$\begin{cases} \overline{G_i^<} = \sum_q x_{iq} \overline{G_{iq}^<}, \\ \overline{G^<} = \overline{G^r} \left(\Sigma^< + \tilde{\varepsilon}^< \right) \overline{G^a}, \\ \overline{G_i^<} = \left[\overline{G^<} \right]_{ii}, \\ \overline{G_i^<} = \overline{G_i^r} \left(\tilde{\varepsilon}_i^< + \Omega_i^< \right) \overline{G_i^a}, \\ \overline{G_{iq}^<} = \overline{G_{iq}^r} \Omega_i^< \overline{G_{iq}^a}. \end{cases} \tag{2.134}$$

Here $\tilde{\varepsilon}^<$ and $\Omega_i^<$ are lesser coherent potential and lesser coherent interactor which generalize the corresponding quantities in equilibrium. Eq.(2.133) and Eq.(2.134) extend the CPA to the nonequilibrium situation, and will be referred to as the NECPA equations in the rest of this monograph.

Generally the NECPA equations need to be solved numerically and the iterative procedure will be presented in Section 6.7. If the disorder concentration is sufficiently low, an approximate analytical solution to the NECPA equations can be obtained. In semiconductor devices, even heavy doping concentration 10^{20} cm^{-3} amounts to a very low disorder concentration $x \sim 2 \times 10^{-3}$. For such a low disorder concentration, an analytical solution can be obtained by a perturbation expansion with respect to the small parameter x. Let $q = 0$ label the host material species and $q > 0$ the impurity species. Low disorder concentration implies that $x_{i,q=0} \gg x_{i,q>0}$. The retarded and lesser coherent potentials are obtained up to the first order of $x_{i,q>0}$ as (see Appendix A.5):

$$\tilde{\varepsilon}_i^r \approx \varepsilon_{i0} + \sum_{q>0} x_{iq} t_{iq}^r, \tag{2.135}$$

$$\tilde{\varepsilon}_i^< \approx \sum_{q>0} x_{iq} t_{iq}^r G_{0,ii}^< t_{iq}^a, \tag{2.136}$$

where

$$t_{iq}^r = \left[\left(\varepsilon_{iq} - \varepsilon_{i0} \right)^{-1} - G_{0,ii}^r \right]^{-1}, \tag{2.137}$$

$$G_0^r = \left[E - H_{CC}^0 - \varepsilon^0 - \Sigma^r \right]^{-1}, \tag{2.138}$$

$$G_0^< = G_0^r \Sigma^< G_0^a, \tag{2.139}$$

in which $\varepsilon^0 = diag\left([\varepsilon_{10}, \varepsilon_{20}, \cdots] \right)$ is the on-site energy of the host material. The analytical solution allows one to calculate $\overline{G^r}$ and $\overline{G^<}$ by using the second line of Eq. (2.133) and Eq. (2.134) without any iterative procedure.

www.ebook3000.com

To sum up, the NECPA equations (2.133,2.134) are the central results of this section. In the NECPA equations, the coherent potential and the coherent interactor are generalized to the nonequilibrium situation. We would like to point out that the nonequilibrium coherent potential $\tilde{\varepsilon}$ can be viewed as a new type of self-energy to take into account the disorder effect. Denote $\Sigma_D \equiv \tilde{\varepsilon}$ and the total self-energy reads $\Sigma = \Sigma_D + \Sigma_L + \Sigma_R$. Notice that the disorder self-energy Σ_D has equal status as the lead self-energies Σ_L and Σ_R. In other words, each disorder site can be viewed as a factitious lead. The next section will investigate the properties of such factitious leads.

2.9 Current conservation and dephasing effect

This section discusses the current conservation and the dephasing effect in the presence of disorder scattering [20]. We shall demonstrate that the nonequilibrium coherent potential can be viewed as a dephasing probe which does not conduct current but destroys phase coherence. In this section, we shall consider a general multi-probe system where the central region is connected to an arbitrary number of leads.

Firstly, we prove that the total current is conserved in a multi-probe system. It is straightforward to generalize the current formula Eqs. (2.101,2.102) of a two-probe system to a multi-probe system

$$\begin{aligned} I_\beta &= Q_e \int \frac{dE}{2\pi} 2\mathrm{Re}\ \mathrm{Tr}\left[G^r(E)\,\Sigma_\beta^<(E) + G^<(E)\,\Sigma_\beta^a(E)\right], \\ &= Q_e \int \frac{dE}{2\pi} 2\mathrm{Re}\ \mathrm{Tr}\left[G(E)\,\Sigma_\beta(E)\right]^<, \end{aligned} \tag{2.140}$$

where β is the lead index, I_β is the current flowing out of the lead-β, and Σ_β is the self-energy of the lead-β. As required by the physical meaning, the total current must be zero; otherwise charge will accumulate in the central region. Below it will be proved mathematically that $\sum_\beta I_\beta = 0$.

Proof: (1) Derive two useful identities Eq. (2.142) and Eq. (2.143). By using the Dyson equations

$$\begin{aligned} G^r &= \left(E^+ - H_{CC} - \Sigma^r\right)^{-1}, \\ G^a &= \left(E^- - H_{CC} - \Sigma^a\right)^{-1}, \end{aligned}$$

one obtains

$$(G^a)^{-1} - (G^r)^{-1} = \Sigma^r - \Sigma^a, \tag{2.141}$$

in which the infinitesimal imaginary energy is neglected since it does not appear in the denominator. By multiplying Eq. (2.141) with G^a from the left and G^r from the right, one obtains

$$G^r - G^a = G^a \left(\Sigma^r - \Sigma^a\right) G^r; \tag{2.142}$$

By multiplying Eq. (2.141) with G^r from the left and G^a from the right, one obtains

$$G^r - G^a = G^r \left(\Sigma^r - \Sigma^a\right) G^a. \tag{2.143}$$

(2) Prove that the integrand of Eq. (2.140) is zero after the summation over the lead index. Replacing Σ_β by Σ, the integrand can be simplified as

$$\begin{aligned}
&2\text{Re Tr}\left[G^r\Sigma^< + G^<\Sigma^a\right] \\
&= \text{Tr}\left[G^r\Sigma^< + G^<\Sigma^a + H.c.\right] \\
&= \text{Tr}\left[G^r\Sigma^< + G^<\Sigma^a - \Sigma^<G^a - \Sigma^rG^<\right] \\
&= \text{Tr}\left[G^r\Sigma^< + G^<\Sigma^a - G^a\Sigma^< - G^<\Sigma^r\right] \\
&= \text{Tr}\left[\left(G^r - G^a\right)\Sigma^< - G^<\left(\Sigma^r - \Sigma^a\right)\right] \\
&= \text{Tr}\left[G^a\left(\Sigma^r - \Sigma^a\right)G^r\Sigma^< - G^r\Sigma^<G^a\left(\Sigma^r - \Sigma^a\right)\right] \\
&= 0,
\end{aligned}$$

where Eq. (2.91) and Eq. (2.142) are used in the derivation.

QED.

An interesting inference from the current conservation is that the transmission coefficient $T_{LR}(E)$ defined by Eq. (2.106) is equal to $T_{RL}(E)$ defined by 2.108 in two-probe systems.

Secondly, we investigate a special type of lead in which the net current is always zero. This type of lead is called Büttiker probe which can simulate dephasing effect in the central region [22]. Although the net current is zero, electrons can still flow into and out of a Büttiker probe in a dynamic balance. As a result electrons may lose their phase memory due to inelastic collisions inside the Büttiker probe. For example, in Ref. [21], two phenomenological self-energies are proposed for the Büttiker probe

$$\Sigma_\alpha = \alpha \cdot G, \tag{2.144}$$

and

$$\Sigma_\gamma = \gamma \cdot diag\left[diag(G)\right], \tag{2.145}$$

www.ebook3000.com

where α and γ are constants. Below it is verified that the net current is indeed zero for the phenomenological self-energies.

Proof: (1) Inserting the self-energy Σ_α into Eq. (2.140), the integrand can be simplified as

$$\begin{aligned} & 2\text{Re Tr}\left[G^r\Sigma_\alpha^< + G^<\Sigma_\alpha^a\right] \\ \sim\ & 2\text{Re Tr}\left[G^rG^< + G^<G^a\right] \\ =\ & \text{Tr}\left[G^rG^< + G^<G^a + H.c.\right] \\ =\ & \text{Tr}\left[G^rG^< + G^<G^a - G^<G^a - G^rG^<\right] \\ =\ & 0. \end{aligned}$$

(2) Inserting the self-energy Σ_γ into Eq. (2.140), the integrand can be simplified as

$$\begin{aligned} & 2\text{Re Tr}\left[G^r\Sigma_\gamma^< + G^<\Sigma_\gamma^a\right] \\ \sim\ & 2\text{Re Tr}\left[diag\left(G^r\right)diag\left(G^<\right) + diag\left(G^<\right)diag\left(G^a\right)\right] \\ =\ & \text{Tr}\left[diag\left(G^r\right)diag\left(G^<\right) + diag\left(G^<\right)diag\left(G^a\right) + H.c\right] \\ =\ & \text{Tr}\left[diag\left(G^r\right)diag\left(G^<\right) + diag\left(G^<\right)diag\left(G^a\right) - diag\left(G^<\right)diag\left(G^a\right)\right. \\ & \left. - diag\left(G^r\right)diag\left(G^<\right)\right] \\ =\ & 0. \end{aligned}$$

QED.

Finally, we show that the nonequilibrium coherent potential $\tilde{\varepsilon}$ solved from the NECPA equations can be viewed as a Büttiker probe. Previously we have defined $\overline{G}$ as disorder averaged Green's function and regarded $\tilde{\varepsilon}$ as an effective medium. Now we change the point of view and regard $\overline{G}$ as the Green's function of a multi-probe system. In the multi-probe system, besides the left and right leads, there are many Büttiker probes connected to the disorder sites. Below it is verified that the net current is indeed zero in each Büttiker probe.

Proof: (1) The Büttiker probe connected to the ith disorder site has the self-energy $\Sigma_i = diag\left([0,\cdots,0,\tilde{\varepsilon}_i,0,\cdots,0]\right)$. Inserting Σ_i to Eq. (2.140), the integrand is obtained as

$$F_i \equiv 2\text{Re Tr}\left[\overline{G}\Sigma_i\right]^< = 2\text{Re Tr}\left[\overline{G_i}\tilde{\varepsilon}_i\right]^<, \tag{2.146}$$

where $\overline{G_i} \equiv \left[\overline{G}\right]_{ii}$. The goal is to prove that $F_i = 0$.

(2) Derive a useful relation Eq. (2.153). According to the complex-time NECPA Eqs. (2.132), the complex-time Green's function $\overline{G_i}$ and the

complex-time conditional Green's function $\overline{G_{iq}}$ satisfy

$$\overline{G_i} = \sum_q x_{iq} \overline{G_{iq}}, \tag{2.147}$$

$$\overline{G_i} = \left(E - \tilde{\varepsilon}_i - \Omega_i\right)^{-1}, \tag{2.148}$$

$$\overline{G_{iq}} = \left(E - \varepsilon_{iq} - \Omega_i\right)^{-1}. \tag{2.149}$$

By multiplying $\left(\overline{G_i}\right)^{-1}$ to Eq. (2.147) from the right, one obtains

$$\sum_q x_{iq} \overline{G_{iq}} \left(\overline{G_i}\right)^{-1} = 1. \tag{2.150}$$

By eliminating Ω_i in Eqs. (2.148,2.149), one obtains

$$\left(\overline{G_i}\right)^{-1} - \left(\overline{G_{iq}}\right)^{-1} = -\tilde{\varepsilon}_i + \varepsilon_{iq}. \tag{2.151}$$

By multiplying Eq. (2.151) with $\overline{G_{iq}}$ from the left, one obtains

$$\overline{G_{iq}} \left(\overline{G_i}\right)^{-1} - 1 = -\overline{G_{iq}}\tilde{\varepsilon}_i + \overline{G_{iq}}\varepsilon_{iq}. \tag{2.152}$$

By applying the weighted summation $\sum_q x_{iq}$ to Eq. (2.152) and using Eqs. (2.150,2.147), one obtains the useful relation

$$\overline{G_i}\tilde{\varepsilon}_i = \sum_q x_{iq} \overline{G_{iq}} \varepsilon_{iq}. \tag{2.153}$$

(3) Inserting Eq. (2.153) to Eq. 2.146), F_i can be simplified as

$$\begin{aligned}
F_i &= 2\text{Re Tr}\left[\overline{G_i}\tilde{\varepsilon}_i\right]^< \\
&= 2\text{Re Tr}\left[\sum_q x_{iq}\overline{G_{iq}}\varepsilon_{iq}\right]^< \\
&= \sum_q x_{iq} 2\text{Re Tr}\left[\overline{G_{iq}}\varepsilon_{iq}\right]^< \\
&= \sum_q x_{iq} \text{Tr}\left[\overline{G_{iq}^<}\varepsilon_{iq} + H.c.\right] \\
&= \sum_q x_{iq} \text{Tr}\left[\overline{G_{iq}^<}\varepsilon_{iq} - \varepsilon_{iq}\overline{G_{iq}^<}\right] \\
&= 0.
\end{aligned}$$

QED.

So we have proved that the current is conserved even in the presence of disorder scattering. The effect of disorder scattering is to break the phase coherence of electron waves. As an illustration, we investigate an

www.ebook3000.com

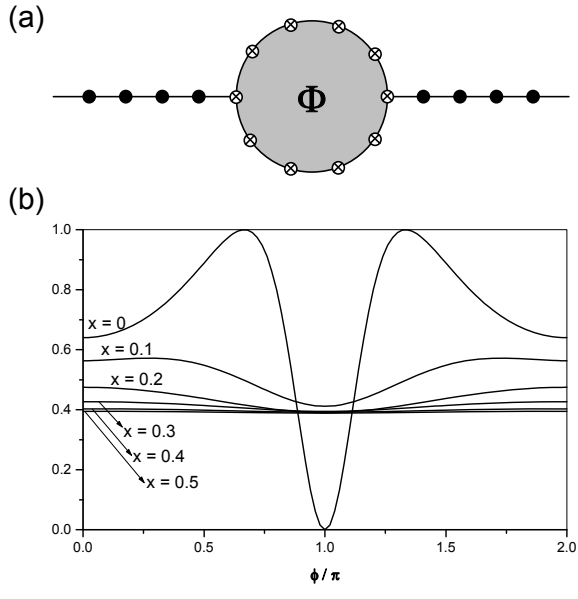


Fig. 2.6 (a) Aharonov–Bohm apparatus to illustrate the dephasing effect due to disorder scattering. The black dots and the crossed dots represent clean sites and disorder sites respectively. Φ is the magnetic flux which generates a phase shift ϕ when the electron wave travels one circle around the apparatus. (b) Conductance as a function of ϕ for different disorder concentration x. The clean sites have the on-site energy $\varepsilon_0 = 0$ and the nearest neighbor coupling $t = 1$. The disorder sites have the on-site energy $\varepsilon_1 = 0$ with probability $p_1 = 1 - x$ or $\varepsilon_2 = 0.5$ with probability $p_2 = x$.

Aharonov–Bohm apparatus whose arms are composed of random disorder sites (see Fig. 2.6). In the clean limit ($x = 0$), the conductance can reach a minimum of 0 and a maximum of 1 due to quantum interference. With the increase of disorder concentration, the averaged conductance is suppressed and the interference pattern disappears gradually. The reason is that disorder scattering not only reduces the amplitude of the scattered wave but also adds a random phase shift to the scattered wave. By averaging disorder configurations, the scattered wave loses its phase coherence.

To sum up, we have proved the current conservation in a multi-probe system and shown that the nonequilibrium coherent potential $\tilde{\varepsilon}$ can be regarded as the self-energy of a Büttiker probe. We would like to point out that the dephasing mechanism discussed in this section needs to be understood as an "effective dephasing". The phase coherence is lost either

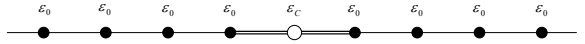


Fig. 2.7 A sketch of the toy model in which a scatterer ε_C is connected to two semi-infinite leads (see the text for details).

due to the average over disorder configurations or due to the partition of system and environment.

2.10 A toy model

The quantum transport theory in disordered two-probe systems has been presented in the previous sections. It will be instructive to apply the theory to a toy model which is sufficiently simple to solve but still captures some essential features. The toy model is sketched in Fig. 2.7 and is described as follows. The left and right leads are two semi-infinite tight-binding chains with on-site energy $\varepsilon_0 = 0$ and nearest neighbor coupling $t_0 = 1$. The central region is a single site with on-site energy ε_C and coupling t_C to both leads.

The two-probe Hamiltonian of the toy model is

$$H_\infty = \begin{pmatrix} \ddots & \ddots & & & & & & \\ \ddots & \varepsilon_0 & t_0 & & & & & \\ & t_0 & \varepsilon_0 & t_C & & & & \\ & & t_C & \varepsilon_C & t_C & & & \\ & & & t_C & \varepsilon_0 & t_0 & & \\ & & & & t_0 & \varepsilon_0 & \ddots & \\ & & & & & \ddots & \ddots & \end{pmatrix}, \tag{2.154}$$

which is an infinitely large matrix. The retarded Green's function of the central region is defined by

$$G^r(E) = \left[\left(E^+ - H_\infty\right)^{-1}\right]_{CC},$$

which is a complex number. By using the concept of self-energy, $G^r(E)$ can be reduced to

$$G^r(E) = \frac{1}{E - \varepsilon_C - \Sigma_L^r(E) - \Sigma_R^r(E)}, \tag{2.155}$$

where $\Sigma_\beta^r(E)$ $(\beta = L, R)$ is the lead self-energy.

www.ebook3000.com

The lead self-energy can be calculated analytically by solving the surface Green's function from Eq. (2.117)

$$g_\beta^r(E) = \left[E^+ - \varepsilon_0 - t_0^2 g_\beta^r(E)\right]^{-1}. \tag{2.156}$$

Eq. (2.156) is a quadratic equation and the two roots can be solved analytically. Notice that only one of the roots is $g_\beta^r(E)$ while the other one is $g_\beta^a(E)$. $g_\beta^r(E)$ and $\Sigma_\beta^r(E)$ are derived as

$$g_\beta^r(E) = \frac{1}{t_0}\xi\left(\frac{E^+ - \varepsilon_0}{t_0}\right), \tag{2.157}$$

$$\Sigma_\beta^r(E) = \frac{t_C^2}{t_0}\xi\left(\frac{E^+ - \varepsilon_0}{t_0}\right). \tag{2.158}$$

The dimensionless complex function $\xi(z)$ is defined by

$$\xi(z) = \frac{z - \mathrm{i}\sqrt{4 - z^2}}{2}, \tag{2.159}$$

where the branch cut of $\sqrt{z}$ is chosen as $\mathrm{Re}\sqrt{z} > 0$. On the real axis, $\xi(z)$ is reduced to

$$\xi\left(x^+\right) = \begin{cases} \frac{1}{2}\left(x - i\sqrt{4 - x^2}\right) & |x| < 2 \\ \frac{1}{2}\left(x - sgn(x)\sqrt{x^2 - 4}\right) & |x| > 2 \end{cases}. \tag{2.160}$$

With the expression of the lead self-energy, $G^r(E)$ is obtained as

$$G^r(E) = \frac{1}{E - \varepsilon_C - 2\frac{t_C^2}{t_0}\xi\left(\frac{E^+ - \varepsilon_0}{t_0}\right)}. \tag{2.161}$$

Before studying physical quantities, let us first take a look at the mathematical property of $G^r(E)$. Remember that the retarded Green's function is analytical in the upper half plane and all the singularities reside in the lower half plane. But where are the singularities located exactly in this toy model? Analysis of Eq. (2.161) reveals that there are two types of singularities: branch cuts and poles. The branch cuts come from the square root in the lead self-energies and the poles come from the roots of the denominator (see Appendix A.2 for details). Those singularities are located on the lower side of the real axis and can be mapped into the eigenvalues of the two-probe Hamiltonian H_∞: The branch cuts correspond to the bands of the leads and the poles correspond to the bound states of the central region. As a verification, we compare the eigenvalues of H_∞ to the singularities of $G^r(E)$ in Fig. 2.8. In the calculation, H_∞ is truncated to a finite sized Hamiltonian H_{M+1+M} where the semi-infinite leads are replaced by finite

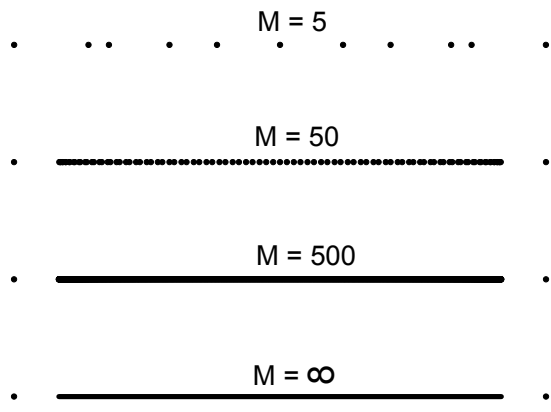


Fig. 2.8 The eigenvalues of truncated two-probe Hamiltonian ($M = 5, 50, 500$) and the singularities of retarded Green's function ($M = \infty$). The parameters of the toy model are: $\varepsilon_0 = 0$, $t_0 = 1$, $\varepsilon_C = 0$, $t_C = 1.5$.

leads with M sites. In the limit $M \to \infty$, the discrete eigenvalues indeed converge to the poles and the branch cuts of $G^r(E)$.

With the obtained $G^r(E)$ in Eq. (2.161), one can calculate the transmission coefficient and the density of states

$$T(E) = \Gamma_L(E)\,\Gamma_R(E)\,|G^r(E)|^2\,, \tag{2.162}$$

$$D(E) = -\frac{1}{\pi}\mathrm{Im}\; G^r(E)\,, \tag{2.163}$$

where the linewidth function $\Gamma_\beta(E)$ is obtained as

$$\Gamma_\beta(E) = -2\;\mathrm{Im}\Sigma^r_\beta(E) = 2\frac{t_C^2}{t_0}(-)\,\mathrm{Im}\left[\xi\left(\frac{E^+ - \varepsilon_0}{t_0}\right)\right]. \tag{2.164}$$

The behaviors of $T(E)$ and $D(E)$ in different parameter regimes are discussed in Appendix A.2.

So far we have studied the properties of a clean two-probe system. We are more interested in a disordered two-probe system and wish to check the accuracy of the NECPA theory. Since the NECPA is exact for a single disorder site, we consider a variation of the toy model in which the central region has two disorder sites (see the inset of Fig. 2.9). The Hamiltonian

www.ebook3000.com

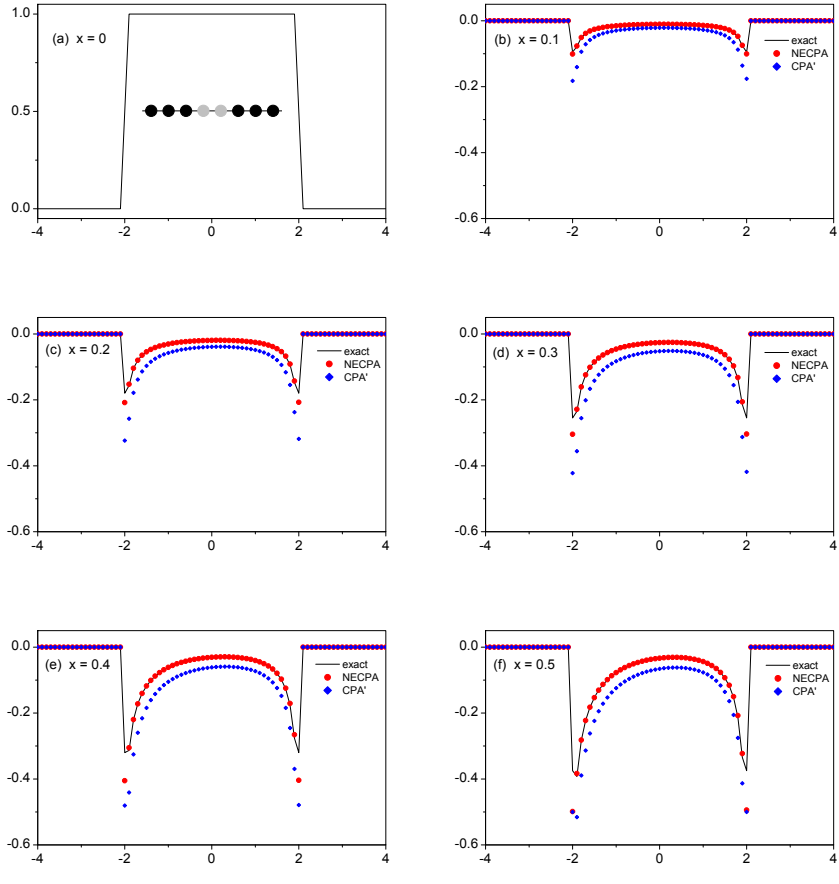


Fig. 2.9 Disorder averaged transmission coefficient $\overline{T(E)}$ in the modified toy model. (a) to (f) correspond to the disorder concentration $x = 0,\ 0.1,\ \cdots,\ 0.5$. Notice that the background transmission of $x = 0$ has been subtracted from (b) to (f) for clarity. The inset of (a) shows the modified toy model. The parameters of the modified toy model are: $\varepsilon_0 = 0$, $\varepsilon_A = 0$, $\varepsilon_B = 0.5$, $t_0 = 1$.

of the disordered two-probe system reads

$$H_\infty = \begin{pmatrix} \ddots & \ddots & & & & & & \\ \ddots & \varepsilon_0 & t_0 & & & & & \\ & t_0 & \varepsilon_0 & t_0 & & & & \\ & & t_0 & \varepsilon_1 & t_0 & & & \\ & & & t_0 & \varepsilon_2 & t_0 & & \\ & & & & t_0 & \varepsilon_0 & t_0 & \\ & & & & & t_0 & \varepsilon_0 & \ddots \\ & & & & & & \ddots & \ddots \end{pmatrix}, \tag{2.165}$$

where the on-site energies ε_1 and ε_2 are random variables taking values ε_A or ε_B with probability $1-x$ or x respectively. Thanks to the simplicity of the model, the disorder-average can be done *exactly* by an enumeration of disorder configurations (AA, AB, BA, BB). For comparison, we calculate the disorder averaged transmission coefficient $\overline{T(E)}$ by using both the NECPA equations and the exact averaging.

Fig. 2.9 shows $\overline{T(E)}$ at various disorder concentrations x calculated with three methods: the exact solution, the NECPA method, and the CPA' method. Here the CPA' method is to neglect the lesser coherent potential $\tilde{\varepsilon}^<$ in the NECPA method. In the clean limit $x = 0$ (Fig. 2.9a), $\overline{T(E)}$ has integer steps which coincides with the conducting channel number as expected. With increasing disorder concentration, the sharp steps become milder and milder due to the disorder scattering. To emphasize the disorder effect, the background transmission $T_0(E)$ of $x = 0$ has been subtracted from $\overline{T(E)}$ of $x > 0$. In Fig. 2.9b to Fig. 2.9f, results from the NECPA method agree very well with the exact solution in a full range of disorder concentration, providing a verification of the NECPA theory. In contrast, the results of CPA' method has a noticeable deviation from the exact solution, indicating the necessity of generalizing the CPA theory to the NECPA theory.

To sum up, we have investigated a toy model of a two-probe system which is analytically solvable. By playing with the toy model, we have demonstrated some essential features of the two-probe Hamiltonian, the lead self-energy, the Green's function, and the physical quantities. By comparing it to the exact solution, we have also verified the accuracy of the NECPA theory in the toy model. In the next chapter, we shall combine atomistic modeling with the NECPA theory and extend the toy model to simulate a realistic electronic device.

www.ebook3000.com

Bibliography

[1] H. Haug and A. -P. Jauho, *Quantum Kinetics in Transport and Optics of Semiconductor* (Springer-Verlag, Berlin, 1996).

[2] M. Gell-Mann and F. Low, Phys. Rev. **84**, 350 (1951).

[3] D. C. Langreth, *in Linear and Nonlinear Electron Transport in Solids*, NATO Advanced Study Institute, Series B: Physics, Vol. 17, edited by J. T. Devreese and V. E. van Doren (Plenum Press, New York, 1976).

[4] In the time domain decomposition, the time variables can be viewed as matrix indices and hence the integral over the inner time variable can also be viewed as matrix multiplication over the inner time index. See Y. Zhu, J. Maciejko, T. Ji, H. Guo and J. Wang, Phys. Rev. B **71**, 075317 (2004).

[5] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B **63**, 121104(R) (2001).

[6] Y. Zhu, Q. F. Sun, T. H. Lin, Phys. Rev. B **66**, 085306 (2002).

[7] There are two exceptions to the argument: (1) At bound states of a two-probe system, Γ is also infinitesimal. (2) If a two-probe system is sufficiently large, many small terms may add up to a finite contribution. To safely drop the first term of the Keldysh equation, one needs to adopt zero imaginary energy in the central region ($\delta = 0$) and small but finite imaginary energy in the leads.

[8] J. Wang and H. Guo, Phys. Rev. B **79**, 045119 (2009).

[9] A. P. Jauho, N. S. Wingreen, Y. Meir, Phys. Rev. B **50**, 5528 (1994).

[10] Q. F. Sun, B. G. Wang, J. Wang, T. H. Lin, Phys. Rev. B **61**, 4754 (2000).

[11] Y. Zhu, L. Liu, H. Guo, Phys. Rev. B **88**, 205415 (2013).

[12] B. Velický, Phys. Rev. **184**, 614 (1969).

[13] K. Carva, I. Turek, J. Kudrnovský, O. Bengone, Phys. Rev. B **73**, 144421 (2006).

[14] Y. Ke, K. Xia, H. Guo, Phys. Rev. Lett. **100**, 166805 (2008).

[15] A. V. Kalitsov, M. G. Chshiev, J. P. Velev, Phys. Rev. B **85**, 235111 (2012).

[16] P. Soven, Phys. Rev. **156**, 809 (1967).

[17] D. W. Taylor, Phys. Rev. **156**, 1017 (1967).

[18] We only consider the situation in which disorder sites are distributed in a finite region. If disorder sites extend to infinity in a lead, scattering waves won't be able to propagate along the lead.

[19] Y. Zhu, L. Liu, H. Guo, Phys. Rev. B **88**, 085420 (2013).

[20] Y. Zhu, L. Liu, H. Guo, unpublished (2013).

[21] R. Golizadeh-Mojarad and S. Datta, Phys. Rev. B **75**, 081301 (2007).

[22] M. Büttiker, Phys. Rev. B **33**, 3020 (1986); IBM J. Res. Dev. **32**, 72 (1988).

Chapter 3

The NECPA-LMTO method

In Chapter 2, we have developed the NECPA theory for the quantum transport in disordered two-probe systems. In the derivation of the formalism, it is assumed that the Hamiltonian is known *a priori*. For example, in the toy model of Section 2.9, the parameters are chosen as $\varepsilon_0 = 0$ and $t_0 = 1$. But those numbers are somehow arbitrary and just for the purpose of illustration. How do we include the atomic information in the Hamiltonian, e.g., assigning this site to be a Co atom and that site to be a Cu atom? This topic will be the focus of this chapter where an implementation of DFT, the LMTO method, is combined with the NECPA theory to solve a two-probe Hamiltonian self-consistently. Consequently the NECPA-DFT theory is reduced to the NECPA-LMTO method in this specific implementation. We shall see that the Hamiltonian elements in Eq. (2.1) are replaced by matrix blocks to represent different atoms. At the end of the chapter, a flowchart and a complete set of formulas will be provided, serving as a blueprint for the numerical implementations.

3.1 Kohn–Sham equation

Density functional theory (DFT) is the theoretical ground of electronic structure calculations in solid state physics. It has been successfully applied to study the electronic structure of crystals, surfaces, interfaces, etc. In this chapter we shall combine the DFT with the NECPA theory to study quantum transport in disordered two-probe systems. Our starting point is the Kohn–Sham equation [1] of DFT.

The Kohn–Sham equation aims to solve the interacting electron gas in an external field, and the equation looks exactly the same as the Schrödinger

www.ebook3000.com

equation (in atomic units)

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}\left(\mathbf{r}\right)\right]\Psi_i\left(\mathbf{r}\right) = E_i\Psi_i\left(\mathbf{r}\right), \tag{3.1}$$

in which the effective potential V_{eff} is defined by

$$V_{eff}\left(\mathbf{r}\right) = V_{ext}\left(\mathbf{r}\right) + \int d\mathbf{r}' \frac{\rho\left(\mathbf{r}'\right)}{\left|\mathbf{r}-\mathbf{r}'\right|} + V_{XC}\left[\rho\left(\mathbf{r}\right)\right]. \tag{3.2}$$

Here V_{eff} has three terms: the external potential V_{ext}, the Hartree potential, and the exchange-correlation potential V_{XC}. The external potential is the potential of external field applied to the electron gas. The Hartree potential is the Coulomb potential generated by the electron charge density $\rho\left(\mathbf{r}\right)$. The exchange-correlation potential in the soul of DFT and deserves further discussion below. Remember in the original problem we wish to solve the interacting electron gas. The Kohn–Sham equation says that the many-body problem can be reduced to an effective single-particle problem. One just needs to solve a Schrödinger-like equation with an effective potential in which the electron-electron Coulomb interaction is treated in a mean-field manner. All the complexities are hidden in the exchange-correlation potential V_{XC} which considers the effects beyond the mean-field picture and other non-classical corrections. We won't go to the details of the derivation of V_{XC}; instead we provide some explicit expressions of V_{XC} in Appendix A.9.

$\rho\left(\mathbf{r}\right)$ is the total electron charge density which can be constructed with the solutions of the Kohn–Sham equation

$$\rho\left(\mathbf{r}\right) = \sum_i n_i \left|\Psi_i\left(\mathbf{r}\right)\right|^2, \tag{3.3}$$

where n_i is the occupation number of the Kohn–Sham state $\Psi_i\left(\mathbf{r}\right)$. In equilibrium the occupation of the states is determined by Fermi-Dirac statistics and hence $n_i = f\left(E_i\right)$ where f is the Fermi function. In nonequilibrium the occupation of the states is determined by the steady state condition which will be discussed later. Since $\Psi_i\left(\mathbf{r}\right)$ appears in $\rho\left(\mathbf{r}\right)$ which determines $V_{eff}\left(\mathbf{r}\right)$, the Kohn–Sham equation (3.1) is actually a nonlinear equation which needs to be solved self-consistently. Namely one can make an initial guess of $\rho\left(\mathbf{r}\right)$, calculate $V_{eff}\left(\mathbf{r}\right)$ with Eq. (3.2), solve $\Psi_i\left(\mathbf{r}\right)$ with Eq. (3.1), and update $\rho\left(\mathbf{r}\right)$ with Eq. (3.3). The iteration continues until $\rho\left(\mathbf{r}\right)$ does not change any more.

One can solve the Kohn–Sham equation in the real space as presented in Eq. (3.1), or in the reciprocal space by taking a Fourier transform of

Eq. (3.1), or in any space of a complete basis set. Here we choose to work in the atomic orbital space where $\Psi_i(\mathbf{r})$ is expanded in terms of atomic orbitals. Since the atomic orbital is already an approximate solution of the Kohn–Sham equation in a single atom, the solution of the whole system can be constructed from a linear combination of just a few atomic orbitals. For example, to solve a Cu bulk system, more than 10^3 real-space points are needed for each Cu atom, while 9 atomic orbitals (1 s-orbital, 3 p-orbitals, 5 d-orbitals) are sufficient to describe a Cu atom in the LMTO method. The gain is obvious. More importantly the atomic orbital is localized around the atom center and hence is a natural choice to solve two-probe systems which conceptually rely on the partition of the leads and the central region. In contrast, other popular basis set such as the plane wave is not specially localized and hence is not directly applicable to study quantum transport.

There are still many choices for the type of atomic orbitals, e.g., Gaussian basis, fire-ball basis, etc. In this monograph, we shall adopt the linear muffin-tin orbitals (LMTO) as our basis set [2–4]. Compared to other atomic orbitals, the LMTO method has following advantages: (1) In the LMTO method, the atomic information only appears in the diagonal elements of Hamiltonian and hence is compatible with the NECPA theory. (2) In the LMTO method, the orbitals are solved dynamically in each self-consistent iteration and hence is more adaptive and accurate than the static atomic orbitals. (3) In the LMTO method, by using the screening transform the effective orbital length is much shorter than other basis sets and hence the computational cost is much lower. The major drawback of the LMTO method is that it works only with close-packed structures such as FCC, BCC, and HCP. To apply the LMTO method to non-close-packed structures such as diamond crystal, graphene sheet, and interface of two crystals, one has to fill the vacuum with proper empty spheres. Fortunately the sphere filling schemes are available in the literature for most non-close-packed structures and some of the schemes are provided in Appendix A.17. The rest of this chapter will focus on the LMTO method, and some contents are adopted from Ref. [4].

3.2 Muffin-tin orbital

In solid-state materials, the potential is oscillatory in the vicinity of atomic nucleus and flat in the interstitial region. It is natural to divide the space into many non-overlapping atomic spheres surrounding the nucleus plus the interstitial region. The potential inside the atomic sphere is assumed

www.ebook3000.com

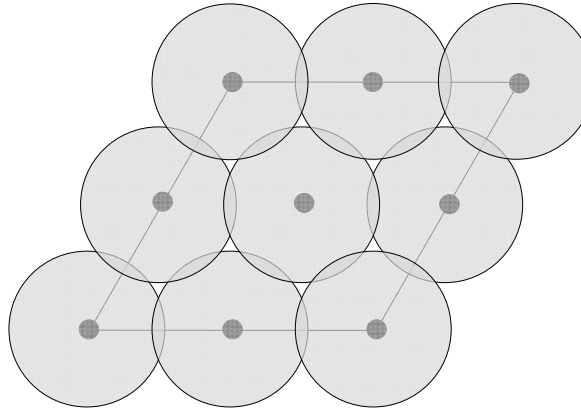


Fig. 3.1 Schematic diagram to illustrate the atomic sphere approximation in the LMTO method. A parallelogram unit cell is approximated by slightly overlapped atomic spheres, and the total volume of the atomic spheres is equal to that of the unit cell.

to be spherically symmetric and the potential in the interstitial region is assumed to be a constant. Such an approximated potential is referred to as the muffin-tin potential. A further approximation to the muffin-tin potential is to inflate the atomic spheres so that they "fill" the whole space of the unit cell, $\sum_i \frac{4\pi}{3} R_i^3 = V_{cell}$, where R_i is the atomic sphere radius and V_{cell} is the unit cell volume (see Fig. 3.1). Since a unit cell cannot be filled completely with spheres, the atomic spheres must overlap a little bit to satisfy the volume constraint. The idea of replacing a unit cell by slightly overlapped spheres is called atomic sphere approximation (ASA).

We first focus on one particular atomic sphere. Inside the sphere, one can solve the Kohn–Sham equation for a given energy E

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}(r) - E\right]\varphi(\mathbf{r}, E) = 0. \tag{3.4}$$

The solution to a spherically symmetric potential can be obtained by separating the radial part and the angular part

$$\varphi(\mathbf{r}, E) = \varphi(r, E)\, Y_L(\Omega).$$

$\varphi(r, E)$ satisfies the radial equation with a physical boundary condition

$$\left[-\frac{1}{2}\partial_r^2 + \frac{1}{r}\partial_r + \frac{l(l+1)}{2r^2} + V_{eff}(r) - E\right]\varphi(r, E) = 0, \tag{3.5}$$

$$\varphi(r \to 0, E) \sim r^l, \tag{3.6}$$

where l is the angular momentum quantum number. $\varphi(r,E)$ is normalized such that $\int_0^R \varphi^2(r,E)\, r^2 dr = 1$ where R is the atomic sphere radius. $Y_L(\Omega)$ is the real-valued spherical harmonics in which $\Omega = (\theta,\phi)$ and $L = (l,m)$ are the solid angle and the corresponding angular momentum quantum numbers. Notice that real-valued spherical harmonics $Y_L(\Omega)$ is related to the conventional spherical harmonics $\tilde{Y}_L(\Omega)$ by a unitary transform

$$Y_L(\Omega) = \begin{cases} \frac{(-1)^m}{\sqrt{2}}\left[\tilde{Y}_L(\Omega) + \tilde{Y}_L^*(\Omega)\right] & m > 0, \\ \tilde{Y}_L(\Omega) & m = 0, \\ \frac{\mathrm{i}}{\sqrt{2}}\left[\tilde{Y}_L(\Omega) - \tilde{Y}_L^*(\Omega)\right] & m < 0. \end{cases} \tag{3.7}$$

Notice that the set of real-valued spherical harmonics is orthonormal, i.e., $\int Y_L(\Omega)\, Y_{L'}(\Omega)\, d\Omega = \delta_{LL'}$. The analytical expressions of $Y_L(\Omega)$ and $\tilde{Y}_L(\Omega)$ for $l = 0,1,2$ are provided in Appendix A.10.

Outside atomic spheres, V_{eff} is a constant V_0. It is further assumed that $E = V_0$ in the LMTO method. So the Kohn–Sham equation is reduced to the Laplace equation

$$\nabla^2 \varphi(\mathbf{r}) = 0,$$

which has two independent solutions $K_L(\mathbf{r})$ and $J_L(\mathbf{r})$,

$$\begin{aligned} K_L(\mathbf{r}) &= K_l(r)\, Y_L(\Omega), \\ K_l(r) &\equiv \left(\frac{\omega}{r}\right)^{l+1}, \end{aligned} \tag{3.8}$$

$$\begin{aligned} J_L(\mathbf{r}) &= J_l(r)\, Y_L(\Omega), \\ J_l(r) &\equiv \frac{1}{2(2l+1)}\left(\frac{r}{\omega}\right)^{l}, \end{aligned} \tag{3.9}$$

where ω is the Wigner–Seitz radius (see Section 3.13.5). To avoid the divergence in the limit of $r \to \infty$, only the irregular solution $K_L(\mathbf{r})$ is adopted outside the atomic spheres.

Now we construct the muffin-tin orbital $\Psi(\mathbf{r},E)$ with $\varphi(\mathbf{r},E)$, $K_L(\mathbf{r})$, and $J_L(\mathbf{r})$: The orbital head inside the atomic sphere is a linear combination of $\varphi(\mathbf{r},E)$ and $J_L(\mathbf{r})$; The orbital tail outside the atomic sphere is just $K_L(\mathbf{r})$

$$\Psi_L(\mathbf{r},E) = \begin{cases} N_l(E)\,\varphi_L(\mathbf{r},E) + P_l(E)\, J_L(\mathbf{r}) & r < R \\ K_L(\mathbf{r}) & r > R \end{cases}. \tag{3.10}$$

To smoothly connect the head and the tail, the wave function is required to be continuous in the function value as well the first derivative at $r = R$.

www.ebook3000.com

The coefficients $N_l(E)$ and $P_l(E)$ are obtained as

$$N_l(E) \equiv \frac{\omega}{2}\frac{1}{\{\varphi_l, J_l\}}, \tag{3.11}$$

$$P_l(E) \equiv \frac{\{K_l, \varphi_l\}}{\{J_l, \varphi_l\}}, \tag{3.12}$$

where $\{\varphi_1, \varphi_2\}$ is the Wronskian of two radial functions at $r = R$, which is defined by

$$\{\varphi_1(r), \varphi_2(r)\} = \left\{r^2\left[\varphi_1(r)\varphi_2'(r) - \varphi_1'(r)\varphi_2(r)\right]\right\}_{r=R}. \tag{3.13}$$

Notice that the Wronskian of J_l and K_l is a constant

$$\{J_l(r), K_l(r)\} = -\frac{\omega}{2}. \tag{3.14}$$

To sum up, Eq. (3.10) defines the muffin-tin orbital (MTO) and is the central result of this section. The MTO will be used to solve the Kohn–Sham equation in Section 3.4.

3.3 Structure constant

There is a useful mathematical theorem to expand K_L of one center into a series of J_L of another center [2, 3], and the coefficient of the expansion is called structure constant. In this section, the MTO will be rewritten to an equivalent form by using the structure constant.

Theorem: Consider two centers $\mathbf{r}_{i_1}$ and $\mathbf{r}_{i_2}$. For $\mathbf{r}$ in the vicinity of $\mathbf{r}_{i_2}$, K_{L_1} can be expanded as

$$K_{L_1}(\mathbf{r} - \mathbf{r}_{i_1}) = -\sum_{L_2} S_{i_1 L_1, i_2 L_2} J_{L_2}(\mathbf{r} - \mathbf{r}_{i_2}), \tag{3.15}$$

where $|\mathbf{r} - \mathbf{r}_{i_2}| < |\mathbf{r}_{i_1} - \mathbf{r}_{i_2}|$. The structure constant S is defined by

$$S_{i_1 L_1, i_2 L_2} \equiv \sum_m (-1)^{l_2+1} \frac{8\pi(2l-1)!!}{(2l_1-1)!!(2l_2-1)!!} C_{LL_1L_2} K_L(\mathbf{r}_{i_2} - \mathbf{r}_{i_1}), \tag{3.16}$$

where $l = l_1 + l_2$ and $m = -l, \cdots, l$. $C_{LL_1L_2}$ is the Gaunt coefficient defined by

$$C_{LL_1L_2} \equiv \int Y_L(\Omega) Y_{L_1}(\Omega) Y_{L_2}(\Omega)\, d\Omega. \tag{3.17}$$

We won't present the proof of the theorem but refer interested readers to Ref. [2, 3]. Below we shall discuss the mathematical properties of the structure constant and apply the theorem to the muffin-tin orbital.

By the definition Eq. (3.16), the structure constant S has following properties: (1) S is symmetric with respect to the subscript i_1L_1 and i_2L_2, namely

$$S_{i_1L_1,i_2L_2} = S_{i_2L_2,i_1L_1}.$$

So the expansion in Eq. (3.15) can also be written as

$$K_{L_1}\left(\mathbf{r}-\mathbf{r}_{i_1}\right) = -\sum_{L_2} J_{L_2}\left(\mathbf{r}-\mathbf{r}_{i_2}\right) S_{i_2L_2,i_1L_1}. \tag{3.18}$$

(2) S is uniquely determined by the positions of the expansion centers. Given an atomic structure (e.g., FCC, BCC), one can calculate the matrix elements of S without knowing any atomic information. That is why S is called the structure constant. (3) The elements of S decay with the distance in a power law

$$S_{i_1L_1,i_2L_2} \sim \left(\frac{\omega}{|\mathbf{r}_{i_1}-\mathbf{r}_{i_2}|}\right)^{l_1+l_2+1}.$$

In Section 3.7, by using the screening transform, the decay can be accelerated from the power law to the exponential law.

Now we apply the theorem Eq. (3.18) to the muffin-tin orbital defined by Eq. (3.10). Recall that in the ASA the real space is approximated by many slightly overlapped spheres (see Fig. 3.1). Inside the i-th sphere, Ψ_{iL} is a linear combination of φ_{iL} and J_L. Outside the i-th sphere, Ψ_{iL} is just K_L which can be expanded into a series of $J_{L'}$ of the i'-th atomic sphere ($i' \neq i$)

$$\Psi_{iL}\left(\mathbf{r},E\right) = \begin{cases} N_{il}\left(E\right)\varphi_{iL}\left(\mathbf{r}-\mathbf{r}_i,E\right) + P_{il}\left(E\right)J_L\left(\mathbf{r}-\mathbf{r}_i\right) & |\mathbf{r}-\mathbf{r}_i| < R_i \\ -\sum_{L'} J_{L'}\left(\mathbf{r}-\mathbf{r}_{i'}\right) S_{i'L',iL} & |\mathbf{r}-\mathbf{r}_{i'}| < R_{i'} \end{cases}. \tag{3.19}$$

3.4 Tail cancelation

In this section, we shall solve the Kohn–Sham equation (3.4) with the muffin-tin orbitals derived in Eq. (3.19). In an atomic orbital space, the wave function will be expanded into a linear combination of atomic orbitals and the differential equation will be reduced to a nonlinear eigenvalue problem.

In the muffin-tin orbital space, the Kohn–Sham wave function $\Psi\left(\mathbf{r}\right)$ can be expanded as

$$\Psi\left(\mathbf{r}\right) = \sum_{iL} \Psi_{iL}\left(\mathbf{r},E\right)\xi_{iL}, \tag{3.20}$$

www.ebook3000.com

where ξ_{iL} is the coefficient. By inserting muffin-tin orbital Eq. (3.19) into Eq. (3.20), the wave function $\Psi(\mathbf{r})$ inside the i-th atomic sphere ($|\mathbf{r}-\mathbf{r}_i| < R_i$) is obtained as

$$\Psi(\mathbf{r}) = \sum_L \left[N_{il}(E)\,\varphi_{iL}(\mathbf{r}-\mathbf{r}_i, E) + P_{il}(E)\,J_L(\mathbf{r}-\mathbf{r}_i)\right]\xi_{iL} - \sum_{i'\neq i}\sum_{L'} J_L(\mathbf{r}-\mathbf{r}_i)\,S_{iL,i'L'}\xi_{i'L'}, \tag{3.21}$$

where the first term comes from the head of the i-th atomic sphere and the second term comes from the tails of other atomic spheres. By inserting the wave function (3.21) into the Kohn–Sham equation (3.4), one obtains

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}(|\mathbf{r}-\mathbf{r}_i|) - E\right]\Psi(\mathbf{r}) = 0.$$

Since $\varphi_{iL}(\mathbf{r}-\mathbf{r}_i, E)$ is already a solution of the Kohn–Sham equation, all the J_L terms must cancel with each other, leading to

$$\sum_{i'L'}\left[P_{il}(E)\,\delta_{ii'}\delta_{LL'} - S_{iL,i'L'}\right]\xi_{i'L'} = 0. \tag{3.22}$$

Notice that the diagonal block of the structure constant has been assigned to zero, i.e., $S_{iL,iL'} \equiv 0$, to take into account the constraint $i' \neq i$. Eq. (3.22) can be written in a compact form

$$\left[P(E) - S\right]\xi = 0, \tag{3.23}$$

where the potential function $P(E)$ is defined by

$$\left[P(E)\right]_{i_1L_1,i_2L_2} = \delta_{i_1i_2}\delta_{L_1L_2}P_{i_1l_1}(E).$$

Eq. (3.23) is referred to as the KKR-ASA equation and is the central result of this section. In Eq. (3.23), the first term is a diagonal matrix and contains all the atomic information, and the second term has nonzero off-diagonal elements and contains all the geometric information. We shall see in Section 3.10 that this feature is essential in applying the NECPA theory.

3.5 Energy linearization

The KKR-ASA equation (3.23) defines a nonlinear eigenvalue problem which is much more difficult to solve than linear eigenvalue problems. In this section, we shall reduce the nonlinear eigenvalue problem to a linear eigenvalue problem by using the energy linearization of MTO.

Suppose $\Psi_{iL}(\mathbf{r}, E)$ can be expanded linearly around an energy center E_{il}^0

$$\Psi_{il}(r, E) = \phi_{il}(r) + \dot{\phi}_{il}(r)\left(E - E_{il}^0\right) + \cdots \tag{3.24}$$

where $\dot{\phi}$ is the energy derivative $\frac{\partial}{\partial E}\left[\Psi_{il}(r, E)\right]$ at $E = E_{il}^0$. As a result the Wronskians in Eq. (3.12) can be approximated by

$$\left\{K_l(r), \Psi_{il}(r, E)\right\} \approx \left\{K_l(r), \phi_{il}(r)\right\} + \left\{K_l(r), \dot{\phi}_{il}(r)\right\}\left(E - E_{il}^0\right), \tag{3.25}$$

$$\left\{J_l(r), \Psi_{il}(r, E)\right\} \approx \left\{J_l(r), \phi_{il}(r)\right\} + \left\{J_l(r), \dot{\phi}_{il}(r)\right\}\left(E - E_{il}^0\right). \tag{3.26}$$

By inserting Eqs. (3.25,3.26) into Eq. (3.12), $P(E)$ can be obtained as

$$P(E) = \frac{E - C}{\Delta + \gamma(E - C)}, \tag{3.27}$$

where C, Δ, γ are diagonal matrices defined by

$$\begin{aligned} C_{i_1L_1,i_2L_2} &= \delta_{i_1L_1,i_2L_2} C_{il_1}, \\ \Delta_{i_1L_1,i_2L_2} &= \delta_{i_1L_1,i_2L_2} \Delta_{il_1}, \\ \gamma_{i_1L_1,i_2L_2} &= \delta_{i_1L_1,i_2L_2} \gamma_{il_1}, \end{aligned}$$

and the diagonal elements are

$$C_{il} = E_{il}^0 - \frac{\left\{K_l(r), \phi_{il}(r)\right\}}{\left\{K_l(r), \dot{\phi}_{il}(r)\right\}}, \tag{3.28}$$

$$\sqrt{\Delta_{il}} = \frac{\sqrt{\omega}}{\left\{K_l(r), \dot{\phi}_{il}(r)\right\}}, \tag{3.29}$$

$$\gamma_{il} = \frac{\left\{J_l(r), \dot{\phi}_{il}(r)\right\}}{\left\{K_l(r), \dot{\phi}_{il}(r)\right\}}. \tag{3.30}$$

Here $\{C_{il}\}$, $\{\Delta_{il}\}$, $\{\gamma_{il}\}$ are called potential parameters which uniquely determine the potential function $P(E)$ for a given E. Notice that in the derivation of the potential parameter Δ_{il}, we have used the following Wronskian identity

$$\left\{J_l(r), \phi_{il}(r)\right\}\left\{K_l(r), \dot{\phi}_{il}(r)\right\} - \left\{J_l(r), \dot{\phi}_{il}(r)\right\}\left\{K_l(r), \phi_{il}(r)\right\} = \omega, \tag{3.31}$$

of which the proof is provided in Appendix A.13.

By using the energy linearization of MTO, the nonlinear eigenvalue problem of Eq. (3.23) can be reduced to a linear eigenvalue problem

$$\left(E - H_{orth}\right)\tilde{\xi} = 0, \tag{3.32}$$

www.ebook3000.com

where the orthogonal Hamiltonian H_{orth} is defined by

$$H_{orth} = C + \sqrt{\Delta}\left(S^{-1} - \gamma\right)^{-1}\sqrt{\Delta}. \tag{3.33}$$

Proof: By using Eq. (3.27), $P(E) - S$ in Eq. (3.23) can be factorized as follows

$$\begin{aligned} P(E) - S &= \frac{E - C}{\Delta + \gamma(E - C)} - S \\ &= \left[\Delta + \gamma(E - C)\right]^{-1}\left[E - C - \Delta S - \gamma(E - C)S\right] \\ &= \left[\Delta + \gamma(E - C)\right]^{-1}\left[(E - C)(1 - \gamma S) - \Delta S\right] \\ &= \left[\Delta + \gamma(E - C)\right]^{-1}\sqrt{\Delta}\left[(E - C)\frac{1}{\sqrt{\Delta}}(1 - \gamma S) - \sqrt{\Delta}S\right] \\ &= \left[\Delta + \gamma(E - C)\right]^{-1}\sqrt{\Delta}\left[(E - C) - \sqrt{\Delta}S(1 - \gamma S)^{-1}\sqrt{\Delta}\right] \\ &\quad \times \frac{1}{\sqrt{\Delta}}(1 - \gamma S) \\ &= \left[\Delta + \gamma(E - C)\right]^{-1}\sqrt{\Delta}\left[E - H_{orth}\right]\frac{1}{\sqrt{\Delta}}(1 - \gamma S), \end{aligned} \tag{3.34}$$

where C, Δ, γ are diagonal matrices and hence are commutative. By eliminating the prefactor $\left[\Delta + \gamma(E - C)\right]^{-1}\sqrt{\Delta}$ in Eq. (3.34) and replacing ξ by $\tilde{\xi} \equiv \frac{1}{\sqrt{\Delta}}(1 - \gamma S)\xi$, Eq. (3.23) is reduced to Eq. (3.32). QED.

To sum up, by using the energy linearization of MTO, the potential function $P(E)$ is reduced to Eq. (3.27), and the KKR-ASA equation is reduced to a linear eigenvalue problem Eq. (3.32).

3.6 LMTO Green's function

In this section, we shall discuss the Green's functions in the LMTO method. We shall define two types of Green's functions, physical Green's function and auxiliary Green's function, and establish the relation between them.

We first define the physical Green's function. In analogy to Eq. (2.55), we define the retarded Green's function by using the Hamiltonian H_{orth} (Eq. (3.36))

$$G^r(E) \equiv \left(E^+ - H_{orth}\right)^{-1}, \tag{3.35}$$

where H_{orth} is defined by

$$H_{orth} = C + \sqrt{\Delta}\left(S^{-1} - \gamma\right)^{-1}\sqrt{\Delta}. \tag{3.36}$$

The Green's function defined by Eq. (3.35) is called physical Green's function because all physical quantities can be expressed in terms of G^r.

However, G^r is not compatible with the NECPA theory presented in Chapter 2.

To carry out disorder average, it is necessary to define the retarded Green's function based on the KKR-ASA equation

$$\mathcal{G}^r(E) \equiv \left[P\left(E^+\right) - S\right]^{-1}, \tag{3.37}$$

where $P(E)$ is defined by

$$P(E) \equiv \frac{E - C}{\Delta + \gamma(E - C)}. \tag{3.38}$$

The Green's function defined by Eq. (3.37) is called auxiliary Green's function which acts as a true Green's function in the NECPA-LMTO formalism (see Section 3.10). Notice that P only has diagonal elements and contains the atomic information, while S has off-diagonal elements and contains the geometric information. This nice property makes $\mathcal{G}^r$ compatible with the NECPA theory.

The connection from the physical Green's function to the auxiliary Green's function is made by the following relation

$$G^r(E) = \lambda(E) + \mu(E)\,\mathcal{G}^r(E)\,\mu(E), \tag{3.39}$$

where λ and μ are defined by

$$\lambda(E) \equiv \frac{\gamma}{\Delta + \gamma(E - C)}, \tag{3.40}$$

$$\mu(E) \equiv \frac{\sqrt{\Delta}}{\Delta + \gamma(E - C)}. \tag{3.41}$$

Proof: The ingredients of G^r and $\mathcal{G}^r$ have already appeared in Eq. (3.34). By inverting Eq. (3.34), we can derive the relation between them

$$\begin{aligned} G^r(E) &= \left(E^+ - H_{orth}\right)^{-1} \\ &= \frac{1}{\sqrt{\Delta}}(1 - \gamma S)\left[P\left(E^+\right) - S\right]^{-1}\frac{\sqrt{\Delta}}{\Delta + \gamma(E - C)} \\ &= \frac{1}{\sqrt{\Delta}}(1 - \gamma S)\left[P\left(E^+\right) - S\right]^{-1}\mu(E) \\ &= \frac{1}{\sqrt{\Delta}}(1 - \gamma S)\,\mathcal{G}^r(E)\,\mu(E). \end{aligned} \tag{3.42}$$

Comparing Eq. (3.42) to Eq. (3.39), we need to prove that

$$\frac{1}{\sqrt{\Delta}}(1 - \gamma S) = \mu(E) + \frac{\lambda(E)}{\mu(E)}\left[\mathcal{G}^r(E)\right]^{-1}. \tag{3.43}$$

www.ebook3000.com

To proceed, inserting the definitions of μ, λ, P and $\mathcal{G}^r$, Eqs. (3.41,3.40,3.38,3.37), into the RHS of Eq. (3.43)

$$\begin{aligned} RHS &= \mu(E) + \frac{\lambda(E)}{\mu(E)}\left[P\left(E^+\right) - S\right] \\ &= \frac{\sqrt{\Delta}}{\Delta+\gamma(E-C)} + \frac{\gamma}{\sqrt{\Delta}}\left[\frac{E-C}{\Delta+\gamma(E-C)} - S\right] \\ &= \frac{\sqrt{\Delta}}{\Delta+\gamma(E-C)} + \frac{1}{\sqrt{\Delta}}\left[\frac{\gamma(E-C)}{\Delta+\gamma(E-C)} - \gamma S\right] \\ &= \frac{\sqrt{\Delta}}{\Delta+\gamma(E-C)} + \frac{1}{\sqrt{\Delta}}\left[\frac{\Delta+\gamma(E-C)-\Delta}{\Delta+\gamma(E-C)} - \gamma S\right] \\ &= \frac{1}{\sqrt{\Delta}}(1-\gamma S), \end{aligned}$$

which equals to the LHS of Eq. (3.43). QED.

The above proof is a little bit lengthy and not instructive at all. Is it possible to make a "numerical proof" to replace the boring algebra? Here's the idea: Assign E to be a random complex number, C, $\sqrt{\Delta}$, γ to be random real diagonal matrices, and S to be a random real full matrix. The physical Green's function G can be calculated with either Eq. (3.32) or Eq. (3.39). By generating many random samplings of E, C, $\sqrt{\Delta}$, γ, and S, we can test if G is always the same calculated from the two different equations. If the error is always within a tolerance (as expected), we can further estimate the probability of violating the equality. If the probability is so small that it is unlikely to happen within the lifetime of our universe, we may claim that the equality is numerically proved. In Appendix A.14, we attempt to provide such a "numerical proof".

Eq. (3.39) and Eq. (3.37) have defined the physical Green's function and the auxiliary Green's function for general systems. In two-probe systems, Eq. (3.39) remains unchanged while Eq. (3.37) needs to include the self-energy term $\tilde{\Sigma}^r(E)$

$$\mathcal{G}^r(E) = \left[P(E) - S - \tilde{\Sigma}^r(E)\right]^{-1}, \tag{3.44}$$

where $\tilde{\Sigma}^r(E)$ is to take into account the influences of semi-infinite leads. In addition to the retarded Green's function, we can also define the lesser Green's function. In analogy to the derivation of the NECPA equations (see Section 2.8), one can first remove the superscript r in Eqs. (3.39,3.44) to obtain the complex-time Green's function (Fourier transformed)

$$G(E) = \lambda(E) + \mu(E)\,\mathcal{G}(E)\,\mu(E), \tag{3.45}$$

$$\mathcal{G}(E) = \left[P(E) - S - \tilde{\Sigma}(E)\right]^{-1}. \tag{3.46}$$

Afterward one can apply the Langreth theorem to Eqs. (3.45,3.46) and obtain the retarded and lesser Green's function

$$G^r(E) = \lambda(E) + \mu(E)\,\mathcal{G}^r(E)\,\mu(E), \tag{3.47}$$

$$\mathcal{G}^r(E) = \left[P(E) - S - \tilde{\Sigma}^r(E)\right]^{-1}, \tag{3.48}$$

$$G^<(E) = \mu(E)\,\mathcal{G}^<(E)\,\mu(E), \tag{3.49}$$

$$\mathcal{G}^<(E) = \mathcal{G}^r(E)\,\tilde{\Sigma}^<(E)\,\mathcal{G}^a(E). \tag{3.50}$$

Notice that the poles of the Green's functions are determined by the eigenvalue equation 3.32, and the zeros of the denominator $\Delta + \gamma(E - C)$ in $\lambda(E)$ and $\mu(E)$ are not singular at all. For this reason, λ and μ can be treated like a constant

$$\lambda^< = 0, \quad \lambda^r = \lambda^a = \lambda,$$
$$\mu^< = 0, \quad \mu^r = \mu^a = \mu,$$

in the derivation of Eqs. (3.47,3.48,3.49,3.50).

To sum up, we have defined two types of Green's functions in this section, the physical Green's function G and the auxiliary Green's function $\mathcal{G}$. In Section 3.8, physical quantities will be expressed in terms of G; In Section 3.10, the NECPA theory will be applied to $\mathcal{G}$.

3.7 Screening transform

In this section, we shall discuss the screening transform of physical Green's function [5]. In the expression of physical Green's function, all the quantities except for the structure constant S are diagonal matrices (see Eq. (3.39)). It is the off-diagonal elements of S that determine the effective radius of an atom. The larger the effective radius is, the more neighbors the atom has, and the more costly the calculation will be. We have seen in Section 3.3 that the elements of S decays in a power law as a function of the distance. Here we present the screening transform which converts the power law decay into an exponential decay and keeps the physical Green's function unchanged.

Theorem: The physical Green's function G^r will not change by the transform $\gamma \longrightarrow \gamma_\alpha$ and $S \longrightarrow S_\alpha$

$$\gamma_\alpha \equiv \gamma - \alpha, \tag{3.51}$$

$$S_\alpha^{-1} \equiv S^{-1} - \alpha, \tag{3.52}$$

where α is an arbitrary diagonal matrix.

www.ebook3000.com

Proof: It has been proved in Section 3.6 that the physical Green's function has two equivalent forms, Eq. (3.35) and Eq. (3.39). It will be much easier to work with the former one. Due to Eqs. (3.35,3.36), the goal is to prove

$$\left(S^{-1}-\gamma\right)^{-1}=\left(S_{\alpha}^{-1}-\gamma_{\alpha}\right)^{-1}.$$

This is obvious from the definition of S_α and γ_α in Eqs. (3.51,3.52). QED.

According to the theorem, the physical Green's function is independent of the choice of α. The idea of screening transform is to find a proper diagonal matrix α so that the off-diagonal elements of S_α decay super fast. It is further assumed that the diagonal elements of α only depend on the angular momentum quantum number l

$$\alpha_{i_1L_1,i_2L_2}=\delta_{i_1L_1,i_2L_2}\alpha_{l_1}^{opt},$$

where α_l^{opt} is called screening constant. The values of α_l^{opt} are obtained by a numerical optimization for a number of atomic structures such as FCC, BCC, and SC to achieve the best screening effect [4, 5]

α_l^{opt}	$l=0$	$l=1$	$l=2$	$l=3$
$l_{\max}=0$	0.2143	0	0	0
$l_{\max}=1$	0.2872	0.02582	0	0
$l_{\max}=2$	0.3485	0.05303	0.01071	0
$l_{\max}=3$	0.3851	0.07321	0.02248	0.00607

(3.53)

where $l_{\max}$ is the maximum angular momentum quantum number of the atomic site.

As an illustration of the screening effect, Fig. 3.2 shows the amplitude of S matrix elements as a function of the distance. One can see that the S matrix elements decay according to a power law before the screening and according to an exponential law after the screening. Although the BCC structure is used in the example, the screening constant of Eq. (3.53) is applicable to all close-packed structures as long as ω is taken as the Wigner–Seitz radius (see Section 3.13.5). The exponential decay allows one to truncate the screened structure constant S_α if the matrix elements are smaller than a given tolerance. It turns out that atomic sites beyond the second nearest neighbors are usually negligible in closed packed structures (see Fig. 3.2).

To sum up, the physical Green's function remains unchanged by the screening transform Eqs. (3.51,3.52,3.53). After the screening transform, the structure constant is very sparse and the computational cost of Green's

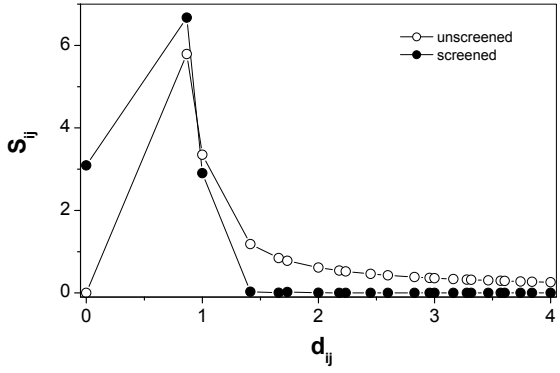


Fig. 3.2 The amplitude of structure constant as the function of site distance before and after the screening transform. In the calculation, $l_{\max} = 2$ and the BCC structure is used with the lattice constant $a = 1$. The amplitude of structure constant between the sites i_1 and i_2 is defined by $||S_{i_1 i_2}|| \equiv \max svd\left(S_{i_1 i_2}\right)$.

function is greatly reduced. The algorithm for calculating Green's function of sparse Hamiltonian will be presented in Chapter 7. So far the screening transform has been presented as a mathematical trick. Actually there are physical considerations underneath the black magic. Interested readers are referred to Ref. [4, 5] for further discussion.

3.8 Physical quantities

Having defined the physical Green's function in Section 3.6, we proceed to express physical quantities in terms of the physical Green's function. The most important physical quantities in quantum transport calculations are the charge density, the density of states, and the transmission coefficient.

The total charge density $\rho\left(\mathbf{r}\right)$ is the sum of the charge density of each atomic sphere

$$\rho\left(\mathbf{r}\right) = \sum_i \rho_i\left(\mathbf{r} - \mathbf{r}_i\right). \tag{3.54}$$

Here $\rho_i\left(\mathbf{r}\right)$ is the charge density of the i-th atomic sphere which can be decomposed into the core electron density and the valence electron density

$$\rho_i\left(\mathbf{r}\right) = \rho_i^{core}\left(\mathbf{r}\right) + \rho_i^{val}\left(\mathbf{r}\right). \tag{3.55}$$

The core electron density $\rho_i^{core}\left(\mathbf{r}\right)$ comes from the occupation of the core orbitals $\Psi_{inL}^{core}\left(\mathbf{r}\right) = \phi_{inl}^{core}\left(r\right) Y_L\left(\Omega\right)$ where $\phi_{inl}^{core}\left(r\right)$ is the bound state so-

www.ebook3000.com

lution of Eq. (3.5) and n is the principal quantum number. Since the core orbitals are fully occupied, the core electron density can be obtained as

$$\rho_i^{core}(\mathbf{r}) = \sum_{nL} \left[\Psi_{inL}^{core}(\mathbf{r})\right]^2 = \frac{1}{4\pi} \sum_{nl} (2l+1) \left[\phi_{inl}^{core}(r)\right]^2, \tag{3.56}$$

where $\sum_m Y_L^2(\Omega) = 2l+1$ is used in the derivation. The valence orbitals, in contrast, are partially occupied and the occupations are determined by the nonequilibrium density matrix which is proportional to $G^<$. By using a transform from the orbital space to the real space [4], the valence electron density $\rho_i^{val}(\mathbf{r})$ can be obtained as

$$\rho_i^{val}(\mathbf{r}) = \sum_{LL'} \int \frac{dE}{2\pi} \Psi_{iL}(\mathbf{r}, E) (-\mathrm{i}) G_{iL,iL'}^<(E) \Psi_{iL'}(\mathbf{r}, E), \tag{3.57}$$

where $\Psi_{iL}(\mathbf{r}, E) \equiv \Psi_{il}(r, E) Y_L(\Omega)$ is the linearized valence muffin-tin orbital (see Eq. (3.24)).

The density of states $D(E)$ can be obtained as

$$D(E) = -\frac{1}{\pi} \mathrm{Im} \sum_i \mathrm{Tr}\, G_{ii}^r(E), \tag{3.58}$$

which follows from the definition of the retarded Green's function. The transmission coefficient $T(E)$ can be obtained as

$$T(E) = \mathrm{Tr}\, G^r(E) \Gamma_L(E) G^a(E) \Gamma_R(E), \tag{3.59}$$

which is the general transmission coefficient formula Eq. (2.110) applied to the LMTO method.

In practical calculations, it is more convenient to calculate the transmission coefficient directly from the auxiliary Green's function rather than the physical Green's function. It is interesting to note that the expression of $T(E)$ in terms of the auxiliary Green's function is formally the same as Eq. (3.59), namely

$$T(E) = \mathrm{Tr}\, \mathcal{G}^r(E) \tilde{\Gamma}_L(E) \mathcal{G}^a(E) \tilde{\Gamma}_R(E), \tag{3.60}$$

where $\mathcal{G}$ is the auxiliary Green's function and $\tilde{\Gamma}_\beta$ is defined by

$$\tilde{\Gamma}_\beta \equiv \mathrm{i} \left(\tilde{\Sigma}_\beta^r - \tilde{\Sigma}_\beta^a\right), \tag{3.61}$$

in which $\tilde{\Sigma}_\beta^r$ is the auxiliary lead self-energy

$$\begin{aligned} \tilde{\Sigma}_\beta^r &\equiv (P-S)_{C\beta} \mathcal{G}_{\beta\beta}^r (P-S)_{\beta C}, \\ \mathcal{G}_{\beta\beta}^r &\equiv \left[\left(P^+ - S\right)_{\beta\beta}\right]^{-1}. \end{aligned} \tag{3.62}$$

The derivation of Eq. (3.60) is provided in Appendix A.15. Here we would like to point out that Eqs. (3.60,3.61,3.62) imply a substitution rule in the general transmission coefficient formulas Eqs. (2.110,2.104,2.113)

$$G \longrightarrow \mathcal{G},$$
$$E^{+} - H \longrightarrow P^{+} - S.$$

We shall see in Section 3.10 that the substitution rule is also applicable to the NECPA formula.

Finally we would like discuss the calculation of electrostatic potential $V(\mathbf{r})$ and derive the expressions of multipoles in each atomic sphere. Remember that we have derived the total charge density $\rho(\mathbf{r})$ in Eqs. (3.54,3.55,3.56,3.57). Given $\rho(\mathbf{r})$, there are two equivalent approaches to calculate $V(\mathbf{r})$: One is to solve the Poisson equation with a proper boundary condition

$$\nabla^2 V(\mathbf{r}) = -4\pi\rho(\mathbf{r}), \tag{3.63}$$

and the other is to integrate the Coulomb potential directly

$$V(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|}. \tag{3.64}$$

In the ASA, it is assumed that the charge density inside an atomic sphere is spherically symmetric and the deviation from spherical symmetry can be described by a dipole sitting at the atomic center. As a result, the integral in Eq. (3.64) is reduced to a summation of monopole and dipole potentials which is much simpler than solving a differential equation. Therefore the integral approach is adopted to calculate $V(\mathbf{r})$ in the LMTO method while the differential approach is used to calculate the potential correction in the presence of an external field (see Appendix A.22). Below we shall derive the expressions of the symmetrized charge density, the monopole (the charge), and the dipole in an atomic sphere.

The charge density in an atomic sphere is composed of the core electron density and the valence electron density

$$\rho_i(r) = \rho_i^{core}(r) + \rho_i^{val}(r).$$

Since the core electron density $\rho_i^{core}(\mathbf{r}) = \rho_i^{core}(r)$ is already spherically symmetric (see Eq. (3.56)), we only need to symmetrize the valence electron

www.ebook3000.com

density $\rho_i^{val}(\mathbf{r})$ by averaging over the angular part

$$\begin{aligned}\rho_i^{val}(r) &\equiv \frac{\int d\Omega \rho_i^{val}(\mathbf{r})}{\int d\Omega} \\ &= \frac{1}{4\pi}\sum_{LL'}\int d\Omega \int \frac{dE}{2\pi}\Psi_{iL}(\mathbf{r},E)(-\mathrm{i})G^{<}_{iL,iL'}(E)\Psi_{iL'}(\mathbf{r},E) \\ &= \frac{1}{4\pi}\sum_{LL'}\int d\Omega \\ &\quad \times \int \frac{dE}{2\pi}\Psi_{il}(r,E)Y_L(\Omega)(-\mathrm{i})G^{<}_{iL,iL'}(E)\Psi_{il'}(r,E)Y_{L'}(\Omega) \\ &= \frac{1}{4\pi}\sum_{L}\int \frac{dE}{2\pi}(-\mathrm{i})G^{<}_{iL,iL}(E)\Psi^2_{il}(r,E), \end{aligned} \tag{3.65}$$

where $\int d\Omega Y_L(\Omega)Y_{L'}(\Omega) = \delta_{LL'}$ is used in the derivation.

The monopole (the charge) in an atomic sphere is a summation of the valence electron charge, the core electron charge, and the nuclear charge

$$\begin{aligned} Q_i &= Q_i^{val} + Z_i^{core} - Z_i \\ &= \int_0^{R_i}\rho_i^{val}(r)4\pi r^2 dr - (Z_i - Z_i^{core}) \\ &= \sum_L \int \frac{dE}{2\pi}(-\mathrm{i})G^{<}_{iL,iL}(E)\int_0^{R_i}\Psi^2_{il}(r,E)r^2dr - Z_i^{val} \\ &= \sum_L \int \frac{dE}{2\pi}(-\mathrm{i})G^{<}_{iL,iL}(E) - Z_i^{val}, \end{aligned} \tag{3.66}$$

where the normalization $\int_0^{R_i}\Psi^2_{il}(r,E)r^2dr = 1$ is used in the derivation. Here Z_i is the atomic number (nuclear charge), Z_i^{core} is the core electron number, Z_i^{val} is the valence electron number, and $Z_i = Z_i^{core} + Z_i^{val}$ due to the charge neutrality. Eq. (3.66) indicates that the charge calculated in the real space coincides with the one calculated in the orbital space.

The dipole in an atomic sphere comes entirely from the valence electron density since the core electron density is spherically symmetric. By definition, the dipole is obtained as

$$\begin{aligned}\mathbf{P}_i &= \int d\Omega \int_0^{R_i} r^2 dr \rho_i^{val}(\mathbf{r})\,\mathbf{r} \\ &= \sum_{LL'} \int_0^{R_i} r^2 dr \int d\Omega \int \frac{dE}{2\pi} \Psi_{iL}(\mathbf{r},E)(-\mathrm{i})G^{<}_{iL,iL'}(E)\Psi_{iL'}(\mathbf{r},E)\,\mathbf{r} \\ &= \sum_{LL'} \int d\Omega \frac{\mathbf{r}}{r} Y_L(\Omega) Y_{L'}(\Omega) \\ &\quad \times \int \frac{dE}{2\pi} \int_0^{R_i} r^3 dr \Psi_{il}(r,E)\Psi_{il'}(r,E)(-\mathrm{i})G^{<}_{iL,iL'}(E) \\ &= \int_0^{R_i} r^3 dr \sum_{LL'} \mathbf{\Theta}_{LL'} F_{i,LL'}(r), \end{aligned} \tag{3.67}$$

where $\Theta_{LL'}$ is defined by

$$\mathbf{\Theta}_{LL'} \equiv \int d\Omega \frac{\mathbf{r}}{r} Y_L(\Omega) Y_{L'}(\Omega), \tag{3.68}$$

and $F_{LL'}(r)$ is defined by

$$F_{iLL'}(r) \equiv \int \frac{dE}{2\pi} \Psi_{il}(r,E)\Psi_{il'}(r,E)(-\mathrm{i})G^{<}_{iL,iL'}(E). \tag{3.69}$$

Notice that the unit vector $\frac{\mathbf{r}}{r}$ can be expressed by the spherical harmonics (see Appendix A.10)

$$\frac{\mathbf{r}}{r} = \left(\frac{x}{r}, \frac{y}{r}, \frac{z}{r}\right) = \sqrt{\frac{4\pi}{3}} \left[Y_{1,1}(\Omega), Y_{1,-1}(\Omega), Y_{1,0}(\Omega)\right].$$

Consequently $\Theta_{LL'}$ can be expressed by the Gaunt coefficient defined in Eq. (3.17)

$$\mathbf{\Theta}_{LL'} = \sqrt{\frac{4\pi}{3}} \left[C_{(1,1),L,L'}, C_{(1,-1),L,L'}, C_{(1,0),L,L'}\right]. \tag{3.70}$$

We note that the expressions of the charge density, the monopole, and the dipole contain energy integrals of $G^{<}(E)$. To simplify the expressions, we define the double energy moment $M^{k_1k_2}_{i,L_1L_2}$ by

$$M^{k_1k_2}_{i,L_1L_2} \equiv \int \frac{dE}{2\pi} (-\mathrm{i}) G^{<}_{iL_1,iL_2}(E) \left(E - E^0_{il_1}\right)^{k_1} \left(E - E^0_{il_2}\right)^{k_2}. \tag{3.71}$$

By using the second order expansion of the MTO

$$\Psi_{il}(r,E) = \phi_{il}(r) + \dot{\phi}_{il}(r)\left(E - E^0_{il}\right) + \frac{1}{2}\ddot{\phi}_{il}(r)\left(E - E^0_{il}\right)^2 + \cdots \tag{3.72}$$

Eq. (3.65) can be simplified as

$$\rho_i(r) = \frac{1}{4\pi} \sum_L M^{00}_{i,LL}\phi^2_{il}(r) + 2M^{10}_{i,LL}\phi_{il}(r)\dot{\phi}_{il}(r) + M^{20}_{i,LL}\left[\dot{\phi}^2_{il}(r) + \ddot{\phi}_{il}(r)\right]; \tag{3.73}$$

www.ebook3000.com

Eq. (3.66) can be simplified as

$$Q_i = \left(\sum_L \mathrm{Tr}\ M^{00}_{i,LL} \right) - Z^{val}; \tag{3.74}$$

Eq. (3.67) can be simplified as

$$\begin{aligned} F_{i,L_1L_2}(r) = {} & M^{00}_{i,L_1L_2} \phi_{il_1}(r)\, \phi_{il_2}(r) \\ & + M^{10}_{i,L_1L_2} \dot{\phi}_{il_1}(r)\, \phi_{il_2}(r) + M^{01}_{i,L_1L_2} \phi_{il_1}(r)\, \dot{\phi}_{il_2}(r) \\ & + M^{11}_{i,L_1L_2} \dot{\phi}_{il_1}(r)\, \dot{\phi}_{il_2}(r) + \frac{1}{2} M^{20}_{i,L_1L_2} \ddot{\phi}_{il_1}(r)\, \phi_{il_2}(r) \\ & + \frac{1}{2} M^{02}_{i,L_1L_2} \phi_{il_1}(r)\, \ddot{\phi}_{il_2}(r). \end{aligned} \tag{3.75}$$

To calculate the double energy moment $M^{k_1k_2}_{i,L_1L_2}$, we define the energy moment $\tilde{M}^k_i$ by

$$\tilde{M}^k_i \equiv \int \frac{dE}{2\pi} (-\mathrm{i})\, G^<_{ii}(E)\, E^k. \tag{3.76}$$

Notice that $M^{k_1k_2}_{i,L_1L_2}$ is a linear combination of $\tilde{M}^k_{i,L_1L_2}$ and $\tilde{M}^k_i$ can be evaluated numerically with the contour integral technique (see Section 5.6 and 6.5). So we are done with the spherically symmetrized quantities.

To sum up, we have derived various physical quantities in terms of Green's function in the LMTO method. The charge density, the density of states, and the transmission coefficient are obtained in Eqs. (3.55,3.56,3.57), Eq. (3.58), and Eq. (3.60), respectively. In the ASA, the spherically symmetrized charge density, the monopole, and the dipole are obtained and simplified in terms of energy moments in Eq. (3.73), Eq. (3.74), and Eqs. (3.67,3.70,3.75), respectively.

3.9 Periodicity and Fourier transform

So far we have discussed the LMTO method for general atomic systems. Very often an atomic system may have translational symmetry in one or more dimensions. In this section, we shall investigate how to adapt the general LMTO method to a periodic system.

Consider a periodic two-probe system which has translational symmetry in the lateral dimensions (see Fig. 3.3). The unit cell of the periodic two-probe system is a quasi-1d two-probe system extending to positive and negative infinity in the transport dimension. The unit cell is repeated periodically in the lateral dimensions, and the images are indexed by integer

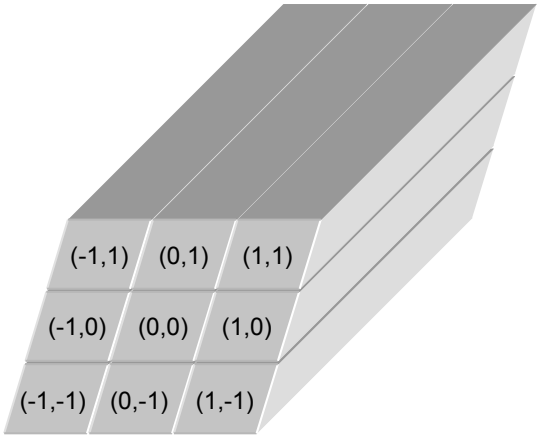


Fig. 3.3 A sketch of a periodic two-probe system in which $I \equiv (I_x, I_y)$ is the unit cell index in the lateral dimensions.

pairs $I \equiv (I_x, I_y)$. The origin of the image (I_x, I_y) has the displacement $I_x\mathbf{a}_x + I_y\mathbf{a}_y$ where $\mathbf{a}_x$ and $\mathbf{a}_y$ are the unit cell vectors.

Due to the periodicity, the structure constant $S_{I_1 I_2}$ between two unit cells I_1 and I_2 only depends on the index difference $I_1 - I_2$, namely, $S_{I_1 I_2} = S_{I_1 - I_2} \equiv S_I$. Consequently S_I can be Fourier transformed to $S(k)$

$$S(k) = \sum_I e^{-\mathrm{i}k\cdot I} S_I, \tag{3.77}$$

where $k = (k_x, k_y)$ is the dimensionless wave vector defined by $k_x \equiv \mathbf{k}\cdot\mathbf{a}_x$ and $k_y \equiv \mathbf{k}\cdot\mathbf{a}_y$ in which $\mathbf{k}$ is the wave vector. Inversely $S(k)$ can be Fourier transformed back to S_I

$$S_I = \int_{BZ} \frac{d^2k}{(2\pi)^2} e^{\mathrm{i}k\cdot I} S(k), \tag{3.78}$$

where BZ represents the 2d Brillouin zone defined by $k_x \in [0, 2\pi]$ and $k_y \in [0, 2\pi]$.

Similarly the Green's function $G_{I_1 I_2} = G_{I_1 - I_2} \equiv G_I$ can be Fourier transformed to $G(k)$

$$G(k) = \sum_I e^{-\mathrm{i}k\cdot I} G_I, \tag{3.79}$$

or inversely

$$G_I = \int_{BZ} \frac{d^2k}{(2\pi)^2} e^{\mathrm{i}k\cdot I} G(k). \tag{3.80}$$

www.ebook3000.com

The Green's function of a single unit cell can be obtained by setting $I = 0$ in Eq. (3.80)

$$G_0 = \int_{BZ} \frac{d^2k}{(2\pi)^2} G(k). \tag{3.81}$$

Due to the periodicity, physical quantities are identical in all unit cells. So one can calculate physical quantities in any one of the them. For example, the transmission coefficient can be calculated in the unit cell I_0

$$\begin{aligned}
T(E) &= \mathrm{Tr}\left[\mathcal{G}^r(E)\tilde{\Gamma}_L(E)\mathcal{G}^a(E)\tilde{\Gamma}_R(E)\right]_{I_0I_0} \\
&= \mathrm{Tr}\sum_{I_1,I_2,I_3}\left[\mathcal{G}^r(E)\right]_{I_0I_1}\left[\tilde{\Gamma}_L(E)\right]_{I_1I_2}\left[\mathcal{G}^a(E)\right]_{I_2I_3}\left[\tilde{\Gamma}_R(E)\right]_{I_3I_0} \\
&= \mathrm{Tr}\sum_{I_1,I_2,I_3}\left[\int_{BZ}\frac{d^2k_1}{(2\pi)^2}e^{\mathrm{i}k_1\cdot(I_0-I_1)}\mathcal{G}^r(E,k_1)\right] \\
&\quad\times\left[\int_{BZ}\frac{d^2k_2}{(2\pi)^2}e^{\mathrm{i}k_2\cdot(I_1-I_2)}\tilde{\Gamma}_L(E,k_2)\right]\left[\int_{BZ}\frac{d^2k_3}{(2\pi)^2}e^{\mathrm{i}k_3\cdot(I_2-I_3)}\mathcal{G}^a(E,k_3)\right] \\
&\quad\times\left[\int_{BZ}\frac{d^2k_4}{(2\pi)^2}e^{\mathrm{i}k_4\cdot(I_3-I_0)}\tilde{\Gamma}_R(E,k_4)\right] \\
&= \mathrm{Tr}\int_{BZ}\frac{d^2k}{(2\pi)^2}\mathcal{G}^r(E,k)\tilde{\Gamma}_L(E,k)\mathcal{G}^a(E,k)\tilde{\Gamma}_R(E,k),
\end{aligned} \tag{3.82}$$

where $\mathcal{G}^r(E,k)$ and $\tilde{\Gamma}_\beta(E,k)$ are the Fourier transform of $\mathcal{G}^r(E)$ and $\tilde{\Gamma}_\beta(E)$ respectively. Notice that $\sum_I e^{-\mathrm{i}k\cdot I} = \delta(k)$ is used in the above derivation.

To sum up, the general LMTO method needs two minor changes to adapt to periodic systems. The first change is to carry out Fourier transform to obtain k-dependent structure constant and Green's function. The second change is to integrate over the Brillouin zone to obtain physical quantities of a unit cell.

3.10 NECPA-LMTO formalism

We have developed the NECPA theory in Chapter 2 to calculate the quantum transport in disordered two-probe systems. We have also developed the LMTO method in previous sections to calculate the electronic structure from first principles. The goal of this section is to combine the two techniques together.

Remember that the auxiliary Green's function has the nice property that diagonal elements carry atomic information while off-diagonal elements

carry geometric information. Since the NECPA theory is formulated for diagonal disorder, it is natural to apply the NECPA theory to the auxiliary Green's function. Comparing the definition of the auxiliary Green's function $\mathcal{G} = (P - S)^{-1}$ to the definition of the conventional Green's function $G = (E - H)^{-1}$, one obtains a *substitution rule* to apply the NECPA theory to the LMTO method

$$E - H \longleftrightarrow P - S. \tag{3.83}$$

It means that the variable $E-H$ in the NECPA theory needs to be replaced by $P - S$ in the LMTO method.

By applying the rule to the NECPA Eqs. (2.133,2.134) and using the Fourier transform Eq. (3.81), one derives the NECPA-LMTO equations for a periodic disordered two-probe system

$$\begin{cases} \overline{\mathcal{G}_i^r}(E) = \sum_q x_{iq} \overline{\mathcal{G}_{iq}^r}(E), \\ \overline{\mathcal{G}^r}(E,k) = \left[\tilde{P}^r(E) - S(k) - \Sigma^r(E,k)\right]^{-1}, \\ \overline{\mathcal{G}^r}(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \overline{\mathcal{G}^r}(E,k), \\ \overline{\mathcal{G}_i^r}(E) \equiv \left[\overline{\mathcal{G}^r}(E)\right]_{ii}, \\ \overline{\mathcal{G}_i^r}(E) = \left[\tilde{P}_i^r(E) - \Omega_i^r(E)\right]^{-1}, \\ \overline{\mathcal{G}_{iq}^r}(E) = \left[P_{iq}(E) - \Omega_i^r(E)\right]^{-1}, \end{cases} \tag{3.84}$$

$$\begin{cases} \overline{\mathcal{G}_i^<}(E) = \sum_q x_{iq} \overline{\mathcal{G}_{iq}^<}(E), \\ \overline{\mathcal{G}^<}(E,k) = \overline{\mathcal{G}^r}(E,k)\left[-\tilde{P}^<(E) + \Sigma^<(E,k)\right]\overline{\mathcal{G}^a}(E,k), \\ \overline{\mathcal{G}^<}(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \overline{\mathcal{G}^<}(E,k), \\ \overline{\mathcal{G}_i^<}(E) \equiv \left[\overline{\mathcal{G}^<}(E)\right]_{ii}, \\ \overline{\mathcal{G}_i^<}(E) = \overline{\mathcal{G}_i^r}(E)\left[-\tilde{P}_i^<(E) + \Omega_i^<(E)\right]\overline{\mathcal{G}_i^a}(E), \\ \overline{\mathcal{G}_{iq}^<}(E) = \overline{\mathcal{G}_{iq}^r}(E)\,\Omega_i^<(E)\,\overline{\mathcal{G}_{iq}^a}(E), \end{cases} \tag{3.85}$$

www.ebook3000.com

where i is the atomic site index, q is the atomic species index, and x_{iq} is the probability that the site-i is occupied by the species-q. $\tilde{P}^{\lambda}(E)$ is the coherent potential and $\Omega_i^{\lambda}(E)$ is the coherent interactor ($\lambda = r, <$ to represent retarded and lesser quantities respectively). $\tilde{P}^{\lambda}(E)$ and $\Omega_i^{\lambda}(E)$ are unknowns and to be solved from the NECPA-LMTO equations.

The major difference between the NECPA equations and the NECPA-LMTO equations is that the on-site energy ε_{iq} in the former is replaced by the potential function $P_{iq}(E)$ in the latter. $P_{iq}(E)$ is a $(l_{\max}+1)^2$-by-$(l_{\max}+1)^2$ diagonal matrix (see Eqs. (3.38,3.51))

$$P_{iq}(E) \equiv \frac{E - C_{iq}}{\Delta_{iq} + (\gamma_{iq} - \alpha)(E - C_{iq})}, \tag{3.86}$$

where $l_{\max}$ is the maximum angular momentum quantum number of the atomic site. Consequently all the on-site variables (e.g., $\tilde{P}^{\lambda}(E)$ and $\Omega_i^{\lambda}(E)$) are also $(l_{\max}+1)^2$-by-$(l_{\max}+1)^2$ matrices. Since the derivation of the NECPA equations does not assume that the on-site variables are scalars, the formulation in Chapter 2 is transferable directly to the LMTO method.

Notice that the NECPA-LMTO equations are for the auxiliary Green's function. To calculate disorder-averaged physical quantities, we need to calculate the physical Green's function. By using Eqs. (3.47,3.49), the diagonal blocks of physical Green's function are obtained as

$$\overline{G_i^r} = \sum_q x_{iq} \left(\lambda_{iq} + \mu_{iq} \overline{\mathcal{G}_{iq}^r} \mu_{iq} \right), \tag{3.87}$$

$$\overline{G_i^<} = \sum_q x_{iq} \mu_{iq} \overline{\mathcal{G}_{iq}^<} \mu_{iq}. \tag{3.88}$$

where $\overline{\mathcal{G}_{iq}^r}$ and $\overline{\mathcal{G}_{iq}^<}$ are the conditional auxiliary Green's functions solved from Eqs. (3.84,3.85). The off-diagonal blocks of physical Green's function are more complicated and involve further approximation, here we simply present the results and refer interested readers to Ref. [4] for a heuristic derivation

$$\overline{G_{ii'}^r} \approx \sum_{qq'} x_{iq} x_{i'q'} \mu_{iq} \left[\overline{\mathcal{G}_{iq}} \left(\overline{\mathcal{G}_i} \right)^{-1} \overline{\mathcal{G}_{ii'}} \left(\overline{\mathcal{G}_{i'}} \right)^{-1} \overline{\mathcal{G}_{i'q'}} \right]^r \mu_{i'q'}, \tag{3.89}$$

$$\overline{G_{ii'}^<} \approx \sum_{qq'} x_{iq} x_{i'q'} \mu_{iq} \left[\overline{\mathcal{G}_{iq}} \left(\overline{\mathcal{G}_i} \right)^{-1} \overline{\mathcal{G}_{ii'}} \left(\overline{\mathcal{G}_{i'}} \right)^{-1} \overline{\mathcal{G}_{i'q'}} \right]^< \mu_{i'q'}, \tag{3.90}$$

where $[\cdots]^r$ and $[\cdots]^<$ need to be expanded with the Langreth theorem.

To sum up, we have derived the NECPA-LMTO equations (3.84,3.85) for the auxiliary Green's function by using a substitution rule. The physical

Green's function can be obtained from the conditional auxiliary Green's function which is solvable from the NECPA-LMTO equations.

3.11 Self-consistent calculation

So far we have derived all the theoretical elements required to carry out atomistic quantum transport simulation: The NECPA theory for quantum transport in disordered open system and the LMTO method for atomistic modeling of nanostructure. In this and the next section, we shall collect all the formulas in their final form to make a summary of the theoretical formalism. The summary will be the starting point for the numerical implementation discussed in following chapters.

Generally speaking an atomistic quantum transport simulation proceeds in two steps. The first step is to solve the nonequilibrium Hamiltonian self-consistently. The second step is to calculate quantum transport based on the obtained Hamiltonian. The first step is referred to as the self-consistent calculation, and the second step is referred to as the post-analysis calculation. This section is to summarize the theoretical formalism of self-consistent calculation, and the next section will summarize the theoretical formalism of post-analysis calculation.

3.11.1 *Flowchart*

Before explaining technical details, we would like to present an overall picture of the NECPA-LMTO self-consistent calculation (see Fig. 3.4): For a given LMTO Hamiltonian (potential parameters and structure constant), the disorder-averaged charge density can be calculated by using the NECPA theory; For a given charge density, the LMTO Hamiltonian can be constructed by using the LMTO method within the DFT. The two steps constitute a self-consistent loop. The first step relies on quantum mechanics, nonequilibrium statistics and the nonequilibrium coherent potential approximation. The second step relies on density functional theory and the approximations made in the LMTO method.

We also assume that the atom centers $\{\mathbf{r}_i\}$ and compositions $\{x_{iq}, Z_{iq}\}$ are known as the input, where i is the atomic site index and q is the chemical species index. An atomic site i can be occupied by one or more species of chemical elements. A chemical element is characterized by its nuclear charge Z_{iq}, and the occupation probability of the element is x_{iq}. Notice that $\sum_q x_{iq} = 1$ due to the probability normalization. In the ASA, each species

www.ebook3000.com

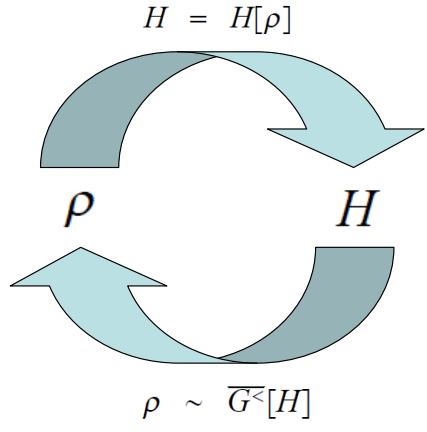


Fig. 3.4 A sketch of the NECPA-LMTO self-consistent calculation: The NECPA theory is used in the step from the Hamiltonian to the charge density; The LMTO method is used in the step from the charge density to the Hamiltonian.

has its own atomic sphere with the radius R_{iq}, and the total atomic sphere volume is equal to the unit cell volume, namely, $\frac{4\pi}{3}\sum_{iq} x_{iq} R_{iq}^3 = V_{cell}$.

The procedure for solving a two-probe Hamiltonian is as follows: (1) Solve the left lead Hamiltonian with a bulk self-consistent calculation; (2) Solve the right lead Hamiltonian with a bulk self-consistent calculation; (3) Solve the two-probe Hamiltonian with a two-probe self-consistent calculation. So we need to carry out two distinct types of calculations, bulk self-consistent calculation and two-probe self-consistent calculation. Although the two types of calculations are very different in concept, the self-consistent loops share many similarities. To avoid the redundancy, we construct a "unified" flowchart and point out the differences if necessary.

With the overall picture in mind, we plot a complete flowchart in Fig. 3.5 which contains all the technical details of the NECPA-LMTO method. There are 12 steps in the flowchart and they will be explained in the following 12 subsections. In particular step-6 is to solve the NECPA equations and step-10 is to calculate the atomic potential with the DFT, which are key steps of the self-consistent loop. The formulas of each step have been developed in previous sections; it is time to thread individual "pearls" into a "necklace". For convenience, some formulas are repeated or rewritten and some symbols are re-defined or overloaded. Also note that the spin degree of freedom is omitted for simplicity of notation (see Section 3.13.1).

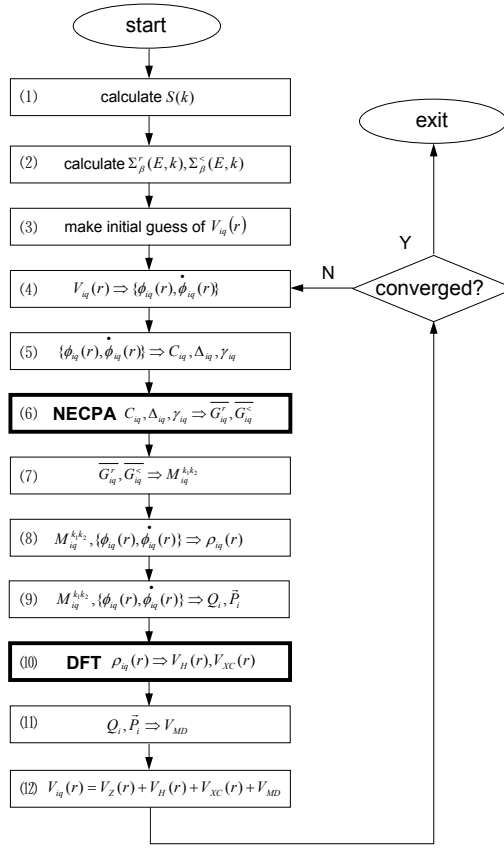


Fig. 3.5 The flowchart of the self-consistent calculation for both bulk and two-probe systems. The formulas used in each step are summarized in the corresponding subsections.

3.11.2 *Step-1 calculate structure constant*

Given atom centers $\{\mathbf{r}_i\}$, the canonical structure constant $\tilde{S}$ can be calculated by

$$\tilde{S}_{i_1L_1,i_2L_2}\equiv\sum_m(-1)^{l_2+1}\frac{8\pi\,(2l-1)\,!!}{(2l_1-1)\,!!\,(2l_2-1)\,!!}C_{LL_1L_2}\times\left(\frac{\omega}{|\mathbf{r}_{i_2}-\mathbf{r}_{i_1}|}\right)^{l+1}Y_L\left(\Omega_{i_2i_1}\right)\left(1-\delta_{i_1i_2}\right),\qquad(3.91)$$

www.ebook3000.com

$$C_{LL_1L_2} \equiv \int Y_L(\Omega) Y_{L_1}(\Omega) Y_{L_2}(\Omega) d\Omega, \tag{3.92}$$

where $l = l_1 + l_2$, ω is the Wigner–Seitz radius defined by Eq. (3.181), $\Omega_{i_2 i_1}$ is the solid angle of $\mathbf{r}_{i_2} - \mathbf{r}_{i_1}$, and $Y_L(\Omega)$ is the real-valued spherical harmonics defined by Eq. (3.7).

The screened structure constant S is obtained with the screening transform

$$S = \left[\tilde{S}^{-1} - \alpha\right]^{-1}, \tag{3.93}$$

where the screening constant α is a diagonal matrix

$$\alpha_{i_1 L_1, i_2 L_2} = \delta_{i_1 i_2} \delta_{L_1 L_2} \alpha_{l_1}^{opt}, \tag{3.94}$$

and the values of α_l^{opt} are listed in Eq. (3.53). The off-diagonal elements of S decay exponentially with the distance $|\mathbf{r}_{i_1} - \mathbf{r}_{i_2}|$. It is assumed that $S_{i_1 L_1, i_2 L_2} = 0$ if $|\mathbf{r}_{i_1} - \mathbf{r}_{i_2}| > \lambda\omega$ where λ is a dimensionless cutoff parameter.

Two-probe systems are assumed to be periodic in the lateral dimensions (see Fig. 3.3). Due to the periodicity, S can be Fourier transformed into $S(k)$

$$S(k) = \sum_I e^{-\mathrm{i}k \cdot I} S_I, \tag{3.95}$$

where $S_I = S_{I_1 - I_2} = S_{I_1 I_2}$ is the structure constant between the two unit cells I_1 and I_2. $I \equiv (I_x, I_y)$ is the unit cell index taking values $(0,0)$, $(0, \pm 1)$, $(\pm 1, 0)$, $(\pm 1, \pm 1)$, etc.; $k \equiv (k_x, k_y)$ is the dimensionless wave vector taking values in the 2d Brillouin zone $k_x \in [0, 2\pi]$ and $k_y \in [0, 2\pi]$. Bulk systems are assumed to be periodic in all the three dimensions and hence $I \equiv (I_x, I_y, I_z)$ and $k \equiv (k_x, k_y, k_z)$.

3.11.3 *Step-2 calculate self-energy*

This step is only applicable to two-probe systems. After the lead self-consistent calculations, the lead Hamiltonians are available. The goal of this step is to calculate the lead self-energies $\Sigma_\beta^r(E,k)$ and $\Sigma_\beta^<(E,k)$ from the lead Hamiltonians.

First of all, a semi-infinite lead can be divided into a series of unit cells along the transport direction, see Fig. 2.3. The lead unit cell is sufficiently

thick that only neighboring unit cells have nonzero structure constant. Consequently $S_L(k)$ and $S_R(k)$ can be written in a block tridiagonal form

$$S_L(k) = \begin{pmatrix} \ddots & \ddots & & \\ \ddots & S_L^0(k) & S_L^-(k) & \\ & S_L^+(k) & S_L^0(k) & S_L^-(k) \\ & & S_L^+(k) & S_L^0(k) \end{pmatrix}, \tag{3.96}$$

$$S_R(k) = \begin{pmatrix} S_R^0(k) & S_R^+(k) & & \\ S_R^-(k) & S_R^0(k) & S_R^+(k) & \\ & S_R^-(k) & S_R^0(k) & \ddots \\ & & \ddots & \ddots \end{pmatrix}, \tag{3.97}$$

where each diagonal matrix block corresponds to a lead unit cell.

The retarded surface Green's function of the lead-β can be solved from

$$\mathcal{G}_{\beta\beta}^r(E,k) = \left[P_\beta(E) - S_\beta^0(k) - S_\beta^+(k)\,\mathcal{G}_{\beta\beta}^r(E,k)\,S_\beta^-(k)\right]^{-1}, \tag{3.98}$$

where $P_\beta(E)$ is the potential function of the lead-β. $P_\beta(E)$ can be constructed with the lead potential parameters by using Eq. (3.114).

Finally, the lead self-energies $\Sigma_\beta^r(E,k)$ and $\Sigma_\beta^<(E,k)$ are obtained as

$$\Sigma_\beta^r(E,k) = S_{C\beta}(k)\,\mathcal{G}_{\beta\beta}^r(E,k)\,S_{\beta C}(k), \tag{3.99}$$

$$\Sigma_\beta^<(E,k) = f_\beta(E)\left[\Sigma_\beta^a(E,k) - \Sigma_\beta^r(E,k)\right], \tag{3.100}$$

where $S_{C\beta}(k)$ is the $C\beta$ block of $S(k)$. $f_\beta(E)$ is the Fermi function of the lead-β

$$f_\beta(E) = \frac{1}{e^{(E-\mu_\beta)/k_BT}+1}. \tag{3.101}$$

Notice that $\Sigma_\beta^a(E,k)$ is the Hermitian conjugate of its retarded counterpart, namely, $\Sigma_\beta^a(E,k) = \left[\Sigma_\beta^r(E,k)\right]^\dagger$.

3.11.4 *Step-3 make an initial guess*

Make an initial guess of $V_{iq}(r)$ either from the solution of an isolated atom or the solution of some previous bulk or two-probe self-consistent calculations.

www.ebook3000.com

3.11.5 *Step-4 calculate atomic orbital*

Given a potential $V_{iq}(r)$, solve the core orbitals $\left\{\Psi^{core}_{iqnL}(\mathbf{r})\right\}$ and the valence orbitals $\{\Psi_{iqL}(\mathbf{r},E)\}$ of the atomic site-i chemical species-q. Since the potential is spherically symmetric, the angular momentum is conserved, and hence $\Psi^{core}_{iqnL}(\mathbf{r}) = \phi^{core}_{iqnl}(r)\,Y_L(\Omega)$ and $\Psi_{iqL}(\mathbf{r},E) = \phi_{iql}(r,E)\,Y_L(\Omega)$ where $Y_L(\Omega)$ is the real-valued spherical harmonics defined by Eq. (3.7). Here n is the principal quantum number and $L=(l,m)$ is a composite of angular momentum quantum numbers.

For the core orbitals, $\phi^{core}_{iqnl}(r)$ is the bound state solution of the radial Schrödinger equation

$$\left\{-\frac{1}{2}\partial_r^2+\frac{l(l+1)}{2r^2}+\left[V_{iq}(r)-E\right]\right\}\chi(r)=0, \tag{3.102}$$

or the radial scalar relativistic equation (see Section 3.13.4)

$$\left\{-\frac{1}{2}\partial_r^2+\frac{l(l+1)}{2r^2}+M(r)\left[V_{iq}(r)-E\right]-\frac{\alpha_c^2}{4M(r)}V'_{iq}(r)\left(\partial_r-\frac{1}{r}\right)\right\}\chi(r)=0, \tag{3.103}$$

where $\chi(r)\equiv r\phi(r)$ and $\chi(r)$ is normalized to $\int_0^{R_{iq}}\chi^2(r)\,dr=1$. In the scalar relativistic equation, $M(r)$ is defined by $M(r)\equiv 1-\frac{\alpha_c^2}{2}\left[V_{iq}(r)-E\right]$ and $\alpha_c\approx\frac{1}{137.036}$ is the fine structure constant.

For the valence orbitals, $\phi_{iql}(r,E)$ is the solution of the radial Schrödinger equation (3.102) or the radial scalar-relativistic equation (3.103) at energy E with the boundary condition $\phi_{iql}(r\to 0,E)\sim r^l$. The normalization of valence orbitals is the same as that of core orbitals. $\phi_{iql}(r,E)$ is further expanded around the energy center E^0_{iql} (see Section 3.13.3)

$$\phi_{iql}(r,E)=\phi_{iql}(r)+\dot{\phi}_{iql}(r)\left(E-E^0_{iql}\right)+\frac{1}{2}\ddot{\phi}_{iql}(r)\left(E-E^0_{iql}\right)^2+\cdots, \tag{3.104}$$

where $\phi_{iql}(r)$, $\dot{\phi}_{iql}(r)$, $\ddot{\phi}_{iql}(r)$ are defined by

$$\phi_{iql}(r)\equiv\left[\phi_{iql}(r,E)\right]_{E=E^0_{iql}}, \tag{3.105}$$

$$\dot{\phi}_{iql}(r)\equiv\left[\frac{\partial}{\partial E}\phi_{iql}(r,E)\right]_{E=E^0_{iql}}, \tag{3.106}$$

$$\ddot{\phi}_{iql}(r)\equiv\left[\frac{\partial^2}{\partial E^2}\phi_{iql}(r,E)\right]_{E=E^0_{iql}}. \tag{3.107}$$

3.11.6 *Step-5 calculate potential parameter*

Given radial functions of the valence orbitals $\{\Psi_{iqL}(\mathbf{r}, E)\}$, the potential parameters $\{C_{iql}, \Delta_{iql}, \gamma_{iql}\}$ can be calculated with the Wronskians at $r = R_{iq}$

$$C_{iql} = E^0_{iql} - \frac{\{K_l(r), \phi_{iql}(r)\}}{\left\{K_l(r), \dot{\phi}_{iql}(r)\right\}}, \tag{3.108}$$

$$\sqrt{\Delta_{iql}} = \sqrt{\omega}\frac{1}{\left\{K_l(r), \dot{\phi}_{iql}(r)\right\}}, \tag{3.109}$$

$$\gamma_{iql} = \frac{\left\{J_l(r), \dot{\phi}_{iql}(r)\right\}}{\left\{K_l(r), \dot{\phi}_{iql}(r)\right\}}, \tag{3.110}$$

where

$$K_l(r) \equiv \left(\frac{\omega}{r}\right)^{l+1}, \tag{3.111}$$

$$J_l(r) \equiv \frac{1}{2(2l+1)}\left(\frac{r}{\omega}\right)^{l}, \tag{3.112}$$

$$\{f_1(r), f_2(r)\} \equiv \left\{r^2\left[f_1(r) f_2'(r) - f_1'(r) f_2(r)\right]\right\}_{r=R_{iq}}, \tag{3.113}$$

and ω is the Wigner–Seitz radius defined by Eq. (3.181).

By using the potential parameters, the potential function $P_{iq}(E)$ and two other quantities $\lambda_{iq}(E)$ and $\mu_{iq}(E)$ are obtained as

$$P_{iq}(E) = \frac{E - C_{iq}}{\Delta_{iq} + (\gamma_{iq} - \alpha)(E - C_{iq})}, \tag{3.114}$$

$$\lambda_{iq}(E) = \frac{\gamma_{iq} - \alpha}{\Delta_{iq} + (\gamma_{iq} - \alpha)(E - C_{iq})}, \tag{3.115}$$

$$\mu_{iq}(E) = \frac{\sqrt{\Delta_{iq}}}{\Delta_{iq} + (\gamma_{iq} - \alpha)(E - C_{iq})}, \tag{3.116}$$

where C_{iq}, Δ_{iq}, γ_{iq}, α are diagonal matrices

$$C_{iq,L_1L_2} = \delta_{L_1L_2} C_{iql_1}, \tag{3.117}$$

$$\Delta_{iq,L_1L_2} = \delta_{L_1L_2} \Delta_{iql_1}, \tag{3.118}$$

$$\gamma_{iq,L_1L_2} = \delta_{L_1L_2} \gamma_{iql_1}, \tag{3.119}$$

$$\alpha_{L_1L_2} = \delta_{L_1L_2} \alpha^{opt}_{l_1}, \tag{3.120}$$

www.ebook3000.com

in which α_l^{opt} is the screening constant and the values are listed in Eq. (3.53).

It is worth mentioning that only valence orbitals are included in the potential parameters and the Green's functions. The core orbitals are assumed to be fully occupied and their influences are taken into account through the charge density (see Section 3.11.9).

3.11.7 *Step-6 solve the NECPA equations*

Having calculated the potential parameters, the coherent potential and the auxiliary Green's function can be solved from the NECPA-LMTO equations.

In two-probe systems, the NECPA-LMTO equations are

$$\begin{cases} \overline{\mathcal{G}_i^r}(E) = \sum_q x_{iq} \overline{\mathcal{G}_{iq}^r}(E), \\ \overline{\mathcal{G}^r}(E,k) = \left[\tilde{P}^r(E) - S(k) - \Sigma^r(E,k)\right]^{-1}, \\ \overline{\mathcal{G}^r}(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \overline{\mathcal{G}^r}(E,k), \\ \overline{\mathcal{G}_i^r}(E) \equiv \left[\overline{\mathcal{G}^r}(E)\right]_{ii}, \\ \overline{\mathcal{G}_i^r}(E) = \left[\tilde{P}_i^r(E) - \Omega_i^r(E)\right]^{-1}, \\ \overline{\mathcal{G}_{iq}^r}(E) = \left[P_{iq}(E) - \Omega_i^r(E)\right]^{-1}, \end{cases} \tag{3.121}$$

$$\begin{cases} \overline{\mathcal{G}_i^<}(E) = \sum_q x_{iq} \overline{\mathcal{G}_{iq}^<}(E), \\ \overline{\mathcal{G}^<}(E,k) = \overline{\mathcal{G}^r}(E,k)\left[-\tilde{P}^<(E) + \Sigma^<(E,k)\right]\overline{\mathcal{G}^a}(E,k), \\ \overline{\mathcal{G}^<}(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \overline{\mathcal{G}^<}(E,k), \\ \overline{\mathcal{G}_i^<}(E) \equiv \left[\overline{\mathcal{G}^<}(E)\right]_{ii}, \\ \overline{\mathcal{G}_i^<}(E) = \overline{\mathcal{G}_i^r}(E)\left[-\tilde{P}_i^<(E) + \Omega_i^<(E)\right]\overline{\mathcal{G}_i^a}(E), \\ \overline{\mathcal{G}_{iq}^<}(E) = \overline{\mathcal{G}_{iq}^r}(E)\,\Omega_i^<(E)\,\overline{\mathcal{G}_{iq}^a}(E), \end{cases} \tag{3.122}$$

where $\tilde{P}_i^{\lambda}(E)$ is the coherent potential and $\Omega_i^{\lambda}(E)$ is the coherent interactor with the superscript $\lambda = r, <$. The 2d Brillouin zone for the dimensionless wave vector is $k_x \in [0, 2\pi]$ and $k_y \in [0, 2\pi]$. $\overline{\mathcal{G}^a}(E,k)$, $\overline{\mathcal{G}_i^a}(E)$, and $\overline{\mathcal{G}_{iq}^a}(E)$ can be obtained by the Hermitian conjugate of their retarded counterparts. Within the NECPA, $\tilde{P}^{\lambda}(E)$ is assumed to be block-diagonal, namely $\tilde{P}_{i_1 i_2}^{\lambda}(E) = \delta_{i_1 i_2}\tilde{P}_{i_1}^{\lambda}(E)$. The unknown variables $\overline{\mathcal{G}_i^{\lambda}}(E)$, $\overline{\mathcal{G}_{iq}^{\lambda}}(E)$, $\tilde{P}_i^{\lambda}(E)$, $\Omega_i^{\lambda}(E)$ can be solved iteratively from the above NECPA-LMTO equations.

In bulk systems, the NECPA-LMTO equations are reduced to the CPA-LMTO equations

$$\begin{cases} \overline{\mathcal{G}_i^r}(E) = \sum_q x_{iq}\overline{\mathcal{G}_{iq}^r}(E), \\ \overline{\mathcal{G}^r}(E,k) = \left[\tilde{P}^r(E) - S(k)\right]^{-1}, \\ \overline{\mathcal{G}^r}(E) = \int_{BZ} \frac{d^3k}{(2\pi)^3}\overline{\mathcal{G}^r}(E,k), \\ \overline{\mathcal{G}_i^r}(E) \equiv \left[\overline{\mathcal{G}^r}(E)\right]_{ii}, \\ \overline{\mathcal{G}_i^r}(E) = \left[\tilde{P}_i^r(E) - \Omega_i^r(E)\right]^{-1}, \\ \overline{\mathcal{G}_{iq}^r}(E) = \left[P_{iq}(E) - \Omega_i^r(E)\right]^{-1}, \end{cases} \tag{3.123}$$

where the self-energy term also vanishes. The 3d Brillouin zone for the dimensionless wave vector is $k_x \in [0, 2\pi]$, $k_y \in [0, 2\pi]$, $k_z \in [0, 2\pi]$.

If the disorder concentration is very low, the solution to the NECPA-LMTO equations can be approximated by an analytical expression. Let $q = 0$ represent the host atom species and $q > 0$ the impurity atom species. Low disorder concentration implies that $x_{i,q=0} \gg x_{i,q>0}$ and $\sum_q x_{iq} = 1$. The coherent potential can be obtained up to the first order of $x_{i,q>0}$

$$\tilde{P}_i^r(E) \approx P_{i0}(E) - \sum_{q>0} x_{iq} t_{iq}^r(E), \tag{3.124}$$

$$\tilde{P}_i^<(E) \approx -\sum_{q>0} x_{iq} t_{iq}^r(E)\,\mathcal{G}_{0,ii}^<(E)\,t_{iq}^a(E), \tag{3.125}$$

where the scattering amplitude $t_{iq}^r(E)$ is defined by

$$t_{iq}^r(E) = \left[\left(P_{i0}(E) - P_{iq}(E)\right)^{-1} - \mathcal{G}_{0,ii}^r(E)\right]^{-1}, \tag{3.126}$$

www.ebook3000.com

in which $P_{i0}(E)$ and $P_{iq}(E)$ are the potential functions of the host atom species and the impurity atom species respectively. The unperturbed auxiliary Green's function $\mathcal{G}_0^r(E)$ and $\mathcal{G}_0^<(E)$ are defined by

$$\mathcal{G}_0^r(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathcal{G}_0^r(E,k), \tag{3.127}$$

$$\mathcal{G}_0^<(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathcal{G}_0^r(E,k)\,\Sigma^<(E,k)\,\mathcal{G}_0^a(E,k), \tag{3.128}$$

$$\mathcal{G}_0^r(E,k) = \left[P_0(E) - S(k) - \Sigma^r(E,k)\right]^{-1}. \tag{3.129}$$

3.11.8 *Step-7 calculate energy moment*

Having solved the auxiliary Green's function from the NECPA-LMTO equations, the diagonal blocks of physical Green's function can be calculated as follows.

For two-probe systems, $\overline{G_{iq}^r}(E)$ and $\overline{G_{iq}^<}(E)$ can be obtained as

$$\overline{G_{iq}^r}(E) = \lambda_{iq}(E) + \mu_{iq}(E)\,\overline{\mathcal{G}_{iq}^r}(E)\,\mu_{iq}(E), \tag{3.130}$$

$$\overline{G_{iq}^<}(E) = \mu_{iq}(E)\,\overline{\mathcal{G}_{iq}^<}(E)\,\mu_{iq}(E), \tag{3.131}$$

where $\lambda_{iq}(E)$ and $\mu_{iq}(E)$ are defined by Eqs. (3.115,3.116).

For bulk systems, $\overline{G_{iq}^r}$ can be calculated with Eq. (3.130), and $\overline{G_{iq}^<}$ can be calculated with the fluctuation-dissipation theorem

$$\overline{G_{iq}^<}(E) = f(E)\left[\overline{G_{iq}^a}(E) - \overline{G_{iq}^r}(E)\right], \tag{3.132}$$

where $f(E)$ is the Fermi function and $\overline{G_{iq}^a}(E) = \left[\overline{G_{iq}^r}(E)\right]^\dagger$.

The energy moment $\tilde{M}_{iq}^k$ is calculated by an energy integral of $\overline{G_{iq}^<}(E)$

$$\tilde{M}_{iq}^k \equiv \int \frac{dE}{2\pi}(-\mathrm{i})\,\overline{G_{iq}^<}(E)\,E^k. \tag{3.133}$$

The double energy moment $M_{iq}^{k_1k_2}$ is defined by

$$M_{iq,L_1L_2}^{k_1k_2} \equiv \int \frac{dE}{2\pi}(-\mathrm{i})\left[\overline{G_{iq}^<}(E)\right]_{L_1L_2}\left(E - E_{iql_1}^0\right)^{k_1}\left(E - E_{iql_2}^0\right)^{k_2}, \tag{3.134}$$

which can be reduced to a linear combination of $\tilde{M}_{iq}^k$.

3.11.9 *Step-8 calculate charge density*

The charge density $\rho_{iq}(r)$ of the atomic site-i species-q can be calculated by using the atomic orbitals and the energy moments. $\rho_{iq}(r)$ has contributions

from both the core electron density and the valence electron density

$$\rho_{iq}(r) = \rho_{iq}^{core}(r) + \rho_{iq}^{val}(r), \tag{3.135}$$

$$\rho_{iq}^{core}(r) = \frac{1}{4\pi}\sum_{nl}(2l+1)\left[\phi_{iqnl}^{core}(r)\right]^2, \tag{3.136}$$

$$\rho_{iq}^{val}(r) = \frac{1}{4\pi}\sum_{L} M_{iq,LL}^{00}\phi_{iql}^2(r) + 2M_{iq,LL}^{10}\phi_{iql}(r)\,\dot{\phi}_{iql}(r) + M_{iq,LL}^{20}\left[\dot{\phi}_{iql}^2(r) + \phi_{iql}(r)\,\ddot{\phi}_{iql}(r)\right]. \tag{3.137}$$

Similarly the kinetic energy density $t_{iq}(r)$ of the atomic site-i species-q is obtained as [6]

$$t_{iq}(r) = t_{iq}^{core}(r) + t_{iq}^{val}(r), \tag{3.138}$$

$$t_{iq}^{core}(r) = \frac{1}{4\pi}\frac{1}{2}\sum_{nl}\sum_{\alpha=A,B}(2l+1)\,\chi_{iqnl}^{core,\alpha}(r)^2, \tag{3.139}$$

$$t_{iq}^{val}(r) = \frac{1}{4\pi}\frac{1}{2}\sum_{L}\sum_{\alpha=A,B} M_{iq,LL}^{00}\chi_{iql}^{\alpha}(r)^2 + 2M_{iq,LL}^{10}\chi_{iql}^{\alpha}(r)\dot{\chi}_{iql}^{\alpha}(r) + M_{iq,LL}^{20}\left[\dot{\chi}_{iql}^{\alpha}(r)^2 + \chi_{iql}^{\alpha}(r)\ddot{\chi}_{iql}^{\alpha}(r)\right], \tag{3.140}$$

where $\chi_{iqnl}^{core,\alpha}(r)$ and $\chi_{iql}^{\alpha}(r)$ are defined by

$$\chi_{iqnl}^{core,A}(r) \equiv \frac{d}{dr}\phi_{iqnl}^{core}(r),$$
$$\chi_{iqnl}^{core,B}(r) \equiv \frac{\sqrt{l(l+1)}}{r}\phi_{iqnl}^{core}(r),$$
$$\chi_{iql}^{A}(r) \equiv \phi_{iql}'(r),$$
$$\chi_{iql}^{B}(r) \equiv \frac{\sqrt{l(l+1)}}{r}\phi_{iql}(r).$$

3.11.10 *Step-9 calculate charge and dipole*

The averaged charge and dipole of the i-th atomic sphere can be calculated by using the atomic orbitals and energy moments. The averaged charge Q_i is obtained as

$$Q_i = \sum_q x_{iq}\left(\sum_L \mathrm{Tr}\; M_{iq,LL}^{00} - Z_{iq}^{val}\right), \tag{3.141}$$
$$Z_{iq}^{val} \equiv Z_{iq} - Z_{iq}^{core},$$

www.ebook3000.com

where Z_{iq} and Z_{iq}^{core} are the total electron number and the core electron number of the atomic site-i species-q, respectively. The averaged dipole $\mathbf{P}_i$ is obtained as

$$\mathbf{P}_i = \sum_q x_{iq} \int_0^{R_{iq}} r^3 dr \sum_{L_1 L_2} \mathbf{\Theta}_{L_1 L_2} F_{iq,L_1 L_2}(r), \tag{3.142}$$

where $\mathbf{\Theta}_{L_1 L_2}$ is defined by

$$\mathbf{\Theta}_{L_1 L_2} = \sqrt{\frac{4\pi}{3}} \left[C_{(1,1)L_1 L_2}, C_{(1,-1)L_1 L_2}, C_{(1,0)L_1 L_2} \right], \tag{3.143}$$

and $F_{iq,L_1 L_2}(r)$ is defined by

$$\begin{aligned} F_{iq,L_1 L_2}(r) = & M^{00}_{iq,L_1 L_2} \phi_{iql_1}(r) \phi_{iql_2}(r) \\ & + M^{10}_{iq,L_1 L_2} \dot{\phi}_{iql_1}(r) \phi_{iql_2}(r) + M^{01}_{iq,L_1 L_2} \phi_{iql_1}(r) \dot{\phi}_{iql_2}(r) \\ & + M^{11}_{iq,L_1 L_2} \dot{\phi}_{iql_1}(r) \dot{\phi}_{iql_2}(r) + \frac{1}{2} M^{20}_{iq,L_1 L_2} \ddot{\phi}_{iql_1}(r) \phi_{iql_2}(r) \\ & + \frac{1}{2} M^{02}_{iq,L_1 L_2} \phi_{iql_1}(r) \ddot{\phi}_{iql_2}(r). \end{aligned} \tag{3.144}$$

The Gaunt coefficient $C_{LL_1L_2}$ has been defined by Eq. (3.92).

3.11.11 *Step-10 calculate atomic potential with DFT*

Given the atomic charge density $\rho_{iq}(r)$, the atomic potential can be calculated by

$$V_Z(r) = -\frac{Z_{iq}}{r}, \tag{3.145}$$

$$V_H(r) = \int_0^r \frac{4\pi r'^2 \rho_{iq}(r')}{r} dr' + \int_r^{R_{iq}} \frac{4\pi r'^2 \rho_{iq}(r')}{r'} dr', \tag{3.146}$$

$$V_{XC}(r) = V_{XC}\left[\rho_{iq}(r)\right], \tag{3.147}$$

where V_Z is the Coulomb potential of the nuclear, V_H is the Hartree potential of the electrons, and V_{XC} is the XC-potential of the electrons. The expression of V_{XC} depends on the type of XC-functional. Some commonly used XC-functionals are provided in Appendix A.9.

3.11.12 *Step-11 calculate Madelung potential*

Given the atomic charge Q_i and dipole $\mathbf{P}_i$, the Madelung potential can be calculated by

$$V_{MD} = \sum_{j \neq i} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{j \neq i} \frac{\mathbf{P}_j \cdot (\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}, \tag{3.148}$$

which takes into account the electrostatic potential from other atom spheres.

There is an arbitrary constant that can be added to the Madelung potential. For bulk systems, the constant does not matter. For two-probe systems, the constant is chosen such that the Madelung potential of the central region lines up with the Madelung potentials of the leads.

3.11.13 *Step-12 calculate total potential*

The potential $V_{iq}(r)$ is the summation of the atomic potential and the Madelung potential

$$V_{iq}(r) = V_Z(r) + V_H(r) + V_{XC}(r) + V_{MD}, \tag{3.149}$$

where V_Z, V_H, V_{XC}, V_{MD} have been obtained in step-10 and step-11.

3.12 Post-analysis calculation

Once the Hamiltonian is obtained from the self-consistent calculation, one can proceed to analyze the electronic structure and the transport property of the system. In this section, the formulas for various post-analysis calculations are summarized.

3.12.1 *Density of states*

The density of states calculation is applicable to both bulk and two-probe systems. The disorder averaged density of states $\overline{D}(E)$ is obtained as

$$\overline{D}(E) = -\frac{1}{\pi} \mathrm{Im} \sum_{iq} x_{iq} \mathrm{Tr}\, \overline{G^r_{iq}}(E), \tag{3.150}$$

where $\overline{G^r_{iq}}(E)$ is calculated with $\overline{\mathcal{G}^r_{iq}}(E)$ by using Eq. (3.130). $\overline{\mathcal{G}^r_{iq}}(E)$ can be solved from the CPA-LMTO equations (3.123) for bulk systems or the NECPA-LMTO equations (3.121) for two-probe systems.

www.ebook3000.com

3.12.2 *Transmission coefficient*

The transmission coefficient calculation is only applicable to two-probe systems. The disorder-averaged transmission coefficient $\overline{T}(E)$ is obtained as

$$\begin{aligned}\overline{T}(E) &= \mathrm{Tr}\left[\overline{\mathcal{G}^r(E)\Gamma_L(E)\mathcal{G}^a(E)\Gamma_R(E)}\right]_{00} \\ &= \int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\ (-\mathrm{i})\,\overline{\mathcal{G}^r\mathrm{i}\Gamma_L\mathcal{G}^a}(E,k)\,\Gamma_R(E,k) \\ &= \int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\ (-\mathrm{i})\,\overline{\mathcal{G}_L^<}(E,k)\,\Gamma_R(E,k)\,. \end{aligned} \tag{3.151}$$

The linewidth function $\Gamma_\beta(E,k)$ is defined by

$$\Gamma_\beta(E,k) \equiv -\mathrm{i}\left[\Sigma_\beta^a(E,k) - \Sigma_\beta^r(E,k)\right], \tag{3.152}$$

where $\Sigma_\beta^r(E,k)$ is the lead self-energy defined by Eq. (3.99). The left lesser auxiliary Green's function $\overline{\mathcal{G}_L^<}(E,k)$ is defined by

$$\overline{\mathcal{G}_L^<}(E,k) \equiv \left[\overline{\mathcal{G}^<}(E,k)\right]_{f_L=1,\ f_R=0},$$

which can be solved from the NECPA-LMTO equations (3.121,3.122) by assigning $f_L(E)=1$ and $f_R(E)=0$. Eq. (3.151) represents the transmission coefficient of scattering waves from the left lead to the right lead.

Alternatively the transmission coefficient can also be calculated by

$$\begin{aligned}\overline{T}(E) &= \mathrm{Tr}\left[\overline{\mathcal{G}^r(E)\Gamma_R(E)\mathcal{G}^a(E)\Gamma_L(E)}\right]_{00} \\ &= \int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\ (-\mathrm{i})\,\overline{\mathcal{G}^r\mathrm{i}\Gamma_R\mathcal{G}^a}(E,k)\,\Gamma_L(E,k) \\ &= \int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\ (-\mathrm{i})\,\overline{\mathcal{G}_R^<}(E,k)\,\Gamma_L(E,k)\,. \end{aligned} \tag{3.153}$$

The right lesser auxiliary Green's function $\overline{\mathcal{G}_R^<}(E,k)$ is defined by

$$\overline{\mathcal{G}_R^<}(E,k) \equiv \left[\overline{\mathcal{G}^<}(E,k)\right]_{f_R=1\ f_L=0},$$

which can be solved from the NECPA-LMTO equations (3.121,3.122) by assigning $f_R(E)=1$ and $f_L(E)=0$. Eq. (3.153) represents the transmission coefficient of scattering waves from the right lead to the left lead. It can be proved that Eq. (3.151) and Eq. (3.153) are equivalent due to current conservation (see Section 2.9).

The transmission coefficient can be further decomposed into the specular part and the diffusive part. By using the second line of Eq. (3.122), $\overline{\mathcal{G}_L^<}(E,k)$ and $\overline{\mathcal{G}_R^<}(E,k)$ are obtained as

$$\overline{\mathcal{G}_L^<}(E,k) = \overline{\mathcal{G}^r}(E,k)\left[\mathrm{i}\Gamma_L(E,k) + \tilde{P}_L^<(E)\right]\overline{\mathcal{G}^a}(E,k)\,, \tag{3.154}$$

$$\overline{\mathcal{G}_R^<}(E,k) = \overline{\mathcal{G}^r}(E,k)\left[\mathrm{i}\Gamma_R(E,k) + \tilde{P}_R^<(E)\right]\overline{\mathcal{G}^a}(E,k)\,. \tag{3.155}$$

Consequently the transmission coefficient can be decomposed as

$$\overline{T}(E) = \overline{T}_s(E) + \overline{T}_d(E), \tag{3.156}$$

where $\overline{T}_s(E)$ and $\overline{T}_d(E)$ are defined by

$$\overline{T}_s(E) \equiv \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\left[\overline{\mathcal{G}^r}(E,k)\,\Gamma_L(E,k)\,\overline{\mathcal{G}^a}(E,k)\,\Gamma_R(E,k)\right], \tag{3.157}$$

$$\overline{T}_d(E) \equiv \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\left[\overline{\mathcal{G}^r}(E,k)\,\Lambda_L(E)\,\overline{\mathcal{G}^a}(E,k)\,\Gamma_R(E,k)\right], \tag{3.158}$$

for the left to right scattering wave, and $\overline{T}_s(E)$ and $\overline{T}_d(E)$ are defined by

$$\overline{T}_s(E) \equiv \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\left[\overline{\mathcal{G}^r}(E,k)\,\Gamma_R(E,k)\,\overline{\mathcal{G}^a}(E,k)\,\Gamma_L(E,k)\right], \tag{3.159}$$

$$\overline{T}_d(E) \equiv \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\left[\overline{\mathcal{G}^r}(E,k)\,\Lambda_R(E)\,\overline{\mathcal{G}^a}(E,k)\,\Gamma_L(E,k)\right], \tag{3.160}$$

for the right to left scattering wave. Here $\Lambda_\beta(E) \equiv -\mathrm{i}\tilde{P}^<_\beta(E)$ is also referred to as the vertex correction. $\overline{T}_s(E)$ is called the specular transmission coefficient whose scattering wave conserves the transverse momentum; $\overline{T}_d(E)$ is called the diffusive transmission coefficient whose scattering wave does not conserve the transverse momentum. The physical meaning of the two terms will be further explained in Section 8.5 and Appendix A.16 by connecting them to the scattering states approach.

If the impurity concentration is very low, the transmission coefficient formula, Eq. (3.151), can be simplified by using the low concentration approximation. Assume that each disorder site is occupied mainly by the host atom ($q = 0$) and that the concentration of impurity atoms ($q > 0$) is very low. It is assumed that $x_{i0} \gg x_{i,q>0}$ and $\sum_q x_{iq} = 1$. $\overline{T}(E)$ can be calculated analytically by a perturbative expansion up to $O(x_{i,q>0})$ (see Appendix A.5)

$$\overline{T}(E) \approx T_0(E) + \sum_{i,q>0} x_{iq}\left[Y^\alpha_{iq}(E) + Y^\beta_{iq}(E) + Y^\gamma_{iq}(E)\right], \tag{3.161}$$

where T_0 is the transmission coefficient in the clean limit

$$T_0(E) = \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\left[\mathcal{G}^r_0(E,k)\,\Gamma_L(E,k)\,\mathcal{G}^a_0(E,k)\,\Gamma_R(E,k)\right], \tag{3.162}$$

www.ebook3000.com

and Y_{iq}^{α}, Y_{iq}^{β}, and Y_{iq}^{γ} are corrections due to the disorder scattering by the atomic site-i species-q

$$Y_{iq}^{\alpha}(E)=\int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\left\{t_{iq}^{a}(E)[\mathcal{G}_0^{a}(E,k)\Gamma_R(E,k)\mathcal{G}_0^{r}(E,k)\Gamma_L(E,k)\mathcal{G}_0^{a}(E,k)]_{ii}\right\}, \tag{3.163}$$

$$Y_{iq}^{\beta}(E)=\int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\left\{t_{iq}^{r}(E)[\mathcal{G}_0^{r}(E,k)\Gamma_L(E,k)\mathcal{G}_0^{a}(E,k)\Gamma_R(E,k)\mathcal{G}_0^{r}(E,k)]_{ii}\right\}, \tag{3.164}$$

$$Y_{iq}^{\gamma}(E)=\int_{BZ}\frac{d^2k}{(2\pi)^2}\mathrm{Tr}\left\{t_{iq}^{r}(E)\left[\mathcal{G}_0^{r}(E,k)\Gamma_L(E,k)\mathcal{G}_0^{a}(E,k)\right]_{ii}t_{iq}^{a}(E)\left[\mathcal{G}_0^{a}(E,k)\Gamma_R(E,k)\mathcal{G}_0^{r}(E,k)\right]_{ii}\right\}, \tag{3.165}$$

in which $t_{iq}^{r}(E)$ and $\mathcal{G}_0^{r}(E,k)$ are defined by Eqs. (3.126,3.129) respectively. It is worth mentioning that $Y_{iq}^{\alpha}=\left(Y_{iq}^{\beta}\right)^{*}$ and $Y_{iq}^{\gamma}=\left(Y_{iq}^{\gamma}\right)^{*}$. Notice that the summation over i,q in Eq. (3.161) indicates the contribution from the atomic site-i and species-q.

Finally the electric current flowing through a two-probe system can be calculated by integrating the transmission coefficient

$$\overline{I}\equiv\overline{I_L}=-\overline{I_R}=Q_e\int\frac{dE}{2\pi}\overline{T}(E)\left[f_L(E)-f_R(E)\right], \tag{3.166}$$

where $f_L(E)$ and $f_R(E)$ are the Fermi functions of the left and right leads. The factor $Q_e=-1$ is to take into account the electron having a negative charge.

3.12.3 *Transmission variation*

The transmission variation calculation is only applicable to two-probe systems. Besides the statistical average, one may also be interested in the variation of transmission coefficient due to disorder scattering. The transmission variation δT is defined by

$$\delta T(E)\equiv\sqrt{\overline{T^2(E)}-\overline{T(E)}^2}. \tag{3.167}$$

Although δT can be calculated by using the CPA diagrammatic technique [7], the calculation is too costly for an atomic simulation.

If the impurity concentration is very low, the calculation of transmission variation can be greatly simplified by using the low concentration approximation. Assume that each disorder site is occupied mainly by the host

atom ($q = 0$) and that concentration of impurity atoms ($q > 0$) is very low. It is assumed that $x_{i0} \gg x_{i,q>0}$ and $\sum_q x_{iq} = 1$. $\delta\overline{T}(E)$ can be calculated analytically by a perturbative expansion up to $O(x_{i,q>0})$ (see Appendix A.5)

$$\delta T = \sqrt{\sum_{i,q>0} x_{iq} \left(Y_{iq}^{\alpha} + Y_{iq}^{\beta} + Y_{iq}^{\gamma}\right)^2}, \tag{3.168}$$

where Y_{iq}^{α}, Y_{iq}^{β}, Y_{iq}^{γ} have been defined by Eqs. (3.163,3.164,3.165). Notice that the summation over i, q in Eq. (3.168) indicates the contribution from the atomic site-i and species-q.

It is worth mentioning that Eq. (3.151) and Eq. (3.168) are derived for transmission coefficient and transmission variation of a single unit cell in the transverse dimensions. For a supercell containing $\mathcal{N}$ unit cells, $\overline{T}$ scales with $\mathcal{N}$ and δT scales with $\sqrt{\mathcal{N}}$.

Finally the current variation can be estimated with the aid of transmission variation

$$\delta\overline{I} \approx Q_e \int \frac{dE}{2\pi} \delta\overline{T}(E) \left[f_L(E) - f_R(E)\right], \tag{3.169}$$

where $f_L(E)$ and $f_R(E)$ are the Fermi functions of the left and right leads.

3.12.4 *Band structure*

The band structure calculation is only applicable to clean bulk systems. The band structure is a plot of eigenvalues as a function of wave vector k, where k is on the lines connecting symmetry points. The eigenvalue problem is defined by

$$\left[E - H_{orth}(k)\right] \Psi = 0, \tag{3.170}$$

$$H_{orth}(k) \equiv C + \sqrt{\Delta} \left[S^{-1}(k) - (\gamma - \alpha)\right]^{-1} \sqrt{\Delta}, \tag{3.171}$$

where C, Δ, γ, α are the potential parameter matrix (diagonal) and screening constant matrix (diagonal) defined by Eqs. (3.117,3.118,3.119,3.120). Notice that the species index q should be set to 0 for clean systems.

3.12.5 *CPA band structure*

The CPA band structure is only applicable to disordered bulk systems. In the presence of random disorder, there is no rigorous periodicity in bulk

www.ebook3000.com

systems, and the concept of band structure is no longer valid. On the other hand, if the doping concentration is not too high, the electronic structure is still very close to that of the host material. The effects of doping are two-fold: Modify the shape of band structure slightly and induce a finite lifetime to each Bloch state. Therefore it is necessary to generalize the concept of band structure of clean bulk systems to CPA band structure of disordered bulk systems.

Although the Hamiltonian is not periodic, the Green's function after disorder-average still has translational symmetry. One can calculate the physical Green's function $\overline{G^r}(E,k)$ and define the k-resolved density of states (also known as the Bloch spectral function)

$$D(E,k) \equiv -\frac{1}{\pi}\mathrm{Im\ Tr}\,\overline{G^r}(E,k). \tag{3.172}$$

After some algebra [4], $D(E,k)$ is reduced to

$$D(E,k) = -\frac{1}{\pi}\sum_i \mathrm{Im\ Tr}\Big[\Lambda_i(E) - \tilde{M}_i(E)\,\overline{\mathcal{G}_i^r}(E)\,M_i(E) + \tilde{M}_i(E)\,\overline{\mathcal{G}_i^r}(E,k)\,M_i(E)\Big], \tag{3.173}$$

$$\Lambda_i(E) = \sum_q x_{iq}\left[\lambda_{iq}(E) + \mu_{iq}(E)\,\overline{\mathcal{G}_{iq}^r}(E)\,\mu_{iq}(E)\right], \tag{3.174}$$

$$\tilde{M}_i(E) = \sum_q x_{iq}\mu_{iq}(E)\,\overline{\mathcal{G}_{iq}^r}(E)\left[\overline{\mathcal{G}_i^r}(E)\right]^{-1}, \tag{3.175}$$

$$M_i(E) = \sum_q x_{iq}\left[\overline{\mathcal{G}_i^r}(E)\right]^{-1}\overline{\mathcal{G}_{iq}^r}(E)\,\mu_{iq}(E), \tag{3.176}$$

where λ_{iq} and μ_{iq} have been defined by Eqs. (3.115,3.116). $\overline{\mathcal{G}_i^r}(E)$, $\overline{\mathcal{G}_{iq}^r}(E)$ and $\overline{\mathcal{G}_i^r}(E,k) \equiv \left[\overline{\mathcal{G}^r}(E,k)\right]_{ii}$ are to be solved from the CPA-LMTO equations (3.123).

$D(E,k)$ is a good substitute of the band structure in disordered bulk systems: In the clean limit, $D(E,k)$ is a summation of δ-functions over the Bloch states defined by Eq. (3.170). In the presence of disorder, the δ-functions are broadened due to disorder scattering and the traces of the broadened δ-functions constitute the CPA band structure.

3.13 Miscellaneous issues

There are some miscellaneous issues that were not covered or not fully addressed in Section 3.11 and 3.12. In this section we shall discuss those issues including spin degree of freedom, Fermi level, linearization center, scalar relativistic equation, and Wigner–Seitz radius.

3.13.1 *Spin degree of freedom*

For simplicity of notation, the electron's spin was not considered in the formulas of Section 3.11 and 3.12. So the variables in those sections should be understood as the variables of one spin species. It is straightforward to include the spin degree of freedom by adding the spin index σ to the Green's functions and the physical quantities. For example, by including the spin degree of freedom, we have spin-resolved potential parameter $C_{iql\sigma}$, spin-resolved atomic orbital $\Psi_{iql\sigma}$, spin-resolved atomic potential $V_{i\sigma}(r)$, spin-resolved charge density $\rho_{i\sigma}(r)$, spin-resolved Green's function $\overline{G_{i\sigma}}(E)$, spin-resolved transmission coefficient $T_{\sigma}(E)$, etc.

Throughout the monograph, it is assumed that electron's spin has a global polarization direction. The assumption is referred to as collinear spin polarization in the literature. In that situation, a variable X is split into $X_{\uparrow}$ and $X_{\downarrow}$ to represent spin-$\uparrow$ and spin-$\downarrow$ respectively. Calculations are carried out almost in parallel for the two spin species. The only coupling between spin-$\uparrow$ and spin-$\downarrow$ variables occurs in the exchange-correlation potential

$$[V_{XC\uparrow}(r), V_{XC\downarrow}(r)] = V_{XC}[\rho_{iq\uparrow}(r), \rho_{iq\downarrow}(r)], \tag{3.177}$$

where V_{XC} not only depends on the charge density but also on the spin polarization. Therefore Eq. (3.147) needs to be replaced by Eq. (3.177).

A special case of the collinear spin polarization is that the system is not spin polarized at all. It is unnecessary to calculate two sets of variables for spin-$\uparrow$ and spin-$\downarrow$ since they are exactly the same. Instead one only needs to work with one set of the variables and multiply the results by a spin degeneracy factor 2 in the calculations of occupation number and electric current.

3.13.2 *Fermi level*

In bulk systems, the position of the Fermi level is determined by the charge neutrality of the unit cell. Electrons are filled into the band structure all the way to the Fermi level, and the number of electrons should be equal to the number of nuclei. Therefore the position of the Fermi level is determined by the condition that the total charge of the unit cell is zero

$$\sum_i Q_i = 0, \tag{3.178}$$

where Q_i is defined by Eq. (3.141).

In two-probe systems, the concept of the Fermi level is no longer valid. The nonequilibrium statistics of the central region is determined by the left

www.ebook3000.com

and right leads which are in local equilibrium. At zero bias voltage, the whole system is in a global equilibrium, and the lead Fermi levels line up with each other, namely $\mu_L = \mu_R$. By applying a bias voltage $V = V_L - V_R$, the lead Fermi levels are shifted to

$$\mu_L - \mu_R = Q_e V, \tag{3.179}$$

where $Q_e = -1$ is to take into account the electron having a negative charge. As a result, the energy zeros of the lead Hamiltonians need to be shifted so that the lead Fermi levels satisfy Eq. (3.179).

3.13.3 *Linearization center*

In Eqs. (3.105,3.106,3.107), the MTO $\Psi_{ilq}(r, E)$ is expanded around the energy E^0_{ilq} which is referred to as the linearization center. How do we choose a proper linearization center for each orbital?

In self-consistent calculations, it is natural to define the linearization center E^0_{ilq} by the ratio of energy moments

$$E^0_{ilq} \equiv \frac{\sum_m \int \frac{dE}{2\pi} (-\mathrm{i}) \left[\overline{G^<_{iq}}(E)\right]_{LL} E}{\sum_m \int \frac{dE}{2\pi} (-\mathrm{i}) \left[\overline{G^<_{iq}}(E)\right]_{LL}} = \frac{\sum_m \tilde{M}^1_{iq,LL}}{\sum_m \tilde{M}^0_{iq,LL}}, \tag{3.180}$$

which is an average of the valence band energy weighted by the occupation number. Since self-consistent calculations are mainly determined by the occupation of the valence bands, the above E^0_{ilq} is a good choice for the purpose of self-consistent calculations.

In post-analysis calculations (e.g., transmission coefficient), not only valence bands but also conduction bands may have impact on the physical quantities. Nevertheless one may still use the same linearization center as the one in the self-consistent calculation and see if the linearization approximation is acceptable within the energy range of interest. It turns out that E^0_{ilq} defined by Eq. (3.180) is pretty good for most metallic materials not only in the self-consistent calculations but also in the post-analysis calculations. Sometimes, however, the linearization may lead to considerable error in the post-analysis calculations of some semiconductors and insulators. In that situation, one needs to abandon the linearization approximation and replace E^0_{ilq} by E in Eqs. (3.105,3.106,3.107,3.108). By making the replacement in the LMTO, the formalisms of the MTO are recovered, and the error due to the linearization is eliminated completely.

3.13.4 *Scalar relativistic equation*

The Kohn–Sham equation (3.1) is a Schrödinger-like equation. One may expect that in heavy atoms the core electrons move so fast that relativistic effects are not negligible. To take into account the relativistic effects, one can replace the Schrödinger-like equation with the scalar relativistic equation [8]. The scalar relativistic equation makes some relativistic corrections proportional to α_c^2 where α_c is the fine structure constant. On the one hand, the scalar relativistic equation takes into account the most important relativistic effects (e.g., the mass-velocity effect and the Darwin effect) except for the spin-orbital interaction. On the other hand, the complexity of the scalar relativistic equation is nearly the same as that of the Schrödinger equation.

3.13.5 *Wigner–Seitz radius*

A constant ω appears in the expressions of $J_l(r)$ and $K_l(r)$ and the canonical structure constant $\tilde{S}_{i_1L_1,i_2L_2}$ (see Eqs.(3.111,3.112,3.91)). Since ω is just a normalization factor, it is allowed to take arbitrary value and even be site-dependent. In the screening transform, however, it is assumed that ω adopts the Wigner–Seitz radius because the screening constant α_l^{opt} in Eq. (3.53) is optimized for such convention. In bulk systems, the Wigner–Seitz radius is defined by

$$\frac{4\pi}{3}\omega^3 = \frac{V}{N}, \tag{3.181}$$

where V is the unit cell volume and N is the number of atomic sites. In two-probe systems, the Wigner–Seitz radius is not well defined. It is recommended to use a unified ω in the left lead, the right lead, and the central region, namely

$$\omega_L = \omega_R = \omega_C = \omega_0, \tag{3.182}$$

where ω_0 is an averaged Wigner–Seitz radius.

Bibliography

[1] K. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[2] O. K. Andersen, in *The Electronic Structure of Complex Systems*, edited by P. Phariseau and W. M. Temmerman (Plenum, New York, 1984).
[3] H. L. Skriver, *The LMTO Method* (Springer, Berlin, 1984).
[4] I. Turek, V. Drchal, J. Kudrnovský, M. Šob, and P. Weinberger, *Electronic Structure of Disordered Alloys, Surfaces and Interfaces* (Kluwer Academic, Dordrecht, 1997).

www.ebook3000.com

[5] O. K. Andersen, O. Jepsen, PRL **53**, 2571 (1984).
[6] Analogously to the charge density $\rho(\mathbf{r}) = \sum_i \psi_i^*(\mathbf{r})\,\psi_i(\mathbf{r})$, the kinetic energy density is defined by $t(\mathbf{r}) = \frac{1}{2}\sum_i \nabla\psi_i^*(\mathbf{r}) \cdot \nabla\psi_i(\mathbf{r})$. The derivation of Eqs. (3.138,3.139,3.140) is similar to that of Eqs. (3.135,3.136,3.137).
[7] Y. Zhu, L. Liu, and H. Guo, Phys. Rev. B **88**, 085420 (2013).
[8] D. D. Koelling and B. N. Harmon, J. Phys. C: Solid State Phys. **10**, 3107 (1977).

Chapter 4

NanoDsim: the package design

NanoDsim, short for nanoelectronic device simulator, is a modeling package for atomistic simulation of quantum transport and electronic properties of nanostructures. The theoretical foundation of NanoDsim has been presented in Chapter 2 to 3. The goal of Chapter 4 to 7 is to convert the formalism of Section 3.11 and 3.12 into a software package. Therefore Chapter 2 to 3 read more like a physics book while Chapter 4 to 7 read more like a computer programming book. Particularly in this chapter we shall explain the programming language, the programming style, and the design of NanoDsim.

4.1 Do you speak MATLAB?

When you talk to a human, you may speak English, French, Spanish, Mandarin, etc. When you communicate with a computer, you may speak C, Java, PHP, C++, Python, etc. Those computer languages have been widely used in writing softwares and websites. But what is the proper language for the purpose of scientific computing? Fortran used to be the most popular language among the senior generation of scientists and is still being used by many researchers. Nevertheless we would like to recommend a new language of scientific computing called **MATLAB** to the younger generation of scientists. In fact, MATLAB is considered to be a fourth generation language while Fortran belongs to the third.

What makes MATLAB so unique for scientific computing? Let us first take a look at the human language. Nearly all languages have similar elements and structures (noun, verb, adjective, etc.). The real difference comes from the culture that is linked to a specific language. A language may have a large number of phrases, idioms, stories, and literature, which make an

www.ebook3000.com

expression short, vivid and precise. The culture of human language roughly corresponds to the supporting libraries and functions of computer language. For scientific computing, MATLAB does have rich libraries, toolboxes, and functions. As a result, a program written in MATLAB is extremely short and simple.

Another important criterion for scientific computing language is the capability to avoid bugs. "To err is human". Human programmer keeps on making mistakes and creating bugs in every M lines of coding (M depends on the programmer's experience and skills). MATLAB provides an excellent platform to lower the frequency of bug occurrence: (1) The code written in MATLAB is short and hence you have less chances to make mistakes; (2) MATLAB has a well-designed debugger which allows you to find bugs quickly; (3) MATLAB is a scripting language which allows one to write a code from scratch and experiment.

A common misconception is that MATLAB is only good for small research projects but not suitable for serious software development. In this monograph, we attempt to demonstrate that MATLAB is equally good for writing complicated computing software like NanoDsim. MATLAB provides a user-friendly graphic interface which integrates text editor window, file system window, command line window, figure plot window, helper window, etc. MATLAB provides a convenient debugging tool and profiling tool which are very useful for software development. MATLAB supports object oriented programming which will be discussed further in Section 4.4. Finally software written in MATLAB can be compiled into a standalone application which can run without the MATLAB environment.

One may wonder if a code written in MATLAB is slower than other "low-level" languages such as C and Fortran. This is not necessarily true. On the one hand, as a scripting language MATLAB does have an overhead to interpret the scripting commands. On the other hand, the problem can be bypassed by using the vectorization technique and hybrid programming. Before considering those advanced techniques, you just write your code straightforwardly. If the calculation is already very fast, you don't need to care about the difference between 0.01s and 1s. Your time is much more precious than the computer's. If the calculation turns out to be very slow, you have to worry about the difference between 1 hour and 10 days. Now you need to find the bottlenecks of your code using MATLAB's profiler. Usually the bottlenecks of a long code are less than 10% in terms of coding lines. So you don't have to work on the whole code, you only focus on the performance of the 10%. For those bottlenecks, you may use

the vectorization technique (see Section 4.2) or replace it with a low-level function written in C or Fortran (see Section 4.3). Theoretically a code written in MATLAB should be as efficient as any code written in C or Fortran.

The best way to learn a new language is through examples. Below are 10 simple examples to demonstrate the power of MATLAB

Example01 solve linear equation array mm*xx = bb

```
mm = [1+i, 2, 3; 1, 2+i, 3; 1, 2, 3+i];    %coefficient matrix
bb = [1; 2; 3];                             %constant vector
xx = linsolve(mm, bb)                       %solution vector
```

Example02 solve nonlinear equation f(x) = 0

```
f = @(x) cos(x) - x;                        %definition of f(x)
fzero(f, [0, pi])                           %root in [0, pi]
```

Example03 calculate matrix eigenvalue

```
mm = [0, 1, 0; 1, 0, 1; 0, 1, 0];           %matrix
ee = eig(mm)                                %eigenvalues
```

Example04 numerical integral

```
f = @(x) sin(x);                            %integrand
quad(f, 0, pi)                              %result
```

Example05 numerical minimization

```
f = @(x) (x-1) .^2 + 6;                     %cost function
[x0, y0] = fminbnd(f, -3, 3)                %minimum in [-3, 3]
```

Example06 polynomial fitting

```
xx = linspace(-1, 2, 101);                  %x data
yy = -xx.^3 + xx.^2 + xx - 1;               %y data
dd = 0.1 * randn(size(xx));                 %random error
n = 6;                                      %polynomial degree
pp = polyfit(xx, yy+dd, n);                 %polynomial fit
```

www.ebook3000.com

```
yy_fit = polyval(pp, xx);                    %fitting curve
figure                                       %plot
hold on
scatter(xx, yy+dd, 'r')
plot(xx, yy_fit, 'b')
legend({'original data', 'fitting curve'})
```

Example07 sparse matrix vs full matrix

```
%% generate matrix
n = 2000;                                    %matrix dimension
x_full = diag(ones(1, n-1), -1) + ...
         diag(ones(1, n-1), +1);             %full tri-diagonal matrix
x_sparse = sparse(x_full);                   %sparse tri-diagonal matrix

%% test memory usage
v1 = whos('x_full');                         %memory usage of full matrix
fprintf('size of x(full) = %5.2f MB \n', v1.bytes/2^20)
v2 = whos('x_sparse');                       %memory usage of sparse matrix
fprintf('size of x(sparse) = %5.2f MB \n', v2.bytes/2^20)

%% test operation time
tic                                          %time of full matrix operation
x_full * x_full;
t1 = toc;
fprintf('time of x(full) = %7.4f s \n', t1)
tic                                          %time of sparse matrix operation
x_sparse * x_sparse;
t2 = toc;
fprintf('time of x(sparse) = %7.4f s \n', t2)
```

Example08 fast Fourier transform

```
n = 200;                                     %time sampling number
ratio = 2.0;                                 %signal noise ratio
frequency = 50;                              %frequency
signal = ratio * sin(frequency * 2*pi/n * (0:n-1));
                                             %sinusoidal signal
noise = randn(1, n);                         %random noise
zz = signal + noise;                         %signal plus noise
yy = fft(zz);                                %analyze frequency spectrum
m = round(n/2) + 1;
spectrum = 2 * abs(yy(1:m));
figure                                       %plot frequency spectrum
plot(0:m-1, spectrum)
xlabel('frequency')
```

Example09 solve ordinary difference equation

```
m = 1;                                          %mass
k = 1;                                          %stiffness
x0 = 1;                                         %initial position
v0 = 0;                                         %initial velocity
T = 4*pi;                                       %time duration
F = @(x) -k * x;                                %force
NewtonEquation = @(t, y) [y(2); 1/m*F(y(1))];
                                                %Newton's equation
y0 = [x0, v0];                                  %initial value
[tt, yy] = ode45(NewtonEquation, [0, T], y0);
                                                %solve ODE
xx = yy(:, 1); vv = yy(:, 2);
figure                                          %plot solution
plot(tt, xx)
xlabel('t')
ylabel('x')
```

Example10 surface plot, color plot, and contour plot

```
%% prepare data
x = linspace(-pi, pi, 101);
y = linspace(-pi, pi, 101);
[xx, yy] = ndgrid(x, y);                        %x-mesh and y-mesh
zz = cos(xx) + cos(yy);                         %z = cos(x) + cos(y)

%% surface plot
figure
surf(xx, yy, zz)
shading interp
xlim([-pi, pi])
ylim([-pi, pi])

%% color plot
figure
surface(xx, yy, zz)
shading interp
axis square
```

www.ebook3000.com

```
xlim([-pi, pi])
ylim([-pi, pi])
colorbar

%% contour plot
figure
contour(xx, yy, zz, 'ShowText','on')
axis square
```

These examples are the "phrases" and "idioms" provided by MATLAB. Similar examples also appear in a standard numerical computation book, e.g., Ref. [1]. With MATLAB, you can throw away the 800-page book, and focus on your own research. More examples and explanations can be found in MATLAB user's guide. Don't forget that the NanoDsim package itself is also an example of scientific computing with MATLAB.

4.2 MATLAB: vectorization technique

Since MATLAB is highly optimized for operations on vectors and matrices, vectorized MATLAB code may have higher efficiency and shorter length. The idea of vectorization is to operate on the whole vector (matrix) in one go to avoid the loop over individual vector (matrix) elements.

For example, to evaluate $\sin x$ in $[0, \pi]$, one can do it with a for-loop in a similar way as in C and Fortran

```
%% for-loop method
x = linspace(0, pi, 100);  %x array from 0 to pi with 100 points
y1 = zeros(size(x));       %initialize y array
for ii = 1:length(x)       %for-loop over x elements
    y1(ii) = sin(x(ii));
end %ii
```

Alternatively one can do the job with MATLAB vectorization

```
%% vectorization method
x = linspace(0, pi, 100);  %x array from 0 to pi with 100 points
y2 = sin(x);               %calculate y array with vectorization
```

Notice that the operation $sin(\cdots)$ is applied to each element of the vector x. Obviously the vectorization method is much simpler than the for-loop method.

This simple example also illustrates another MATLAB programming trick, pre-allocation. In the for-loop method, the statement *y1 = zeros(size(x))* is to pre-allocate a block of memory for the variable *y1*. Although MATLAB does not require variable declaration, it is a good practice to pre-allocate large arrays by using *zeros*. The reason is that MATLAB needs contiguous blocks of memory to store arrays. Pre-allocation can avoid repeatedly resizing array and moving data.

MATLAB implements plenty of functions to support vectorization: The operators and functions *.**, *./*, *.^*, *sin*, *log*, *exp*, *sqrt* are useful to construct vectorized expressions; The functions *sum*, *prod*, *diff*, *cumsum* are useful to carry out vectorized calculations; The functions *find*, *any*, *all* are useful to make logic judgement; The functions *ones*, *zeros*, *rand*, *diag* are useful to construct matrices; The functions *repmat*, *reshape*, *ndgrid*, *squeeze*, *kron* are useful to manipulate matrices. With the aid of these functions, one can write MATLAB code in a vectorized style.

Let's take a look at a more realistic example. In the local density approximation, the correlation energy density can be parameterized by [2]

$$\varepsilon_C(\rho) = \begin{cases} \frac{\alpha_1 t}{t+\alpha_2\sqrt{t}+\alpha_3} & t < 1; \\ -\beta_1 \ln t + \beta_2 - \beta_3 \frac{\ln t}{t} + \beta_4 \frac{1}{t} & t > 1. \end{cases} \tag{4.1}$$

where t is defined by $t = \left(\frac{4\pi}{3}\rho\right)^{1/3}$, and the constants are $\alpha_1 = -0.1423$, $\alpha_2 = 1.0529$, $\alpha_3 = 0.3334$, $\beta_1 = 0.0311$, $\beta_2 = -0.0480$, $\beta_3 = 0.0020$, $\beta_4 = -0.0116$.

As a comparison, we shall implement this formula with both for-loop and vectorization method. The for-loop code is as follows:

```
function yy = epsC_ForLoop(nn)
a1 = -0.1423; a2 =  1.0529; a3 =  0.3334;
b1 =  0.0311; b2 = -0.0480; b3 =  0.0020; b4 = -0.0116;
yy = zeros(size(nn));
for ii = 1:numel(nn)
    ni = nn(ii);
    t = (4*pi/3 * ni) ^(1/3);
    if t < 1
        yi = a1*t / (t + a2*sqrt(t) + a3);
    else
        yi = -b1*log(t) + b2 - b3*log(t)/t + b4/t;
    end
    yy(ii) = yi;
```

www.ebook3000.com

```
end %ii
end %epsC_ForLoop
```

The vectorization code is as follows

```
function yy = epsC_Vectorized(nn)
a1 = -0.1423; a2 =  1.0529; a3 =  0.3334;
b1 =  0.0311; b2 = -0.0480; b3 =  0.0020; b4 = -0.0116;
yy = zeros(size(nn));
tt = (4*pi/3 * nn).^(1/3);
index1 = find(tt <= 1);                                  %conditional indexing
index2 = find(tt > 1);                                   %conditional indexing
tt1 = tt(index1);
tt2 = tt(index2);
yy1 = a1*tt1 ./ (tt1 + a2*sqrt(tt1) + a3);               %vectorized expression
yy2 = -b1*log(tt2) + b2 - b3*log(tt2)./tt2 + b4./tt2;    %vectorized expression
yy(index1) = yy1;
yy(index2) = yy2;
end %epsC_Vectorized
```

Notice that the conditional judgements $t < 1$ and $t > 1$ are achieved by the function *find* which picks up the indexes satisfying the logical expression. Also note that the operator $*$ and / are replaced by $.*$ and ./ in the vectorized expressions.

Finally we would like to mention that it is not always necessary to vectorize a MATLAB for-loop. Due to MATLAB's JIT-accelerator after version 6.5, simple for-loop with scalar variable operations and built-in functions are as efficient as vectorized code.

4.3 MATLAB: hybrid programming

MATLAB is a high-level language in the sense that one line of MATLAB code amounts to several lines of C or Fortran code. Although C and Fortran were also referred to as the high-level languages, they are actually low-level languages compared to MATLAB. A hybrid programming of high-level and low-level languages provides a good compromise between readability and efficiency: One can write the majority of the code with the high-level language which is simpler and shorter, and optimize the bottlenecks with the low-level language which is more flexible but less maintainable.

Consider the following situations: (1) One already has a complicated C or Fortran code and it is impractical to rewrite it completely in MATLAB; (2) A few lines of MATLAB code are identified as the major bottleneck and it is too complicated or even impossible to vectorize that part; (3) A few

lines of MATLAB code consumes a huge amount of memory and needs to be optimized by using flexible C pointers. In these situations, MATLAB provides an interface to integrate the C or Fortran code to the MATLAB code.

As a demonstration, let us rewrite *epsC_ForLoop.m* in Section 4.2 into a Fortran-code and a C-code. The C-code (Fortran-code) consists of two parts: the gateway routine and the computational routine. The computational routine is the code doing the real calculations and is quite similar to the corresponding MATLAB code. The gateway routine is an interface between the low-level language (C or Fortran) and the high-level language (MATLAB). In the gateway routine, one needs to create new MATLAB variables to host the output arguments, get the pointers to both the input and output arguments, and call the computational routine to operate on the input variables to obtain the output variables.

For example, the MATLAB code *epsC_ForLoop.m* can be rewritten as the C-code *epsC_Ccode.c*

```
/* usage::   yy = epsC_Ccode(nn) */
/* compile:: mex epsC_Ccode.c    */

#include "mex.h"
#include "math.h"

/************************** gateway routine **************************/
void mexFunction(int OutputNumber, mxArray *OutputPointer[],
                 int InputNumber, const mxArray *InputPointer[])
{
/* declaration */
double *nn;                     /* pointer to the nn matrix data       */
double *yy;                     /* pointer to the yy matrix data       */
int dim1, dim2, dim;            /* internal variables                  */

/* create a new matlab variable yy */
dim1 = mxGetM(InputPointer[0]); /* get dim1 of the nn matrix data      */
dim2 = mxGetN(InputPointer[0]); /* get dim2 of the nn matrix data      */
dim = dim1 * dim2;
OutputPointer[0] = mxCreateDoubleMatrix(dim1, dim2, mxREAL);
                                /*create a new matlab variable yy      */

/* call computational routine */
nn = mxGetPr(InputPointer[0]);  /* get pointer to the nn matrix data   */
yy = mxGetPr(OutputPointer[0]); /* get pointer to the yy matrix data   */
calculate(yy, nn, dim);         /* call computational routine          */
}

/************************ computational routine ************************/
```

www.ebook3000.com

```
calculate(double *yy, double *nn, int dim)
{
int ii;
double ni, yi, t;
const double pi = 3.14159265358979323846264433;
const double a1 = -0.1423, a2 =  1.0529, a3 =  0.3334;
const double b1 =  0.0311, b2 = -0.0480, b3 =  0.0020, b4 = -0.0116;
for (ii=0; ii<dim; ii++) {
    ni = nn[ii];
    t = pow(4*pi/3 * ni, 1.0/3.0);
    if (t < 1)
        yi = a1*t / (t + a2*sqrt(t) + a3);
    else
        yi = -b1*log(t) + b2 - b3*log(t)/t + b4/t;
    yy[ii] = yi;
}
}
```

The MATLAB code *epsC_ForLoop.m* can also be rewritten as the Fortran-code *epsC_Fcode.f90*

```
!usage  :: yy = epsC_Fcode(nn)
!compile:: mex epsC_Fcode.f90

!************************** gateway routine **************************!
subroutine mexFunction(OutputNumber, OutputPointer, &
                       InputNumber, InputPointer)
!declaration
implicit none
integer:: OutputPointer(*), InputPointer(*) !pointer to output and input arguments
integer:: OutputNumber, InputNumber         !number of output and input arguments
integer:: mxGetM, mxGetN, mxGetPr           !pointer to mx-functions
integer:: mxCreateDoubleMatrix              !pointer to mx-functions
integer:: nn_pointer                        !pointer to the nn matrix data
integer:: yy_pointer                        !pointer to the yy matrix data
integer:: dim1, dim2, dim                   !interal variables

!create a new matlab variable yy
dim1 = mxGetM(InputPointer(1))              !get dim1 of the nn matrix data
dim2 = mxGetN(InputPointer(1))              !get dim2 of the nn matrix data
dim = dim1 * dim2
OutputPointer(1) = mxCreateDoubleMatrix(dim1, dim2, 0)
                                            !create a new matlab variable yy
                                            !0 means mxREAL

!call computational routine
nn_pointer = mxGetPr(InputPointer(1))       !get pointer to the nn matrix data
yy_pointer = mxGetPr(OutputPointer(1))      !get pointer to the yy matrix data
call calculate(%val(yy_pointer), %val(nn_pointer), dim)
                                            !call computation routine
end subroutine mexFunction

!************************ computational routine ************************!
subroutine calculate(yy, nn, dim)
integer:: dim
real*8 :: nn(dim)
```

```
real*8 :: yy(dim)
real*8 :: ni, yi, t
integer:: ii
real*8, parameter:: pi = 3.14159265358979323846264433d0
real*8, parameter:: a1 = -0.1423d0, a2 =  1.0529d0, a3 =  0.3334d0
real*8, parameter:: b1 =  0.0311d0, b2 = -0.0480d0, b3 =  0.0020d0, b4 = -0.0116d0
do ii = 1, dim
   ni = nn(ii)
   t = (4*pi/3 * ni) ** (1d0/3d0)
   if (t < 1) then
      yi = a1*t / (t + a2*sqrt(t) + a3);
   else
      yi = -b1*log(t) + b2 - b3*log(t)/t + b4/t;
   end if
   yy(ii) = yi
enddo
end subroutine calculate
```

A few comments are as follows. (1) Except for some minor differences, the C-code *epsC_Ccode.c* and the Fortran-code *epsC_Fcode.f90* look quite similar. Note that the index of C-array starts from 0 while the index of Fortran-array starts from 1. (2) MATLAB matrix elements can be referred to by 1d indices. For example, M is a 2×3 matrix

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \end{pmatrix},$$

whose elements can be referred to as $M(1) = M_{11}$, $M(2) = M_{21}$, $M(3) = M_{12}$, $M(4) = M_{22}$, $M(5) = M_{13}$, $M(6) = M_{23}$. (3) The C-code and the Fortran-code can be compiled by using the command *mex* [3]. The compiled functions *epsC_Ccode* [4] and *epsC_Fcode* [5] can be called within MATLAB as if they were MATLAB functions. (4) It is a good practice to write an m-code first and then convert it into a C-code or Fortran-code like what we did for *epsC_Forloop.m*. Although you waste some time writing the code twice, you will save a lot in the debugging stage where you can simply check the C-code or Fortran-code against the m-code.

Now let us turn to a real example encountered in software development. We have a large 3d grid, e.g., $256 \times 256 \times 256$, and a small 3d grid, e.g., $64 \times 64 \times 64$. We wish to add the data of the small grid to the large grid. The operation can be easily implemented by a vectorized MATLAB function *IndexPlusEqual_Mcode.m* ("plus equal" refers to the "+=" operation)

```
function GridData = IndexPlusEqual_Mcode(GridData,indexX,indexY,indexZ,SubGridData)
GridData(indexX,indexY,indexZ) = GridData(indexX,indexY,indexZ) + SubGridData;
end %IndexPlusEqual_Mcode
```

www.ebook3000.com

Since the operation will be carried out many times, this simple operation was found to be a serious bottleneck. Further study reveals that MATLAB makes a temporary copy of the large grid data and copies it back after adding the small grid data. This is obviously not efficient. Can we operate directly on the large grid data to avoid the redundant copy? The pointer concept in C makes it possible, and the m-code can be rewritten into the C-code *IndexPlusEqual_Ccode.c*

```
/* usage::   IndexPlusEqual(GridData,indexX,indexY,indexZ,SubGridData) */
/* compile:: mex IndexPlusEqual.c                                     */

#include "mex.h"

/************************** gateway routine **************************/
void mexFunction(int OutputNumber, mxArray *OutputPointer[],
                 int InputNumber, const mxArray *InputPointer[])
{
/* declaration */
double *GridData, *SubGridData;                  /* pointers to input arguments */
double *indexX, *indexY, *indexZ;                /* pointers to input arguments */
const int *Dimensions, *SubDimensions;           /* grid data dimensions        */

/* call computational routine */
GridData    = mxGetPr(InputPointer[0]);          /* get pointer to GridData     */
SubGridData = mxGetPr(InputPointer[4]);          /* get pointer to SubGridData  */
indexX = mxGetPr(InputPointer[1]);               /* get pointer to indexX       */
indexY = mxGetPr(InputPointer[2]);               /* get pointer to indexY       */
indexZ = mxGetPr(InputPointer[3]);               /* get pointer to indexZ       */
Dimensions = mxGetDimensions(InputPointer[0]);   /* get GridData dimensions     */
SubDimensions = mxGetDimensions(InputPointer[4]); /* get SubGridData dimensions  */
OutputPointer[0] = InputPointer[0];              /* set output pointer          */
calculate(GridData, indexX, indexY, indexZ, SubGridData,
          Dimensions, SubDimensions);            /* call computational routine  */
}

/********************** computational routine **********************/
calculate(double *GridData, double *indexX, double *indexY, double *indexZ,
          double *SubGridData, int *Dimensions, int *SubDimensions)
{
/* In MATLAB, (i1,i2,i3) is mapped to I = 1 + (i1-1) + n1*(i2-1) + n1*n2*(i3-1)  */
/* In C, (i1,i2,i3) is mapped to I = i1 + n1*i2 + n1*n2*i3                       */
int i1, i2, i3;
int a1, a2, a3;
int b1, b2, b3;
const int dim1 = Dimensions[0];
const int dim2 = Dimensions[1];
const int dim3 = Dimensions[2];
const int sdim1 = SubDimensions[0];
const int sdim2 = SubDimensions[1];
const int sdim3 = SubDimensions[2];
const int g1 = 1, g2 = dim1,  g3 = dim1*dim2;
const int h1 = 1, h2 = sdim1, h3 = sdim1*sdim2;
for (i3=0; i3<sdim3; i3++) {
    a3 = g3*(indexZ[i3]-1);
    b3 = h3*i3;
    for (i2=0; i2<sdim2; i2++) {
        a2 = g2*(indexY[i2]-1);
        b2 = h2*i2;
        for (i1=0; i1<sdim1; i1++) {
```

```
            a1 = indexX[i1]-1;
            b1 = i1;
            GridData[a1+a2+a3] += SubGridData[b1+b2+b3];
        }
    }
}
}
```

A few comments are as follows. (1) The C-code works on the input argument *GridData* directly and makes no extra copy, and hence the memory cost is reduced by half. (2) In MATLAB the elements of a 3d array can be referred to by 1d indices. The 3d index (i_1, i_2, i_3) is mapped to the 1d index $I = 1 + (i_1 - 1) + n_1 (i_2 - 1) + n_1 n_2 (i_3 - 1)$ where $1 \leqslant i_1 \leqslant n_1$, $1 \leqslant i_2 \leqslant n_2$, $1 \leqslant i_3 \leqslant n_3$. In C the array index starts from 0, and the 3d index $(\tilde{i}_1, \tilde{i}_2, \tilde{i}_3)$ is mapped to the 1d index $\tilde{I} = i_1 + n_1 i_2 + n_1 n_2 i_3$ where $0 \leqslant \tilde{i}_1 \leqslant n_1 - 1$, $0 \leqslant \tilde{i}_2 \leqslant n_2 - 1$, $0 \leqslant \tilde{i}_3 \leqslant n_3 - 1$. (3) A test run shows that the C-code is faster than the m-code by one or two orders of magnitude. The speed up is overwhelming and this part is no longer a bottleneck.

We have seen that the hybrid programming of high-level and low-level languages makes it possible to write high quality code with good readability and excellent performance. Another possibility of using hybrid programming is to call an existing C or Fortran code from MATLAB. An example is provided in Section 7.5 where an interface is created to call MPI commands from MATLAB code.

4.4 MATLAB: object oriented programming

The nature of a computer program is a sequence of commands. For the sake of human programmers, the commands are re-grouped into some meaningful units such as functions and objects. In scientific computing, most simple tasks are procedure-like: one follows step-1, step-2, ..., to accomplish a calculation. Designing code in terms of procedures are called procedure-oriented programming (POP). On the other hand, complicated tasks may involve individual pieces such as systems, atoms, and orbitals encountered in the NECPA-LMTO problem. These pieces are called objects which have certain properties and operations. Designing code in terms of objects is called object-oriented programming (OOP). In this section, we shall illustrate some essential features of OOP in MATLAB with two simple examples.

The three pillars of OOP are **encapsulation**, **inheritance**, and **polymorphism** [**6**]. Let us first take a look at the first pillar, encapsulation.

www.ebook3000.com

Essentially encapsulation means to pack up a data structure and the corresponding operations together. In the first example, we wish to create a new data type to enable arithmetic operations of rational numbers. In OOP this can be achieved by creating a class *@rational* which defines a new type of objects. A rational number is composed of a numerator and a denominator which are the data structures (*properties*) of *@rational*. The arithmetic operations are addition, subtraction, multiplication, and division which are the operations (*methods*) of *@rational*. Encapsulation enforces that one can only operate on an object's data structures through its methods. In this sense, the concept of encapsulation is analogous to a user-defined data type.

To implement the class in MATLAB, we first create a new folder called *@rational*. In the folder *@rational*, we create an m-file named *rational.m* to define the class

```
classdef rational
    properties
        numerator
        denominator
    end %properties

    methods
        function object = rational(string)
            [numerator, denominator] = parse(string);
            [numerator, denominator] = reduce(numerator, denominator);
            object.numerator = numerator;
            object.denominator = denominator;
        end %rational
    end %methods
end %classdef
```

The definition of the class *@rational* has two blocks, *properties* and *methods*. The block *properties* contains the data structures of the class. The block *methods* contains the operations of the class. Besides the arithmetic operations, there is a special method called class constructor. The mission of the class constructor is to create one object of this class. In this example, the class constructor is the function *rational*, which analyzes the input string and obtain the numerator and denominator. For example, in the execution of *rational('4/6')*, the string '4/6' is parsed as 4 and 6, and reduced to 2 and 3 as the numerator and denominator respectively. Notice that the constructor calls two internal functions *parse* and *reduce* which are located in the sub-folder *@rational/private*. The functions are called

private functions and can only be used within the class.

Now we are ready to implement the arithmetic operations. In MATLAB, the methods (except for the constructor) can be included either in the *method* block of the class definition or as independent m-files in the class folder. For example, addition is implemented in the m-file *@rational/plus.m*

```
function obj3 = plus(obj1, obj2)
n1 = obj1.numerator;
m1 = obj1.denominator;
n2 = obj2.numerator;
m2 = obj2.denominator;
n3 = n1 * m2 + n2 * m1;
m3 = m1 * m2;
[n3, m3] = reduce(n3, m3);
obj3 = rational(sprintf('%d / %d', n3, m3));
end %plus
```

Similarly substraction, multiplication, division, and the negative sign are implemented in *@rational/minus.m*, *@rational/mtimes.m*, *@rational/mrdivide.m*, and *@rational/uminus.m*, respectively. In addition, a user-defined display method for a rational number is implemented in *@rational/display.m*. To test the new class, we run the script *test_rational.m* and get the following results:

```
x1 = rational('4 / 3');
x2 = rational('-1 / 2 ');
x3 = rational('4');
y1 = x1 + x2
y2 = x1 - x2
y3 = x1 * x2
y4 = - x1 / x2
y5 = - x3
y6 = -x1 * x2 - x2 / x1 + x3
>>
5 / 6
11 / 6
-2 / 3
8 / 3
-4
121 / 24
```

www.ebook3000.com

We can see that the MATLAB operators +, −, ∗, / and the display method have been overloaded successfully by these new methods designed for the rational numbers.

Now let us turn to inheritance, the second pillar of OOP. Like biological inheritance, OOP inheritance has two aspects: heredity and mutation. On the one hand, the child class may possess all the properties and methods of the parent class; On the other hand, the child class may have new properties and methods of its own. In the second example, we wish to plot various shapes with different colors. We start from a very simple class *@shape_circle* which can plot a unit circle, change the circle size, and move the circle center. The class is implemented as follows:

```
classdef shape_circle
    properties
        xy
    end %properties

    methods
        function object = shape_circle
            tt = linspace(0, 2*pi, 101);
            object.xy = [cos(tt); sin(tt)];
        end %shape_circle

        function object = rescale(object, factor)
            xy = object.xy;
            xy(1, :) = xy(1, :) * factor(1);
            xy(2, :) = xy(2, :) * factor(2);
            object.xy = xy;
        end %rescale

        function object = translate(object, delta)
            xy = object.xy;
            xy(1, :) = xy(1, :) + delta(1);
            xy(2, :) = xy(2, :) + delta(2);
            object.xy = xy;
        end %translate

        function plot(object)
            xy = object.xy;
```

```
            plot(xy(1,:), xy(2,:))
        end %plot
    end %methods
end %classdef
```

The class *@shape_circle* has one property *xy* to store the coordinates, and three methods *plot*, *rescale*, *translate* to achieve the designed functions. Suppose the class has been well tested, highly optimized, and used extensively in your art design package. You are very satisfied with it until one day you wish to fill the circle with color. Now you have three choices: (1) rewrite the *plot* method in the old class *@shape_circle*; (2) create an independent new class *@shape_circle_color*; (3) create a new class *@shape_circle_color* by the inheritance of the old class *@shape_circle*. Choice (1) is a bad idea because whenever you do something you always have chances to make mistakes. The revised class may induce new bugs into the existing package. Choice (2) is better than choice (1) because all the existing functions won't be affected. Even if you make a mistake in *@class_circle_color*, the error will be localized to those new "colorful" functions. The problem with choice (2) is that some codes have to be duplicated which is inconvenient for the future maintenance. Let's say a new version of MATLAB renames the function *sin* to *sine*. You must update the function name in both *@shape_circle* and *@shape_circle_color*. In other words, you have to maintain two classes since they are independent. In contrast, by using the class inheritance, choice (3) makes it possible to build up a new child class without touching the existing parent class. The child class inherits all the methods and properties of parent class and modifies some parent methods if necessary.

In MATLAB, the class inheritance is implemented by the syntax *classdef ChildClassName < ParentClassName*. It is also required to call the parent class constructor in the child class constructor by the syntax *object = object@ParentClassName(arguments)*. For example, *@shape_circle_color* inheriting *@shape_circle* is implemented as

```
classdef shape_circle_color < shape_circle
    properties
        color
    end %properties

    methods
        function object = shape_circle_color(color)
            if ~isstr(color) & ~isvector(color)
```

www.ebook3000.com

```
                error('unknown color format')
            end
            object.color = color;
        end %shape_circle_color

        function plot(object)
            color = object.color;
            xy = object.xy;
            fill(xy(1,:), xy(2,:), color)
        end %plot
    end %methods
end %classdef
```

The child class *@shape_circle_color* inherits the property *xy* and three methods *plot*, *rescale*, *translate* from its parent class *@shape_circle*. In addition, the child class has a new field *color* and its own method *plot*. Notice that the child method *plot* will overload the parent method *plot*. Analogously we can create the class *@shape_square* and *@shape_triangle* for squares and triangles, and extend them to *@shape_square_color* and *@shape_triangle_color* by using the inheritance.

So far we have had a taste of OOP. Now let us apply the programming skill to do some art work. The task is to plot a Christmas tree. We don't have to start from zero since we already have some objects in the art design package. We shall create a new class *@shape_XmasTree* by including the existing colored shapes as pieces. One class containing other classes as its parts is called *aggregation* in OOP. The class *@shape_XmasTree* is implemented as follows (some lines are omitted to save space, see the research code for a full implementation)

```
classdef shape_XmasTree
    properties
        pieces
    end %properties

    methods
        function object = shape_XmasTree
            %brown trunk
            pieces{1} = shape_square_color([0.7, 0.3, 0.0]);
            pieces{1} = rescale(pieces{1}, 1/sqrt(2)*[1, 1.5]);
            pieces{1} = translate(pieces{1}, [0, 0.75]);
```

```
            ......
            %decoration-6
            pieces{10} = shape_circle_color('w');
            pieces{10} = rescale(pieces{10}, [0.1, 0.1]);
            pieces{10} = translate(pieces{10}, [-0.4, 4.5]);
            object.pieces = pieces;
        end %shape_XmasTree

        function object = rescale(object, factor)
            pieces = object.pieces;
            for ii = 1:length(pieces)
                pieces{ii} = rescale(pieces{ii}, factor);
            end %ii
            object.pieces = pieces;
        end %rescale

        function object = translate(object, delta)
            pieces = object.pieces;
            for ii = 1:length(pieces)
                pieces{ii} = translate(pieces{ii}, delta);
            end %ii
            object.pieces = pieces;
        end %translate

        function plot(object)
            pieces = object.pieces;
            for ii = 1:length(pieces)
                plot(pieces{ii});
            end %ii
        end %plot
    end %methods
end %classdef
```

The class *@shape_XmasTree* also supports three methods *plot*, *rescale*, *translate*. Although the contents of the methods are very different from those implemented in the previous shape classes, the methods share the same name. This is called polymorphism which is the last pillar of OOP. To be brief, polymorphism means the same verb may have different meanings when applied to different nouns. You can open a door, open a file, open

www.ebook3000.com

a bottle, open an account, open your mouth, open your mind, etc. We have seen that methods are encapsulated in its class, and hence identical method names won't cause any ambiguity. On the contrary, we can take the advantage of identical method names: Whenever we see a shape class, we know it can be plotted; Whenever we see a numerical data type, we know it can be operated on with $+$, $-$, $*$, $/$.

The above is a brief introduction to the OOP in MATLAB. We have demonstrated three key components of OOP with two simple examples. There is yet another important ingredient: your own creativity. To conclude the section, we complete the art work with the following script

```
MyTree1 = shape_XmasTree;
MyTree2 = shape_XmasTree;
MyTree1 = rescale(MyTree1, [0.8,1]);
MyTree2 = rescale(MyTree2, [0.8,1]);
MyTree1 = translate(MyTree1, [-1.5, 0]);
MyTree2 = translate(MyTree2, [+1.5, 0]);

figure
hold on
plot(MyTree1)
plot(MyTree2)
axis image
axis off
```

4.5 NanoDsim: overall design

In this section, we shall present the overall design of the NanoDsim package which implements the NECPA-LMTO formalism of Section 3.11 and 3.12. We shall adopt MATLAB as our working language which has been reviewed in previous sections.

Since it is a scientific computing package, the emphasis is a little bit different from general programming. We put high premium on the performance and efficiency, and push the hardware to its limits to gain simulation capability. Moreover the most challenging part of scientific computing is sophisticated algorithms, which are more like procedures than objects. Therefore we don't constrain ourselves to think only in terms of objects. The package will be a hybrid of OOP and POP whichever is more natural and efficient.

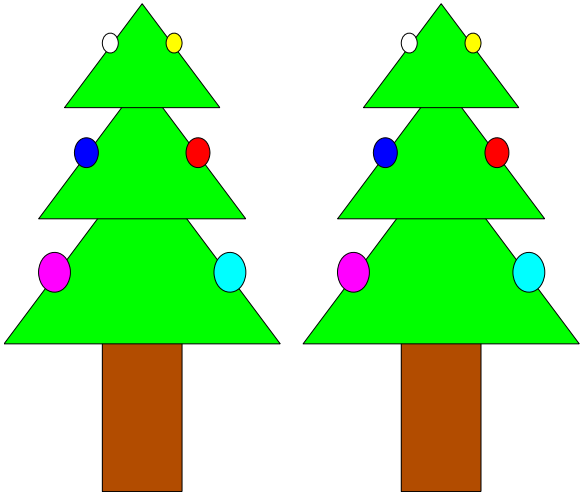


Fig. 4.1 Christmas trees plotted with shape objects which are designed by using MATLAB OOP.

In NanoDsim, we have two distinct types of systems: bulk systems and two-probe systems. Consequently we have two dsim-classes *@class_cpaBulk* and *@class_necpaTwoProbe*. A bulk system or a two-probe system is composed of many atomic sites, which are modeled by the dsim-class *@class_cpaAtom* or *@class_necpaAtom*. Each atomic site is composed of single or multiple species of atoms, which are modeled by the dsim-class *@class_lmtoAtom*. Each atom is composed of many linear muffin-tin orbitals, which are modeled by the dsim-class *@class_lmtoOrbital*. In addition, we also need the solution of isolated atoms to provide an initial guess for bulk and two-probe systems. An isolated atom is modeled by the dsim-class *@class_lmtoElement*.

In NanoDsim, we have two distinct types of calculations: self-consistent calculations and post-analysis calculations. The self-consistent calculations are organized by the dsim-solvers *@SCFsolver_Atom*, *@SCFsolver_Bulk* and *@SCFsolver_TwoProbe* for isolated atoms, bulk systems, and two-probe systems, respectively. The dsim-solvers invoke the methods of dsim-classes to accomplish the self-consistent calculation in the flowchart of Fig. 3.5. The outcome of a dsim-solver is an LMTO Hamiltonian. The post-analysis calculations are organized by the dsim-calculators *@calculator_** where * refers to a calculation type (e.g, band_EIG, dos_CPA, trans_CPA). All the dsim-

www.ebook3000.com

calculators support a method *calculate* which can operate on the LMTO Hamiltonian generated by dsim-solvers. The outcome of a dsim-calculator is the relevant physical quantities.

Some functions are shared by many different classes, and some algorithms are too complicated to implement as class methods. These functions and algorithms are separated as supporting libraries and made public to all the dsim-classes and dsim-solvers. In addition to the supporting libraries, there are two auxiliary folders: One is *Accessory* in which small independent tools are placed; The other is *Environment* in which NanoDsim environment variables are placed.

Last but not least, we have the main function *nanodsim* which serves as an entrance to the package. The mission of *nanodsim* is to initialize classes, solvers, calculators and supporting libraries, and call the *solve* and *calculate* method of the dsim-solver and dsim-calculator to carry out the simulation. The user is supposed to access all the functionalities of NanoDsim through the key command *nanodsim*.

At the end of this section, the design of NanoDsim is summarized by Fig. 4.2.

4.6 NanoDsim: dsim-solvers

In this section, we shall discuss dsim-solvers which carry out self-consistent calculations. The goal of the self-consistent calculation is to solve a nonlinear equation array

$$X = F[X], \tag{4.2}$$

where X is a vector and F is a nonlinear function. To solve the equation array, one first makes an initial guess $X = X_0$ and then iterates $F[X]$ to obtain a series of X_n. The iteration continues until $|X_{n+1} - X_n| < \varepsilon$ where ε is a given tolerance.

In the NECPA-LMTO formalism, X is the assemble of atomic potentials $\{V_{iq}(r)\}$ and linearization centers $\left\{E^0_{ilq}\right\}$. F is implicitly defined by the procedures from step-4 to step-12 in the flowchart of Fig. 3.5. Notice that step-6 is distinct from other steps because it needs an additional inner loop to solve the NECPA equations. In principle the self-consistent calculation should be a double loop: The outer loop is for the F iteration and the inner loop is for the NECPA iteration. However, the NECPA iteration turns out to be the bottleneck of the whole self-consistent calculation. Even a single NECPA iteration is much more difficult than all other steps. To reduce

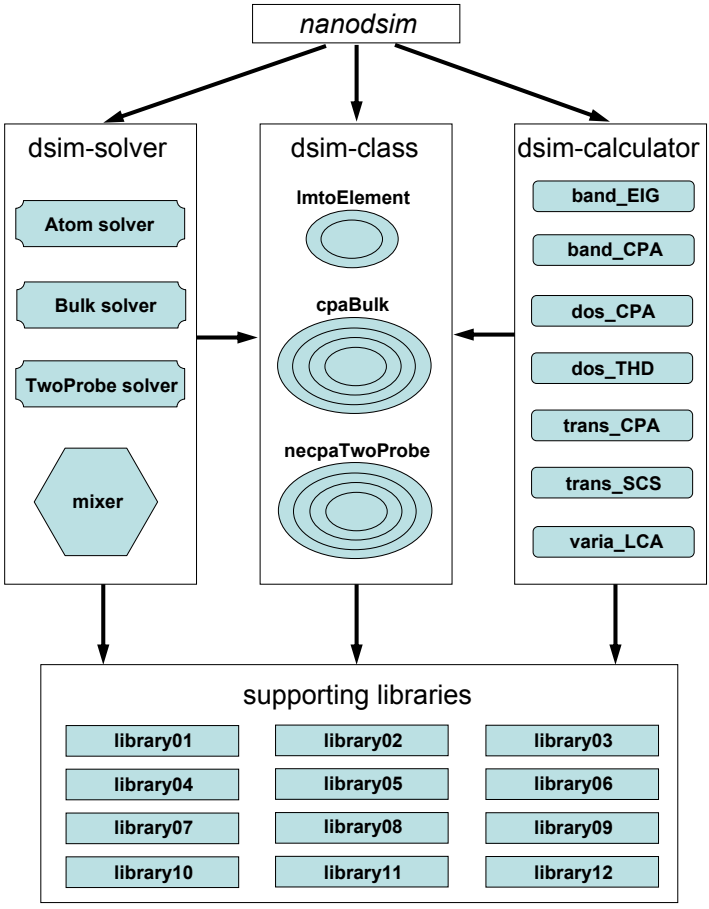


Fig. 4.2 The overall design of NanoDsim package. The key command *nanodsim* controls the interactions of dsim-classes, dsim-solvers, and dsim-calculators. Dsim-classes have their inner structures represented by the nested circles. Dsim-solvers and dsim-calculators call the methods of dsim-classes to carry out self-consistent and post-analysis calculations. Dsim-classes, dsim-solvers, and dsim-calculators can access the 12 supporting libraries which implement the managing and computational algorithms.

the computational cost, the double loop is converted to a big single loop where the NECPA inner loop is merged into the F outer loop. The idea is based on the following observation: At the beginning of the F iteration, the atomic potentials $\{V_{iq}(r)\}$ are far from the final solution. It does not make too much sense to solve the NECPA equations accurately at this stage.

www.ebook3000.com

Instead one can iterate the NECPA equations only once to obtain a rough estimate of the coherent potentials. With more steps of the F iteration, the atomic potential $\{V_{iq}(r)\}$ gets more and more accurate, and the solution of the NECPA equations also gets converged. For the same reason, the inner loop for solving the Fermi level μ in bulk systems is also merged into the outer F iteration. The nonlinear function F used in different types of systems is summarized as follows:

solver type	**nonlinear function**
Atom	$V_{iq}(r) = F[V_{iq}(r)]$
Bulk	$\left[\{V_{iq}(r)\}, \left\{E^0_{ilq}\right\}, \{\Omega^r_i\}, \mu\right] = F\left[\{V_{iq}(r)\}, \left\{E^0_{ilq}\right\}, \{\Omega^r_i\}, \mu\right]$
TwoProbe	$\left[\{V_{iq}(r)\}, \left\{E^0_{ilq}\right\}, \{\Omega^r_i, \Omega^<_i\}\right] = F\left[\{V_{iq}(r)\}, \left\{E^0_{ilq}\right\}, \{\Omega^r_i, \Omega^<_i\}\right]$

(4.3)

To start an iteration, we need to construct a proper initial guess X_0. The general idea of constructing an initial guess is to expand the "territory" little by little from the known solutions. To solve an isolated atom, the initial guess of $V_{iq}(r)$ is constructed by using an empirical charge density $\rho_{iq}(r)$

$$\rho_{iq}(r) = \rho_0 e^{-ar}, \tag{4.4}$$

where $a = \frac{1}{2}Z_{iq}^{1/4} + 1$ and ρ_0 is chosen such that $N_s \int_0^{R_{iq}} 4\pi r^2 \rho_{iq}(r)\, dr = Z_{iq}$ to satisfy the charge neutrality ($N_s = 2$ is the spin degeneracy). To solve a clean bulk system, the initial guess is constructed by using the solutions of single atoms. To solve a disordered bulk system, the initial guess is constructed by using the solution of the clean bulk system. To solve an equilibrium two-probe system, the initial guess is constructed by using the atoms in bulk systems with similar chemical environment. To solve a nonequilibrium two-probe system, the initial guess is constructed by using the equilibrium two-probe system. In addition, in disordered systems, the initial guess of coherent potentials is constructed by solving the NECPA equations for the initial guess of $\{V_{iq}(r)\}$ and $\left\{E^0_{ilq}\right\}$.

Having the nonlinear function F and the initial guess X_0, one can start the iteration, $X_1 = F[X_0]$, $X_2 = F[X_1]$, $\cdots$, $X_{n+1} = F[X_n]$. Very often, however, such simple iteration does not converge for a complicated nonlinear function F. A better strategy to make a good trial solution X is to use a mixture of the previous solutions rather than the immediate previous solution. In other words, the new trial solution X is constructed by considering a period of iteration history. The predicting algorithm is called

mixing, and the class implementing a mixing algorithm is called mixer. The simplest mixer is the linear mixer, which constructs a new trial solution X by the linear combination $\alpha X_{n-1} + (1-\alpha) X_{n-2}$ where $0 < \alpha < 1$ is the mixing rate. More advanced and delicate mixers are discussed in Appendix A.20. Since different types of variables (e..g, $\{V_{iq}(r)\}$ and $\left\{E^0_{ilq}\right\}$) may have very different convergence behavior, each of them are appointed an independent mixer.

In NanoDsim, the self-consistent calculation is organized by dsim-solvers. A dsim-solver has three basic properties: *mixer* (the mixers of variables), *tolerance* (the tolerances of variables), and *maxstep* (the maximum iteration step). A dsim-solver supports six methods: *initialize* (set up *mixer* and *tolerance*), *solve* (organize the iteration), *SCF_f0* (carry out the initial iteration step), *SCF_f* (carry out a complete iteration step), and *SCF_get* (get the variable values), *SCF_set* (set the variables values). A typical *solve* method is implemented as follows:

```
function system = solve(solver, system)
%% get
mixer = solver.mixer;
tolerance = solver.tolerance;
maxstep = solver.maxstep;

%% initial iteraiton
system = SCF_f0(solver, system);
data = SCF_get(solver, system);

%% SCF iteration
for step = 1:maxstep
    %mix
    [mixer, newdata] = mix(mixer, data);
    %iteration
    system = SCF_set(solver, system, newdata);
    system = SCF_f(solver, system);
    data = SCF_get(solver, system);
    %estimate error
    maxdiff = @(x1,x2) max(max(max(abs(x1-x2))));
    delta = maxdiff(data, newdata);
    %check convergence
    if delta < tolerance
```

www.ebook3000.com

```
        break
    end
end %step
end %solve
```

In *solve* method, *solver* and *system* refer to objects of dsim-solver and dsim-class respectively. The statement *mix(mixer, data)* is to make a new trial solution X by mixing the solution of the last iteration with the previous iteration history. The statement *SCF_f(solver, system)* is to carry out one iteration step which is the key step of the *solve* method. Since the initial iteration step can be very different from a complete iteration step, the method *SCF_f0* is called before the iteration loop. The method *SCF_get* is to get X from the dsim-class, and the method *SCF_set* is to set X to the dsim-class.

4.7 NanoDsim: dsim-calculators

In this section, we shall discuss dsim-calculators which carry out post-analysis calculations. The goal of the post-analysis calculation is to calculate physical quantities by using the self-consistent LMTO-Hamiltonian. In NanoDsim, the post-analysis calculation is organized by dsim-calculators. A dsim-calculator has no properties and supports only one method, *calculate*. The syntax of *calculate* method is *calculate(calculator, parameter)* where *calculator* and *parameter* refer to dsim-calculator and its parameters.

Although calculators may have very different algorithms for calculating various physical quantities, all the implementations share the same name *calculate*. This is another example of polymorphism as discussed in Section 4.4. The advantage is that a new developer is allowed to create his/her own dsim-calculator to extend NanoDsim functionality without touching any NanoDsim code.

NanoDsim supports seven post-analysis calculators which is summarized as follows:

calculator	**physical quantities**	**applied system**
@calculator_band_EIG	bandstructure	clean bulk system
@calculator_band_CPA	k-resolved density of states	disordered bulk system
@calculator_dos_CPA	density of states	bulk and two-probe system
@calculator_dos_THD	density of states	clean bulk system
@calculator_trans_CPA	transmission coefficient	two-probe system
@calculator_trans_SCS	transmission coefficient	clean two-probe system
@calculator_varia_LCA	transmission variation	two-probem system

The specific algorithm of each calculator will be discussed in Chapter 5 and 6.

4.8 NanoDsim: dsim-classes

In NanoDsim, there are seven dsim-classes which define objects of LMTO orbitals, LMTO atoms, bulk systems and two-probe systems. The relation between dsim-classes is summarized as follows: *@class_lmtoElement* ⊃ *@class_lmtoOrbital*, *@class_cpaBulk* ⊃ *@class_cpaAtom* ⊃ *@class_lmtoAtom* ⊃ *@class_lmtoOrbital*, *@class_necpaTwoProbe* ⊃ *@class_necpaAtom* ⊃ *@class_lmtoAtom* ⊃ *@class_lmtoOrbital*, where ⊃ refers to class aggregation. In this section, we shall analyze *@class_lmtoElement* in detail and leave other dsim-classes for Chapter 5 and 6.

The mission of *@class_lmtoElement* is to produce a physical charge density $\rho_{iq}(r)$ for the given atomic sphere radius R_{iq}, the nuclear charge Z_{iq}, and the occupation configuration. *@class_lmtoElement* has 12 properties which are summarized as follows (using the element Fe as an example to illustrate the data structure)

Property	**Meaning**	**Data Structure**
AtomInfo	atomic number valence electron number atomic symbol atom sphere radius radial mesh point number maximum angular momentum core electron shell structure valence electron shell structure	*AtomInfo.Z = 26* *AtomInfo.Zvalence = 8* *AtomInfo.symbol = 'Fe'* *AtomInfo.radius = 2.667* *AtomInfo.number = 400* *AtomInfo.Lmax = 2* *AtomInfo.shell.Core = 'Ar'* *AtomInfo.shell.Valence = '4s2 4p0 3d6'*
SpinType	spin type	*SpinType = neutral_spin*
XCfunctional	XC-functional	*XCfunctional = XCfunctional_LDA_PZ81*
equation	radial equation	*equation = SchrodingerEquation*
OrbitalSet	atomic orbital set	*OrbitalSet(1).spin{1} = class_lmtoOrbital* *OrbitalSet(2).spin{1} = class_lmtoOrbital* ..... *OrbitalSet(8).spin{1} = class_lmtoOrbital*
CoreIndex	core orbital indices	*CoreIndex = [1,2,3,4,5]*
ValenceIndex	valence orbital indices	*ValenceIndex = [6,7,8]*
rrData	radial mesh: $r_i = R(i)$	*rrData = zeros(1,400)*
drData	radial mesh: $dr_i = R'(i)$	*drData = zeros(1,400)*
vvData	atomic potential	*vvData.spin{#} = zeros(1,400)*
rhoData	charge density	*rhoData.spin{#} = zeros(1,400)*
Parameter	element parameters	*N/A*

www.ebook3000.com

@class_lmtoElement has 6 methods: *initialize*, *get*, *set*, *calculatePsi*, *calculateRho*, *calculateV*, which are explained as follows.

The method *initialize* is to initialize an object by doing six things: (1) Set up the property of *AtomInfo*. (2) Generate the radial mesh by calling the library function *generate_rmesh*. (3) Set up the initial guess of $\rho_{iq}(r)$ according to Eq. (4.4). Notice that $\rho_{iq}(r)$ is all zeros in an empty sphere. (4) Set up *CoreIndex* and *ValenceIndex*. The indices and occupations of core orbitals and valence orbitals are read from the string *shell.Core* and *shell.Valence* by using the internal function *parseShell*. (5) Handle the case of a VCA atom in which the effective nuclear charge and the total electron number are not an integer [7]. (6) Set up the atomic orbital set. Each atomic orbital is an object of *@class_lmtoOrbital* (See Section 5.1 for more details).

The methods *set* and *get* are to set and get properties to and from an object. Although the properties of an object are accessible directly (by default the properties are public), it is a good practice to access the data of an object through the *set* and *get* methods. The reason is that *set* and *get* provide a flexible interface which allows one to refer to a property with different names, to change the data structure, or even to do some data processing. For example, *@class_lmtoElement* does not have the property *ValenceOrbitalSet*. With the aid of *get* method, we can get the valence orbitals by using *get(object, 'ValenceOrbitalSet')* as if *ValenceOrbitalSet* were a property. In the *get* method, it is actually implemented by two lines

```
ValenceIndex = object.ValenceIndex;
value = object.OrbitalSet(ValenceIndex);
```

The method *calculatePsi* is to solve the radial equation of core and valence orbitals defined by Eq. (3.102). For core orbitals, the boundary condition is to form a bound state. For valence orbitals, the boundary condition is

$$\frac{\chi'_{iq}(R_{iq})}{\chi_{iq}(R_{iq})} = -\frac{l}{R_{iq}}, \tag{4.5}$$

which matches the logarithmic derivative of $K_l(r)$ at $r = R_{iq}$. Notice that the linearization center E^0_{ilq} is not needed in calculation of valence orbital.

The method *calculateRho* is to calculate the total charge density for given orbitals and occupations. The formula is similar to that of

Eqs. (3.135,3.136,3.137) except that the occupations of valence orbitals have been determined by the shell configuration. Namely $\rho_{iq}(r)$ is obtained as

$$\rho_{iq}(r) = \frac{1}{4\pi} \sum_{l} N_{iql} \phi_{iql}^2(r), \tag{4.6}$$

where N_{iql} is the occupation number of the orbital $\phi_{iql}(r)$ and the factor $\frac{1}{4\pi}$ comes from the angular part.

The methods *calculateV* is to calculate the atomic potential for a given charge density. The atomic potential is composed of three parts, the nuclear potential, the Hartree potential, and the exchange-correction potential, defined by Eqs. (3.145,3.146,3.147), respectively. The Hartree potential is calculated using the library function *SphericalHartree*. The exchange-correlation potential is calculated using the library class *@XCfunctional_** where * refers to the XC-functional type (e.g., LDA_PZ81).

The methods of *@class_lmtoElement* are organized by the solver *@SCFsolver_Atom* to solve the charge density of an isolated atom. For example, *SCF_f* of *@SCFsolver_Atom* carries out one self-consistent iteration by calling three methods of *@class_lmtoElement*.

```
function system = SCF_f(solver, system)
system = calculatePsi(system);
system = calculateRho(system);
system = calculateV(system);
end %SCF_f
```

Despite of the simplicity, *@class_lmtoElement* and *@SCFsolver_Atom* illustrate the interactions between dsim-classes and dsim-solvers. In short, dsim-solvers organize the self-consistent iteration by calling the methods of dsim-classes. The methods of dsim-classes implement their functions by calling the supporting libraries, which will be the subject of the next section.

4.9 NanoDsim: supporting libraries

We have seen in Section 4.8 that *@class_lmtoElement* uses library functions and classes to carry out sophisticated calculations. Thus all the complexities are hidden in the supporting libraries. NanoDsim has 12 supporting libraries and their contents are summarized as follows.

www.ebook3000.com

Library	Description	Content	Reference
01	general	*AngularMomentum* *SpinType* *NumericalIntegral* *NumericalDerivative* *SchemeManager* *ClassManager* *ASA_occupancy*	Chapter 4
02	structure constant	*GauntCoefficient* *SphereicalHarmonics* *StructureConstant* *SupercellMaker*	Chapter 5
03	electrostatic solver	*MadelungConstant* *SphericalHartree*	Chapter 5
04	surface Green's function	*SurfaceGreenFunction*	Chapter 6
05	radial equation solver	*RadialEquation* *RadialMesh*	Chapter 5
06	contour integral	*GaussianQuadrature* *@IntegralPath* *Esampling* *K-sampling*	Chapter 5
07	principal layer algorithm	*PrincipalLayer* *LayerPartition*	Chapter 7
08	input / output	*InputManager* *OutputManager*	Chapter 4
09	memory / harddisk	*MemoryManager* *MemoryMonitor* *FileManager* *HarddiskVariable* *DataManager*	Chapter 7
10	database	*@PeriodicTable* *@CONSTANT*	Chapter 4
11	XC-functional	*@XCfunctional_LDA_PZ81* *@XCfunctional_GGA_PBE96* *@XCfunctional_MBJ_TB09*	Appendix
12	parallelization	*JobManager* *InterfaceMPI*	Chapter 7

In this section, we shall briefly discuss *Library01*, *Library08*, *Library10*, and leave the discussion of other libraries to Chapter 5, 6, 7.

Library01 contains some general functions and classes, including the subfolders *AngularMomentum*, *SpinType*, *NumericalIntegral*, *NumericalDerivative*, *SchemeManager*, *ClassManager*, *ASA_occupancy*. The subfolder *AngularMomentum* contains four functions related to the angular momentum operations. The subfolder *SpinType* contains two classes *@collinear_spin* and *@neutral_spin* to classify collinear spin type and neutral spin type. The subfolder *NumericalIntegral* and *NumericalDerivative* contain some useful functions to evaluate the integral and derivative of numerical functions. The subfolder *SchemeManager* contains a status class *@scheme_mto* to control the MTO/LMTO scheme of potential parameters. The subfolder *ClassManager* contains a function *constructClass* which is an interface to different class constructors. The subfolder *ASA_occupancy* contains a function *evaluateASA* which evaluates the unit cell occupation rate for a given ASA scheme.

Library08 has two subfolders, *InputManger* and *OutputManger*, controlling the input and output of NanoDsim. The subfolder *InputManger* contains a function *parseInputFile* which analyzes the input files and parses the input parameters. The subfolder *OutputManger* contains four classes: *@scheme_info* (control the output level of detail), *@scheme_plot* (turn on/off

the function of figure plot), *@scheme_log* (output the work log to screen and/or file), *@report_scf* (output the detailed information of self-consistent iteration).

Library10 contains two database classes *@CONSTANT* and *@PeriodicTable*. *@CONSTANT* is a collection of various physical constants. For example, *get(CONSTANT, 'aB')* provides the value of Bohr radius ($\approx 0.529\mathring{A}$); *get(CONSTANT, 'Ha')* provides the value of Hartree energy ($\approx 27.2eV$). *@PeriodicTable* is a periodic table which can tell the atomic number and the shell structure of an element. For example, *getAtomicNumber(PeriodicTable, 'Fe')* provides the atomic number of Fe which is 26; *getShellStructure(PeriodicTable, 'Fe')* provides the shell structure of Fe which is $[Ar]4s^2 3d^6$.

4.10 NanoDsim: implementation and debugging

In previous sections, we have described the plan of NanoDsim package. The remaining task is to find proper algorithms to implement the design class by class, solver by solver, calculator by calculator, and library by library. We shall discuss the detailed implementations and algorithms in Chapter 5 to 7, and in this section we shall focus on the debugging. As mentioned in Section 4.1, human programmers are prone to making mistakes in coding such as typos, syntax errors, unexpected special cases, and missing factors or terms. As a result, bugs are inevitable if a program is sufficiently long and sophisticated. An intriguing question is how to eliminate all the bugs and make a stable and reliable package.

We don't have a general answer to the question. Nevertheless we have some tips according to our experiences of developing NanoDsim. Tip 1: Use high-level language (e.g., MATLAB) as much as possible to reduce the number of bugs. The reason is that high-level language code is much shorter than the corresponding low-level language code and hence there is less chance of making mistakes. Tip 2: MATLAB has a powerful debugger which can check variable values or run a script at any break point. It is worth mentioning that the debugger can also trace the flow of the code. Tip 3: Read the code three times after you write it. It is amazing that a considerable number of bugs can be found by this primitive method. Tip 4: Do the implementation version by version. In the first basic version, try not to use fancy tricks to complicate the code. The only goal of the basic version is to do things right. In the following upgraded versions, you can optimize the code with various tricks and check the validity against the basic version.

www.ebook3000.com

To sum up, a recommended working procedure is as follows: write the code with a high-level language; read it carefully three times; run it through by eliminating syntax errors; correct it with debugger and independent verifications; improve the performance by making optimization.

Although careful reading may eliminate most syntax errors, it does not help too much in finding logical errors. Due to psychological inertia, very often we are blind to our own mistakes. To pick up the logical bugs, we need to design some independent verifications including local tests and global tests. The local tests are designed to check individual functions and classes, while the global tests are designed to check the cooperation of functions and classes.

Let us first design some local tests of NanoDsim. (1) To test the structure constant in *Library02*, we compare the LHS and RHS of the identity Eq. (3.15). (2) To test the Madelung potential of *Library03*, we compare the Madelung potential to the Hartree potential solved by Poisson equation. (3) To test the surface Green's function in *Library04*, we work on some special Hamiltonians having analytical solutions (see e.g., Section 2.9). (4) To test the radial equation solver in *Library05*, we work on some special potentials having analytical solutions (e.g., Hydrogen atom or harmonic potential well). (5) To test the contour integral in *Library06*, we work on some special integrands which can be evaluated analytically. (6) To test the principal layer algorithm in *Library07*, we compare the results to the direct inverse of full matrices. (7) To test the iterative algorithm of solving the NECPA equations, we check the identity $\overline{\mathcal{G}^<} = \overline{\mathcal{G}^a} - \overline{\mathcal{G}^r}$ for the special case of $f_L \equiv f_R \equiv 1$.

Next we design some global tests of NanoDsim. (1) To verify the calculation of clean bulk systems, we compare the results of two approaches, Green's function approach and wave function approach. (2) To verify the calculation of disordered bulk systems, we compare the results of two methods, the CPA method and the supercell method. (3) To verify the calculation of clean two-probe systems, we compare the results of two models for a perfect crystal, the bulk model and the two-probe model (see Fig. 4.3). (4) To verify the calculation of disordered two-probe systems, we compare the results of two methods, the NECPA method and the supercell method (see Section 8.5). (5) To verify the calculation of density of states, we compare the results of two approaches, Green's function approach and wave function approach (see Section 5.10). (6) To verify the calculation of transmission coefficient, we compare the results of two approaches, Green's function approach and wave function approach (see Section 6.9).

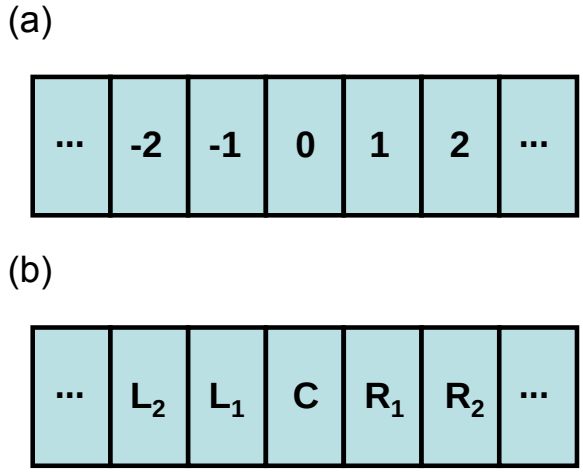


Fig. 4.3 A test of two-probe calculation by using two different points of view for a periodic bulk system. (a) From the bulk point of view, the periodic bulk system is composed of unit cells. (b) From the two-probe point of view, the periodic bulk system can be partitioned into the left lead, the central region, and the right lead.

To sum up, we have emphasized the importance of debugging after the implementation. For a simple program, the implementation may take 70% of the effort and the debugging may take the other 30%. For a complicated package like NanoDsim, the percentages can even be reversed. In the following chapters, we shall focus on the implementation details. Here we would like to point out that the work of debugging is by no means easier than the implementation.

Bibliography

[1] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1987).
[2] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
[3] There are some complexities in the compilation: (1) One needs to install a MATLAB-supported C or Fortran compiler and set it up as the default compiler by using the command *mex -setup*. (2) Some Fortran compilers do not support *%val* and one has to bypass it by using *mxCopyPtrToReal8* and *mxCopyReal8ToPtr* in the gateway routine. (3) In some occasions, it is necessary to edit the file of *fullfile(matlabroot, 'bin', 'mexopts.sh')* to set up the compilation options. (4) The compilation depends on the type of platform

www.ebook3000.com

(Windows, Linux, etc.), the MATLAB version, and the compiler. Please refer to MATLAB's documents for more information.

[4] To compile *epsC_Ccode.c* in Linux 64-bit (AMD) system installed with gcc 4.4.6, use the command *mex epsC_Ccode.c* in MATLAB (R2013b), and obtain *epsC_Ccode.mexa64*.

[5] To compile *epsC_Fcode.f90* in Linux 64-bit (AMD) system installed with gcc 4.4.6, edit mexopts.sh to set up *FC='gfortran'* and *FFLAGS='-fPIC -fno-omit-frame-pointer -fexceptions -fdefault-integer-8'* in the glnxa64 block, use the command *mex epsC_Fcode.f90* in MATLAB (R2013b), and obtain *epsC_Fcode.mexa64*.

[6] T. Sintes, *Sams Teach Yourself Object Oriented Programming in 21 Days* (Sams, Indiana, 2002).

[7] For example, consider a Si bulk doped with 10^{-4} P atoms. From the VCA point of view, a virtual "Si-P" atom has the nuclear charge 14.0001 and the same amount of electron charge.

Chapter 6

NanoDsim: two-probe systems

In this chapter, we shall discuss how to implement dsim-classes, dsim-solvers, and dsim-calculators for two-probe systems. Some algorithms in two-probe systems are similar to those of bulk systems, such as calculating the structure constant, solving the radial equation, and evaluating the complex contour integral. For a discussion of these algorithms, we refer readers to Chapter 5. Other algorithms are unique to two-probe systems, such as calculating the lead self-energies and evaluating the real-axis integral. These algorithms are the focus of this chapter. Section 6.1 and 6.2 will be devoted to the design of dsim-classes and dsim-solvers for two-probe systems. Section 6.3 to 6.9 will be devoted to various algorithms used in both self-consistent and post-analysis calculations. Section 6.10 will be devoted to the verification of the algorithms implemented for two-probe systems.

6.1 Two-probe classes

As mentioned in Section 4.8, there are four dsim-classes for two-probe systems. The hierarchy of the dsim-classes is @class_necpaTwoProbe $\supset$ @class_necpaAtom $\supset$ @class_lmtoAtom $\supset$ @class_lmtoOrbital, where $\supset$ refers to the class aggregation. The design of @class_lmtoAtom and @class_lmtoOrbital have been explained in Section 5.1. The design of @class_necpaTwoProbe and @class_necpaAtom will be explained in the following subsections.

www.ebook3000.com

6.1.1 *@class_necpaTwoProbe*

The class @class_necpaTwoProbe has 21 properties which are summarized in Table 6.1. In the column of data structure, # stands for an array index, * stands for a string, and *[1]* stands for a scalar.

Property	**Meaning**	**Data Structure**
AtomSet	atomic sites of central region	*AtomSet(#) = @class_necpaAtom*
AtomSetL	atomic sites of left lead	*AtomSetL(#) = @class_cpaAtom*
AtomSetR	atomic sites of right lead	*AtomSetR(#) = @class_cpaAtom*
StructureConstant	structure constant	*StructureConstant.(XX)(#).Displacement = zeros(1,3)* *StructureConstant.(XX)(#).data = zeros(#,#)* *XX = 'CC', 'CL', 'CR', 'L11', 'L12', 'R11', 'R12'*
MadelungConstant	Madelung constant	*MadelungConstant.(XX) = zeros(#,#,4)* *XX = 'CC', 'LC', 'RC'*
MadelungPotential	Madelung potential	*MadelungPotential.(X) = zeros(1,#)* *X = 'C', 'L', 'R', 'C0', 'L0', 'R0', 'leadL', 'leadR'*
equation	radial equation	*equation = @*Equation*
SpinType	spin type	*SpinType = @*_spin*
XCfunctional	XC-functional	*XCfunctional = @XCfunctional_**
UnitCellVector	unit cell vector	*UnitCellVector.(X) = zeros(3,3)* *X = 'C', 'L', 'R'*
ElementDatabase	element database	*ElementDatabase(#) = @class_lmtoElement*
LengthOmega	Wigner–Seitz radius	*LengthOmega.(X) = [1]* *X = 'C', 'L', 'R'*
ComplexContour	complex contour	*ComplexContour(#) = @IntegralPath*
RealAxis	real-axis	*RealAxis = @IntegralPath*
Selfenergy	lead self-energy	*Selfenergy.(X) = @SigmaBlock_** *X = 'L', 'R'*
SkData	Fourier transform	*SkData = @SkBlock_**
Partition	principal layer partition	*Partition(#).SiteIndex = zeros(1,#)* *Partition(#).MatrixIndex = zeros(1,#)*
JobManager	parallel job manager	*JobManager = @manager_parajob*
JobManager_site	parallel job manager	*JobManager_site = @manager_parajob*
Quantity	physical quantities	···
Parameter	system parameters	···

(6.1)

The most important properties are *AtomSet*, *Selfenergy*, and *StructureConstant*. *AtomSet* is an object array of @class_necpaAtom, which is a list of atomic sites in the central region. The object @class_necpaAtom contains the potential parameters and the nonequilibrium coherent potential of a single atomic site. *Selfenergy* is a data structure containing the left and right lead self-energies. *StructureConstant* is a structure array, containing the structure constant matrices of different region pairs and unit cell displacements (see Fig. 3.3). The diagonal blocks from *AtomSet*, the off-diagonal matrices from *StructureConstant*, and the matrix blocks of *Selfenergy* constitute the effective LMTO Hamiltonian.

The class @class_necpaTwoProbe has 35 methods which are summarized in Table 6.2.

	Method	Description
1	*class_necpaTwoProbe*	CLS: class constructor
2	*get*	CLS: get value of a given variable
3	*set*	CLS: set value to a given variable
4	*initialize*	INIT: initialize the class
5	*setupParameter*	INIT: set up system parameters
6	*prepareElementData*	INIT: prepare *ElementTable*
7	*prepareAtomicData*	INIT: prepare *AtomSet* with isolated atom solutions
8	*updateAtomicData*	INIT: update *AtomSet* with user provided initial guess
9	*preparePartition*	INIT: prepare principal layer partition
10	*prepareStructureConstant*	INIT: prepare structure constant
11	*generateContour*	INIT: generate integral contour
12	*prepareSkData*	INIT: prepare Fourier transformed structure constant
13	*prepareMadelungPotential*	INIT: prepare Madelung constant and Madelung potential
14	*prepareLeadData*	INIT: prepare left and right lead data
15	*prepareSelfenergy*	INIT: prepare lead self-energies
16	*summarize*	SUM: summarize the class
17	*CPA_solution_lowX*	CPA: solve NECPA equations in low concentration limit
18	*CPA_iteration*	CPA: iterate NECPA equations in one step
19	*CPA_solution*	CPA: solve NECPA equations to a given accuracy
20	*calculatePsi*	CAL: calculate atomic orbital
21	*calculateCDG*	CAL: calculate potential parameter
22	*calculateEM*	CAL: calculate energy moment
23	*calculateRho*	CAL: calculate charge density
24	*calculateDrho*	CAL: calculate charge density derivative
25	*calculateKED*	CAL: calculate kinetic energy density
26	*calculateCharge*	CAL: calculate atomic charge
27	*calculateDipole*	CAL: calculate atomic dipole
28	*calculateE0*	CAL: calculate valence linearization center
29	*calculateV*	CAL: calculate potential
30	*isClean*	MISC: check if the system only contains clean sites
31	*isLowX*	MISC: check if low concentration limit is adopted
32	*isEquilibrium*	MISC: check if the system is in equilibrium
33	*clear*	MISC: clear some memory consuming fields
34	*moveVariableToHDVP*	MISC: move some memory consuming variables to hard disk
35	*collectAtom*	MISC: collect atomic data after parallel calculation

(6.2)

The 35 methods can be classified into six groups: The first group CLS is composed of method 1 to 3, which are the methods of standard class operation. The second group INIT is composed of method 4 to 15, which are the methods used in the initialization. The third group SUM is composed of method 16, which is the method used in the summarization. The fourth group CPA is composed of method 17 to 19, which are the methods used in solving the CPA equations. The fifth group CAL is composed of method 20 to 29, which are the methods used in the calculations of various physical

www.ebook3000.com

quantities. The sixth group MISC is composed of method 30 to 35, which are some miscellaneous supportive methods.

6.1.2 *@class_necpaAtom*

The class @class_necpaAtom is a child class of the parent class @class_cpaAtom. In addition to the inherited properties and methods, @class_necpaAtom has its own properties and methods which are summarized in Table 6.3 and Table 6.4.

Property	**Meaning**	**Data Structure**
necpaData	NECPA data	*necpaData.sampleE = zeros(1,#)* *necpaData.weightE = zeros(1,#)* *necpaData.tiltP_r(#).spin{#} = zeros(#,#)* *necpaData.tiltP_d(#).spin{#} = zeros(#,#)* *necpaData.Omega_r(#).spin{#} = zeros(#,#)* *necpaData.Omega_d(#).spin{#} = zeros(#,#)* *necpaData.gr(#).spin{#} = zeros(#,#)* *necpaData.gd(#).spin{#} = zeros(#,#)*

(6.3)

	Method	**Description**
1	*class_necpaAtom*	CLS: class constructor
2	*get*	CLS: get value of a given variable
3	*set*	CLS: set value to a given variable
4	*calcCPA_tiltP_r*	CPA: calculate $\tilde{P}_i^r(E)$ in the NECPA iteration
5	*calcCPA_tiltP_r_lowX*	CPA: calculate $\tilde{P}_i^r(E)$ in low concentration limit
6	*calcCPA_Omega_r*	CPA: calculate $\Omega_i^r(E)$ in the NECPA iteration
7	*calcCPA_gr*	CPA: calculate $\overline{\mathcal{G}_i^r}(E)$ in the NECPA iteration
8	*calcCPA_tiltP_d*	CPA: calculate $\tilde{P}_i^<(E)$ in the NECPA iteration
9	*calcCPA_tiltP_d_lowX*	CPA: calculate $\tilde{P}_i^<(E)$ in low concentration limit
10	*calcCPA_Omega_d*	CPA: calculate $\Omega_i^<(E)$ in the NECPA iteration
11	*calcCPA_gd*	CPA: calculate $\overline{\mathcal{G}_i^<}(E)$ in the NECPA iteration
12	*getEnergyIndex*	CPA: get index of E
13	*get_CPAdata*	CPA: get NECPA data from *necpaData*
14	*set_CPAdata*	CPA: set NECPA data to *necpaData*
15	*calculateEM*	CAL: calculate energy moment
16	*addRealAxisIntegral*	CAL: evaluate the real-axis integral
17	*isEquilibrium*	MISC: check if the system is in equilibrium
18	*clear*	MISC: clear some memory consuming fields
19	*shiftEnergy*	MISC: shift the energy of the atomic site

(6.4)

Most of the new properties and methods of @class_necpaAtom are related to the nonequilibrium calculation. We shall see in Section 6.5 that the calculation in two-probe systems can be divided into the equilibrium part and the nonequilibrium part. The equilibrium part is handled by the parent class @class_cpaAtom, while the nonequilibrium part is handled by the new properties and methods of the child class @class_necpaAtom. To be precise, the energy integral of the density matrix is divided into a complex contour integral and a real-axis integral. On the complex contour, the CPA equations are solved by the methods of @class_cpaAtom, the obtained coherent potential is stored in @class_cpaAtom's property *cpaData*, and the energy integral is evaluated by using @class_cpaAtom's method *ContourIntegral*. On the real axis, the NECPA equations are solved by the methods of @class_necpaAtom, the obtained nonequilibrium coherent potential is stored in @class_necpaAtom's property *necpaData*, and the energy integral is evaluated by @class_necpaAtom's method *addRealAxisIntegral*. Here the dsim-classes @class_cpaAtom and @class_necpaAtom provide another example of class inheritance (see Section 4.4).

6.2 Two-probe solver

The self-consistent calculation for two-probe systems is organized by the dsim-solver @SCFsolver_TwoProbe. @SCFsolver_TwoProbe has 7 methods, *SCFsolver_TwoProbe*, *initialize*, *solve*, *SCF_f*, *SCF_f0*, *SCF_get*, and *SCF_set*. The most important method is *solve* which solves the nonlinear equation array $X = F[X]$ defined by Eq. (4.2). In the method *solve*, the self-consistent iteration is organized by calling the methods *SCF_f0*, *SCF_f*, *SCF_get*, *SCF_set*, and mixing algorithms are used to accelerate the convergence (see Appendix A.20).

Among the 7 methods, the first 3 methods are more or less the same as other dsim-solvers (see Section 4.6). The unique methods related to two-probe systems are *SCF_f0*, *SCF_f*, *SCF_get*, and *SCF_set*. The method *SCF_f* consists of a complete self-consistent iteration. The mapping from the code to the formulas of Section 3.11 is commented as follows.

```
function system = SCF_f(solver, system)
system = calculatePsi(system);              %step-4
system = calculateCDG(system);              %step-5
if isLowX(system)                           %step-6
    system = CPA_solution_lowX(system);
```

www.ebook3000.com

```
else
    system = CPA_iteration(system);
end
system = calculateEM(system);              %step-7
system = calculateRho(system);             %step-8
system = calculateDrho(system);
system = calculateKED(system);
system = calculateCharge(system);          %step-9
system = calculateDipole(system);
system = calculateE0(system);
system = calculateV(system);               %step-10,11,12
end %SCF_f
```

The method *SCF_f0* consists of an incomplete self-consistent iteration. This method is useful at the beginning of a self-consistent calculation where an ignition is needed. Essentially *SCF_f0* solves the atomic orbitals, calculates potential parameters, and solves the NECPA equations to obtain a reasonable initial guess of the nonequilibrium coherent potential. Notice that it is unnecessary to obtain an initial guess of the nonequilibrium coherent potential in clean systems or in the low concentration limit.

```
function system = SCF_f0(solver, system)
if isClean(system) | isLowX(system)
    return
end
print(class_log, 'pre-calculating CPA...\n')
system = calculatePsi(system);
system = calculateCDG(system);
ControlParameter.mixer.Omega_r = Mixer_DoNothing;
ControlParameter.mixer.Omega_d = Mixer_DoNothing;
ControlParameter.maxstep = 60;
ControlParameter.tolerance.Omega_r = 1e-4;
ControlParameter.tolerance.Omega_d = 1e-4;
system = CPA_solution(system, ControlParameter);
print(class_log, '\n')
end %SCF_f0
```

The method *SCF_get* (*SCF_set*) is to get (set) X from (to) an object of @class_necpaTwoProbe. In two-probe systems, X is composed of the variables of $\{V_{iq}(r)\}$, $\left\{E^0_{ilq}\right\}$, $\{\Omega^r_i, \Omega^<_i\}$, and μ, as listed in Table (4.3).

6.3 Ewald sum technique

This section discusses the algorithms for calculating the Madelung potential in two-probe systems. Since two-probe systems are periodic in the transverse dimensions, one can calculate the Madelung potential layer by layer in the transport direction

$$\begin{aligned} V_{MD} &= \sum_{j\neq i} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{\mathbf{P}_j \cdot (\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3} \\ &= \sum_{j\neq i} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j|} - \mathbf{P}_j \cdot \nabla \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \\ &= \sum_{k=-\infty}^{+\infty} \sum_{j\in A_k} \sum_{\mathbf{T}} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j - \mathbf{T}|} - \mathbf{P}_j \cdot \nabla \frac{1}{|\mathbf{r}_i - \mathbf{r}_j - \mathbf{T}|} \\ &= \sum_{k=-\infty}^{+\infty} \sum_{j\in A_k} Q_j \varphi(\mathbf{r}_i - \mathbf{r}_j) - \mathbf{P}_j \cdot \nabla \varphi(\mathbf{r}_i - \mathbf{r}_j), \end{aligned} \tag{6.5}$$

where k is the layer index and $j \in A_k$ means that the atomic site is in the k^{th} 2d unit cell. $\mathbf{T} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2$ is the 2d lattice vector where n_1, n_2 are integers and $\mathbf{a}_1, \mathbf{a}_2$ are the 2d unit cell vectors. To exclude the self-interaction, it is required that $\mathbf{T} \neq 0$ for $\mathbf{r}_i = \mathbf{r}_j$. The function $\varphi(\mathbf{r})$ is defined by

$$\varphi(\mathbf{r}) = \sum_{\mathbf{T}} \frac{1}{|\mathbf{r} - \mathbf{T}|}, \tag{6.6}$$

which satisfies the Poisson equation

$$\nabla^2 \varphi(\mathbf{r}) = -4\pi \sum_{\mathbf{T}} \delta(\mathbf{r} - \mathbf{T}), \tag{6.7}$$

where $\mathbf{T} \neq 0$ for $\mathbf{r} = 0$. The algorithm for evaluating $\varphi(\mathbf{r})$ will be discussed in Section 6.3.1; The summation over k will be discussed in Section 6.3.2; The arbitrary addible constant to the Madelung potential will be discussed in Section 6.3.3.

6.3.1 *2d Madelung potential*

This subsection is devoted to the algorithm for calculating $\varphi(\mathbf{r})$ which is referred to as 2d Madelung constant. In Section 5.4, we have discussed the Ewald sum technique for calculating 3d Madelung constant. Here the technique will be generalized to the 2d case [1–3]. The key idea is still to enclose the point charges with Gaussian packets. The differences from the

www.ebook3000.com

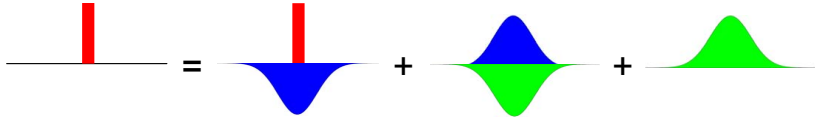


Fig. 6.1 Sketch of the idea for the 2d Ewald sum technique. A point charge (red) is neutralized by a spherical Gaussian packet (blue), and the spherical Gaussian packet (blue) is neutralized by a layer Gaussian packet (green).

3d case are: (i) Two types of Gaussian packet are involved; (ii) Charge neutrality is not required.

The total charge density in the RHS of Eq. (6.7) is split into three terms (see Fig. 6.1):

$$\rho(\mathbf{r}) = \rho_1(\mathbf{r}) + \rho_2(\mathbf{r}) + \rho_3(\mathbf{r}), \tag{6.8}$$

$$\rho_1(\mathbf{r}) = \sum_{\mathbf{T}} \left[\delta(\mathbf{r}-\mathbf{T}) - \rho_\sigma(\mathbf{r}-\mathbf{T})\right], \tag{6.9}$$

$$\rho_2(\mathbf{r}) = \sum_{\mathbf{T}} \rho_\sigma(\mathbf{r}-\mathbf{T}) - \rho_z(\mathbf{r}), \tag{6.10}$$

$$\rho_3(\mathbf{r}) = \rho_z(\mathbf{r}), \tag{6.11}$$

where $\rho_\sigma(\mathbf{r})$ is a spherical Gaussian packet defined by

$$\rho_\sigma(\mathbf{r}) \equiv \sigma^3 \pi^{-\frac{3}{2}} e^{-\sigma^2 r^2},$$

and $\rho_z(\mathbf{r})$ is a layer Gaussian packet defined by

$$\rho_z(\mathbf{r}) \equiv \frac{1}{A}\frac{\sigma}{\sqrt{\pi}} e^{-\sigma^2 z^2}.$$

Notice that $\rho_\sigma(\mathbf{r})$ has been normalized to $\int d^3r \rho_\sigma(\mathbf{r}) = 1$ and $\rho_z(\mathbf{r})$ normalized to $\int dz \rho_z(\mathbf{r}) = \frac{1}{A}$ where A is the area of the 2d unit cell. The potential of $\rho_1(\mathbf{r})$ has been evaluated by Eq. (5.23) in real space; the potential of $\rho_2(\mathbf{r})$ will be solved in the reciprocal space; the potential of $\rho_3(\mathbf{r})$ will be calculated analytically.

The potential of $\rho_2(\mathbf{r})$ can be solved from the Poisson equation

$$\nabla^2 \varphi_2(\mathbf{r}) = -4\pi\rho_2(\mathbf{r}),$$

in the reciprocal space. $\varphi_2(\mathbf{r})$ is obtained as

$$\varphi_2(\mathbf{r}) = 4\pi \int \frac{dk}{2\pi} \sum_{\mathbf{K}\neq\mathbf{0}} \rho(\mathbf{K},k) \frac{e^{i\mathbf{K}\cdot\mathbf{X}} e^{ikz}}{K^2+k^2}, \tag{6.12}$$

where $\rho(\mathbf{K},\mathbf{k})$ is the Fourier transform of $\rho_2(\mathbf{r})$

$$\rho(\mathbf{K},k) = \int dz \frac{1}{A} \int_A d^2X \rho_2(\mathbf{r}) e^{-i\mathbf{K}\cdot\mathbf{X}} e^{-ikz}. \tag{6.13}$$

Here $\mathbf{r} \equiv \mathbf{X} + \mathbf{z}$ is decomposed into the in-plane component $\mathbf{X}$ and the perpendicular component $\mathbf{z} = z\mathbf{e}_z$ where $\mathbf{e}_z$ is the unit vector perpendicular to the 2d lattice plane. $\mathbf{K}$ is the 2d reciprocal lattice vector defined by $\mathbf{K} = m_1\mathbf{b}_1 + m_2\mathbf{b}_2$ where $\mathbf{a}_i \cdot \mathbf{b}_j = 2\pi\delta_{ij}$ and m_i is an integer. $k \in (-\infty, +\infty)$ is the wave vector in the perpendicular direction. Notice that the $\mathbf{K} = \mathbf{0}$ term has been excluded from the RHS of Eq. (6.12) because of the charge neutrality $\int d^3r \rho_2(\mathbf{r}) = 0$. $\rho(\mathbf{K},\mathbf{k})$ can be evaluated analytically as follows

$$\begin{aligned}
\rho(\mathbf{K},k) &= \int dz \frac{1}{A} \int_A d^2X \left[\sum_{\mathbf{T}} \rho_\sigma(\mathbf{r}-\mathbf{T}) - \rho_z(\mathbf{r}) \right] e^{-i\mathbf{K}\cdot\mathbf{X}} e^{-ikz} \\
&= \int dz \frac{1}{A} \left[\int_A d^2X \sum_{\mathbf{T}} \rho_\sigma(\mathbf{r}-\mathbf{T}) - \int_A d^2X \rho_z(\mathbf{r}) \right] e^{-i\mathbf{K}\cdot\mathbf{X}} e^{-ikz} \\
&= \int dz \frac{1}{A} \left[\int d^2X \rho_\sigma(\mathbf{r}) - \int_A d^2X \rho_z(\mathbf{r}) \right] e^{-i\mathbf{K}\cdot\mathbf{X}} e^{-ikz} \\
&= \frac{1}{A}\sigma^3\pi^{-\frac{3}{2}} \int dz \int d^2X e^{-\sigma^2X^2} e^{-\sigma^2z^2} e^{-i\mathbf{K}\cdot\mathbf{X}} e^{-ikz} \\
&\quad - \delta_{\mathbf{K}} \int dz \frac{1}{A} \frac{\sigma}{\sqrt{\pi}} e^{-\sigma^2z^2} e^{-ikz} \\
&= \frac{1}{A} \exp\left[-\frac{K^2+k^2}{4\sigma^2} \right] - \delta_{\mathbf{K}} \frac{1}{A} \exp\left[-\frac{k^2}{4\sigma^2} \right],
\end{aligned} \tag{6.14}$$

where

$$\frac{1}{A} \int_A e^{i\mathbf{K}\cdot\mathbf{X}} d^2X = \delta_{\mathbf{K}}$$

and

$$\int e^{-\sigma^2z^2} e^{-ikz} dz = \frac{\sqrt{\pi}}{\sigma} \exp\left(-\frac{k^2}{4\sigma^2} \right)$$

are used in the derivation.

Substituting Eq. (6.14) to Eq. (6.12), $\varphi_2(\mathbf{r})$ is obtained as

$$\begin{aligned}
\varphi_2(\mathbf{r}) &= \frac{4\pi}{A} \sum_{\mathbf{K}\neq\mathbf{0}} \int \frac{dk}{2\pi} \frac{e^{i\mathbf{K}\cdot\mathbf{X}} e^{ikz}}{K^2+k^2} \exp\left(-\frac{K^2+k^2}{4\sigma^2} \right) \\
&= \frac{4\pi}{A} \sum_{\mathbf{K}\neq\mathbf{0}} \int_0^\infty \frac{dk}{2\pi} \frac{2\cos(\mathbf{K}\cdot\mathbf{X})\cos(kz)}{K^2+k^2} \exp\left(-\frac{K^2+k^2}{4\sigma^2} \right).
\end{aligned} \tag{6.15}$$

www.ebook3000.com

By using the integral ($\alpha > 0$, $\mathrm{Re}\beta > 0$, $\mathrm{Re}\gamma > 0$)

$$\int_0^\infty e^{-\beta x^2} \cos \alpha x \frac{dx}{\gamma^2 + x^2} = \frac{\pi}{4\gamma} e^{\beta\gamma^2} \left[e^{-\gamma\alpha} \mathrm{erfc}\left(\gamma\sqrt{\beta} - \frac{\alpha}{2\sqrt{\beta}} \right) + e^{\gamma\alpha} \mathrm{erfc}\left(\gamma\sqrt{\beta} + \frac{\alpha}{2\sqrt{\beta}} \right) \right],$$

Eq. (6.15) can be simplified as

$$\varphi_2(\mathbf{r}) = \frac{\pi}{A} \sum_{\mathbf{K}\neq\mathbf{0}} \frac{\cos(\mathbf{K}\cdot\mathbf{X})}{K} \left[e^{-Kz} \mathrm{erfc}\left(\frac{K}{2\sigma} - z\sigma \right) + e^{Kz} \mathrm{erfc}\left(\frac{K}{2\sigma} + z\sigma \right) \right]. \tag{6.16}$$

The RHS of Eq. (6.16) is proportional to $\lambda\left(\frac{K}{2\sigma}, \sigma z\right)$ where $\lambda(a,x)$ is a dimensionless function defined by

$$\lambda(a,x) \equiv e^{-2ax} \mathrm{erfc}(a-x) + e^{2ax} \mathrm{erfc}(a+x).$$

Since $\lambda(a,x) \approx \frac{2}{\sqrt{\pi}} \frac{1}{a} e^{-\left(a^2+x^2\right)}$ for $a \gg 1$, the $\mathbf{K}$ terms decay according to $\exp\left[-\left(\frac{K^2}{4\sigma^2} + \sigma^2 z^2 \right) \right]$.

The potential of $\rho_3(\mathbf{r})$ can be calculated analytically as follows

$$\begin{aligned}
\varphi_3(\mathbf{r}) &= \int dz' \rho_3(z') 4\pi \frac{-|z-z'|}{2} \\
&= -2\pi \int dz' \frac{1}{A} \frac{\sigma}{\sqrt{\pi}} e^{-\sigma^2 z'^2} |z-z'| \\
&= -\frac{2\sqrt{\pi}\sigma}{A} \int dz'' e^{-\sigma^2\left(z''+z\right)^2} |z''| \\
&= -\frac{2\sqrt{\pi}\sigma}{A} \int_0^\infty dz'' \left[e^{-\sigma^2\left(z''+z\right)^2} + e^{-\sigma^2\left(z''-z\right)^2} \right] z'' \\
&= -\frac{2\pi}{A} z \mathrm{erf}(\sigma z) - \frac{2\sqrt{\pi}}{A} \frac{1}{\sigma} e^{-\sigma^2 z^2},
\end{aligned} \tag{6.17}$$

where $\mathrm{erf}(x)$ is the error function defined by

$$\mathrm{erfc}(x) \equiv \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.$$

Finally, the solution to Eq. (6.7) is derived as $\varphi(\mathbf{r}) = \varphi_1(\mathbf{r}) + \varphi_2(\mathbf{r}) + \varphi_3(\mathbf{r}) + C$

$$\begin{aligned}
\varphi(\mathbf{r}) = &\sum_{\mathbf{T}} \frac{1}{|\mathbf{r}-\mathbf{T}|} \mathrm{erfc}(\sigma|\mathbf{r}-\mathbf{T}|) \\
&+ \frac{\pi}{A} \sum_{\mathbf{K}\neq\mathbf{0}} \frac{\cos(\mathbf{K}\cdot\mathbf{X})}{K} \left[e^{-Kz} \mathrm{erfc}\left(\frac{K}{2\sigma} - z\sigma \right) + e^{Kz} \mathrm{erfc}\left(\frac{K}{2\sigma} + z\sigma \right) \right] \\
&- \frac{2\pi}{A} z \mathrm{erf}(\sigma z) - \frac{2\sqrt{\pi}}{A} \frac{1}{\sigma} e^{-\sigma^2 z^2} + C,
\end{aligned} \tag{6.18}$$

where C is an arbitrary addible constant which will be discussed in Section 6.3.3.

At $\mathbf{r} = 0$, the $\mathbf{T} = \mathbf{0}$ term needs to eliminated to exclude the self-interaction

$$\varphi(\mathbf{r}=\mathbf{0}) = \lim_{\mathbf{r}\longrightarrow 0}\left[\varphi(\mathbf{r}) - \frac{1}{r}\right].$$

By using the limit

$$\lim_{\mathbf{r}\longrightarrow 0}\frac{1}{|\mathbf{r}|}\mathrm{erfc}(\sigma|\mathbf{r}|) - \frac{1}{r} = -\frac{2}{\sqrt{\pi}}\sigma,$$

one obtains

$$\begin{aligned}\varphi(\mathbf{r}=\mathbf{0}) = &\sum_{\mathbf{T}\neq\mathbf{0}}\frac{1}{|\mathbf{T}|}\mathrm{erfc}(\sigma|\mathbf{T}|) - \frac{2}{\sqrt{\pi}}\sigma \\ &+\frac{\pi}{A}\sum_{\mathbf{K}\neq\mathbf{0}}\frac{\cos(\mathbf{K}\cdot\mathbf{X})}{K}2\mathrm{erfc}\left(\frac{K}{2\sigma}\right) \\ &-\frac{2\sqrt{\pi}}{A}\frac{1}{\sigma}.\end{aligned} \tag{6.19}$$

The gradient of $\varphi(\mathbf{r})$ can be evaluated as

$$\begin{aligned}-\nabla\varphi(\mathbf{r}) = &\sum_{\mathbf{T}}\frac{\mathbf{r}-\mathbf{T}}{|\mathbf{r}-\mathbf{T}|^3}\left[\mathrm{erfc}(\sigma|\mathbf{r}-\mathbf{T}|) + \frac{2}{\sqrt{\pi}}e^{-\sigma^2|\mathbf{r}-\mathbf{T}|^2}\sigma|\mathbf{r}-\mathbf{T}|\right] \\ &+\frac{\pi}{A}\sum_{\mathbf{K}\neq\mathbf{0}}\mathbf{K}\frac{\sin(\mathbf{K}\cdot\mathbf{X})}{K}\left[e^{-Kz}\mathrm{erfc}\left(\frac{K}{2\sigma}-z\sigma\right) + e^{Kz}\mathrm{erfc}\left(\frac{K}{2\sigma}+z\sigma\right)\right] \\ &+\mathbf{e}_z\frac{\pi}{A}\sum_{\mathbf{K}\neq\mathbf{0}}\cos(\mathbf{K}\cdot\mathbf{X})\left[e^{-Kz}\mathrm{erfc}\left(\frac{K}{2\sigma}-z\sigma\right) - e^{Kz}\mathrm{erfc}\left(\frac{K}{2\sigma}+z\sigma\right)\right] \\ &+\mathbf{e}_z\frac{2\pi}{A}\mathrm{erf}(\sigma z),\end{aligned} \tag{6.20}$$

and

$$-\nabla\varphi(\mathbf{r}=\mathbf{0}) = 0. \tag{6.21}$$

Eqs. (6.18,6.19,6.20,6.21) are the central results of this subsection.

To implement the formulas, one needs to find an optimal value of σ and determine the summation limits of $\mathbf{T}$ and $\mathbf{K}$. Notice that $\mathbf{T}$ terms decay according to $\exp\left[-\sigma^2\left(T^2+z^2\right)\right]$ and $\mathbf{K}$ terms decay according to $\exp\left[-\left(\frac{K^2}{4\sigma^2}+\sigma^2z^2\right)\right]$. If σ is too small, there will be too many $\mathbf{T}$ terms; If σ is too large, there will be too many $\mathbf{K}$ terms. The optimal choice

www.ebook3000.com

of σ is to make the numbers of $\mathbf{T}$ terms and $\mathbf{K}$ terms balanced. Assume that the summation limits of $\mathbf{T}$ and $\mathbf{K}$ are determined by $|\mathbf{T}| \leqslant T_{\max}$ and $|\mathbf{K}| \leqslant K_{\max}$ respectively. Let the minimum $\mathbf{T}$ term and $\mathbf{K}$ term decay to a given tolerance ε and require that the number of $\mathbf{T}$ terms equals to that of $\mathbf{K}$ terms

$$\exp\left[-\left(\sigma^2 T_{\max}^2 + \sigma^2 z^2\right)\right] = \exp\left[-\left(\frac{K_{\max}^2}{4\sigma^2} + \sigma^2 z^2\right)\right] = \varepsilon,$$

$$\frac{\pi T_{\max}^2}{A} = \frac{\pi K_{\max}^2}{(2\pi)^2 / A},$$

one obtains

$$\sigma = \frac{\sqrt{\pi}}{\sqrt{A}}, \tag{6.22}$$

$$T_{\max} = N\frac{1}{\sigma}, \tag{6.23}$$

$$K_{\max} = 2N\sigma, \tag{6.24}$$

where $N = \sqrt{-\ln \varepsilon}$ is an accuracy control parameter.

6.3.2 *Surface Madelung potential*

In the previous subsection, we have learned how to calculate the Madelung potential of a 2d lattice. In two-probe systems, one needs to add up the contribution of all the 2d lattices from $k = -\infty$ to $k = +\infty$ along the transport direction. In a practical calculation, however, one can only take into account a finite number of lattices. The question is whether the infinite summation $\sum_{k=-\infty}^{+\infty}$ can be replaced by a finite summation $\sum_{k=-K_1}^{+K_2}$ where K_1 and K_2 are some positive integer. It turns out that the truncation is not trivial at all because the Coulomb potential is a long range interaction.

To have an insight to the problem, let us first investigate a simple model. Consider the electrostatic potential of uniformly charged planes stacked along the z direction. Assume that the adjacent planes have opposite charge density (see Fig. 6.2a). The total charge density is

$$\rho(z) = \frac{1}{2\pi} \sum_{k=-\infty}^{+\infty} [2\mathrm{mod}(k, 2) - 1]\, \delta(z - k), \tag{6.25}$$

where $\mathrm{mod}(m, n)$ represents the modulus of m divided by n. By using the Gauss theorem, the electrostatic potential of $\rho(z)$ is obtained as

$$V(z) = -\sum_{k=-\infty}^{+\infty} [2\mathrm{mod}(k, 2) - 1]\, |z - k|. \tag{6.26}$$

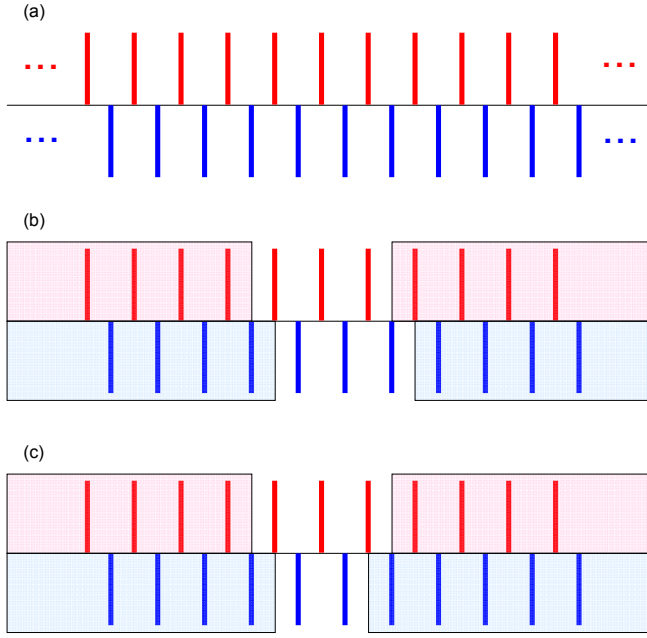


Fig. 6.2 Schematic diagram of the electrostatic problem defined by Eq. (6.25). (a) Uniformly charged planes are stacked along the z direction. The adjacent planes (red and blue) have equal distance but opposite charge density. (b) Six charged planes are considered explicitly and other charged planes are replaced by jellium. (c) Five charged planes are considered explicitly and other charged planes are replaced by jellium.

Suppose the summation is truncated from $\sum_{k=-\infty}^{+\infty}$ to $\sum_{k=-K_1}^{+K_2}$. It is shown in Fig. 6.3 that the resulting electrostatic potential strongly depends on the parity of K_1 and K_2 no matter how large they are. If K_1+K_2+1 is an even integer, the potential will be stair-like (the dashed line in Fig. 6.3b); If K_1+K_2+1 is an odd integer, the potential will be zigzag (the dashed line in Fig. 6.3c). The reason is that the series of Eq. (6.26) does not converge due to the long range nature of Coulomb interaction. It is necessary to take into account the contribution of layers all the way to the infinities.

In two-probe systems, the Madelung potential has three contributions: the potential of the left semi-infinite lead, the potential of the central region, and the potential of the right semi-infinite lead. The potential of the central region can be calculated by a finite summation of 2d Madelung potentials, while the potentials of the left and right leads involve the contribution of

www.ebook3000.com

layers all the way to the infinities. The potential of semi-infinite 3d point charge lattices is called the surface Madelung potential. The electrostatic problem of the surface Madelung potential is defined as follows: A 2d lattice with unit cell vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ are stacked along the $\mathbf{a}_3$ direction, and the displacement is 0, $\mathbf{a}_3$, $2\mathbf{a}_3$, $3\mathbf{a}_3$, etc. In each 2d lattice, a group of point charges $\{Q_i\}$ are located on the center $\{\mathbf{r}_i\}$ relative to the lattice site. The 2d lattice is assumed to be charge neutral, namely, $\sum_i Q_i = 0$. To calculate the surface Madelung potential, the key idea is to separate the potential into the near field potential and the far field potential [4].

The near field potential takes into account the contributions of those 2d lattices close to the field point, i.e., $z \sim \frac{1}{\sigma}$, where z is the distance from the field point to the lattice plane and σ is defined by Eq. (6.22). Because the field point is close to the 2d lattice, it can see the structural details of the lattice. So the near field potential must be calculated accurately layer by layer with the 2d Ewald sum technique discussed in Section 6.3.1.

The far field potential takes into account the contributions of those 2d lattices far away from the field point, i.e., $z \gg \frac{1}{\sigma}$. Because the field point is far away from the 2d lattice, it cannot see the details of the lattice. The 2d charged lattice can be replaced by a uniformly charged plane. One can verify that in the limit $|z| \gg \frac{1}{\sigma}$ Eq. (6.18) is reduced to

$$\varphi(\mathbf{r}) = -\frac{2\pi}{A}|z| + C,$$

which is indeed identical to the potential of a uniformly charged plane. As a further simplification, the uniformly charged plane is inflated into a uniformly charged layer with a finite thickness d, where d is the distance between the 2d lattices. Since the field point is far away from the 2d lattice plane, the inflation does not change the potential at the field point due to the Gauss theorem. As a result, the semi-infinite 3d lattice is replaced by a uniformly charged jellium in the far field potential. Due to the charge neutrality $\sum_i Q_i = 0$, the jelliums of different Q_i cancel with each other deep inside the surfaces. One only needs to consider the jellium potential from different jellium surfaces to $z = z_0$ where z_0 is a reference point (see Fig. 6.2b and 6.2c). The jellium potential is obtained as

$$V_{jellium}(x) = -2\pi\rho_0 \cdot u(x, D),$$
$$u(x, D) \equiv \begin{cases} x^2 + \frac{1}{4}D^2 & |x| < \frac{1}{2}D \\ |x|\, D & |x| > \frac{1}{2}D \end{cases},$$

where D is the thickness of the jellium, ρ_0 is the charge density of the jellium, and x is the distance from the field point to the jellium center.

Adding up the near field potential and the far field potential, the surface Madelung potential is obtained as

$$V_s(\mathbf{r}) = V_1(\mathbf{r}) + V_2(\mathbf{r}), \tag{6.27}$$

$$V_1(\mathbf{r}) = \sum_i Q_i \sum_{k=0}^{K-1} \varphi(\mathbf{r} - \mathbf{r}_i - k\mathbf{a}_3), \tag{6.28}$$

$$V_2(\mathbf{r}) = -\sum_i 2\pi\rho_i \cdot u(z - z_i^c, d_i). \tag{6.29}$$

$V_1(\mathbf{r})$ is the near field potential, in which $\varphi(\mathbf{r})$ is defined by Eqs. (6.18,6.19). $V_2(\mathbf{r})$ is the far field potential, in which ρ_i, d_i and z_i^c are the charge density, the thickness, and the center of the jellium for Q_i

$$\rho_i = \frac{Q_i}{Ad} sgn(z_i^s - z_0),$$
$$d_i = |z_i^s - z_0|,$$
$$z_i^c = \frac{z_i^s + z_0}{2},$$

where A is the area of the 2d unit cell and d is the spacing between the 2d lattices. z_i^s is the jellium surface point

$$z_i^s = \left[\mathbf{r}_i + \left(K - \frac{1}{2}\right)\mathbf{a}_3\right] \cdot \mathbf{e}_z,$$

where $\mathbf{e}_z$ is a unit vector perpendicular to the lattice plane and in the *opposite* direction of $\mathbf{a}_3$. z_0 is the reference point and can be assigned to zero, because the surface potential is *independent* of the choice of z_0. The gradient of $V_s(\mathbf{r})$ is obtained as

$$-\nabla V_s(\mathbf{r}) = -\nabla V_1(\mathbf{r}) - \nabla V_2(\mathbf{r}), \tag{6.30}$$

$$-\nabla V_1(\mathbf{r}) = \sum_i Q_i \sum_{k=0}^{K-1} \left[-\nabla\varphi(\mathbf{r} - \mathbf{r}_i - k\mathbf{a}_3)\right], \tag{6.31}$$

$$-\nabla V_2(\mathbf{r}) = \sum_i 2\pi\rho_i d_i^2 \cdot u'\left(\frac{z - z_i^c}{d_i}\right)\mathbf{e}_z. \tag{6.32}$$

Eqs. (6.27–6.32) are the central results of this subsection.

Finally, we come back to the electrostatic problem defined by Eq. (6.25). To fix up the discrepancy due to different truncations, we take into account the effect of semi-infinite leads by including the jellium correction. It is shown in Fig. 6.3 that the jellium-corrected potential in the central region is identical to the periodic solution and independent of the parity of K_1 and K_2.

www.ebook3000.com

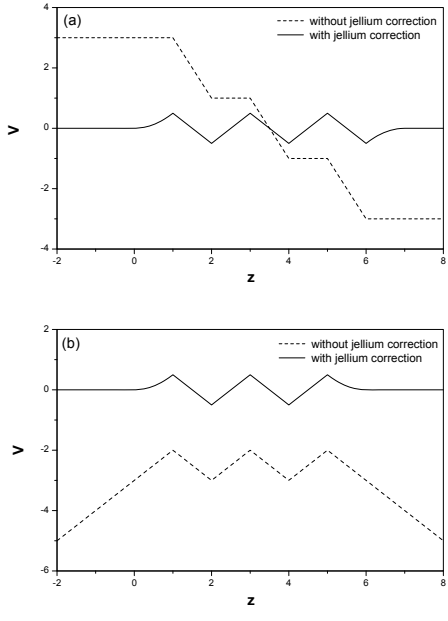


Fig. 6.3 The Madelung potential of the uniformly charged planes with alternative charge densities (see Fig. 6.2a). (a) The Madelung potential of six layers with and without jellium correction (see Fig. 6.2b). (b) The Madelung potential of five layers with and without jellium correction (see Fig. 6.2c).

6.3.3 *Boundary condition*

In this subsection, we shall first derive the Madelung potential of two-probe systems up to a constant, and then determine the constant by using the boundary condition.

In two-probe systems, the Madelung potential has three contributions: the contribution of the left lead, the contribution of the central region, and the contribution of the right lead. As a result, the Madelung potential of the central region V_C is obtained as

$$\begin{aligned} V_C = & M_{CL}Q_L + M_{CC}Q_C + M_{CR}Q_R + \\ & M^x_{CL}P^x_L + M^x_{CC}P^x_C + M^x_{CR}P^x_R + \\ & M^y_{CL}P^y_L + M^y_{CC}P^y_C + M^y_{CR}P^y_R + \\ & M^z_{CL}P^z_L + M^z_{CC}P^z_C + M^z_{CR}P^z_R + C, \end{aligned} \tag{6.33}$$

where Q is the column vector of charge, $P^{x,y,z}$ is the column vector of x, y, z

component of dipole, M is the Madelung constant matrix, and $M^{x,y,z}$ is the gradient component of Madelung constant matrix. The subscript L, C, R refers to the left lead, the central region, and the right lead, respectively. M_{CC} can be calculated with the 2d Madelung constant, namely, $(M_{CC})_{ij} = \varphi\left(\mathbf{r}_i^C - \mathbf{r}_j^C\right)$ where $\varphi(\mathbf{r})$ is defined by Eqs. (6.18,6.19). M_{CL} and M_{CR} can be calculated with the surface Madelung constant, namely, $(M_{CL})_{ij} = \varphi_s\left(\mathbf{r}_i^C, \mathbf{r}_j^L\right)$ and $(M_{CR})_{ij} = \varphi_s\left(\mathbf{r}_i^C, \mathbf{r}_j^R\right)$ where $\varphi_s(\mathbf{r}, \mathbf{r}_j) \equiv [V_s(\mathbf{r})]_{Q_i=\delta_{ij}}$ and $V_s(\mathbf{r})$ is defined by Eqs. (6.27,6.28,6.29). $M^{x,y,z}$ can be calculated with the gradient of the corresponding Madelung constant matrix M, namely, $M^x = -\partial_x M$, $M^y = -\partial_y M$, $M^z = -\partial_z M$. C is an arbitrary addible constant which will be focus of this subsection.

In bulk systems, C amounts to the energy zero and hence is arbitrary. In two-probe systems, C is determined by the boundary condition and hence is not arbitrary. The boundary condition is to match the potentials of the central region and the leads: On the one hand, the lead Madelung potential can be determined from the bulk Madelung potential by shifting the Fermi level according to Eq. (6.149) (see Section 6.8). Suppose the obtained lead Madelung potential is V_β where $\beta = L, R$ is the lead index. On the other hand, the lead Madelung potential can be calculated with the Madelung constant matrices analogously to Eq. (6.33)

$$\begin{aligned}\tilde{V}_\beta = {} & M_{\beta L}Q_L + M_{\beta C}Q_C + M_{\beta R}Q_R + \\ & M^x_{\beta L}P^x_L + M^x_{\beta C}P^x_C + M^x_{\beta R}P^x_R + \\ & M^y_{\beta L}P^y_L + M^y_{\beta C}P^y_C + M^y_{\beta R}P^y_R + \\ & M^z_{\beta L}P^z_L + M^z_{\beta C}P^z_C + M^z_{\beta R}P^z_R + C,\end{aligned} \tag{6.34}$$

where $\beta = L, R$ represents the lead unit cell adjacent to the central region. The constant C can be determined by matching the potential $\left(\tilde{V}_L, \tilde{V}_R\right)$ to (V_L, V_R). If a lead unit cell contains more than one atom, V_β and $\tilde{V}_\beta$ should be understood as the average over atomic sites in the unit cell.

The match of $\left(\tilde{V}_L, \tilde{V}_R\right)$ to (V_L, V_R) defines an over-determined problem. A simple solution is to match the potential average

$$\frac{\tilde{V}_L + \tilde{V}_R}{2} = \frac{V_L + V_R}{2}. \tag{6.35}$$

Once self-consistency is reached, V_L and V_R will match to $\tilde{V}_L$ and $\tilde{V}_R$ simultaneously. Another solution is to replace the constant C by an auxiliary linear potential $C(z)$ [5]

$$C(z) \equiv az + b, \tag{6.36}$$

www.ebook3000.com

where z is the atomic coordinate in the transport direction and the coefficients a and b are determined by

$$\begin{cases} \tilde{V}_L = V_L \\ \tilde{V}_R = V_R \end{cases} . \tag{6.37}$$

Once self-consistency is reached, $a = 0$ and $C(z)$ will be independent of z. Both matching schemes, Eq. (6.35) and Eqs. (6.36,6.37), have been implemented in NanoDsim.

To sum up, the Madelung potential in two-probe systems can be calculated by Eq. (6.33) in which the Madelung constant matrices are calculated by Eqs. (6.18–6.21) and Eqs. (6.27–6.32). The arbitrary addible constant is determined by Eq. (6.35) or Eqs. (6.36,6.37). In addition to the bias voltage between the left and right leads, one may also be interested in the gate voltage applied to the central region. The effect of the gate voltage can be taken into account by solving the Poisson equation in real space, which is presented in Appendix A.22.

6.4 Surface Green's function

This section discusses the algorithms for calculating the surface Green's function and the lead self-energy. In some special cases, the surface Green's function can be solved analytically which is discussed in Section 6.4.1. Generally the surface Green's function has to be solved numerically. There are two types of algorithms, recursive method and eigenvalue method, which are discussed in Section 6.4.2 and 6.4.3, respectively. Finally a few comments are made in Section 6.4.4.

6.4.1 *Analytically solvable case*

The surface Green's function is defined by

$$g_s^r \equiv [G^r]_{00} , \tag{6.38}$$

$$G^r = \left(E^+ - H\right)^{-1} , \tag{6.39}$$

$$H = \begin{pmatrix} H_0 & H_+ & & \\ H_- & H_0 & H_+ & \\ & H_- & H_0 & \ddots \\ & & \ddots & \ddots \end{pmatrix} , \tag{6.40}$$

where H is a block tridiagonal matrix and $[\cdots]_{00}$ means to take the first diagonal block (for convenience the matrix block index starts from 0). Due

to the Hermitian property of H, $(H_0)^\dagger = H_0$ and $(H_+)^\dagger = H_-$. In Section 2.7, the equation of surface Green's function has been obtained as

$$g_s^r = \left(E^+ - H_0 - H_+ g_s^r H_-\right)^{-1}. \tag{6.41}$$

Consider a special case in which $H_+ = H_- = tI$ where t is a real number and I is an identity matrix. In this situation, the surface Green's function can be solved analytically by diagonalizing H_0 [6]. Since H_0 is Hermitian, it can be decomposed as

$$\begin{aligned} H_0 &= U\Lambda U^\dagger, \\ U &= \left(|\Psi_1\rangle, |\Psi_2\rangle, \cdots, |\Psi_n\rangle\right), \\ \Lambda &= diag\left([\lambda_1, \lambda_2, \cdots, \lambda_n]\right), \end{aligned} \tag{6.42}$$

where $\{\lambda_i\}$ and $\{|\Psi_i\rangle\}$ are eigenvalues and eigenvectors of H_0. Notice that $\{|\Psi_i\rangle\}$ have been orthonormalized to $\langle\Psi_i|\Psi_j\rangle = \delta_{ij}$ and hence $UU^\dagger = 1$. By making a unitary transform $U^\dagger(\cdots)U$, Eq. (6.41) can be diagonalized as

$$\tilde{g}_s^r = \left(E^+ - \Lambda - t^2 \tilde{g}_s^r\right)^{-1}, \tag{6.43}$$

where $\tilde{g}_s^r \equiv U^\dagger g_s^r U$ is a diagonal matrix. Thus the matrix equation (6.41) is reduced to decoupled scalar equations. The solution to Eq. (6.43) is obtained as

$$\tilde{g}_s^r = \frac{1}{t} diag\left[\xi\left(\frac{E^+ - \lambda_i}{t}\right)\right],$$

where $\xi(z)$ is defined by Eq. (2.159) or Eq. (2.160). The analytical solution to Eq. (6.41) is derived by making an inverse unitary transform

$$g_s^r = U \cdot \frac{1}{t} diag\left[\xi\left(\frac{E^+ - \lambda_i}{t}\right)\right] \cdot U^\dagger. \tag{6.44}$$

As an illustration, we shall calculate the surface Green's function for three tight-binding models. Model 1 is a semi-infinite ribbon (Fig. 6.4a); model 2 is a semi-infinite cylinder (Fig. 6.4b); and model 3 is a semi-infinite 2d mesh (Fig. 6.4c). In the three tight-binding models, the on-site energy is $\varepsilon_0 = 0$ and the nearest neighbor coupling is $t = 1$. The Hamiltonian blocks of model 1 are

$$H_0 = \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & 0 & 1 \\ & & & 1 & 0 \end{pmatrix}_{N\times N}, \tag{6.45}$$

www.ebook3000.com

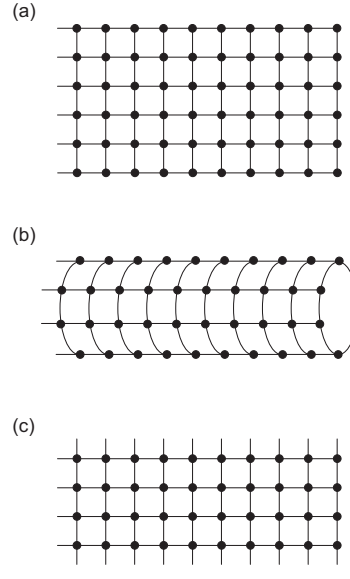


Fig. 6.4 Three tight-binding models whose surface Green's function can be solved analytically.

and $H_{\pm} = I$ is an $N \times N$ identity matrix. By using the eigen-decomposition (see Appendix A.12), the surface Green's function is obtained as

$$g_s^r = U \cdot \xi \left(E^+ - \Lambda\right) \cdot U^\dagger, \tag{6.46}$$

$$\Lambda_{jk} = \delta_{jk} 2 \cos \frac{j\pi}{N+1},$$

$$U_{jk} = \sqrt{\frac{2}{N+1}} \sin \frac{jk\pi}{N+1}.$$

The Hamiltonian blocks of model 2 are

$$H_0 = \begin{pmatrix} 0 & 1 & & & 1 \\ 1 & 0 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & 0 & 1 \\ 1 & & & 1 & 0 \end{pmatrix}_{N \times N}, \tag{6.47}$$

and $H_{\pm} = I$ is an $N \times N$ identity matrix. By using the eigen-decomposition (see Appendix A.12), the surface Green's function is obtained as

$$\begin{aligned} g_s^r &= U \cdot \xi \left(E^+ - \Lambda\right) \cdot U^\dagger, \\ \Lambda_{jk} &= \delta_{jk} 2 \cos \frac{2\pi}{N} j, \\ U_{jk} &= \frac{1}{\sqrt{N}} \exp\left(\mathrm{i} \frac{2\pi}{N} jk\right). \end{aligned} \tag{6.48}$$

The Hamiltonian of model 3 after Fourier transform is

$$H_0(k) = 2 \cos k, \tag{6.49}$$

and $H_{\pm}(k) = 1$. By taking inverse Fourier transform, the surface Green's function is obtained as

$$\begin{aligned} \left(g_s^r\right)_{i_1 i_2} &= \int_{-\pi}^{+\pi} \frac{dk}{2\pi} e^{\mathrm{i}k(i_1 - i_2)} g_s^r(k), \\ g_s^r(k) &\equiv \xi\left(E^+ - 2\cos k\right), \end{aligned} \tag{6.50}$$

where i_1 and i_2 are site indices. It can be proved that both Eq. (6.46) and Eq. (6.48) are reduced to Eq. (6.50) in the limit $N \to \infty$ [7].

6.4.2 *Recursive method*

In this subsection, we shall discuss the recursive method for calculating the surface Green's function. Notice that g_s^r appears in both the LHS and the RHS of Eq. (6.41). A simple idea is to make an initial guess of g_s^r and substitute g_s^r into the RHS of Eq. (6.41) to obtain a new g_s^r. Afterward substitute the new g_s^r into the RHS of Eq. (6.41) again to obtain another g_s^r. The iteration continues until g_s^r is fully converged. The simple iterative algorithm is summarized as follows

$$\begin{aligned} g_s^r &= \lim_{n \to \infty} Y_n, \\ Y_1 &= \left(E + \mathrm{i}\eta - H_0\right)^{-1}, \\ Y_n &= \left[\left(E + \mathrm{i}\eta - H_0 - H_+ Y_{n-1} H_-\right)\right]^{-1}, \end{aligned} \tag{6.51}$$

where η is a small positive number. Y_n is actually the surface Green's function of a finite system which contains n unit cells. As long as n is sufficiently large, Y_n should converge to g_s^r. But how large is sufficiently

www.ebook3000.com

large? As we know, the density of states of a finite system is composed of many δ-functions, each of which corresponds to an eigenvalue. The width of the δ-function is of the order of η. To make those δ-functions merge into a continuous spectrum, n needs to be comparable to $\frac{W}{\eta M}$ where W and M are the bandwidth and the orbital number of the lead unit cell. As an estimate, $W = 1$, $\eta = 10^{-6}$, $M = 100$. It takes about $n \sim \frac{W}{\eta M} = 10000$ recursions to converge the surface Green's function.

A more sophisticated recursive algorithm was invented in Ref. [8, 9], which reduces the number of recursions significantly. The recursive algorithm is described as follows

$$
\begin{aligned}
g_s^r &= \lim_{n\to\infty} \left[h_s^{(n)}\right]^{-1}, \qquad (6.52)\\
h_+^{(0)} &= -H_+,\\
h_-^{(0)} &= -H_-,\\
h_0^{(0)} &= E + \mathrm{i}\eta - H_0,\\
h_s^{(0)} &= E + \mathrm{i}\eta - H_0,\\
h_+^{(n)} &= -h_+^{(n-1)} \left[h_0^{(n-1)}\right]^{-1} h_+^{(n-1)},\\
h_-^{(n)} &= -h_-^{(n-1)} \left[h_0^{(n-1)}\right]^{-1} h_-^{(n-1)},\\
h_0^{(n)} &= h_0^{(n-1)} - h_+^{(n-1)} \left[h_0^{(n-1)}\right]^{-1} h_-^{(n-1)} - h_-^{(n-1)} \left[h_0^{(n-1)}\right]^{-1} h_+^{(n-1)},\\
h_s^{(n)} &= h_s^{(n-1)} - h_+^{(n-1)} \left[h_0^{(n-1)}\right]^{-1} h_-^{(n-1)}.
\end{aligned}
$$

After n recursions, 2^n unit cells are taken into account in the surface Green's function. As an estimate, for $W = 1$, $\eta = 10^{-6}$, $M = 100$, it takes about $n \tilde{} \log_2 \frac{W}{\eta M} = 13$ recursions to converge the surface Green's function. Compared to the simple recursive algorithm, the gain is overwhelming.

The idea behind the algorithm of Eq. (6.52) is analogous to the renormalization group: Firstly, the recursive relation of $\left\{G_{0,0}^r, G_{0,1}^r, G_{0,2}^r, G_{0,3}^r, \cdots\right\}$ is derived; Secondly, the recursive relation is generalized to $\left\{G_{0,0}^r, G_{0,2}^r, G_{0,4}^r, G_{0,6}^r \cdots\right\}$; Thirdly, the recursive relation is generalized to $\left\{G_{0,0}^r, G_{0,4}^r, G_{0,8}^r, G_{0,12}^r, \cdots\right\}$; so on and so forth. With the increase of "cluster size", the form of the recursive relation remains unchanged while the coefficients are renormalized, see Fig. 6.5.

The derivation of Eq. (6.52) is presented as follows: By using Eq. (6.39), one obtains $h \cdot G^r = 1$ where h is defined as $h \equiv E^+ - H$. By taking the

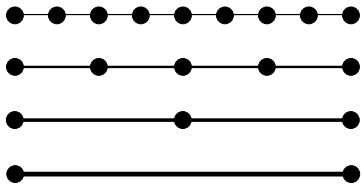


Fig. 6.5 Schematic diagram of the recursive method for the surface Green's function. The key idea is analogous to the renormalization group.

first column of the above matrix equation, one derives

$$\begin{pmatrix} h_s & h_+ & & \\ h_- & h_0 & h_+ & \\ & h_- & h_0 & \ddots \\ & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \tag{6.53}$$

where $h_s = h_0 \equiv E^+ - H_0$, $h_+ \equiv -H_+$, $h_- \equiv -H_-$, and $g_n \equiv G^r_{n,0}$. Eq. (6.53) leads to the recursive relation of $\{g_0, g_1, g_2, g_3, \cdots\}$

$$\begin{aligned} h_s g_0 &= 1 - h_+ g_1, \\ h_0 g_i &= -h_- g_{i-1} - h_+ g_{i+1} \quad (i \geqslant 1). \end{aligned} \tag{6.54}$$

By eliminating the odd terms $\{g_1, g_3, g_5, g_7, \cdots\}$ in Eq. (6.54), one derives the recursive relation of $\{g_0, g_2, g_4, g_6, \cdots\}$

$$\begin{aligned} h'_s g_0 &= 1 - h'_+ g_2, \\ h'_0 g_{2i} &= -h'_- g_{2(i-1)} - h'_+ g_{2(i+1)} \quad (i \geqslant 1), \end{aligned} \tag{6.55}$$

where the coefficients are

$$\begin{aligned} h'_+ &= -h_+ h_0^{-1} h_+, \\ h'_- &= -h_- h_0^{-1} h_-, \\ h'_0 &= h_0 - h_+ h_0^{-1} h_- - h_- h_0^{-1} h_+, \\ h'_s &= h_s - h_+ h_0^{-1} h_-. \end{aligned} \tag{6.56}$$

Comparing Eq. (6.54) and Eq. (6.55), one can see that the series $\{g_0, g_1, g_2, g_3, \cdots\}$ and $\{g_0, g_2, g_4, g_6, \cdots\}$ satisfy the same recursive relation except that the coefficients are renormalized by Eq. (6.56).

We continue to eliminate the odd terms in $\{g_0, g_2, g_4, g_6, \cdots\}$ to obtain a new recursive relation of $\{g_0, g_4, g_8, g_{12}, \cdots\}$. This time we don't have to

www.ebook3000.com

do any derivation but simply renormalize h'_s, h'_0, h'_+, h'_- to h''_s, h''_0, h''_+, h''_- by using Eq. (6.56). It is straightforward to generalize this procedure to obtain the recursive relation of the $\{g_0, g_{2^n}, g_{2\times 2^n}, g_{3\times 2^n}, \cdots\}$

$$\begin{aligned} h_s^{(n)} g_0 &= 1 - h_+^{(n)} g_{2^n}, \\ h_0^{(n)} g_{2^n i} &= -h_-^{(n)} g_{2^n(i-1)} - h_+^{(n)} g_{2^n(i+1)} \quad (i \geqslant 1), \end{aligned} \tag{6.57}$$

where the coefficients $h_s^{(n)}$, $h_0^{(n)}$, $h_+^{(n)}$, $h_-^{(n)}$ are defined by Eq. (6.52). For sufficiently large n, $g_{2^n} = G^r_{2^n,0} \approx 0$, and the surface Green's function g_s^r is obtained from Eq. (6.57) as $g_s^r = g_0 \approx \left[h_s^{(n)}\right]^{-1}$.

6.4.3 *Eigenvalue method*

In this subsection, we shall discuss the eigenvalue method for calculating the surface Green's function [10,11]. Instead of solving Eq. (6.41), the surface Green's function will be constructed by using the Bloch states. The idea is analogous to solving an ordinary differential equation: One first derives all possible solutions of the equation without considering the boundary condition. Afterward the solutions are linearly combined to satisfy the boundary condition.

The Bloch states are the eigensolutions of the 1d periodic system

$$\begin{pmatrix} \ddots & \ddots & & & \\ \ddots & H_0 & H_+ & & \\ & H_- & H_0 & H_+ & \\ & & H_- & H_0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ \phi_{-1} \\ \phi_0 \\ \phi_{+1} \\ \vdots \end{pmatrix} = E \begin{pmatrix} \vdots \\ \phi_{-1} \\ \phi_0 \\ \phi_{+1} \\ \vdots \end{pmatrix}, \tag{6.58}$$

or equivalently

$$\begin{pmatrix} \ddots & \ddots & & & \\ \ddots & h_0 & h_+ & & \\ & h_- & h_0 & h_+ & \\ & & h_- & h_0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ \phi_{-1} \\ \phi_0 \\ \phi_{+1} \\ \vdots \end{pmatrix} = 0, \tag{6.59}$$

where $h_0 \equiv E - H_0$ and $h_\pm \equiv -H_\pm$. Eq. (6.59) leads to the recursive relation

$$h_- \phi_{n-1} + h_0 \phi_n + h_+ \phi_n = 0. \tag{6.60}$$

Assume that the solution to Eq. (6.60) has the form $\phi_n = \lambda^n \phi_0$. Eq. (6.60) is reduced to

$$\left(h_- \lambda^{-1} + h_0 + h_+ \lambda\right) \phi_0 = 0, \tag{6.61}$$

which defines a quadratic eigenvalue problem. By introducing an auxiliary variable $\omega_0 \equiv \lambda \varphi_0$, Eq. (6.61) is reduced to a generalized eigenvalue problem

$$\begin{pmatrix} 0 & 1 \\ h_- & h_0 \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \omega_0 \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & -h_+ \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \omega_0 \end{pmatrix}. \tag{6.62}$$

Eq. (6.62) has $2N$ eigensolutions where N is the size of h_0. The $2N$ eigensolutions can be classified into four categories: (1) the left-decaying mode with $|\lambda| > 1$, (2) the right-decaying mode with $|\lambda| < 1$, (3) the left-traveling mode with $|\lambda| = 1$ and negative group velocity, and (4) the right-traveling mode with $|\lambda| = 1$ and positive group velocity. The left-decaying mode and the left-traveling mode together are called the left-moving mode, and the right-decaying mode and the right-traveling mode together are called the right-moving mode. By adding an infinitesimal imaginary part $\mathrm{i}0^+$ to E, $|\lambda|$ is slightly larger (smaller) than 1 for the left-traveling (right-traveling) mode. Therefore the left-moving and right-moving mode can be classified by $|\lambda| > 1$ and $|\lambda| < 1$ in a unified manner. Due to the time-reversal symmetry, there are N left-moving modes and N right-moving modes [12]. Sort the $2N$ eigensolutions of Eq. (6.62) in ascending order of $|\lambda|$. The first N eigensolutions are right-moving, and $\{\Lambda_+, \Psi_+\}$ are defined by

$$\begin{aligned} \Lambda_+ &= diag\left(\left[\lambda^{(1)}, \lambda^{(2)}, \cdots, \lambda^{(N)}\right]\right), \\ \Psi_+ &= \left(\varphi_0^{(1)}, \varphi_0^{(2)}, \cdots, \varphi_0^{(N)}\right). \end{aligned} \tag{6.63}$$

The last N eigensolutions are left-moving, and $\{\Lambda_-, \Psi_-\}$ are defined by

$$\begin{aligned} \Lambda_- &= diag\left(\left[\lambda^{(N+1)}, \lambda^{(N+2)}, \cdots, \lambda^{(2N)}\right]\right), \\ \Psi_- &= \left(\varphi_0^{(N+1)}, \varphi_0^{(N+2)}, \cdots, \varphi_0^{(2N)}\right). \end{aligned} \tag{6.64}$$

Notice that $\{\Lambda_\pm, \Psi_\pm\}$ satisfy the equation

$$h_- \Psi_\pm \Lambda_\pm^{-1} + h_0 \Psi_\pm + h_+ \Psi_\pm \Lambda_\pm = 0. \tag{6.65}$$

Now we are ready to construct the surface Green's function with $\{\Lambda_\pm, \Psi_\pm\}$. Comparing Eq. (6.53) and Eq. (6.59), one can see that the two equations are nearly identical except for the first row. It is assumed

www.ebook3000.com

that the solution to Eq. (6.53) is a linear combination of the solutions to Eq. (6.59)

$$g_n = [\Psi_+ (\Lambda_+)^n] C_+ + [\Psi_- (\Lambda_-)^n] C_-, \tag{6.66}$$

where $C_\pm$ are some unknown coefficient matrices. For $n \geqslant 1$, it is easy to verify the recursive relation

$$h_- g_{n-1} + h_0 g_n + h_+ g_{n+1} = 0,$$

where Eq. (6.65) is used in the derivation. So we only need to worry about the two "boundaries", namely, $n \to \infty$ and $n = 0$. For $n \to \infty$, $g_n \to 0$, and hence $C_- = 0$ because of $|\lambda_-| > 1$. For $n = 0$, by using the first row of Eq. (6.53), one obtains

$$h_0 \Psi_+ C_+ + h_+ \Psi_+ \Lambda_+ C_+ = 1. \tag{6.67}$$

With the aid of Eq. (6.65), Eq. (6.67) is further reduced to

$$-h_- \Psi_+ \Lambda_+^{-1} C_+ = 1. \tag{6.68}$$

Consequently g_0 is obtained as

$$g_0 = \Psi_+ C = -\Psi_+ \Lambda_+ \Psi_+^{-1} h_-^{-1}. \tag{6.69}$$

Eq. (6.69) is not numerically favorable because $h_- = -H_-$ may not be invertible. To bypass the problem [13, 14], we further derive the lead self-energy due to the surface Green's function

$$\Sigma = h_+ g_0 h_- = -h_+ \Psi_+ \Lambda_+ \Psi_+^{-1}, \tag{6.70}$$

which does not involve any numerical instability. By using the lead self-energy, the surface Green's function is obtained as

$$g_s^r = \left(h_0 + h_+ \Psi_+ \Lambda_+ \Psi_+^{-1}\right)^{-1}, \tag{6.71}$$

which is the central result of this subsection.

6.4.4 *A few comments*

A few comments on calculating the surface Green's function are in order.

First of all, the definition of the surface Green's function, Eqs. (6.38,6.39,6.40), are for the Hamiltonian represented in an orthogonal basis set. It is straightforward to generalize the formulas to nonorthogonal Hamiltonian or LMTO Hamiltonian. From the mathematical point of view, the surface Green's function is nothing but the first diagonal block of the inverse of a semi-infinite block tridiagonal matrix. The derived algorithms, Eq. (6.52) or Eq. (6.71), do not depend on the details of the semi-infinite

block tridiagonal matrix. One simply needs to adapt the definition of h with respect to the Hamiltonian type

orthogonal	**nonorthogonal**	**LMTO**
$h \equiv E^{+} - H$	$h \equiv E^{+}S - H$	$h \equiv P\left(E^{+}\right) - S\left(k\right)$

(6.72)

where S is the overlap matrix in nonorthogonal Hamiltonian, $P\left(E\right)$ is the potential function and $S\left(k\right)$ is the Fourier transformed structure constant in an LMTO Hamiltonian. Also note that the derived algorithms are for the surface Green's function of the left surface of the right lead. For the right surface of the left lead, one simply needs to exchange h_{+} and h_{-} in the algorithms.

Secondly, we would like to make a comparison of the recursive method and the eigenvalue method. The cost of the recursive method is mainly determined by the number of recursions. According to Eq. (6.52), each recursion involves 9 multiplications or inversions of an $N \times N$ matrix. So the total cost of the recursive method is about $9Mt_N$ where M is the number of recursions and t_N is the cost of multiplication or inversion of an $N \times N$ matrix. The cost of the eigenvalue method is mainly determined by solving the generalized eigenvalue problem defined by Eq. (6.62). So the total cost of the eigenvalue method is about $\tilde{t}_{2N}$ which is the cost of a generalized $2N \times 2N$ eigenvalue problem. Numerical tests indicate that $\tilde{t}_{2N}/t_N \approx 5 \times 10^2$ for $N = 100$ and $\tilde{t}_{2N}/t_N \approx 1 \times 10^3$ for $N = 1000$. It is inferred that the crossover between the two methods occurs around $M = 50$ to 100. Generally speaking, on the complex contour, the recursive method is superior because the recursion step is rather small ($M \sim 10$) due to the finite imaginary part of E. On the real axis, the eigenvalue method is superior because the imaginary part of E is infinitesimal and M can be very large at some special energies. The numerical instability of the two methods are also complementary. The recursive method is less accurate if h_0^{-1} is close to singular, while the eigenvalue method is less accurate if $h_{\pm}^{-1}$ is close to singular. Therefore *NanoDsim* adopts a hybrid algorithm in the implementation of surface Green's function calculation *calcSurfaceGr.m*. The recursive method is used first with the constraint $M \leqslant 100$. If the solution converges within the maximum number of recursions, the calculation will stop there. Otherwise the code automatically switches to the eigenvalue method and calculates the surface Green's function again. It is very rare that both methods fail in a surface Green's function calculation.

Finally, it is worth mentioning that Eq. (6.52) or Eq. (6.71) are basic algorithms for calculating surface Green's function. There are a few

www.ebook3000.com

variations in the literature. For example, one can eliminate numerical instability by using the single value decomposition [14, 15]; one can reduce the computational cost considerably if a lead unit cell is composed of several principal layers [16,17]; one can construct the lead self-energy by using propagating and slowly decaying evanescent modes [18], etc. Interested readers are referred to the literature for more details.

6.5 Real axis integral

This section discusses the algorithms for evaluating the energy integral of Eq. (3.133) in nonequilibrium two-probe systems. In equilibrium two-probe systems, the integral path can be changed from the real axis to the complex contour by using the fluctuation-dissipation theorem and the analytical properties of retarded Green's function (see Section 5.6). In nonequilibrium two-probe systems, the fluctuation-dissipation theorem does not hold, and the integrand has singularities in both upper and lower complex plane. It is not straightforward to change the integral path from the real axis to the complex contour. Below we shall show that the energy integral can be divided into an equilibrium part and a nonequilibrium part [19]. The equilibrium part can be evaluated on the complex contour, while the nonequilibrium part has to be done on the real axis.

For simplicity of notation and without losing generality, the zeroth energy moment ($m = 0$) is adopted in the discussion and the subscript iq and the disorder average $\overline{(\cdots)}$ are omitted. The task is to calculate the energy integral

$$\rho \equiv (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} G^{<}(E) . \tag{6.73}$$

To have some intuition, let us first consider the case of zero temperature. Assume that the chemical potentials of the left and right leads are μ_L and μ_R, respectively. An important observation is that all the states below $\mu_{\min} \equiv \min(\mu_L, \mu_R)$ are fully occupied and all the states above $\mu_{\max} \equiv \max(\mu_L, \mu_R)$ are fully unoccupied. Only the states between $\mu_{\min}$ and $\mu_{\max}$ are partially occupied and the occupation is determined by nonequilibrium statistics. Therefore the integral of Eq. (6.73) can be divided into two parts: the equilibrium part of the energy range $(-\infty, \mu_{\min})$ and the nonequilibrium part of the energy range $(\mu_{\min}, \mu_{\max})$. The equilibrium part can be evaluated with the complex contour integral discussed in Section 5.6, while the nonequilibrium part has to be done on the real axis.

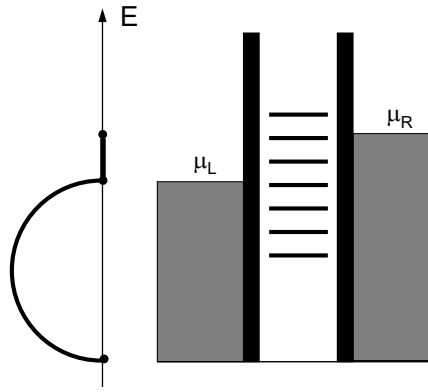


Fig. 6.6 Schematic diagram of the integral path on the complex contour and the real axis. The complex contour is a semicircle from the band bottom to $\mu_{\min}$. The real axis segment is a line from $\mu_{\min}$ to $\mu_{\max}$.

Fortunately the equilibrium part covers the majority of the energy integral and the energy range is of the order of 10 eV. The nonequilibrium part is in a narrow voltage window and the energy range is of the order of 1 eV. So a large portion of Eq. (6.73) can be evaluated on the complex contour and only a small portion has to be evaluated on the real axis, see Fig. 6.6.

Next let us turn to the general case of finite temperature. Analogous to the case of zero temperature, the energy integral can be divided into an equilibrium part and a nonequilibrium part

$$\begin{aligned}
\rho &\equiv (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} G^{<}(E) \\
&= (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} G^{r}(E) \Sigma^{<}(E) G^{a}(E) \\
&= (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} \left[G^{r}(E) f_L(E) \Sigma_L^{ar}(E) G^{a}(E) \right. \\
&\quad \left. + G^{r}(E) f_R(E) \Sigma_R^{ar}(E) G^{a}(E) \right] \\
&= (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} \left[f_0(E) G^{r}(E) \Sigma^{ar}(E) G^{a}(E) \right] \\
&\quad + (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} \left[\Delta f_L(E) G^{r}(E) \Sigma_L^{ar}(E) G^{a}(E) \right. \\
&\qquad \left. + \Delta f_R(E) G^{r}(E) \Sigma_R^{ar}(E) G^{a}(E) \right] \\
&\equiv \rho_{eq} + \rho_{neq}.
\end{aligned} \tag{6.74}$$

www.ebook3000.com

Here $\Sigma_\beta^{ar}(E) \equiv \Sigma_\beta^a(E) - \Sigma_\beta^r(E)$, $\Sigma^{ar}(E) \equiv \Sigma^a(E) - \Sigma^r(E)$, and $\Delta f_\beta(E) \equiv f_\beta(E) - f_0(E)$. The chemical potential of $f_0(E)$ is selected as $\mu_0 = \frac{1}{2}(\mu_L + \mu_R)$. The first term ρ_{eq} can be rewritten as

$$\rho_{eq} = (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} f_0(E) \left[G^a(E) - G^r(E)\right], \tag{6.75}$$

which can be evaluated on the complex contour (see also Eq. (5.68)). The second term ρ_{neq} has to be evaluated on the real axis

$$\begin{aligned}\rho_{neq} = (-\mathrm{i}) \int_{\mu_{\min}-Nk_BT}^{\mu_{\max}+Nk_BT} \frac{dE}{2\pi} \left[\Delta f_L(E) G^r(E) \Sigma_L^{ar}(E) G^a(E)\right. \\ \left. + \Delta f_R(E) G^r(E) \Sigma_R^{ar}(E) G^a(E)\right],\end{aligned} \tag{6.76}$$

where the integral limits have been reduced from $(-\infty, +\infty)$ to a narrow energy window. Here N is the Fermi function cutoff defined by the condition that $f(E) \approx \theta(\mu - E)$ if $|E - \mu| > Nk_BT$.

Finally we would like to investigate the calculation of the nonequilibrium part ρ_{neq}. Although the integral limits are much narrower than the equilibrium part, the integrand may contain sharp features such as van Hove singularities, sharp resonances, and even bound states [20]. In order to capture these sharp features, one can use the adaptive integral method which works efficiently in the post-analysis calculations [21]. For the self-consistent calculations, however, it is impractical to use the adaptive integral method especially in the context of NECPA. One has to use a very dense uniform energy mesh on the real axis, resulting in a heavy computational cost. Is it possible to add a finite imaginary part $\mathrm{i}\eta$ to the real energy E to broaden the sharp features? Our answer is as follows [22].

By using the analytical continuation, $G^r(E)$ and $G^a(E)$ can be extended to $G_\eta^r(E)$ and $G_\eta^a(E)$

$$G_\eta^r(E) \equiv G^r(E + \mathrm{i}\eta), \tag{6.77}$$

$$G_\eta^a(E) \equiv G^a(E - \mathrm{i}\eta). \tag{6.78}$$

In contrast, $G^<(E)$ has singularities on both upper and lower half plane, and we cannot make a similar analytical continuation. Therefore we change the point of view and re-interpret $\mathrm{i}\eta$ as the self-energy of a dephasing probe (see Section 2.9). It is assumed that each atomic orbital couples to a dephasing probe with coupling strength η. The chemical potentials of the dephasing probes are denoted by μ_L, μ_C, and μ_R in the left lead, the central region, and the right lead, respectively (see Fig. 6.7). Notice that μ_L and μ_R are known from the lead chemical potentials while μ_C is to be determined by the current conservation. Namely one needs to find a proper

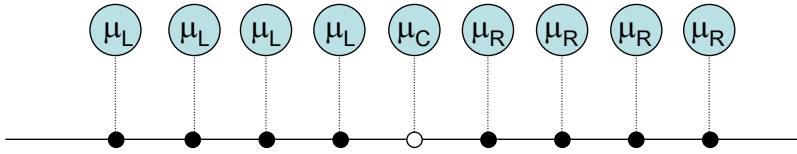


Fig. 6.7 Schematic diagram of connecting dephasing probes to atomic sites of the leads and the central region.

μ_C so that the net current flowing out of the dephasing probes connected to the central region is zero. In practice, we make a further approximation that $\mu_C \approx \mu_0 = \frac{1}{2}(\mu_L + \mu_R)$ and hence $f_C(E)$ is identical to the $f_0(E)$ in Eq. (6.75). As a result, $\Delta f_C(E) = f_C(E) - f_0(E) = 0$, and hence Eq. (6.76) is reduced to

$$\rho_{neq} \approx (-\mathrm{i}) \int_{\mu_{\min}-Nk_BT}^{\mu_{\max}+Nk_BT} \frac{dE}{2\pi} \left[\Delta f_L(E)\, G_\eta^r(E)\, \Sigma_{\eta L}^{ar}(E)\, G_\eta^a(E) + \Delta f_R(E)\, G_\eta^r(E)\, \Sigma_{\eta R}^{ar}(E)\, G_\eta^a(E)\right], \tag{6.79}$$

where $\Sigma_{\eta\beta}^{ar}(E)$ is defined by

$$\Sigma_{\eta\beta}^{ar}(E) \equiv \Sigma_\beta^a(E - \mathrm{i}\eta) - \Sigma_\beta^r(E + \mathrm{i}\eta). \tag{6.80}$$

Last but not least, don't forget to include the dephasing probes in the equilibrium part, and hence Eq. (6.75) is reduced to

$$\rho_{eq} \approx (-\mathrm{i}) \int_{-\infty}^{+\infty} \frac{dE}{2\pi} f_0(E) \left[G_\eta^a(E) - G_\eta^r(E)\right]. \tag{6.81}$$

To sum up, the energy integral of nonequilibrium two-probe systems can be divided into an equilibrium part Eq. (6.81) and a nonequilibrium part Eq. (6.79). The equilibrium part can be evaluated by using the complex contour integral as presented in Section 5.6. The nonequilibrium part can be evaluated on the real axis by broadening the spectrum with dephasing probes.

6.6 k-integral in the Brillouin zone

This section discusses the algorithms for evaluating the k-integral of the Brillouin zone (BZ) in Eq. (3.121) and Eq. (3.122). To evaluate the k-integral in two-probe systems, one needs to select some k-points in the 2d BZ and sum up the contribution from each k-point with a proper weight. This is called k-sampling. Section 6.6.1 is devoted to uniform k-sampling;

www.ebook3000.com

Section 6.6.2 is devoted to k-sampling with geometric symmetry; and Section 6.6.3 is devoted to k-sampling with time-reversal symmetry. The discussion here also applies to bulk systems except that the k-sampling is in the 3d BZ.

6.6.1 *Uniform k-sampling*

Uniform k-sampling is to sample the BZ with a uniform k-grid. It is applicable to any periodic system. Here the key issue is to understand the physical meaning of the uniform k-sampling. Since all the dimensions are independent, it is sufficient to study uniform k-sampling in the 1d case.

Consider a periodic 1d system described by the Hamiltonian $H_{I_1 I_2} = H_{I_1 - I_2} \equiv H_I$ where I_1 and I_2 are the unit cell indices. Due to the periodicity, $H_{I_1 I_2}$ only depends on the difference $I_1 - I_2$, and one can make a Fourier transform

$$H(k) = \sum_I e^{-\mathrm{i}kI} H_I. \tag{6.82}$$

Consequently the Fourier transformed Green's function is obtained as

$$G^r(E,k) = \left[E^+ - H(k)\right]^{-1}, \tag{6.83}$$

and the Green's function of a single unit cell is derived by the k-integral

$$G^r(E) = \int_0^{2\pi} \frac{dk}{2\pi} G^r(E,k). \tag{6.84}$$

To evaluate the k-integral in Eq. (6.84), one can discretize the 1d BZ

$$G^r(E) \approx G^r_N(E) = \frac{1}{N} \sum_{n=0}^{N-1} G^r(E, k_n),$$

where the k-points are selected as

$$k_n = \frac{2\pi}{N} n. \tag{6.85}$$

The discretization of the 1d BZ is not unique. One may select other k-points as the k-sampling, e.g.,

$$\tilde{k}_n = \frac{2\pi}{N}\left(n + \frac{1}{2}\right). \tag{6.86}$$

As long as $N \to \infty$, the k-samplings of Eq. (6.85) and Eq. (6.86) are equivalent. Nevertheless the k-sampling of Eq. (6.85) is superior to Eq. (6.86) because the former has a clear physical meaning even if N is finite. The

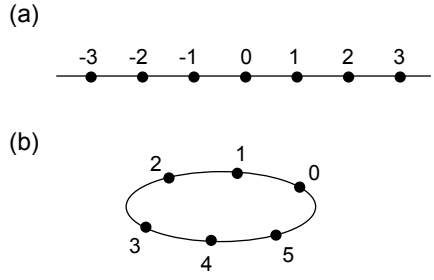


Fig. 6.8 Physical meaning of the uniform k-sampling. (a) Periodic 1d system amounts to using an infinite number of k-points ($N = \infty$) in the Brillouin zone. (b) Cyclic supercell amounts to using a finite number of k-points ($N = 6$) in the Brillouin zone.

physical meaning of Eq. (6.85) is illustrated in Fig. 6.8. By using N discretized k-points, the calculated Green's function $G_N^r(E)$ is equivalent to the Green's function of a finite cyclic system which is composed of N unit cells. With increasing N, the infinitely large periodic system is approached systematically by a series of finite cyclic systems.

In the implementation, it is more convenient to extend the summation limits from $0 \leqslant n \leqslant N-1$ to $0 \leqslant n \leqslant N$ by adding proper weights

$$G_N^r(E) = \frac{1}{N} \sum_{n=0}^{N} G^r(E, k_n)\, w_n, \tag{6.87}$$

where the weights are defined by $w_0 = w_N = \frac{1}{2}$ and $w_{1 \leqslant n \leqslant N-1} = 1$. Notice that $\frac{1}{N} \sum_{n=0}^{N} w_n = 1$ due to the normalization. So far we have discussed the 1d uniform k-sampling. It is straightforward to generalize the 1d k-sampling to 2d or 3d k-sampling. The 2d uniform k-sampling is just a direct product of two 1d k-samplings. The k-points and corresponding weights are shown in Fig. 6.9 in the 2d BZ.

6.6.2 *Symmetric k-sampling*

Very often solid crystals have geometric symmetry which can be used to reduce the k-sampling in the BZ. Notice that the reciprocal space has the same symmetry as the real space. One may work on a small portion of the BZ, and extend the result to the whole BZ by using symmetry operations. In two-probe systems, the symmetry operations are 2d symmetry groups in the transverse dimensions. There are ten point groups in 2d, C_n and D_n, where $n = 1, 2, 3, 4, 6$ is the degree of the rotation axis. C_n

www.ebook3000.com

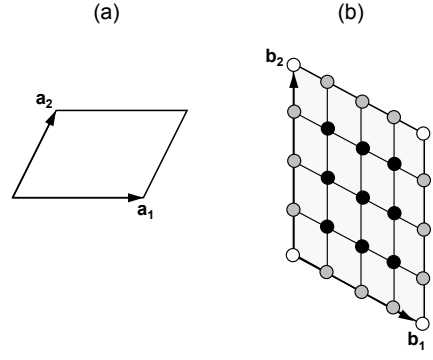


Fig. 6.9 Uniform k-sampling in the 2d Brillouin zone. (a) Unit cell in real space. (b) k-sampling in the 2d Brillouin zone. The black dot, gray dot, and white dot have the weight $1, \frac{1}{2}$, and $\frac{1}{4}$, respectively.

group only contains rotations, e.g., $C_3 = \left\{1, c_3, c_3^2\right\}$, where c_3 is the $\frac{2\pi}{3}$ rotation. D_n group contains both rotations and mirror reflections, e.g., $D_3 = \left\{1, c_3, c_3^2, m, mc_3, mc_3^2\right\}$, where m is the mirror reflection. With the aid of the symmetry group, the BZ can be reduced to $\frac{1}{n_G}$ where n_G is the number of group members. The smallest inequivalent BZ is called irreducible Brillouin zone (IBZ). If a physical quantity is a scalar (e.g., transmission coefficient), one may simply calculate the k-integral in the IBZ and multiply the result by n_G. If the quantity is an operator (e.g., Green's function), one needs to carry out a unitary transform to extend the integral of the IBZ to that of the BZ. The derivation of the unitary transform will be the focus of this subsection.

First of all, we would like to introduce some notations and definitions. A crystal can be constructed from unit cells. Let I denote the unit cell index and i the atom index inside a unit cell. The atom center of (I, i) is denoted by $R_I + s_i$ where R_I is the unit cell displacement and s_i is the atom center relative to unit cell. Each atom has its own atomic orbitals $\{|I, i, \mu\rangle\}$ where μ is the orbital index, e.g. $\mu = s, p_x, p_y, p_z$. The notations $\langle I, i|$ and $|I, i\rangle$ represent the column vector and row vector of atomic orbitals, e.g.,

$$|I, i\rangle = \left(|I, i, s\rangle, |I, i, p_x\rangle, |I, i, p_y\rangle, |I, i, p_z\rangle\right).$$

$$\langle I, i| = \begin{pmatrix} \langle I, i, s| \\ \langle I, i, p_x| \\ \langle I, i, p_y| \\ \langle I, i, p_z| \end{pmatrix}.$$

Suppose $\{u\}$ is a symmetry point group of the crystal. Applying the symmetry operation u to the crystal, the atom (I,i) is mapped to another atom (I',i'). In real space, the atom center is mapped to

$$\begin{aligned} u\cdot(R_I+s_i) &= u\cdot R_I + u\cdot s_i \\ &= u\cdot R_I + (\delta R_i + s_{i'}) \\ &= (u\cdot R_I + \delta R_i) + s_{i'} \\ &\equiv R_{I'} + s_{i'}. \end{aligned}$$

The mapping from R_I to $R_{I'}$

$$R_{I'} = u\cdot R_I + \delta R_i \tag{6.88}$$

is called displacement mapping. In the orbital space, the atomic orbitals $\{|I,i,\mu\rangle\}$ are mapped to

$$u\cdot|I,i,\mu\rangle = \sum_{\mu'}|I',i',\mu'\rangle\, U_{\mu'\mu},$$

where U is a unitary transform. The mapping from $\{|I,i,\mu\rangle\}$ to $\{|I',i',\mu'\rangle\}$ is called orbital mapping, which can be written in a compact form

$$u\cdot|I,i\rangle = |I',i'\rangle\, U. \tag{6.89}$$

Suppose $\hat{H}$ is the Hamiltonian operator and $\hat{G}\equiv\left(E-\hat{H}\right)^{-1}$ is the Green's function operator. Define the Green's function matrix block by

$$G_{I_1i_1,I_2i_2}\equiv\left\langle I_1i_1|\hat{G}|I_2i_2\right\rangle, \tag{6.90}$$

which implies a matrix block, e.g.,

$$\begin{pmatrix} \left\langle I_1i_1,s\middle|\hat{G}\middle|I_2i_2,s\right\rangle & \left\langle I_1i_1,s\middle|\hat{G}\middle|I_2i_2,p_x\right\rangle & \left\langle I_1i_1,s\middle|\hat{G}\middle|I_2i_2,p_y\right\rangle & \left\langle I_1i_1,s\middle|\hat{G}\middle|I_2i_2,p_z\right\rangle \\ \left\langle I_1i_1,p_x\middle|\hat{G}\middle|I_2i_2,s\right\rangle & \left\langle I_1i_1,p_x\middle|\hat{G}\middle|I_2i_2,p_x\right\rangle & \left\langle I_1i_1,p_x\middle|\hat{G}\middle|I_2i_2,p_y\right\rangle & \left\langle I_1i_1,p_x\middle|\hat{G}\middle|I_2i_2,p_z\right\rangle \\ \left\langle I_1i_1,p_y\middle|\hat{G}\middle|I_2i_2,s\right\rangle & \left\langle I_1i_1,p_y\middle|\hat{G}\middle|I_2i_2,p_x\right\rangle & \left\langle I_1i_1,p_y\middle|\hat{G}\middle|I_2i_2,p_y\right\rangle & \left\langle I_1i_1,p_y\middle|\hat{G}\middle|I_2i_2,p_z\right\rangle \\ \left\langle I_1i_1,p_z\middle|\hat{G}\middle|I_2i_2,s\right\rangle & \left\langle I_1i_1,p_z\middle|\hat{G}\middle|I_2i_2,p_x\right\rangle & \left\langle I_1i_1,p_z\middle|\hat{G}\middle|I_2i_2,p_y\right\rangle & \left\langle I_1i_1,p_z\middle|\hat{G}\middle|I_2i_2,p_z\right\rangle \end{pmatrix}.$$

Due to the periodicity, $G_{I_1i_1,I_2i_2}$ only depends on the displacement difference $R_{I_1}-R_{I_2}$, and hence one can make a Fourier transform to obtain $G_{i_1i_2}(k)$

$$G_{i_1i_2}(k)\equiv\sum_{\Delta R}e^{-\mathrm{i}k\cdot\Delta R}G_{I_1i_1,I_2i_2}, \tag{6.91}$$

where $\Delta R\equiv R_{I_1}-R_{I_2}$.

www.ebook3000.com

Next, we would like to establish the mapping between the Green's functions of different k-points that are related by symmetry operations.

Theorem: Suppose a crystal is invariant under the symmetry operation u. Applying u to the crystal, the atom (I,i) is mapped to (I',i'). In real space, $R_I + s_i$ is mapped to $R_{I'} + s_{i'}$ where $R_{I'} = u \cdot R_I + \delta R_i$. In orbital space, $|I,i\rangle$ is mapped to $|I',i'\rangle U$. In the calculation of Green's function, the matrix block $G_{i_1 i_2}\left(u^{-1}k\right)$ is mapped to $G_{i'_1 i'_2}(k)$ by

$$G_{i_1 i_2}\left(u^{-1}k\right) = e^{\mathrm{i}k\cdot\left(\delta R_{i_1} - \delta R_{i_2}\right)} \cdot U_1^{\dagger} G_{i'_1 i'_2}(k)\, U_2. \tag{6.92}$$

Proof: (1) Since the system is invariant under the symmetry operation u, the Hamiltonian operator $\hat{H}$ satisfies

$$\hat{H} = u^{-1} \cdot \hat{H} \cdot u.$$

The Green's function operator $\hat{G}$ is related to $\hat{H}$ by $\hat{G} = \left(E - \hat{H}\right)^{-1}$. As a result, $\hat{G}$ also satisfies

$$\hat{G} = u^{-1} \cdot \hat{G} \cdot u. \tag{6.93}$$

(2) By using the displacement mapping (see Eq. (6.88))

$$\begin{aligned} R_{I'_1} &= u \cdot R_{I_1} + \delta R_{i_1}, \\ R_{I'_2} &= u \cdot R_{I_2} + \delta R_{i_2}, \end{aligned}$$

one obtains

$$u \cdot \Delta R = \Delta R' - \left(\delta R_{i_1} - \delta R_{i_2}\right), \tag{6.94}$$

where $\Delta R \equiv R_{I_1} - R_{I_2}$ and $\Delta R' \equiv R_{I'_1} - R_{I'_2}$,

(3) By definition (see Eq. (6.91)), $G_{i_1 i_2}\left(u^{-1}k\right)$ is the Fourier transform of $G_{I_1 i_1, I_2 i_2}$

$$\begin{aligned} G_{i_1 i_2}\left(u^{-1}k\right) &= \sum_{\Delta R} e^{-\mathrm{i}\left(u^{-1}k\right)\cdot\Delta R} G_{I_1 i_1, I_2 i_2} \\ &= \sum_{\Delta R} e^{-\mathrm{i}\left(u^{-1}k\right)\cdot\Delta R} \left\langle I_1 i_1 | \hat{G} | I_2 i_2 \right\rangle \\ &= \sum_{\Delta R} e^{-\mathrm{i}k\cdot(u\Delta R)} \left\langle I_1 i_1 | \hat{G} | I_2 i_2 \right\rangle, \end{aligned} \tag{6.95}$$

where $\left(u^{-1}k\right)\cdot\Delta R = k\cdot(u\Delta R)$ is used in the derivation. Inserting

Eq. (6.93) and Eq. (6.94) into Eq. (6.95), one obtains

$$\begin{aligned} G_{i_1 i_2}\left(u^{-1}k\right) &= \sum_{\Delta R} e^{-\mathrm{i}k\cdot\Delta R'} e^{\mathrm{i}k\cdot\left(\delta R_{i_1}-\delta R_{i_2}\right)} \left\langle I_1 i_1 | u^{-1}\cdot\hat{G}\cdot u | I_2 i_2 \right\rangle \\ &= \sum_{\Delta R'} e^{-\mathrm{i}k\cdot\Delta R'} e^{\mathrm{i}k\cdot\left(\delta R_{i_1}-\delta R_{i_2}\right)} \cdot U_1^{\dagger} \left\langle I_1' i_1' | \hat{G} | I_2' i_2' \right\rangle U_2 \\ &= e^{\mathrm{i}k\cdot\left(\delta R_{i_1}-\delta R_{i_2}\right)} \cdot U_1^{\dagger} \left[\sum_{\Delta R'} e^{-\mathrm{i}k\cdot\Delta R'} \left\langle I_1' i_1' | \hat{G} | I_2' i_2' \right\rangle\right] U_2 \\ &= e^{\mathrm{i}k\cdot\left(\delta R_{i_1}-\delta R_{i_2}\right)} \cdot U_1^{\dagger} G_{i_1' i_2'}(k)\, U_2. \end{aligned}$$

QED.

An important consequence of the theorem is the mapping between the diagonal blocks of Green's function. For $i_1 = i_2$, Eq. (6.95) is reduced to [23]

$$G_{ii}\left(u^{-1}k\right) = U^{\dagger} G_{i'i'}(k)\, U. \tag{6.96}$$

Consequently the k-integral in the BZ is reduced to

$$\begin{aligned} G_{ii} &= \int_{BZ} \frac{d^2k}{(2\pi)^2} G_{ii}(k) \\ &= \sum_u \int_{IBZ} \frac{d^2k}{(2\pi)^2} G_{ii}\left(u^{-1}k\right) \\ &= \sum_u \int_{IBZ} \frac{d^2k}{(2\pi)^2} U^{\dagger} G_{i'i'}(k)\, U \\ &= \sum_u U^{\dagger} \left[\int_{IBZ} \frac{d^2k}{(2\pi)^2} G_{i'i'}(k)\right] U. \end{aligned} \tag{6.97}$$

Thus the computational cost is reduced to $\frac{1}{n_G}$ by working in the IBZ. Eq. (6.97) is the central result of this subsection.

Finally we would like to discuss some implementation details. The k-integral in the IBZ can be discretized by using a uniform k-sampling similar to Section 6.6.1

$$\int_{IBZ} \frac{d^2k}{(2\pi)^2} G_{i'i'}(k) \approx \frac{1}{N} \sum_n G_{i'i'}(k_n)\, w_n, \tag{6.98}$$

where N is the total number of k-points in the whole BZ, and k_n and w_n are the k-point and weight in the IBZ respectively. The weights are normalized to $\sum_n w_n = \frac{N}{n_G}$. For example, Fig. 6.10 shows the symmetric k-sampling

www.ebook3000.com

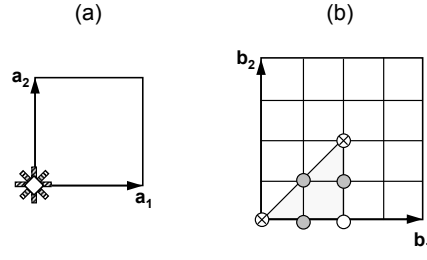


Fig. 6.10 Symmetric k-sampling of D_4 point group. (a) Unit cell in real space and symmetry operations. The c_4 rotation axis and the reflection mirrors are indicated by a diamond and shaded bars. (b) Symmetric k-sampling in the 2d Brillouin zone. The gray dot, white dot, and crossed dot have the weights $\frac{1}{2}$,$\frac{1}{4}$, and $\frac{1}{8}$, respectively.

of D_4 group. Here total number of k-points is $N = 4 \times 4$ and the symmetry group size is $n_G = 8$. The k-points and weights are

k_n	$\mathbf{0}$	$\frac{1}{4}\mathbf{b}_1$	$\frac{1}{4}\mathbf{b}_1 + \frac{1}{4}\mathbf{b}_2$	$\frac{1}{2}\mathbf{b}_1$	$\frac{1}{2}\mathbf{b}_1 + \frac{1}{4}\mathbf{b}_2$	$\frac{1}{2}\mathbf{b}_1 + \frac{1}{2}\mathbf{b}_2$
w_n	$\frac{1}{8}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{8}$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the reciprocal unit cell vectors. For other 2d point groups, the symmetric k-samplings are summarized in Appendix A.18.

Last but not least, don't forget to symmetrize the physical quantities in a self-consistent calculation if symmetric k-sampling is used. The reason is that symmetric k-sampling assumes that the system is strictly invariant under the symmetry operations. However, due to finite accuracy of atomic coordinates and the cutoff approximation in the screened structure constant, the geometric symmetry is approximate. The error may accumulate with the self-consistent iterations and lead to numerical instability. To avoid the problem, one needs to symmetrize the on-site physical quantities

$$\tilde{X}_{ii} = \frac{1}{n_G} \sum_u U^\dagger X_{ii} U. \tag{6.99}$$

6.6.3 *Time-reversal symmetry*

In most circumstances, the Hamiltonian matrix is not only Hermitian but also real. It implies that the system may have another type of symmetry, the time-reversal symmetry (TRS). Essentially the TRS means that if a dynamic process is recorded by a video and the video is played in reverse it will look as real as being played normally. A direct consequence of the TRS

is that the physical quantity of $-k$ can be deduced from that of $+k$ even if the system has no centrosymmetry. In bulk and two-probe systems, as long as no external magnetic field is applied, the Hamiltonian is always real which is a necessary condition of the TRS. However, a real Hamiltonian is insufficient for the TRS and one has to take statistics into account. Below we shall discuss the TRS for two important physical quantities: the density matrix and the transmission coefficient.

The density matrix determines the occupation of atomic orbitals and hence is a key variable. The density matrix ρ is related to $G^<$ by

$$\rho = (-\mathrm{i}) \int \frac{dE}{2\pi} G^< (E) , \tag{6.100}$$

where $G^< (E)$ is an integral over $G^< (E, k)$

$$G^< (E) = \int \frac{d^2 k}{(2\pi)^2} G^< (E, k) . \tag{6.101}$$

So the problem is reduced to whether and how $G^< (E, -k)$ and $G^< (E, +k)$ are related by the TRS. The answer is as follows: In equilibrium, the TRS exists and $G^< (E, -k) = [G^< (E, k)]^T$ where $[\cdots]^T$ represents the operation of matrix transpose; In nonequilibrium, the TRS does not exist, and $G^< (E, -k)$ and $G^< (E, +k)$ are independent. The derivation is as follows.

In equilibrium, due to the fluctuation-dissipation theorem Eq. (2.64), $G^<$ is related to G^r and G^a by $G^< (E) = f (E) [G^a (E) - G^r (E)]$. Since the Hamiltonian is real, it is also symmetric due to the Hermitian property, i.e., $H = H^T$. It follows that $G^r (E) = (E^+ - H)^{-1}$ and $G^a (E) = (E^- - H)^{-1}$ are symmetric, i.e., $G^r (E) = [G^r (E)]^T$ and $G^a (E) = [G^a (E)]^T$. Consequently $G^< (E)$ is symmetric because of the relation to $G^r (E)$ and $G^a (E)$. By making a Fourier transform of $G^< (E)$, one obtains

$$\begin{aligned} G^< (E, -k) &= \sum_I e^{-\mathrm{i}(-k)I} G^<_I (E) \\ &= \sum_{I'} e^{-\mathrm{i}(-k)(-I')} G^<_{-I'} (E) \\ &= \sum_{I'} e^{-\mathrm{i}kI'} \left[G^<_{I'} (E) \right]^T \\ &= \left[\sum_{I'} e^{-\mathrm{i}kI'} G^<_{I'} (E) \right]^T \\ &= \left[G^< (E, +k) \right]^T , \end{aligned} \tag{6.102}$$

where $G^<_{-I} (E) = [G^<_I (E)]^T$ is used in the derivation.

www.ebook3000.com

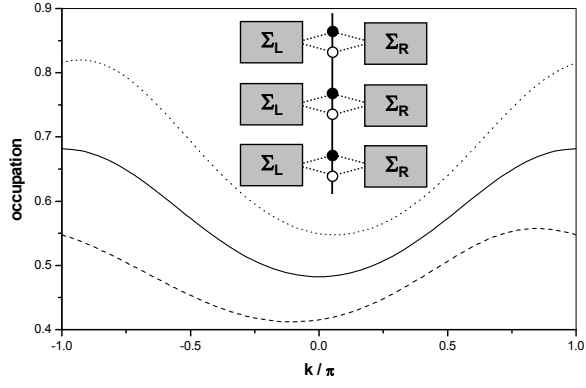


Fig. 6.11 The k-dependent occupation number $N(k)$ for different Fermi factors. The occupation number $N(k)$ is defined by $N(k) \equiv -\mathrm{Im}\ \mathrm{Tr}[G^<(E,k)]$. The Fermi factors are $(f_L, f_R) = (1,0)$, $\left(\frac{1}{2}, \frac{1}{2}\right)$ and $(0,1)$ in the dotted line, solid line, and dashed line, respectively. The two-probe model is shown in the inset and described in the text. Other parameters are: $E = 0$, $\varepsilon_A = 1$, $\varepsilon_B = 0$, $t = 1$, $\Sigma_L^r = -\frac{i}{2}[5, 0.5; 0.5, 1]$, $\Sigma_R^r = -\frac{i}{2}[2, 0.5; 0.5, 4]$.

In nonequilibrium, even though the Hamiltonian is real, the TRS does not exist. The nonequilibrium of chemical potentials results in the current flow and dissipation in the leads. It is the dissipation that destroys the TRS. The waterfall provides a vivid example of classical transport: Water falls from the upper bank and loses most of its kinetic energy in the lower bank. Nobody has ever seen a waterfall time-reversed to a "water rise". A quantum transport example is investigated in Fig. 6.11: The central region is composed of two sites, black dot and white dot, having the on-site energy ε_A and ε_B respectively. The coupling between the black dot and white dot is t. The central region is connected to the left and right leads by the self-energies Σ_L and Σ_R which are constructed deliberately to avoid any geometric symmetry [24]. It is verified that $G^<(E,-k)$ and $G^<(E,+k)$ are not symmetric with respect to $k = 0$ as long as the Fermi functions $f_L \neq f_R$ due to nonequilibrium.

The transmission coefficient determines the quantum transport and hence is another key variable. It turns out that the TRS of transmission coefficient depends on whether the two-probe system contains disorder. In clean two probe systems, the TRS exists and $T(E,k) = T(E,-k)$. In disordered two-probe systems, the TRS does not exist and $T(E,k)$ and $T(E,-k)$ are independent. The derivation is as follows.

In clean two-probe systems, the transmission coefficient $T(E)$ is an integral over $T(E,k)$

$$T(E)=\int_{BZ}\frac{d^2k}{(2\pi)^2}T(E,k), \tag{6.103}$$

where $T(E,k)$ is defined by

$$T(E,k)=\mathrm{Tr}\ [G^r(E,k)\,\Gamma_L(E,k)\,G^a(E,k)\,\Gamma_R(E,k)]. \tag{6.104}$$

The goal is to prove

$$T(E,k)=T(E,-k), \tag{6.105}$$

which will proceeds in two steps. In the first step, we prove that the lead index L and R are interchangeable in Eq. (6.104), namely

$$\begin{aligned}&\mathrm{Tr}\ [G^r(E,k)\,\Gamma_L(E,k)\,G^a(E,k)\,\Gamma_R(E,k)]\\&=\mathrm{Tr}\ [G^r(E,k)\,\Gamma_R(E,k)\,G^a(E,k)\,\Gamma_L(E,k)].\end{aligned} \tag{6.106}$$

By using $\Gamma(E,k)\equiv\Gamma_L(E,k)+\Gamma_R(E,k)$, the LHS and the RHS of Eq. (6.106) are reduced to

$$\begin{aligned}LHS=&\mathrm{Tr}\ [G^a(E,k)\,\Gamma(E,k)\,G^r(E,k)\,\Gamma_L(E,k)]\\&-\mathrm{Tr}\ [G^a(E,k)\,\Gamma_L(E,k)\,G^r(E,k)\,\Gamma_L(E,k)],\end{aligned} \tag{6.107}$$

$$\begin{aligned}RHS=&\mathrm{Tr}\ [G^r(E,k)\,\Gamma(E,k)\,G^a(E,k)\,\Gamma_L(E,k)]\\&-\mathrm{Tr}\ [G^r(E,k)\,\Gamma_L(E,k)\,G^a(E,k)\,\Gamma_L(E,k)],\end{aligned} \tag{6.108}$$

By using Eqs. (2.142,2.143), one obtains

$$G^a(E,k)\,\Gamma(E,k)\,G^r(E,k)=\mathrm{i}\,[G^r(E,k)-G^a(E,k)], \tag{6.109}$$

$$G^r(E,k)\,\Gamma(E,k)\,G^a(E,k)=\mathrm{i}\,[G^r(E,k)-G^a(E,k)]. \tag{6.110}$$

By inserting Eqs. (6.109,6.110) into Eqs. (6.107,6.108), one derives Eq. (6.106). In the second step, we prove Eq. (6.105) by using the transpose properties of Green's functions and linewidth functions. Since the Hamiltonian is real, Green's functions and linewidth functions have the following transpose properties

$$[G^r(E,k)]^T=G^r(E,-k), \tag{6.111}$$

$$[G^a(E,k)]^T=G^a(E,-k), \tag{6.112}$$

$$[\Gamma_L(E,k)]^T=\Gamma_L(E,-k), \tag{6.113}$$

$$[\Gamma_R(E,k)]^T=\Gamma_R(E,-k), \tag{6.114}$$

www.ebook3000.com

which can be derived in analogous to Eq. (6.102). By using Eq. (6.106) and Eqs. (6.111,6.112,6.113,6.114), one can derive Eq. (6.105) as follows

$$\begin{aligned} T(E,k) &= \mathrm{Tr}\ \left[G^r(E,k)\,\Gamma_L(E,k)\,G^a(E,k)\,\Gamma_R(E,k)\right] \\ &= \mathrm{Tr}\ \left[G^r(E,k)\,\Gamma_R(E,k)\,G^a(E,k)\,\Gamma_L(E,k)\right] \\ &= \mathrm{Tr}\ \left[G^r(E,k)\,\Gamma_R(E,k)\,G^a(E,k)\,\Gamma_L(E,k)\right]^T \\ &= \mathrm{Tr}\ \left[\Gamma_L(E,-k)\,G^a(E,-k)\,\Gamma_R(E,-k)\,G^r(E,-k)\right] \\ &= \mathrm{Tr}\ \left[G^r(E,-k)\,\Gamma_L(E,-k)\,G^a(E,-k)\,\Gamma_R(E,-k)\right] \\ &= T(E,-k)\,. \end{aligned} \tag{6.115}$$

QED.

In disordered two-probe systems, the transmission coefficient can be divided into the specular part and the diffusive part (see Eq. (3.156)),

$$T(E) = T_s(E) + T_d(E)\,. \tag{6.116}$$

The specular part is defined by

$$T_s(E) = \int \frac{d^2k}{(2\pi)^2} T_s^{RL}(E,k) = \int \frac{d^2k}{(2\pi)^2} T_s^{LR}(E,k)\,, \tag{6.117}$$

$$T_s^{RL}(E,k) = \mathrm{Tr}\ \overline{G^r}(E,k)\,\Gamma_L(E,k)\,\overline{G^a}(E,k)\,\Gamma_R(E,k)\,, \tag{6.118}$$

$$T_s^{LR}(E,k) = \mathrm{Tr}\ \overline{G^r}(E,k)\,\Gamma_R(E,k)\,\overline{G^a}(E,k)\,\Gamma_L(E,k)\,. \tag{6.119}$$

The diffusive part is defined by

$$T_d(E) = \int \frac{d^2k}{(2\pi)^2} T_d^{RL}(E,k) = \int \frac{d^2k}{(2\pi)^2} T_d^{LR}(E,k)\,, \tag{6.120}$$

$$T_d^{RL}(E,k) = \mathrm{Tr}\ \overline{G^r}(E,k)\,\Delta_L(E)\,\overline{G^a}(E,k)\,\Gamma_R(E,k)\,, \tag{6.121}$$

$$T_d^{LR}(E,k) = \mathrm{Tr}\ \overline{G^r}(E,k)\,\Delta_R(E)\,\overline{G^a}(E,k)\,\Gamma_L(E,k)\,. \tag{6.122}$$

Here the superscript RL (LR) represents the transmission coefficient of the scattering wave from the left (right) lead to the right (left) lead.

For the specular part, one can prove $T_s^{RL}(E,+k) = T_s^{LR}(E,-k)$ in analogy to Eq. (6.115), indicating that this part is reversible. For the diffusive part, $T_d^{RL}(E,+k)$ and $T_d^{LR}(E,-k)$ are independent (see Fig. 6.12), indicating this part is irreversible. From the physics analysis, k is a good quantum number in the left and right leads but is not conserved in the central region due to random disorder. The specular transmission coefficient describes momentum-conserved scattering in which an incoming wave with momentum k goes through the scattering region and conserves its momentum in the outgoing wave. The diffusive transmission describes

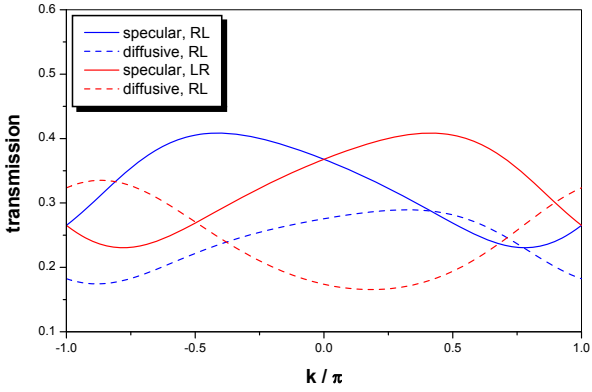


Fig. 6.12 The k-dependent transmission coefficients in a disordered two-probe system. The disordered two-probe system is similar to the inset of Fig. 6.11 except that the on-site energy of the black dot ε_A is a random variable, which may take the value $\varepsilon_A^{(1)} = 3$ and $\varepsilon_A^{(2)} = -3$ with probability $p_A^{(1)} = 0.5$ and $p_A^{(2)} = 0.5$, respectively. Other parameters are: $E = 0$, $\varepsilon_B = 0$, $t = 1$, $\Sigma_L^r = -\frac{\mathrm{i}}{2}[3, 0.5; 0.5, 1]$, $\Sigma_R^r = -\frac{\mathrm{i}}{2}[2, 0.5; 0.5, 4]$.

non-momentum-conserved scattering in which an incoming wave with momentum k is scattered by the random disorder into the outgoing wave with momentum $k' \neq k$. In the diffusive part, the information of k is lost in the disorder scattering, and one cannot recover k from k'. It is the information loss that sets up a time arrow and makes the quantum transport irreversible. A disordered two-probe system is investigated in Fig. 6.12. Here the two-probe model is similar to that of Fig. 6.11 except that the on-site energy of the black dot is a random variable. It is observed that only the specular transmission coefficient has the TRS, namely $T_s^{RL}(E, +k) = T_s^{LR}(E, -k)$. In contrast, the diffusive transmission coefficient $T_d^{RL}(E, +k)$ and $T_d^{LR}(E, -k)$ have very different shapes. Nevertheless $T^{RL}(E) = T^{LR}(E)$, $T_s^{RL}(E) = T_s^{LR}(E)$, and $T_d^{RL}(E) = T_d^{LR}(E)$ after the k-integral, which are the consequence of the current conservation and the TRS of the specular part. Therefore it is unnecessary to put the subscript RL (LR) on the quantities $T(E)$, $T_s(E)$, and $T_d(E)$. Interested readers are referred to Appendix A.16 for further discussions on specular scattering and diffusive scattering.

To sum up, the TRS is applicable to the calculation of density matrix in equilibrium or the transmission coefficient in clean two-probe systems. The TRS can relate the physical quantities of $-k$ and $+k$ even in systems with-

www.ebook3000.com

out the centrosymmetry. As a result, the uniform k-sampling of Fig. 6.9 can be reduced by half. The symmetric k-sampling of Fig. 6.10, however, cannot benefit from the TRS since the centrosymmetry has been included in the D_4 group. Finally it is worth mentioning that the spin degree of freedom is not considered explicitly in the above discussion. Within the scope of this monograph, the spin is regarded as a species index, and the Hamiltonian of each spin species is treated independently. This point of view is valid for the cases of neutral spin and collinear spin.

6.7 NECPA equations

This section discusses the algorithms for solving the NECPA-LMTO equations (3.121,3.122). Eqs. (3.121) has been discussed in Section 5.7, and Eqs. (3.122) will be the focus of this section.

To gain some insights into the mathematical structure, we first do a warm up exercise by studying the analytically solvable model defined in Section 5.7. By applying Eqs. (2.134) to the model, one obtains

$$\overline{G_i^<} = x_A \overline{G_{iA}^<} + x_B \overline{G_{iB}^<}, \tag{6.123}$$

$$\overline{G^<} = \overline{G^r} \left[\begin{pmatrix} \tilde{\varepsilon}_1^< & 0 \\ 0 & \tilde{\varepsilon}_2^< \end{pmatrix} + \begin{pmatrix} \mathrm{i}\Gamma f_1 & 0 \\ 0 & \mathrm{i}\Gamma f_2 \end{pmatrix} \right] \overline{G^a}, \tag{6.124}$$

$$\overline{G_i^<} = \left[\overline{G^<} \right]_{ii}, \tag{6.125}$$

$$\overline{G_i^<} = \overline{G_i^r} \left(\tilde{\varepsilon}_i^< + \Omega_i^< \right) \overline{G_i^a}, \tag{6.126}$$

$$\overline{G_{iA}^<} = \overline{G_{iA}^r} \Omega_i^< \overline{G_{iA}^a}, \tag{6.127}$$

$$\overline{G_{iB}^<} = \overline{G_{iB}^r} \Omega_i^< \overline{G_{iB}^a}, \tag{6.128}$$

where $i = 1, 2$ is the site index. f_1 and f_2 are the Fermi functions of the left and right reservoirs respectively. In nonequilibrium, $f_1 \neq f_2$ which results in $\Omega_1^< \neq \Omega_2^<$ and $\tilde{\varepsilon}_1^< \neq \tilde{\varepsilon}_2^<$.

Eqs. (6.123–6.128) consist a linear equation array. It is straightforward to eliminate all other variables to obtain the linear equations of $\tilde{\varepsilon}_1^<$ and $\tilde{\varepsilon}_2^<$

$$\begin{cases} a_{11}\tilde{\varepsilon}_1^< + a_{12}\tilde{\varepsilon}_2^< = b_1 \\ a_{21}\tilde{\varepsilon}_1^< + a_{22}\tilde{\varepsilon}_2^< = b_2 \end{cases}, \tag{6.129}$$

where the coefficients are

$$a_{11} = \frac{\left|\overline{G^r_{11}}\right|^2}{x_A \frac{\left|\overline{G^r_{1A}}\right|^2}{\left|\overline{G^r_{11}}\right|^2} + x_B \frac{\left|\overline{G^r_{1B}}\right|^2}{\left|\overline{G^r_{11}}\right|^2} - 1}, \tag{6.130}$$

$$a_{12} = -\left|\overline{G^r_{12}}\right|^2, \tag{6.131}$$

$$a_{22} = \frac{\left|\overline{G^r_{22}}\right|^2}{x_A \frac{\left|\overline{G^r_{2A}}\right|^2}{\left|\overline{G^r_{22}}\right|^2} + x_B \frac{\left|\overline{G^r_{2B}}\right|^2}{\left|\overline{G^r_{22}}\right|^2} - 1}, \tag{6.132}$$

$$a_{21} = -\left|\overline{G^r_{21}}\right|^2, \tag{6.133}$$

$$b_1 = \mathrm{i}\Gamma\left(f_1\left|\overline{G^r_{11}}\right|^2 + f_2\left|\overline{G^r_{12}}\right|^2\right), \tag{6.134}$$

$$b_2 = \mathrm{i}\Gamma\left(f_2\left|\overline{G^r_{22}}\right|^2 + f_1\left|\overline{G^r_{21}}\right|^2\right). \tag{6.135}$$

Notice that $\overline{G^r_{ij}}$, $\overline{G^r_{iA}}$, $\overline{G^r_{iB}}$ in the coefficients a_{ij} and b_i can be solved from Eqs. (5.78–5.83) as

$$\overline{G^r_{11}} = \overline{G^r_{22}} = \frac{E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma}{\left(E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma\right)^2 - t_0^2}, \tag{6.136}$$

$$\overline{G^r_{12}} = \overline{G^r_{21}} = \frac{t_0}{\left(E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma\right)^2 - t_0^2}, \tag{6.137}$$

$$\overline{G^r_{1A}} = \overline{G^r_{2A}} = \frac{1}{x_A}\frac{\tilde{\varepsilon}^r - \varepsilon_B}{\varepsilon_A - \varepsilon_B}\overline{G^r_{11}}, \tag{6.138}$$

$$\overline{G^r_{1B}} = \overline{G^r_{2B}} = \frac{1}{x_B}\frac{\tilde{\varepsilon}^r - \varepsilon_A}{\varepsilon_B - \varepsilon_A}\overline{G^r_{11}}, \tag{6.139}$$

where the analytical solution of $\tilde{\varepsilon}^r$ has been derived in Eq. (5.88). Consequently $\tilde{\varepsilon}^<_1$ and $\tilde{\varepsilon}^<_2$ are solved analytically from Eq. (6.129)

$$\tilde{\varepsilon}^<_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}}, \tag{6.140}$$

$$\tilde{\varepsilon}^<_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{12}a_{21}}. \tag{6.141}$$

By inserting the analytical solution of $\tilde{\varepsilon}^<_1$ and $\tilde{\varepsilon}^<_2$ to Eq. (6.124), one obtains disorder-averaged lesser Green's function and other physical quantities. It will be a good exercise to check the equilibrium limit where $f_1 = f_2 = f_0$. In equilibrium, due to the fluctuation-dissipation theorem, $\tilde{\varepsilon}^<_i$ is related to $\tilde{\varepsilon}^r_i$ and $\tilde{\varepsilon}^a_i$ by $\tilde{\varepsilon}^<_i = f_0\left(\tilde{\varepsilon}^a_i - \tilde{\varepsilon}^r_i\right)$. By using Eq. (5.88) for $\tilde{\varepsilon}^r_i$ and Eqs. (6.140,6.141) for $\tilde{\varepsilon}^<_i$, one can prove the above identity after some lengthy algebra [25].

www.ebook3000.com

So much for the analytical work. Now let us solve the same problem numerically with the iterative method. Assume that all the retarded quantities $\tilde{\varepsilon}^r$, $\overline{G^r}$, $\overline{G^r_{iA}}$, $\overline{G^r_{iB}}$ and their advanced counterparts have been solved by using the iterative procedure of Section 5.7. The procedure of solving $\tilde{\varepsilon}^<$ and $\overline{G^<}$ is described as follows:

(1') Make an initial guess of $\tilde{\varepsilon}^<_i$

$$\varepsilon^<_1 = \varepsilon^<_2 = 0;$$

(2') Calculate the diagonal element of $\overline{G^<}$ by using Eqs. (6.124,6.125)

$$\overline{G^<_{ii}} = \left\{ \overline{G^r} \left[\begin{pmatrix} \tilde{\varepsilon}^<_1 & 0 \\ 0 & \tilde{\varepsilon}^<_2 \end{pmatrix} + \begin{pmatrix} \mathrm{i}\Gamma f_1 & 0 \\ 0 & \mathrm{i}\Gamma f_2 \end{pmatrix} \right] \overline{G^a} \right\}_{ii};$$

(3') Update $\Omega^<_i$ by using Eq. (6.126)

$$\Omega^<_i = \frac{\overline{G^<_{ii}}}{\left|\overline{G^r_{ii}}\right|^2} - \tilde{\varepsilon}^<_i;$$

(4') Update $\tilde{\varepsilon}^<_i$ by using Eqs. (6.123,6.126,6.127,6.128)

$$\tilde{\varepsilon}^<_i = x_A \frac{\left|\overline{G^r_{iA}}\right|^2}{\left|\overline{G^r_{ii}}\right|^2} \Omega^<_i + x_B \frac{\left|\overline{G^r_{iB}}\right|^2}{\left|\overline{G^r_{ii}}\right|^2} \Omega^<_i - \Omega^<_i;$$

(5') Go back to the step (2) to repeat the process until $\tilde{\varepsilon}^<_i$ is fully converged.

The analytical solution and the numerical solution of $\tilde{\varepsilon}^<_1$ and $\tilde{\varepsilon}^<_2$ are plotted in Fig. 6.13. One can see that the two solutions are on top of each other, verifying that the iterative method is accurate and reliable.

With the simple model in mind, we proceed to discuss the algorithm for solving the NECPA-LMTO equations (3.122). Analogously to the simple model, we can adopt the iterative method to solve the equations. Here the major differences are (i) $E - H$ is replaced by $P(E) - S(k)$ in the LMTO method and (ii) the integral over k is needed to take into account the periodicity in the transverse dimensions. The procedure is described as follows:

(1') Make an initial guess of $\tilde{P}^<_i(E)$,

$$\tilde{P}^<_i(E) = 0; \tag{6.142}$$

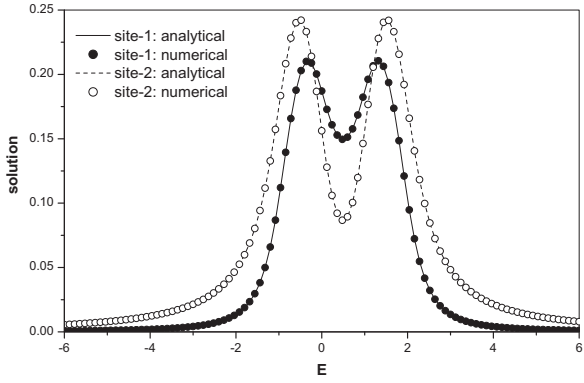


Fig. 6.13 Comparison of the numerical solution and the analytical solution of the lesser coherent potential $\tilde{\varepsilon}_1^<$ and $\tilde{\varepsilon}_2^<$ in the two-site model. Notice that the real part of $\tilde{\varepsilon}_i^<$ is zero and only the imaginary part is plotted. The solid line (black dot) is the analytical (numerical) solution of $\tilde{\varepsilon}_1^<$. The dashed line (white dot) is the analytical (numerical) solution of $\tilde{\varepsilon}_2^<$. The error between the two solutions is less than 2×10^{-6}. Other parameters are $x_A = x_B = 0.5$, $\varepsilon_A = 0$, $\varepsilon_B = 1$, $t_0 = 1$, $\Gamma = 1$, $f_1 = 0.1$, $f_2 = 0.9$.

(2') Calculate $\overline{\mathcal{G}}_i^< (E)$ by using the second, third, and fourth line of Eqs. (3.122)

$$\overline{\mathcal{G}}_i^< (E) = \int_{BZ} \frac{d^3k}{(2\pi)^3} \overline{\mathcal{G}}_i^< (E,k)\,, \tag{6.143}$$

$$\overline{\mathcal{G}}_i^< (E,k) = \left[\overline{\mathcal{G}}^< (E,k)\right]_{ii}\,,$$

$$\overline{\mathcal{G}}^< (E,k) = \overline{\mathcal{G}}^r (E,k) \left[-\tilde{P}^< (E) + \Sigma^< (E,k)\right] \overline{\mathcal{G}}^a (E,k)\,,$$

$$\left[\tilde{P}^< (E)\right]_{ij} = \delta_{ij} \tilde{P}_i^< (E)\,;$$

(3') Update $\Omega_i^< (E)$ by using the fifth line of Eqs. (3.122)

$$\Omega_i^< (E) = \tilde{P}_i^< (E) + \left[\overline{\mathcal{G}}_i^r (E)\right]^{-1} \overline{\mathcal{G}}_i^< (E) \left[\overline{\mathcal{G}}_i^a (E)\right]^{-1}; \tag{6.144}$$

(4') Update $\tilde{P}_i^< (E)$ by using the first, fifth, and sixth line of Eqs. (3.122)

$$\tilde{P}_i^< (E) = \Omega_i^< (E) - \left[\overline{\mathcal{G}}_i^r (E)\right]^{-1} \left[\sum_q x_{iq} \overline{\mathcal{G}}_{iq}^r (E)\, \Omega_i^< (E)\, \overline{\mathcal{G}}_{iq}^a (E)\right] \left[\overline{\mathcal{G}}_i^a (E)\right]^{-1}; \tag{6.145}$$

(5') Go back to the step (2') to repeat the process until $\tilde{P}_i^< (E)$ is fully converged.

www.ebook3000.com

In the above procedure, it is assumed that all the retarded quantities $\tilde{P}^r(E)$, $\overline{\mathcal{G}}^r(E)$, $\overline{\mathcal{G}}_i^r(E)$, $\overline{\mathcal{G}}_{iq}^r(E)$ and their advanced counterparts $\tilde{P}^a(E)$, $\overline{\mathcal{G}}^a(E)$, $\overline{\mathcal{G}}_i^a(E)$, $\overline{\mathcal{G}}_{iq}^a(E)$ have been solved by using the iterative procedure of Section 5.7. In practice, the two iterative procedures can be merged into one, and hence the retarded and the lesser coherent potential are solved simultaneously. The merged procedure has the following 10 steps: (1), (1'), (2), (2'), (3), (3'), (4), (4'), (5), (5'), where (1) to (5) are the iterative steps for solving the retarded coherent potential and (1') to (5') are the iterative steps for solving the lesser coherent potential.

There is an important variation of step (4'), which makes a connection from the NECPA theory to the CPA-NVC theory [26]. By inserting Eq. (6.144) into Eq. (6.145) and eliminating $\Omega_i^<(E)$, after some algebra (see Appendix A.8), one obtains an alternative expression for updating $\tilde{P}_i^<(E)$ in the step (4')

$$\begin{aligned}\Lambda_i(E) = &\sum_q x_{iq} t_{iq}^r(E)\, \overline{\mathcal{G}}_i^<(E)\, t_{iq}^a(E) \\ &- \sum_q x_{iq} t_{iq}^r(E)\, \overline{\mathcal{G}}_i^r(E)\, \Lambda_i(E)\, \overline{\mathcal{G}}_i^a(E)\, t_{iq}^a(E), \end{aligned} \tag{6.146}$$

where $\Lambda_i(E) \equiv -\tilde{P}_i^<(E)$ is the nonequilibrium vertex correction [26] and $t_{iq}^r(E)$ is the scattering amplitude defined by

$$t_{iq}^r(E) \equiv V_{iq}(E) \left[1 - \overline{\mathcal{G}}_i^r(E)\, V_{iq}(E)\right]^{-1}, \tag{6.147}$$

$$V_{iq}(E) \equiv \tilde{P}_i^r(E) - P_{iq}(E). \tag{6.148}$$

Finally we would like to discuss some implementation details. The above iterative procedure has been implemented in the method *CPA_solution* of @class_necpaTwoProbe. To be precise, the method *CPA_solution* calls the methods *calcCPA_gr*, *calcCPA_gd*, *calcCPA_Omega_r*, *calcCPA_Omega_d*, *calcCPA_tiltP_r*, and *calcCPA_tiltP_d* of @class_necpaAtom to carry out the steps (2), (2'), (3), (3'), (4), and (4'), respectively. In the steps (2) and (2'), the k-integral is evaluated by using the private function *integrateBZ_neqb* which is the bottleneck of the NECPA iteration. To evaluate the k-integral, one needs to select k-points in the Brillouin zone, and the k-sampling methods have been discussed in Section 6.6.

In post-analysis calculations, the NECPA equations need to be solved accurately at every energy point (see e.g., Section 6.9). In self-consistent calculations, however, it is unnecessary to solve the NECPA equations accurately in every self-consistent iteration. In the early stage of a self-consistent

calculation, the potential is far away from the correct answer, so it does not make too much sense to solve the NECPA equations to high precision. In analogy to Section 5.7, the NECPA loop and the self-consistent loop can be merged into one big loop, and the atomic potential and nonequilibrium coherent potential are converged simultaneously. The idea has been implemented in the method *CPA_iteration* of @class_necpaTwoProbe. The flow of *CPA_iteration* is quite similar to that of *CPA_solution* except that the NECPA iteration is carried out only once. To use the method *CPA_iteration* in a self-consistent calculation, it is necessary to construct a reasonable initial guess of the nonequilibrium coherent potential. The "warm up" is done by calling *CPA_solution* prior to the self-consistent loop.

In some circumstances, the disorder concentration is very low and it is unnecessary to solve the NECPA equations iteratively. Approximate analytical expressions are available in Eqs. (3.124,3.125). To implement the formulas, one needs to first calculate the unperturbed Green's function $\mathcal{G}_0^r(E)$ and $\mathcal{G}_0^<(E)$ by using Eqs. (3.127,3.128,3.129). Afterward one can proceed to calculate the nonequilibrium coherent potential $\tilde{P}_i^r(E)$ and $\tilde{P}_i^<(E)$ by using Eqs. (3.124,3.125,3.126). Once the $\tilde{P}_i^r(E)$ and $\tilde{P}_i^<(E)$ are available, the disorder-averaged Green's functions $\overline{\mathcal{G}^r}(E)$ and $\overline{\mathcal{G}^<}(E)$ can be evaluated by using the NECPA-LMTO equations (3.121) and (3.122). So the computational cost is nearly twice that of the clean systems but is still much lower than the iterative method. The low concentration limit has been implemented in the method *CPA_solution_lowX* of @class_necpaTwoProbe which calls the method *calcCPA_tiltP_r_lowX* and *calcCPA_tiltP_d_lowX* of @class_necpaAtom.

To sum up, the NECPA equations can be solved with the iterative procedure Eqs. (5.94,5.95,5.96,5.97) and Eqs. (6.142,6.143,6.144,6.145). The procedure has been implemented in the methods *CPA_solution*, *CPA_iteration*, and *CPA_solution_lowX*. The method *CPA_solution* is to solve the NECPA equations for a given LMTO Hamiltonian; The method *CPA_iteration* is to iterate the CPA equations once for a given LMTO Hamiltonian; The method *CPA_solution_lowX* is to evaluate an approximate solution of the NECPA equations in the low concentration limit.

6.8 Fermi level alignment

This section discusses the Fermi level alignment in two-probe systems. In nonequilibrium two-probe systems, the concept of Fermi level is no longer valid. The two leads are in local equilibrium with local chemical potentials

www.ebook3000.com

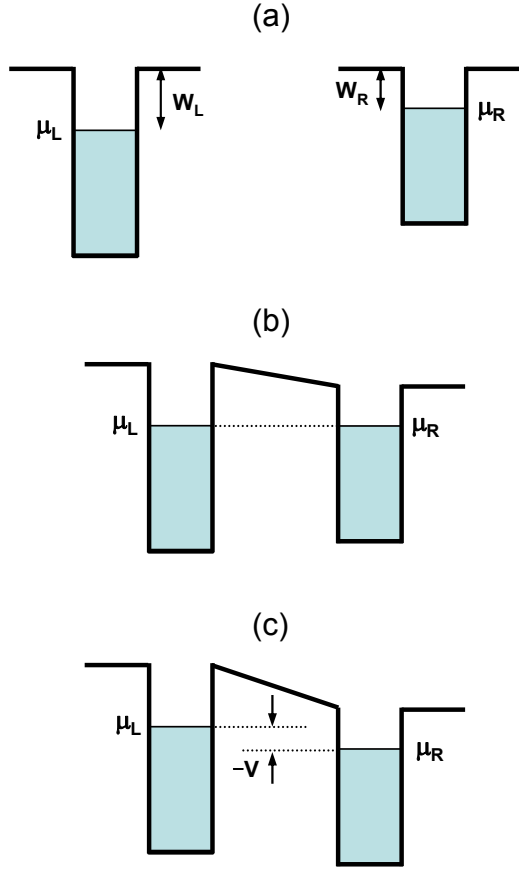


Fig. 6.14 Schematic diagram of the Fermi level alignment. (a) Separate bulk systems: the left and right leads are separated from each other, hence μ_L and μ_R are independent. (b) Equilibrium two-probe system: the left and right leads are connected by some intermediate material, hence $\mu_L = \mu_R$. (c) Nonequilibrium two-probe system: a bias voltage V is applied to the left and right leads, hence $\mu_L - \mu_R = -V$.

μ_L and μ_R respectively. The question is how μ_L and μ_R are aligned with each other in nonequilibrium two-probe systems.

To answer the question, let us do a thought experiment. In Fig. 6.14a, the two leads are separated from each other. Each lead can be considered as a container and electrons fill the container to the Fermi level μ. The distance from μ to the vacuum level is the work function W. In Fig. 6.14b,

the two leads are connected by some intermediate material to form a two-probe system. At the beginning, the vacuum level is flat, and charge is transferred from the lead with higher Fermi level to the lead with lower Fermi level. As a result, a dipole layer is built up at the interface. The electric field of the dipole layer tends to prevent further charge transfer. The dynamic process continues until $\mu_L = \mu_R$ and a new equilibrium is established. The vacuum level is no longer flat and the slope is determined by the work function difference. In Fig. 6.14c, a bias voltage V is applied to the left and right leads. The two-probe system is driven to nonequilibrium, and the chemical potential difference is determined by

$$\mu_L - \mu_R = -V, \tag{6.149}$$

where the minus sign of V accounts for the negative charge of electrons. The dipole layer is either strengthened or suppressed depending on the polarity of the bias voltage. The field of the dipole layer not only compensates for the work function difference but also builds up a voltage drop at the interface. Once the steady state is established, current flows continuously through the two-probe system.

A consequence of the Fermi level alignment is to shift the energy zero of lead Hamiltonians. In a two-probe calculation, the left and right leads are calculated by using bulk systems which may have arbitrary energy zeros. The energy zeros need to be shifted so that the lead Fermi levels satisfy Eq. (6.149). Consequently the lead potential parameters, the lead linearization centers, and the lead Madelung potential need to be shifted by the same amount. The energy shift is done in the method *prepareLead.m* of @class_necpaTwoProbe, where $\mu_L = -\frac{V}{2}$ and $\mu_R = +\frac{V}{2}$ are assumed for convenience.

6.9 Two-probe calculator: transmission coefficient

This section discusses the algorithms for calculating the transmission coefficient in two-probe systems. Essentially there are two approaches to calculating the transmission coefficient: the Green's function approach and the wave function approach. The Green's function approach is implemented in @calculator_trans_CPA and the wave function approach is implemented in @calculator_trans_SCS. Below we shall discuss some implementation details.

In the Green's function approach, the transmission coefficient is calculated by using Eq. (3.151). There is an outer loop over the energy points

www.ebook3000.com

where the disorder-averaged transmission coefficient $\overline{T}(E)$ is calculated. For each energy E, the calculation proceeds in three steps: Step 1 calculate the lead self-energy $\Sigma_\beta^r(E,k)$; Step 2 solve the NECPA equations to evaluate $\overline{\mathcal{G}^r \mathrm{i}\Gamma_L \mathcal{G}^a}$ or $\overline{\mathcal{G}^r \mathrm{i}\Gamma_R \mathcal{G}^a}$; Step 3 calculate $\overline{T}(E)$ by integrating over k in the Brillouin zone. In the implementation, the NECPA equations are solved by overloading the energy integral used in the self-consistent calculations. In step 1, the real-axis is overloaded with the energy point E, and the self-energy is calculated by using the method *prepareSelfenergy* of @class_necpaTwoProbe. In step 2, $\mathrm{i}\Gamma_L$ or $\mathrm{i}\Gamma_R$ is "disguised" as a lesser self-energy $\tilde{\Sigma}^<$

$$\tilde{\Sigma}^< = \tilde{f}_L\left(\Sigma_L^a - \Sigma_L^r\right) + \tilde{f}_R\left(\Sigma_R^a - \Sigma_R^r\right),$$

where $\tilde{f}_L = 1$ and $\tilde{f}_R = 0$ for $\mathrm{i}\Gamma_L$, and $\tilde{f}_R = 1$ and $\tilde{f}_L = 0$ for $\mathrm{i}\Gamma_R$. Afterward the disorder average is evaluated by using the method *CPA_solution* of @class_necpaTwoProbe. In step 3, the k-integral is evaluated by using the private function *calcTrans* of @calculator_trans_CPA. In addition to the transmission coefficient, one may also plot the k-resolved transmission map in the Brillouin zone.

In the wave function approach, the transmission coefficient is calculated by using Eqs. (A.111,A.100) derived in Appendix A.6. Notice that the wave function approach is only applicable to clean two-probe systems. The outer loops over E-points and k-points are organized by the calculator @calculator_trans_SCS. For a given E point and a k point, $T(E,k)$ is calculated by the solver @solver_ScattTrans. The calculation proceeds in three steps: Step 1 solve the bulk states of the left and right leads; Step 2 solve the scattering matrix from the equation of scattering states; Step 3 calculate transmission coefficient by summing up the modulus square of transmission amplitude. In step 1, generalized eigenvalue problems are solved for lead unit cells by using the private function *solveBulkStates* of @solver_ScattTrans. The obtained eigenstates are classified into four categories: left-decaying modes, left-traveling modes, right-decaying modes, and right-traveling modes. All the traveling modes constitute the conducting channels. In step 2, the equation of scattering states is constructed according to Eq. (A.100), and the scattering matrix is solved by using the private function *solveScatteringStates* of @solver_ScattTrans. Notice that the traveling modes need to be normalized properly so that they carry the same amount of charge. In step 3, the transmission coefficient is calculated by summing up the modulus square of transmission amplitude for all conducting channels. As a by-product, the solver may also output the

conducting channel number, the scattering matrix and the scattering wave function.

To sum up, two transmission coefficient calculators have been implemented in NanoDsim. For clean two-probe systems, the two calculators are totally equivalent. The equivalence has been proved analytically in Ref. [28, 29] and will be tested numerically in Section 6.10. For disordered two-probe systems, the wave function-based calculator needs to construct a supercell to simulate the random disorder. In contrast, the Green's function-based calculator works in a unit cell and simulates the disorder effect with the NECPA theory.

6.10 Verification of the implementation

We have discussed all the algorithms for the calculation of bulk systems and two-probe systems. By implementing the algorithms, we developed the *NanoDsim* package which is based on the NECPA-LMTO formalism presented in Section 3.11 and 3.12. However, it is unavoidable for a human programmer to make mistakes. How do we know that the algorithms have been implemented correctly? Some debugging strategies have been discussed in Section 4.10. Here we shall do some global tests to check the consistency between different types of systems or calculations. It is very unlikely that one makes different mistakes and still obtains consistent results.

The first test is to check the consistency between bulk systems and two-probe systems in a self-consistent calculation. The target is a perfect Cu crystal. On the one hand, it is a genuine bulk system by definition. On the other hand, it can also be regarded as a special two-probe system in which the central region is identical to both left and right leads (see Fig. 4.3). So one can simulate the target system with either a bulk self-consistent calculation or a two-probe self-consistent calculation. To make a fair comparison, one needs to use the same k-sampling in the transverse dimensions. It is also required to use a sufficiently large number of k-points in the transport direction for the bulk self-consistent calculation. In the two self-consistent calculations, the converged potential parameters agree very well with each other after aligning the Fermi levels. As a further check, the density of states is calculated for both bulk system and two-probe system, and the results are nearly indistinguishable in Fig. 6.15.

The second test is to check the consistency between the Green's function method and the wave function method in the transmission coefficient cal-

www.ebook3000.com

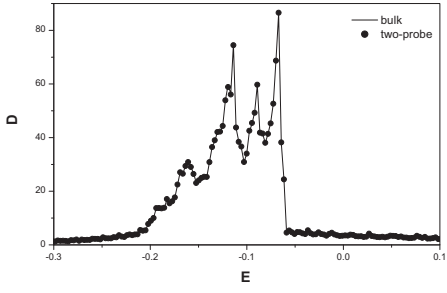


Fig. 6.15 Density of states of a clean Cu crystal. The calculations are carried out in both bulk system (solid line) and two-probe system (dots).

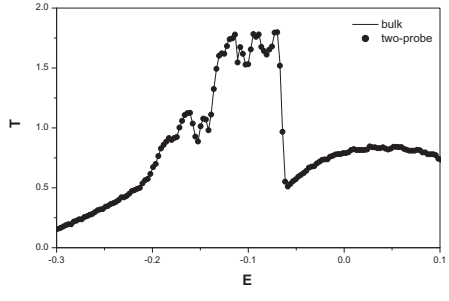


Fig. 6.16 Transmission coefficient of a clean Cu two-probe system in the 111 direction. The calculations are carried out with both the Green's function method (solid line) and the wave function method (dots).

culation. For clean two-probe systems, the target is the clean Cu two-probe system which was investigated in the first test. The two methods of calculating transmission coefficient are distinct in algorithms and implemented independently. Nevertheless it is shown in Fig. 6.16 that the transmission coefficient curves are nearly on top of each other, indicating that the calculators have been implemented correctly for clean two-probe systems. For disordered two-probe systems, the target is a disordered Cu/Co interface which was investigated in Ref. [30]. To simulate the interface roughness, one can work in a unit cell by using the NECPA theory or work in a supercell by averaging disorder configurations. It will be shown in Section 8.5 that the results from the two methods are in an excellent agreement, indicating that the calculators have been implemented correctly for disordered two-probe systems.

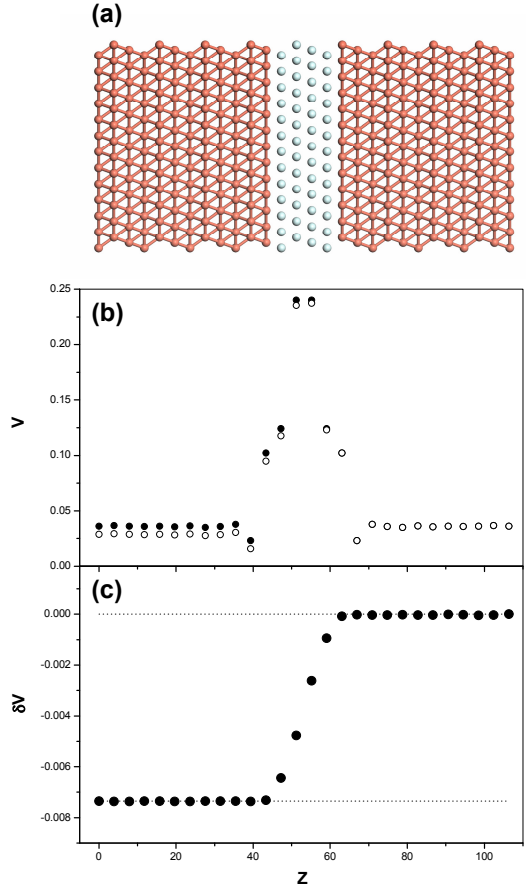


Fig. 6.17 Electrostatic potentials of a Cu-Vac-Cu two-probe system. (a) The atomic structure of the Cu-Vac-Cu two-probe system. (b) The electrostatic potentials of the Cu-Vac-Cu two-probe system in both equilibrium (white dot) and nonequilibrium (black dot). (c) The difference between the electrostatic potentials in equilibrium and nonequilibrium. It is verified that the potential difference is consistent with the applied bias voltage (doted line).

The third test is to check the consistency between the electrostatic potential of a nonequilibrium two-probe system and the applied bias voltage. The system is a Cu-Vac-Cu two-probe system shown in Fig. 6.17a. The two-probe system can be viewed as a perfect Cu crystal in which four layers of Cu atoms are replaced by vacuum spheres. Self-consistent calculations

www.ebook3000.com

are carried out in both equilibrium ($V = 0$) and nonequilibrium ($V = 0.2V$), and the electrostatic potentials are plotted in Fig. 6.17b. One can see that the two potentials have very similar shape, both have a large tunnel barrier (about $5.5V$) in the vacuum region. The minor difference δV between the two potentials is due to the applied bias voltage, which is plotted in Fig. 6.17c. One can see that the shape of δV mimics the potential of a classical capacitor and the potential drop is consistent with the applied bias voltage.

To sum up, self-consistent calculations are verified for both bulk and two-probe systems, in both equilibrium and nonequilibrium situations; transmission coefficient calculations are verified for both Green's function and wave function methods, in both clean and disordered systems.

Bibliography

[1] H. L. Skriver and N. M. Rosengaard, Phys. Rev. B **43**, 9538 (1991).
[2] J. M. MacLaren, S. Crampin, D. D. Vvedensky, and J. B. Pendry, Phys. Rev. B **40**, 12164 (1989).
[3] J. Kudrnovský, I. Turek, V. Drchal, P. Weinberger, S. K. Bose, and A. Pasturel, Phys. Rev. B **47**, 16525 (1993).
[4] Y. Zhu, L. Liu, and H. Guo, unpublished (2008).
[5] From the Poisson equation point of view, $\nabla^2 C(z) = 0$, hence $C(z)$ is nothing but a homogeneous solution to help the Madelung potential satisfy a proper boundary condition.
[6] I. Appelbaum, T. Wang, J. D. Joannopoulos, V. Narayanamurti, Phys. Rev. B **69**, 165301 (2004).
[7] To derive Eq. (6.50) from Eq. (6.48), there is no constraint on i_1 and i_2. To derive Eq. (6.50) from Eq. (6.46), it is further assumed that $i_1 \sim \frac{N}{2}$ and $i_2 \sim \frac{N}{2}$.
[8] M. P. López Sancho, J. M. López Sancho, and J Rubio, J. Phys. F: Mat. Phys. **15**, 851 (1985).
[9] F. Guinea, C. Tejedor, F. Flores, and E. Louis, Phys. Rev. B **28**, 4397 (1983).
[10] S. Sanvito, C. J. Lambert, J. H. Jefferson, A. M. Bratkovsky, Phys. Rev. B **59**, 11936 (1999).
[11] A. Umerski, Phys. Rev. B **55**, 5266 (1997).
[12] Strickly speaking this property is valid even if the time-reversal symmetry is violated. It is easy to verify that if λ is an eigenvalue of Eq. (6.61), $(\lambda^*)^{-1}$ is also an eigenvaue. For $|\lambda| \neq 1$, one of λ and $(\lambda^*)^{-1}$ is left-moving and the other is right-moving. For $|\lambda| = 1$, let $\lambda = e^{ik}$ in which $k \in (-\pi, \pi)$. Eq. (6.61) defines the bandstructure of a 1d bulk system. For a given energy E, the eigenstates are always in pair. One eigenstate has positive group velocity and hence is right-moving, and the other has negative group velocity and hence is left-moving. See Appendix A.6 for more details.

[13] L. Liu, unpublished (2004).
[14] I. Rungger and S. Sanvito, Phys. Rev. B **78**, 035407 (2008).
[15] A. R. Rocha, V. M. García-Suárez, S. Bailey, C. Lambert, J. Ferrer, and S. Sanvito, Phys. Rev. B **73**, 085414 (2006).
[16] M. Luisier, A. Schenk, W. Fichtner, and G. Klimeck, Phys. Rev. B **74**, 205323 (2006).
[17] Y. Zhu, L. Liu, and H. Guo, unpublished (2011).
[18] H. H. B. Sørensen, P. C. Hansen, D. E. Petersen, S. Skelboe, and K. Stokbro, Phys. Rev. B **77**, 155301 (2008).
[19] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B **63**, 245407 (2001).
[20] van Hove singularities come from the lead self-energies, while Sharp resonances and bound states come from the Green's functions. For a numerical example, see Appendix A.2.
[21] S. Agarwal, M. Povolotskyi, T. Kubis, and G. Klimeck, J. Comput. Electron 9, 252 (2010).
[22] Y. Zhu, L. Liu, and H. Guo, unpublished (2015).
[23] K. Koepernik, B. Velický, R. Hayn, and H. Eschrig, Phys. Rev. B **55**, 5717 (1997).
[24] There is a lot of freedom in constructing a retarded self-energy Σ^r. The only constraint is that the resulting linewidth function $\Gamma \equiv \mathrm{i}(\Sigma^r - \Sigma^a)$ must be positive-definite.
[25] To avoid the tedious work, one may use *mathematica* in the symbolic derivation. A few points are worth mentioning: (1) Define retarded and advanced variables separately since *mathematica* does not know that they are complex conjugate. (2) Try not to solve $\tilde{\varepsilon}^r$ from the quadratic equation which induces a square root and makes the simplication complicated. Instead replace $(\tilde{\varepsilon}^r)^2$ and $(\tilde{\varepsilon}^a)^2$ by linear terms deduced from the quadratic equation. A *mathematica* code is available in the *ResearchCode* folder.
[26] Y. Ke, K. Xia, and H. Guo, Phys. Rev. Lett. **100**, 166805 (2008), and associated Supplemental Material (E-PRLTAO-100-020817).
[27] Z. Ren, Ph.D. thesis, Purdue University, 2001.
[28] D. S. Fisher and P. A. Lee, Phys. Rev. B **23**, 6851 (1981).
[29] P. A. Khomyakov, G. Brocks, V. Karpan, M. Zwierzycki, and P. J. Kelly, Phys. Rev. B **72**, 035450 (2005).
[30] K. Xia, M. Zwierzycki, M. Talanana, P. J. Kelly, G. E. W. Bauer, Phys. Rev. B **73**, 064420 (2006).

www.ebook3000.com

Chapter 5

NanoDsim: bulk systems

In this chapter, we shall discuss how to implement dsim-classes, dsim-solvers, and dsim-calculators for bulk systems. Although the theoretical formulas have been derived in Chapter 3, one cannot write computer programs by translating the formulas directly. One needs to design the classes and solvers, and choose proper **algorithms** to implement the formulas. Section 5.1 and 5.2 will be devoted to the design of dsim-classes and dsim-solver for bulk systems. Section 5.3 to 5.10 will be devoted to various algorithms used in the self-consistent and post-analysis calculations.

A good algorithm may accelerate a numerical calculation by several orders of magnitude while hardware improvements usually speed up the same calculation by a small factor. Here's a simple example to illustrate a good algorithm. To compute the summation $1+2+\cdots+100$, a straightforward method is to add them up term by term. A good algorithm, as discovered by C. F. Gauss in his childhood, is to carry out the summation by pairing the numbers

$$\begin{aligned} & 1+2+\cdots+100 \\ =\ & \frac{1}{2}\left[(1+100)+(2+99)+\cdots+(100+1)\right] \\ =\ & \frac{1}{2}\times 101\times 100 \\ =\ & 5050. \end{aligned}$$

We shall see many similar tricks in this chapter which help to implement the theoretical formulas efficiently.

5.1 Bulk classes

As mentioned in Section 4.8, there are four dsim-classes for bulk systems. The hierarchy of the dsim-classes is @class_cpaBulk ⊃ @class_cpaAtom ⊃ @class_lmtoAtom ⊃ @class_lmtoOrbital, where ⊃ refers to the class aggregation. The design of these classes is explained in the following subsections.

5.1.1 *@class_cpaBulk*

The class @class_cpaBulk has 17 properties which are summarized in Table 5.1. In the column of data structure, # stands for an array index, * stands for a string, and *[1]* stands for a scalar.

Property	**Meaning**	**Data Structure**
AtomSet	atomic sites	*AtomSet(#) = @class_cpaAtom*
StructureConstant	structure constant	*StructureConstant(#).Displacement = zeros(1,3)* *StructureConstant(#).data = zeros(#,#)*
MadelungConstant	Madelung constant	*MadelungConstant = zeros(#,#,4)*
MadelungPotential	Madelung potential	*MadelungPotential = zeros(1,#)*
equation	radial equation	*equation = @*Equation*
SpinType	spin type	*SpinType = @*_spin*
XCfunctional	XC-functional	*XCfunctional = @XCfunctional_**
UnitCellVector	unit cell vector	*UnitCellVector = zeros(3,3)*
ElementDatabase	element database	*ElementDatabase(#) = @class_lmtoElement*
LengthOmega	Wigner–Seitz radius	*LengthOmega = [1]*
ComplexContour	complex contour	*ComplexContour(#) = @IntegralPath*
SkData	Fourier transform	*SkData = @SkBlock_**
Partition	principal layer partition	*Partition(#).SiteIndex = zeros(1,#)* *Partition(#).MatrixIndex = zeros(1,#)*
JobManager	parallel job manager	*JobManager = @manager_parajob*
JobManager_site	parallel job manager	*JobManager_site = @manager_parajob*
Quantity	physical quantities	···
Parameter	system parameters	···

(5.1)

The most important properties are *AtomSet* and *StructureConstant*. *AtomSet* is an object array of @class_cpaAtom, which is a list of atomic sites. The object @class_cpaAtom contains the potential parameters and the coherent potential of a single atomic site. *StructureConstant* is a structure array, containing the structure constant matrices for different unit cell displacements (see Fig. 3.3). The diagonal blocks from *AtomSet* and the off-diagonal matrices from *StructureConstant* constitute the LMTO Hamiltonian.

www.ebook3000.com

The class @class_cpaBulk has 34 methods which are summarized in Table 5.2.

	Method	**Description**
1	*class_cpaBulk*	CLS: class constructor
2	*get*	CLS: get value of a given variable
3	*set*	CLS: set value to a given variable
4	*initialize*	INIT: initialize the class
5	*setupParameter*	INIT: set up system parameters
6	*prepareElementData*	INIT: prepare *ElementTable*
7	*prepareAtomicData*	INIT: prepare *AtomSet* with single atom solutions
8	*updateAtomicData*	INIT: update *AtomSet* with user provided initial guess
9	*preparePartition*	INIT: prepare principal layer partition
10	*prepareStructureConstant*	INIT: prepare structure constant
11	*generateContour*	INIT: generate integral contour
12	*prepareSkData*	INIT: prepare Fourier transformed structure constant
13	*prepareMadelungPotential*	INIT: prepare Madelung constant and Madelung potential
14	*summarize*	SUM: summarize the class
15	*CPA_solution_lowX*	CPA: solve CPA equations in low concentration limit
16	*CPA_iteration*	CPA: iterate CPA equations in one step
17	*CPA_solution*	CPA: solve CPA equations to a given accuracy
18	*calculatePsi*	CAL: calculate atomic orbital
19	*calculateCDG*	CAL: calculate potential parameter
20	*calculateEM*	CAL: calculate energy moment
21	*calculateEM_eigen*	CAL: calculate energy moment with eigenvalues
22	*correctFermiEnergy*	CAL: correct Fermi energy to satisfy charge neutrality
23	*calculateRho*	CAL: calculate charge density
24	*calculateDrho*	CAL: calculate charge density derivative
25	*calculateKED*	CAL: calculate kinetic energy density
26	*calculateCharge*	CAL: calculate atomic charge
27	*calculateDipole*	CAL: calculate atomic dipole
28	*calculateE0*	CAL: calculate valence linearization center
29	*calculateV*	CAL: calculate potential
30	*makeSupercell*	MISC: make a supercell of the bulk system
31	*isClean*	MISC: check if the system only contains clean sites
32	*isLowX*	MISC: check if low concentration limit is adopted
33	*clear*	MISC: clear some memory consuming fields
34	*collectAtom*	MISC: collect atomic data after parallel calculation

(5.2)

The 34 methods can be classified into six groups: The first group CLS is composed of methods 1 to 3, which are the methods of standard class operation. The second group INIT is composed of methods 4 to 13, which are the methods used in the initialization. The third group SUM is

composed of method 14, which is the method used in the summarization. The fourth group CPA is composed of methods 15 to 17, which are the methods used in solving the CPA equations. The fifth group CAL is composed of methods 18 to 29, which are the methods used in the calculations of various physical quantities. The sixth group MISC is composed of methods 30 to 34, which are some miscellaneous supportive methods.

5.1.2 *@class_cpaAtom*

The class @class_cpaAtom has 7 properties which are summarized in Table 5.3.

Property	**Meaning**	**Data Structure**
cpaComponent	chemical species	*cpaComponent(#).lmtoAtom = @class_lmtoAtom* *cpaComponent(#).probability = [1]*
cpaData	CPA data	*cpaData.sampleE = zeros(1,#)* *cpaData.weightE = zeros(1,#)* *cpaData.tiltP_r(#).spin{#} = zeros(#,#)* *cpaData.Omega_r(#).spin{#} = zeros(#,#)* *cpaData.gr(#).spin{#} = zeros(#,#)*
MadelungPotential	Madelung potential	*MadelungPotential = [1]*
SpinType	spin type	*SpinType = @*_spin*
center	atomic center	*center = zeros(1,3)*
Lmax	maximum l	*Lmax = [1]*
JobManager	parallel job manager	*JobManager = @manager_parajob*

(5.3)

The most important properties are *cpaComponent* and *cpaData*. *cpaComponent* is a structure array containing all the chemical species of an atomic site. For example, suppose an atomic site has two chemical species Si and P with probability 0.99 and 0.01 respectively. Consequently *cpaComponent* has two array elements: *cpaComponent(1).lmtoAtom* is the Si atom and *cpaComponent(1).probability = 0.99*; *cpaComponent(2).lmtoAtom* is the P atom and *cpaComponent(2).probability = 0.01*. *cpaData* is a data structure with the fields *sampleE*, *weightE*, *tiltP_r*, *Omega_r*, and *gr*, corresponding to the variables E_n, W_n, $\tilde{P}_i^r(E_n)$, $\Omega_i^r(E_n)$, and $\overline{\mathcal{G}_i^r}(E_n)$ in the CPA equations. Here the energy integral in the CPA equations has been replaced by an E-sampling with the energy point E_n and the weight W_n.

The class @class_cpaAtom has 29 methods which are summarized in Table 5.4.

www.ebook3000.com

	Method	**Description**
1	*class_cpaAtom*	CLS: class constructor
2	*get*	CLS: get value of a given variable
3	*set*	CLS: set value to a given variable
4	*initialize*	INIT: initialize the class
5	*calcCPA_tiltP_r*	CPA: calculate $\tilde{P}_i^r(E)$ in the CPA iteration
6	*calcCPA_tiltP_r_lowX*	CPA: calculate $\tilde{P}_i^r(E)$ in low concentration limit
7	*calcCPA_Omega_r*	CPA: calculate $\Omega_i^r(E)$ in the CPA iteration
8	*calcCPA_gr*	CPA: calculate $\overline{\mathcal{G}_i^r}(E)$ in the CPA iteration
9	*getEnergyIndex*	CPA: get index of E
10	*get_CPAdata*	CPA: get CPA data from *cpaData*
11	*set_CPAdata*	CPA: set CPA data to *cpaData*
12	*get_P0_lowX*	CPA: get $P_{i0}(E)$ in low concentration limit
13	*calculatePsi*	CAL: calculate atomic orbital
14	*calculateCDG*	CAL: calculate potential parameter
15	*calculateEM*	CAL: calculate energy moment
16	*calculateRho*	CAL: calculate charge density
17	*calculateDrho*	CAL: calculate charge density derivative
18	*calculateKED*	CAL: calculate kinetic energy density
19	*calculateCharge*	CAL: calculate atomic charge
20	*calculateDipole*	CAL: calculate atomic dipole
21	*calculateE0*	CAL: calculate valence linearization center
22	*calculateV*	CAL: calculate potential
23	*ContourIntegral*	CAL: evaluate the contour integral
24	*correctEnergyMoment*	CAL: correct energy moment
25	*isClean*	MISC: check if the atomic site is a clean site
26	*clear*	MISC: clear some memory-consuming fields
27	*transferAtomicData*	MISC: transfer data between two atomic sites
28	*shiftCenter*	MISC: shift the center of the atomic site
29	*shiftEnergy*	MISC: shift the energy of the atomic site

(5.4)

The 29 methods can be classified into five groups: The first group CLS is composed of methods 1 to 3, which are the methods of standard class operation. The second group INIT is composed of method 4, which is the method used in the initialization. The third group CPA is composed of methods 5 to 12, which are the methods used in solving the CPA equations. The fourth group CAL is composed of methods 13 to 24, which are the methods used in the calculations of various physical quantities. The fifth group MISC is composed of methods 25 to 29, which are some miscellaneous supportive methods.

5.1.3 *@class_lmtoAtom*

The class @class_lmtoAtom has 20 properties which are summarized in Table 5.5.

Property	Meaning	Data Structure
AtomInfo	atomic information	···
SpinType	spin type	*SpinType = @*_spin*
XCfunctional	XC-functional	*XCfunctional = @XCfunctional_**
equation	radial equation	*equation = @*Equation*
OrbitalSet	atomic orbital set	*OrbitalSet(#).spin{#} = @class_lmtoOrbital*
rrData	radial mesh: $r_i = R(i)$	*rrData = zeros(1,#)*
drData	radial mesh: $dr_i = R'(i)$	*drData = zeros(1,#)*
vvData	atomic potential	*vvData.spin{#} = zeros(1,#)*
rhoData	charge density	*rhoData.spin{#} = zeros(1,#)*
ttData	kinetic energy density	*ttData.spin{#} = zeros(1,#)*
DrhoData	charge density derivative (1st)	*DrhoData.spin{#} = zeros(1,#)*
DDrhoData	charge density derivative (2nd)	*DDrhoData.spin{#} = zeros(1,#)*
multipole	multipoles	*multipole.monopole = [1]* *multiple.dipole = zeros(1,3)*
magmom	magnetic moment	*magmom = [1]*
EnergyMoment	energy moment	*EnergyMoment.spin{#}.m0 = zeros(#,#)* *EnergyMoment.spin{#}.m1 = zeros(#,#)* *EnergyMoment.spin{#}.m2 = zeros(#,#)*
rhoAtEf	density matrix at $E = E_f$	*rhoAtEf.spin{#} = zeros(#,#)*
MadelungPotential	Madelung potential	*MadelungPotential = [1]*
CoreIndex	core orbital index	*CoreIndex = zeros(1,#)*
ValenceIndex	valence orbital index	*ValenceIndex = Zeros(1,#)*
Parameter	atom parameters	···

(5.5)

The most important properties are *OrbitalSet*, *vvData*, *rhoData*. *OrbitalSet* is an object array containing all the linear muffin orbitals of the atom. *vvData* and *rhoData* are the spherically symmetrized atomic potential and charge density in the atom sphere. All the above quantities are spin resolved, where spin-up and spin-down components are distinguished by the subfield *spin*{#}.

The class @class_lmtoAtom has 17 methods which are summarized in Table 5.6.

	Method	Description
1	*class_lmtoAtom*	CLS: class constructor
2	*get*	CLS: get value of a given variable
3	*set*	CLS: set value to a given variable
4	*initialize*	INIT: initialize the class
5	*calculatePsi*	CAL: calculate atomic orbital
6	*calculateCDG*	CAL: calculate potential parameter
7	*calculateRho*	CAL: calculate charge density
8	*calculateDrho*	CAL: calculate charge density derivative
9	*calculateKED*	CAL: calculate kinetic energy density
10	*calculateCharge*	CAL: calculate atomic charge
11	*calculateDipole*	CAL: calculate atomic dipole
12	*calculateE0*	CAL: calculate linearization center
13	*calculateV*	CAL: calculate atomic potential
14	*calculatePLM*	CAL: calculate potential function
15	*transferAtomicData*	MISC: transfer data between two atoms
16	*shiftCenter*	MISC: shift the center of the atom
17	*shiftEnergy*	MISC: shift the energy of the atom

(5.6)

www.ebook3000.com

The 17 methods can be classified into four groups: The first group CLS is composed of methods 1 to 3, which are the methods of standard class operation. The second group INIT is composed of method 4, which is the method used in the initialization. The third group CAL is composed of methods 5 to 14, which are the methods used in the calculations of various physical quantities. The fourth group MISC is composed of methods 15 to 17, which are some miscellaneous supportive methods.

5.1.4 *@class-lmtoOrbital*

The class @class_lmtoOrbital has 10 properties which are summarized in Table 5.7.

Property	**Meaning**	**Data Structure**
rrData	radial mesh: $r_i = R(i)$	*rrData = zeros(1,#)*
drData	radial mesh: $dr_i = R'(i)$	*drData = zeros(1,#)*
psiData	radial function $\phi_{ilq}(r)$	*psiData = zeros(1,#)*
psidotData	radial function $\dot{\phi}_{ilq}(r)$	*psidotData = zeros(1,#)*
psidot2Data	radial function $\ddot{\phi}_{ilq}(r)$	*psidot2Data = zeros(1,#)*
DpsiData	radial function $\phi'_{ilq}(r)$	*DpsiData = zeros(1,#)*
DpsidotData	radial function $\dot{\phi}'_{ilq}(r)$	*DpsidotData = zeros(1,#)*
Dpsidot2Data	radial function $\ddot{\phi}'_{ilq}(r)$	*Dpsidot2Data = zeros(1,#)*
boundary	boundary value at $r = R_{iq}$	*boundary.psi = zeros(1,3)* *boundary.psidot = zeros(1,3)*
Parameter	orbital parameters	$\cdots$

(5.7)

The most important properties are *psiData*, *psidotData*, *psidot2Data*, *DpsiData*, *DpsidotData*, *Dpsidot2Data*. They are radial wave functions and derivatives solved from the radial equation.

The class @class_lmtoOrbital has 7 methods which are summarized in Table 5.8.

	Method	**Description**
1	*class_lmtoOrbital*	CLS: class constructor
2	*get*	CLS: get value of a given variable
3	*set*	CLS: set value to a given variable
4	*initialize*	INIT: initialize the class
5	*calculatePsi*	CAL: calculate atomic orbital
6	*calculateCDG*	CAL: calculate potential parameter
7	*shiftEnergy*	MISC: shift the energy of the orbital

(5.8)

The 7 methods can be classified into four groups: The first group CLS is composed of methods 1 to 3, which are the methods of standard class operation. The second group INIT is composed of method 4, which is the method used in the initialization. The third group CAL is composed of methods 5 and 6, which are the methods used in the calculations of various physical quantities. The fourth group MISC is composed of method 7, which is a miscellaneous supportive method.

One may have noticed that many methods in the dsim-classes @class_cpaBulk, @class_cpaAtom, @class_lmtoAtom, @class_lmtoOrbital share the same names. This is another example of polymorphism (see Section 4.4). In the implementation, the methods of @class_cpaBulk and @class_cpaAtom set up the loops over atomic site and chemical species, and call the same name methods of @class_lmtoAtom and @class_lmtoOrbital to carry out the real calculations.

5.2 Bulk solver

The self-consistent calculation for bulk systems is organized by the dsim-solver @SCFsolver_Bulk. @SCFsolver_Bulk has 7 methods, *SCF-solver_Bulk*, *initialize*, *solve*, *SCF_f*, *SCF_f0*, *SCF_get*, and *SCF_set*. The most important method is *solve* which solves the nonlinear equation array $X = F[X]$ defined by Eq. (4.2). In the method *solve*, the self-consistent iteration is organized by calling the methods *SCF_f0*, *SCF_f*, *SCF_get*, and *SCF_set*, and mixing algorithms are used to accelerate the convergence (see Appendix A.20).

Among the 7 methods, the first 3 methods are more or less the same as other dsim-solvers (see Section 4.6). The unique methods related to bulk systems are *SCF_f0*, *SCF_f*, *SCF_get*, and *SCF_set*. The method *SCF_f* consists of a complete self-consistent iteration. The mapping from the code to the formulas of Section 3.11 is described by the in-line comments.

```
function system = SCF_f(solver, system)
system = calculatePsi(system);                %step-4
system = calculateCDG(system);                %step-5
DMmethod = get(system, 'DMmethodScheme');
if strcmp(DMmethod, 'Green function')
    system = generateContour(system);
    if isLowX(system)                         %step-6
        system = CPA_solution_lowX(system);
```

www.ebook3000.com

```
    else
        system = CPA_iteration(system);
    end
    system = calculateEM(system);            %step-7
    system = correctFermiEnergy(system);
else
    if ~isClean(system)
        error('...')
    end
    system = calculateEM_eigen(system);      %step-6',7'
end
system = calculateRho(system);               %step-8
system = calculateDrho(system);
system = calculateKED(system);
system = calculateCharge(system);            %step-9
system = calculateDipole(system);
system = calculateE0(system);
system = calculateV(system);                 %step-10,11,12
end %SCF_f
```

Notice that for clean bulk systems the density matrix can be calculated either with Green's function or with wave function method. The formulas of Green's function method were presented in Section 3.11 and the formulas of wave function method are presented in Appendix A.7.

The method *SCF_f0* consists of an incomplete self-consistent iteration. This method is useful at the beginning of a self-consistent calculation where an ignition is needed. Essentially *SCF_f0* solves the atomic orbitals, calculates potential parameters, and solves the CPA equations to obtain a reasonable initial guess of the coherent potential. Notice that it is unnecessary to obtain an initial guess of the coherent potential in clean systems or in the low concentration limit.

```
function system = SCF_f0(solver, system)
if isClean(system) | isLowX(system)
    return
end
print(class_log, 'pre-calculating CPA...\n')
system = calculatePsi(system);
system = calculateCDG(system);
system = generateContour(system);
```

```
ControlParameter.mixer.Omega_r = Mixer_DoNothing;
ControlParameter.maxstep = 60;
ControlParameter.tolerance.Omega_r = 1e-4;
system = CPA_solution(system, ControlParameter);
print(class_log, '\n')
end %SCF_f0
```

The method *SCF_get* (*SCF_set*) gets (sets) X from (to) an object of @class_cpaBulk. In bulk systems, X is composed of the variables $\{V_{iq}(r)\}$, $\left\{E^0_{ilq}\right\}$, $\{\Omega^r_i\}$, and μ, as listed in Table (4.3).

5.3 Structure constant

This section discusses the algorithms for calculating the structure constant. The structure constant calculations proceed in three steps: (1) Calculate the canonical structure constant $\tilde{S}$ by using Eq. (3.91); (2) Calculate the screened structure constant S by using Eq. (3.93); (3) Calculate the Fourier transformed structure constant by using Eq. (3.95).

The key in the calculation of the canonical structure constant $\tilde{S}$ is to evaluate the real-valued spherical harmonics $Y(\Omega)$ and the corresponding Gaunt coefficients $C_{LL_1L_2}$ defined by Eq. (3.92). To avoid any ambiguity in the convention, both the complex-valued and the real-valued spherical harmonics are explicitly defined in Appendix A.10. The Gaunt coefficients of the complex-valued spherical harmonics $\tilde{C}_{LL_1L_2}$ can be evaluated with the aid of Wigner-3j symbols. Since the real-valued spherical harmonics $Y(\Omega)$ are related to the complex-valued spherical harmonics $\tilde{Y}(\Omega)$ by Eq. (3.7), the Gaunt coefficients $C_{LL_1L_2}$ can be evaluated by a linear combination of $\tilde{C}_{LL_1L_2}$. The formulas for calculating $\tilde{C}_{LL_1L_2}$ and $C_{LL_1L_2}$ are presented in Appendix A.11.

Two numerical tricks are worth mentioning in the calculation of $\tilde{S}$. The first trick is to make a lookup table for the coefficients. Eq. (3.91) can be rewritten as

$$S_{i_1L_1,i_2L_2} = \sum_{m=-l}^{l} A_{LL_1L_2} K_L\left(\mathbf{r}_{i_2} - \mathbf{r}_{i_1}\right), \tag{5.9}$$

where $l = l_1 + l_2$ and the coefficient $A_{LL_1L_2}$ is defined by

$$A_{LL_1L_2} \equiv (-1)^{l_2+1} \frac{8\pi\,(2l-1)\,!!}{(2l_1-1)\,!!\,(2l_2-1)\,!!} C_{LL_1L_2}.$$

www.ebook3000.com

Notice that in atomic simulations the angular momentum quantum number l_1 and l_2 are no greater than 3. So one can pre-calculate all the coefficients $A_{LL_1L_2}$ and save them to a data file. The saved data can be loaded to a *persistent* variable of a MATLAB function. The loading work is done only once at the first call of the function, and the value of the persistent variable will be available for the subsequent calls. The advantage of the lookup table is that the complexity of the coefficient calculation is isolated and hidden. One does not have to worry about the computational cost of generating the lookup table.

The other trick is to use *cache* to recycle the calculated data. Notice that the matrix block $S_{i_1 i_2}$ only depends on $\mathbf{r}_{i_2} - \mathbf{r}_{i_1}$ and a considerable portion of matrix blocks is identical due to the crystal structure. It is a big waste to do repeated calculations and discard the data right away. A better strategy is to recycle the calculated data by using a cache. Before a new calculation, we first check if the data is available in the cache. If so we simply pick it up from the cache. Otherwise we carry out the calculation and store the calculated data in the cache. The cache keeps growing until its size exceeds a pre-defined limit. In that situation, the oldest data is removed from the cache to make room for new data. Research shows that a cache size of 32 MB is good for most applications [1].

Having calculated the canonical structure constant $\tilde{S}$, we proceed to solve the screened structure constant S from

$$S = \left(\tilde{S}^{-1} - \alpha\right)^{-1}, \tag{5.10}$$

where α is a diagonal matrix defined by Eq. (3.53). Since $\tilde{S}$ is an infinitely large matrix, a direct inverse is impossible. The equation has to be solved approximately by using the property that the elements of S decay exponentially with the distance between two atomic sites, see Fig. 3.2. Eq. (5.10) can be rewritten as

$$\left[\frac{1}{\alpha} - \tilde{S}\right] X = \tilde{S}, \tag{5.11}$$

where X is defined by $X \equiv \alpha S$. Due to the decaying property, X_{ij} is negligible if $|\mathbf{r}_i - \mathbf{r}_j|$ is larger than a cutoff radius. Define the cutoff radius by $R_{cut} \equiv \lambda\omega$ where λ is a dimensionless parameter and ω is the Wigner–Seitz radius. It is assumed that $X_{ij} \approx 0$ for $|\mathbf{r}_i - \mathbf{r}_j| > R_{cut}$, which is called cutoff approximation.

For a fixed site j, a finite number of atomic sites are located within the cutoff radius (see Fig. 5.1). Those atomic sites are said to be in the

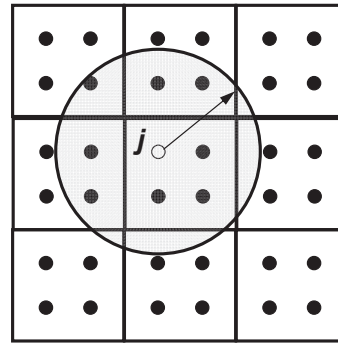


Fig. 5.1 A schematic diagram to illustrate the cutoff approximation in calculating the screened structure constant. The shaded circle indicates the neighborhood of site-j. All other sites outside the neighborhood will be neglected in solving Eq. (5.12).

neighborhood of site j. Inserting site indices to Eq. (5.11), one obtains

$$\sum_k \left[\frac{1}{\alpha} - \tilde{S}\right]_{ik} X_{kj} = \tilde{S}_{ij}. \tag{5.12}$$

Due to the cutoff approximation, site k must be in the neighborhood of site j. Let site i be also in the neighborhood of site j, one obtains a complete set of linear equations to solve the vector $X_{:,j}$ (The meaning of colon notation is similar to that of MATLAB). Let j go over all the atomic sites of a bulk unit cell, one can obtain the complete solution of X. Once X is available, it is straightforward to solve S from X

$$S = \frac{1}{\alpha} X.$$

By definition the structure constant matrix is Hermitian, namely, $S = S^\dagger$. Because of the cutoff approximation, the Hermitian property is slightly violated. To restore the Hermitian property, one needs to symmetrize S by using

$$S \to \frac{1}{2}\left(S + S^\dagger\right). \tag{5.13}$$

Especially in bulk systems the Hermitian property is reduced to $S_I = (S_{-I})^\dagger$ where I is the unit cell displacement. Consequently the symmetrization is reduced to

$$S_I \to \frac{1}{2}\left(S_I + S_{-I}^\dagger\right). \tag{5.14}$$

www.ebook3000.com

Finally, we calculate the Fourier transform of S

$$S(k) = \sum_I e^{-\mathrm{i}kI} S_I. \tag{5.15}$$

Since the structure constant does not contain any atomic information, the calculation can be done outside the self-consistent loop. The calculated $S(k)$ data can be stored either in the memory or in a temporary file. If the size of the bulk system or the number of k-points is too large, one can also calculate $S(k)$ on the fly and discard the data right after the usage. The three schemes of handling $S(k)$ data are implemented in *@SkBlock_memo*, *@SkBlock_hdvp*, and *@SkBlock_calc* of *Library09*.

5.4 Ewald sum technique

This section discusses the algorithms for calculating the Madelung potential in bulk systems. Due to the periodicity in bulk systems, the Madelung potential can be simplified as

$$\begin{aligned} V_{MD} &= \sum_{j\neq i} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{\mathbf{P}_j \cdot (\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3} \\ &= \sum_{j\neq i} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j|} - \mathbf{P}_j \cdot \nabla \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \\ &= \sum_{j\in\Omega} \sum_{\mathbf{T}} \frac{Q_j}{|\mathbf{r}_i - \mathbf{r}_j - \mathbf{T}|} - \mathbf{P}_j \cdot \nabla \frac{1}{|\mathbf{r}_i - \mathbf{r}_j - \mathbf{T}|} \\ &= \sum_{j\in\Omega} Q_j \phi(\mathbf{r}_i - \mathbf{r}_j) - \mathbf{P}_j \cdot \nabla \phi(\mathbf{r}_i - \mathbf{r}_j), \end{aligned} \tag{5.16}$$

where $j \in \Omega$ means that the summation is over the atomic sites of a unit cell. Notice that $\sum_{j\in\Omega} Q_j = 0$ due to the charge neutrality. $\mathbf{T}$ is the lattice vector defined by $\mathbf{T} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3$ where $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ are unit cell vectors and n_1, n_2, n_3 are integers. To exclude the self-interaction, it is required that $\mathbf{T} \neq 0$ for $\mathbf{r}_i = \mathbf{r}_j$. The function $\phi(\mathbf{r})$ is defined by

$$\phi(\mathbf{r}) = \sum_{\mathbf{T}} \frac{1}{|\mathbf{r} - \mathbf{T}|}, \tag{5.17}$$

where $\mathbf{T} \neq 0$ for $\mathbf{r} = 0$. Therefore the calculation of the Madelung potential is reduced to the calculation of $\phi(\mathbf{r})$ which is referred to as the Madelung constant. The physical meaning of the Madelung constant is the Coulomb

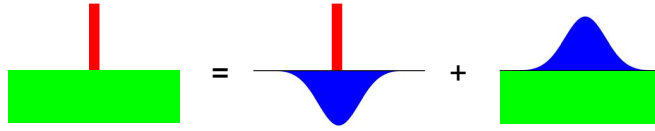


Fig. 5.2 Sketch of the idea for the Ewald sum technique. A point charge (red) with uniform background charge (green) is enclosed by a Gaussian packet (blue). The potential of the enclosed point charge will be calculated in the real space, and the potential of the neutralized Gaussian packet will be calculated in the reciprocal space.

potential generated by unit point charges located on a lattice. Consequently $\phi(\mathbf{r})$ satisfies the Poisson equation

$$\nabla^2\phi(\mathbf{r}) = -4\pi\sum_{\mathbf{T}}\delta(\mathbf{r}-\mathbf{T}). \tag{5.18}$$

A straightforward method of calculating $\phi(\mathbf{r})$ is to do the summation of Eq. (5.17) term by term. The series, however, diverges due to the lack of charge neutrality. To avoid the divergence, one may screen the point charge by a uniform jellium with an opposite charge. Because $\sum_{j\in\Omega} Q_j = 0$, jellium of different Q_j cancels each other and has no net effect on the Madelung potential. Consequently Eq. (5.17) and Eq. (5.18) are modified to

$$\phi(\mathbf{r}) = \sum_{\mathbf{T}}\left[\frac{1}{|\mathbf{r}-\mathbf{T}|} - V_\Omega(\mathbf{r}-\mathbf{T})\right], \tag{5.19}$$

$$\nabla^2\phi(\mathbf{r}) = -4\pi\sum_{\mathbf{T}}\left[\delta(\mathbf{r}-\mathbf{T}) - \frac{1}{\Omega}\right], \tag{5.20}$$

where Ω is the unit cell volume and $V_\Omega(\mathbf{r})$ is the Coulomb potential generated by the jellium unit cell. After the screening, the series does converge but the convergence rate is still rather slow. We need to find an algorithm to calculate the series efficiently, just like C. F. Gauss did for the arithmetic series.

The algorithm is called Ewald sum technique [2]. The key idea is to enclose the point charge by a Gaussian packet as illustrated in Fig. 5.2. The screened charge density $\sum_{\mathbf{T}}\left[\delta(\mathbf{r}-\mathbf{T}) - \frac{1}{\Omega}\right]$ is replaced by $\rho_1(\mathbf{r}) + \rho_2(\mathbf{r})$ where

$$\rho_1(\mathbf{r}) = \sum_{\mathbf{T}}\left[\delta(\mathbf{r}-\mathbf{T}) - \rho_\sigma(\mathbf{r}-\mathbf{T})\right], \tag{5.21}$$

$$\rho_2(\mathbf{r}) = \sum_{\mathbf{T}}\left[\rho_\sigma(\mathbf{r}-\mathbf{T}) - \frac{1}{\Omega}\right]. \tag{5.22}$$

www.ebook3000.com

Here $\rho_\sigma(\mathbf{r})$ is a Gaussian packet

$$\rho_\sigma(\mathbf{r}) \equiv \sigma^3 \pi^{-\frac{3}{2}} e^{-\sigma^2 r^2},$$

which has been normalized to $\int d^3 r \rho_\sigma(\mathbf{r}) = 1$. Notice that the Fourier transform of a Gaussian packet is still a Gaussian packet which converges quickly in both real space and reciprocal space. The potential of $\rho_1(\mathbf{r})$ will be calculated in real space and the potential of $\rho_2(\mathbf{r})$ will be calculated in reciprocal space.

The potential of $\rho_1(\mathbf{r})$ is a summation of the contribution from each screened point charge. The potential of a single screened point charge can be calculated using the Gauss theorem

$$\begin{aligned}\phi_s(\mathbf{r}) &= \frac{1}{r} - \left[\frac{1}{r}\int_0^r 4\pi s^2 \rho_\sigma(s)\, ds + \int_r^\infty \frac{1}{s} 4\pi s^2 \rho_\sigma(s)\, ds\right] \\ &= \frac{1}{r}\mathrm{erfc}(\sigma r),\end{aligned}$$

where $\mathrm{erfc}(x)$ is the complementary error function defined by

$$\mathrm{erfc}(x) \equiv \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.$$

As a result the potential of $\rho_1(\mathbf{r})$ is obtained as

$$\phi_1(\mathbf{r}) = \sum_{\mathbf{T}} \phi_s(\mathbf{r} - \mathbf{T}) = \frac{2}{\sqrt{\pi}} \sum_{\mathbf{T}} \frac{1}{|\mathbf{r} - \mathbf{T}|} \mathrm{erfc}(\sigma |\mathbf{r} - \mathbf{T}|). \tag{5.23}$$

Since $\mathrm{erfc}(x) \approx \frac{1}{\sqrt{\pi}} \frac{1}{x} e^{-x^2}$ for $x \gg 1$, the series in Eq. (5.23) decays extremely quickly with $|\mathbf{r} - \mathbf{T}|$.

The potential of $\rho_2(\mathbf{r})$ is solved from the Poisson equation

$$\nabla^2 \phi_2(\mathbf{r}) = -4\pi \rho_2(\mathbf{r}),$$

whose solution can be obtained by using the Fourier transform

$$\phi_2(\mathbf{r}) = 4\pi \sum_{\mathbf{K} \neq \mathbf{0}} \rho_2(\mathbf{K}) \frac{e^{i\mathbf{K}\cdot\mathbf{r}}}{K^2}. \tag{5.24}$$

Here $\mathbf{K}$ is the reciprocal lattice vector defined by $\mathbf{K} = m_1 \mathbf{b}_1 + m_2 \mathbf{b}_2 + m_3 \mathbf{b}_3$ where $\mathbf{a}_i \cdot \mathbf{b}_j = 2\pi \delta_{ij}$ and m_i is an integer. The $\mathbf{K} = \mathbf{0}$ term is excluded in Eq. (5.24) because the background charge of $\rho_2(\mathbf{r})$ is zero and hence $\rho_2(\mathbf{K} = \mathbf{0}) = 0$. $\rho_2(\mathbf{K})$ is the Fourier transform of $\rho_2(\mathbf{r})$ which can be

evaluated analytically

$$\begin{aligned}\rho_2(\mathbf{K}) &= \frac{1}{\Omega}\int_\Omega d^3r\rho_2(\mathbf{r})\,e^{-i\mathbf{K}\cdot\mathbf{r}} \\ &= \frac{1}{\Omega}\int_\Omega d^3r\left\{\sum_{\mathbf{T}}\left[\rho_\sigma(\mathbf{r}-\mathbf{T})-\frac{1}{\Omega}\right]\right\}e^{-i\mathbf{K}\cdot\mathbf{r}} \\ &= \frac{1}{\Omega}\int d^3r\left[\rho_\sigma(\mathbf{r})-\frac{1}{\Omega}\right]e^{-i\mathbf{K}\cdot\mathbf{r}} \\ &= \frac{1}{\Omega}\exp\left[-\frac{K^2}{4\sigma^2}\right]-\frac{1}{\Omega}\delta_K. \end{aligned} \tag{5.25}$$

Substituting Eq. (5.25) into Eq. (5.24), $\phi_2(\mathbf{r})$ is obtained as

$$\begin{aligned}\phi_2(\mathbf{r}) &= \frac{4\pi}{\Omega}\sum_{\mathbf{K}\neq\mathbf{0}}\frac{e^{i\mathbf{K}\cdot\mathbf{r}}}{K^2}\exp\left[-\frac{K^2}{4\sigma^2}\right] \\ &= \frac{4\pi}{\Omega}\sum_{\mathbf{K}\neq\mathbf{0}}\frac{\cos(\mathbf{K}\cdot\mathbf{r})}{K^2}\exp\left[-\frac{K^2}{4\sigma^2}\right], \end{aligned} \tag{5.26}$$

where the imaginary part is canceled between $\mathbf{K}$ and $-\mathbf{K}$. Each $\mathbf{K}$ term is proportional to $\exp\left[-\frac{K^2}{4\sigma^2}\right]$ and decays extremely quickly with $\frac{K}{2\sigma}$.

Finally the solution to Eq. (5.20) is derived as $\phi_1(\mathbf{r})+\phi_2(\mathbf{r})+C$, where C is an arbitrary constant. To make the solution independent on the choice of σ, let $\partial_\sigma\left[\phi_1(\mathbf{r})+\phi_2(\mathbf{r})+C\right]=0$, and C is derived as $-\frac{\pi}{\Omega}\frac{1}{\sigma^2}$. As a result, $\phi(\mathbf{r})$ is reduced to

$$\begin{aligned}\phi(\mathbf{r}) = &\sum_{\mathbf{T}}\frac{1}{|\mathbf{r}-\mathbf{T}|}\mathrm{erfc}(\sigma|\mathbf{r}-\mathbf{T}|) \\ &+\frac{4\pi}{\Omega}\sum_{\mathbf{K}\neq\mathbf{0}}\frac{\cos\mathbf{K}\cdot\mathbf{r}}{K^2}\exp\left[-\frac{K^2}{4\sigma^2}\right] \\ &-\frac{\pi}{\Omega}\frac{1}{\sigma^2}. \end{aligned} \tag{5.27}$$

At $\mathbf{r}=0$, the $\mathbf{T}=\mathbf{0}$ term needs to be eliminated to exclude the self-interaction

$$\phi(\mathbf{r}=\mathbf{0}) = \lim_{\mathbf{r}\longrightarrow 0}\left[\phi(\mathbf{r})-\frac{1}{r}\right].$$

By using the limit

$$\lim_{\mathbf{r}\longrightarrow 0}\frac{1}{|\mathbf{r}|}\mathrm{erfc}(\sigma|\mathbf{r}|)-\frac{1}{r} = -\frac{2}{\sqrt{\pi}}\sigma,$$

www.ebook3000.com

one obtains

$$\begin{aligned}\phi(\mathbf{r}=\mathbf{0}) = &\sum_{\mathbf{T}\neq\mathbf{0}} \frac{1}{|\mathbf{T}|}\operatorname{erfc}(\sigma|\mathbf{T}|) - \frac{2}{\sqrt{\pi}}\sigma \\ &+\frac{4\pi}{\Omega}\sum_{\mathbf{K}\neq\mathbf{0}} \frac{1}{K^2}\exp\left[-\frac{K^2}{4\sigma^2}\right] \\ &-\frac{\pi}{\Omega}\frac{1}{\sigma^2}. \end{aligned} \tag{5.28}$$

The gradient of $\phi(\mathbf{r})$ can be evaluated as

$$\begin{aligned}-\nabla\phi(\mathbf{r}) = &\sum_{\mathbf{T}} \frac{\mathbf{r}-\mathbf{T}}{|\mathbf{r}-\mathbf{T}|^3}\left[\operatorname{erfc}(\sigma|\mathbf{r}-\mathbf{T}|) + \frac{2}{\sqrt{\pi}}e^{-\sigma^2|\mathbf{r}-\mathbf{T}|^2}\sigma|\mathbf{r}-\mathbf{T}|\right] \\ &+\frac{4\pi}{\Omega}\sum_{\mathbf{K}\neq\mathbf{0}} \mathbf{K}\frac{\sin\mathbf{K}\cdot\mathbf{r}}{K^2}\exp\left[-\frac{K^2}{4\sigma^2}\right], \end{aligned} \tag{5.29}$$

and

$$-\nabla\phi(\mathbf{r}=\mathbf{0}) = 0. \tag{5.30}$$

Eqs. (5.27,5.28,5.29,5.30) are the central results of this section.

To implement the formulas, one needs to find an optimal value of σ and determine the summation limits of $\mathbf{T}$ and $\mathbf{K}$. Notice that $\mathbf{T}$ terms decay according to $\exp\left(-\sigma^2T^2\right)$ and $\mathbf{K}$ terms decay according to $\exp\left(-\frac{K^2}{4\sigma^2}\right)$. If σ is too small, there will be too many $\mathbf{T}$ terms; If σ is too large, there will be too many $\mathbf{K}$ terms. The optimal choice of σ is to make the numbers of $\mathbf{T}$ terms and $\mathbf{K}$ terms balanced. Assume that the summation limits of $\mathbf{T}$ and $\mathbf{K}$ are determined by $|\mathbf{T}| \leqslant T_{\max}$ and $|\mathbf{K}| \leqslant K_{\max}$ respectively. Let the minimum $\mathbf{T}$ term and $\mathbf{K}$ term decay to a given tolerance ε and require that the number of $\mathbf{T}$ terms equals that of $\mathbf{K}$ terms

$$\begin{aligned}\exp\left(-\frac{K_{\max}^2}{4\sigma^2}\right) &= \exp\left(-\sigma^2T_{\max}^2\right) = \varepsilon, \\ \frac{\frac{4\pi}{3}T_{\max}^3}{\Omega} &= \frac{\frac{4\pi}{3}K_{\max}^3}{(2\pi)^3/\Omega},\end{aligned}$$

one obtains

$$\sigma = \sqrt{\pi}\Omega^{-\frac{1}{3}}, \tag{5.31}$$

$$T_{\max} = N\frac{1}{\sigma}, \tag{5.32}$$

$$K_{\max} = 2N\sigma, \tag{5.33}$$

where $N = \sqrt{-\ln\varepsilon}$ is an accuracy control parameter.

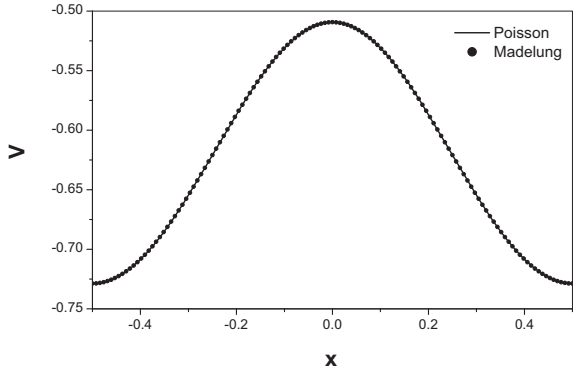


Fig. 5.3 Comparison of electrostatic potential of the target problem defined in the text. The potential is plotted along the line $y = z = \frac{1}{2}$ and $-\frac{1}{2} < x < \frac{1}{2}$. The solid line is for the solution of Poisson equation with 129^3 real-space grid and the circle is for the solution of Madelung potential. The two solutions agree with each other to a precision of 10^{-8}.

As a verification of the implementation, we solve an electrostatic problem with two different methods. The first method is to calculate the Madelung potential with Ewald sum technique. The second method is to solve the Poisson equation with fast Fourier transform (see Appendix A.23). The electrostatic problem is defined as follows: Charge spheres are located on a simple cubic lattice. The charge density of each sphere is spherically symmetric

$$\rho_0(r) = \begin{cases} \frac{\pi^2}{\pi^2-6}\frac{1}{\frac{4\pi}{3}R^3}\left(1+\cos\frac{r}{R}\pi\right) & r < R \\ 0 & r > R \end{cases},$$

where $\rho_0(r)$ is normalized to $\int_0^R \rho_0(r)\, d^3r = 1$. The origin of the lattice is $(0,0,0)$ and the lattice constant is $a = 1$. The radius of charge spheres is chosen as $R = \frac{1}{3}$. Since the line of $y = z = \frac{1}{2}$ does not touch any charge sphere, the potential on the line should be exactly the same for charge spheres and point charges. The potential of charge spheres can be solved with the Poisson equation and the potential of point charges can be calculated with the Madelung potential. In Fig. 5.3, the solutions of the two methods agree with each other to high precision, verifying the results of this section.

www.ebook3000.com

5.5 Radial equation

This section discusses the algorithms for solving the radial equation in an atomic sphere. To solve the radial equation, the first step is to rewrite the second order differential equation into a first order differential equation. The radial Schrödinger equation (3.102) can be rewritten as

$$\frac{d}{dr}\begin{pmatrix}\chi(r)\\ \psi(r)\end{pmatrix}=\begin{pmatrix}0 & 1\\ 2\left[U_{eff}(r)-E\right] & 0\end{pmatrix}\begin{pmatrix}\chi(r)\\ \psi(r)\end{pmatrix}, \tag{5.34}$$

where $\psi(r)$ is an auxiliary variable defined by $\psi(r)\equiv\chi'(r)$ and $U_{eff}(r)$ is the effective potential defined by $U_{eff}(r)\equiv V(r)+\frac{l(l+1)}{2r^2}$ (here the index iq is omitted for simplicity). Similarly, the radial scalar relativistic equation (3.103) can be rewritten as

$$\frac{d}{dr}\begin{pmatrix}\chi(r)\\ \xi(r)\end{pmatrix}=\begin{pmatrix}\frac{1}{r} & 2M(r)\\ W_{eff}(r)-E & -\frac{1}{r}\end{pmatrix}\begin{pmatrix}\chi(r)\\ \xi(r)\end{pmatrix}, \tag{5.35}$$

where $\xi(r)$ is an auxiliary variable defined by $\xi(r)\equiv\frac{1}{2M(r)}\left[\chi'(r)-\frac{1}{r}\chi(r)\right]$ and $W_{eff}(r)$ is the effective potential defined by $W_{eff}(r)\equiv V(r)+\frac{l(l+1)}{2M(r)r^2}$. The dimensionless coefficient $M(r)$ is defined by $M(r)\equiv 1-\frac{\alpha_c^2}{2}\left[V(r)-E\right]$ where $\alpha_c\approx\frac{1}{137.036}$.

Assume that the atomic potential $V(r)$ has the properties

$$V(r\to 0)=\frac{-Z}{r},$$
$$V(r\to\infty)=0.$$

The asymptotic physical solution of $\chi(r)$ and $\psi(r)$ to Eq. (5.34) is obtained as

$$\chi(r\to 0)=r^{l+1}\left(1-\frac{Z}{l+1}r\right), \tag{5.36}$$

$$\psi(r\to 0)=r^{l}\left[(l+1)-\frac{Z}{l+1}(l+2)\,r\right], \tag{5.37}$$

$$\chi(r\to\infty)=e^{-\gamma r}, \tag{5.38}$$

$$\psi(r\to\infty)=-\gamma e^{-\gamma r}, \tag{5.39}$$

where the exponent γ is defined by $\gamma\equiv\sqrt{2\left[V_{eff}(r)-E\right]}$. The asymptotic physical solution of $\chi(r)$ and $\xi(r)$ to Eq. (5.35) is obtained as

$$\chi(r\to 0)=r^{\beta}, \tag{5.40}$$

$$\xi(r\to 0)=r^{\beta}\frac{\beta-1}{\alpha_c^2 Z}, \tag{5.41}$$

$$\chi(r\to\infty)=e^{-\nu r}, \tag{5.42}$$

$$\xi(r\to\infty)=e^{-\nu r}\frac{-1}{2M(r)}\left(\nu+\frac{1}{r}\right), \tag{5.43}$$

where the exponents β and ν are defined by

$$\beta \equiv \sqrt{l(l+1) - \alpha_c^2 Z^2 + 1},$$
$$\nu \equiv \sqrt{2\left[\frac{l\,(l+1)}{2r^2} + M\,(r)\,(V\,(r) - E)\right]}.$$

Eq. (5.34) and Eq. (5.35) belong to the category of first order differential equations

$$\frac{dY}{dr} = F\,(r, Y)\,, \tag{5.44}$$

where r is a scalar and Y and F are vectors. The solution to Eq. (5.44) is determined not only by the equation itself but also by the boundary condition. Two types of boundary conditions are used in the LMTO method. The first type of boundary condition defines an initial value problem. For valence orbitals, the energy E and the value of $\chi\,(r \to 0)$ are known, and the goal is to calculate $\chi\,(r)$ in the range $r \in (0, R)$ where R is the atomic sphere radius. The second type of boundary condition defines a two-point boundary value problem. For core orbitals, the value or derivative of $\chi\,(r)$ at $r = 0$ and $r = R$ are known, the goal is to solve the energy E and $\chi\,(r)$ in the range $r \in (0, R)$. In addition, $\chi\,(r)$ satisfies the normalization

$$\int_0^R \chi^2\,(r) = 1, \tag{5.45}$$

which uniquely determines the solution.

For the first type of boundary condition, one can integrate Eq. (5.44) from $r = 0$ to $r = R$ numerically. To proceed one can discretize the radial coordinate from 0 to R by a uniform radial mesh $\{r_1, r_2, \cdots, r_n\}$ where $r_1 = \varepsilon$ and $r_n = R$ (ε is a small positive number, e.g., $\varepsilon = 2 \times 10^{-5}$). At $r_1 = \varepsilon$, the solution can be approximated by the asymptotic expressions Eqs. (5.36,5.37) or Eqs. (5.40,5.41). At $r_2 = r_1 + \Delta r$, the solution can be approximated by

$$\begin{aligned} Y\,(r_2) &\approx Y\,(r_1) + \frac{dY}{dr}\Delta r \\ &\approx Y\,(r_1) + F\,(r_1, Y_1)\,\Delta r. \end{aligned} \tag{5.46}$$

Repeating this procedure one can obtain the numerical solution all the way to $r_n = R$. The algorithm described in Eq. (5.46) is called Euler's method. As long as Δr is sufficient small, the obtained numerical solution is a good approximation to the exact solution.

The above simple algorithm can be improved from two aspects. The first aspect is the integration method. Euler's method Eq. (5.46) is accurate

www.ebook3000.com

up to the first order of Δr. More advanced and sophisticated algorithms can achieve an accuracy to higher order of Δr. Popular algorithms such as the Runge-Kutta method and the predictor-corrector method are well explained in Ref. [8]. The second aspect is the radial mesh. Notice that Eq. (5.34) and Eq. (5.35) are singular at $r = 0$ and the solutions have rapid oscillations in the vicinity of $r = 0$. To capture the oscillations, one needs to put more radial points in the core region. Therefore the uniform radial mesh needs to be replaced by a logarithmic radial mesh

$$\begin{aligned} r_k &= r(k), \\ \Delta r_k &= r'(k), \end{aligned}$$

where $r(k)$ is defined by

$$r(k) \equiv cke^{\alpha k}, \tag{5.47}$$

and the constant c and α are determined by the constraints $r_1 = \varepsilon$ and $r_n = R$. Consequently the integral over r can be evaluated on the radial mesh by

$$\int_0^R f(r)\, dr \approx \sum_{k=1}^{n} f(r_k)\, \Delta r_k. \tag{5.48}$$

For the second type of boundary condition, the energy E is unknown and one has to find a proper E such that $\chi(r)$ satisfies the boundary condition at both ends. Essentially the algorithm is to carry out a calculation with the first type of boundary condition from one end, and tune the value of E until the trajectory of $\chi(r)$ shoots the target at the other end. For this reason, the algorithm is called the shooting method. In the LMTO method, we have two different second type boundary conditions: (1) logarithmic derivative boundary condition

$$\begin{cases} \chi(r \to 0) = 0 \\ \left.\frac{\chi'(r)}{\chi(r)}\right|_{r=R} = K \end{cases}; \tag{5.49}$$

(2) bound state boundary condition

$$\begin{cases} \chi(r \to 0) = 0 \\ \chi(r \to \infty) = 0 \end{cases}. \tag{5.50}$$

As a result, the details of the shooting algorithm is slightly different for the two kinds.

For the logarithmic derivative boundary condition, one can integrate Eq. (5.34) or Eq. (5.35) from $r = 0$ to $r = R$ for a trial energy E and check

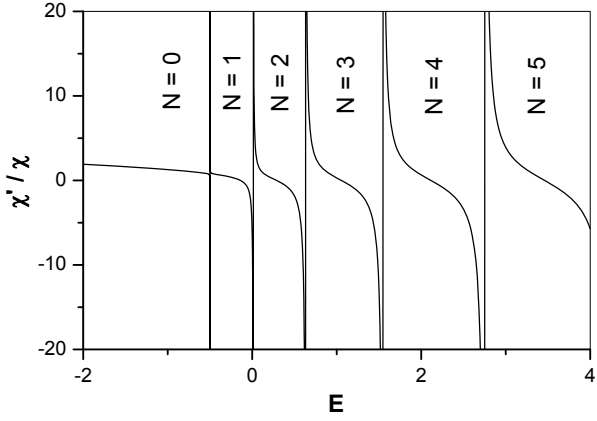


Fig. 5.4 The energy dependence of the logarithmic derivative $\frac{\chi'(r)}{\chi(r)}$ of the hydrogen atom at $r = 6$ for $l = 0$. The logarithmic derivative has several branches. Within each branch, $\chi(r)$ has a definite number of nodes and $\frac{\chi'(r)}{\chi(r)}$ decreases monotonously. Although the curve is calculated for the hydrogen atom, the cotangent-like behavior is a general feature of logarithmic derivative.

the logarithmic derivative at the boundary. If $\frac{\chi'(R)}{\chi(R)}$ is not equal to K, one needs to tune E to satisfy the boundary condition. Notice that the energy dependence of the logarithmic derivative exhibits a cotangent-like behavior: $\frac{\chi'(R)}{\chi(R)}$ as a function of E has several branches within which $\frac{\chi'(R)}{\chi(R)}$ decreases monotonously and $\chi(r)$ has a definite node number (see Fig. 5.4). Given K and the node number, the bound state energy E can be determined by using a bisection algorithm.

For the bound state boundary condition, one can integrate Eq. (5.34) or Eq. (5.35) from $r = 0$ outward and from $r = \infty$ inward with the asymptotic expressions. The two solutions $\chi_1(r)$ and $\chi_2(r)$ will meet at the classical turning point $r = r_c$ where r_c is defined by [3]

$$U_{eff}(r_c) = E, \tag{5.51}$$

or

$$W_{eff}(r_c) = E. \tag{5.52}$$

It is required that $\chi_1(r)$ and $\chi_2(r)$ match smoothly at $r = r_c$, i.e., both function values and the first derivatives are equal to each other. Notice that the logarithmic derivative $\frac{\chi_1'}{\chi_1}$ decreases monotonously with E [4] and $\frac{\chi_2'}{\chi_2}$ increases monotonously with E [5]. The bound state energy E

www.ebook3000.com

can be determined by using a bisection algorithm so that $\frac{\chi_1'}{\chi_1} = \frac{\chi_2'}{\chi_2}$ at $r = r_c$. An even faster algorithm is to estimate the correction of E to fix up the discrepancy between the two solutions. Suppose $\chi_1(r)$ and $\chi_2(r)$ are normalized such that $\chi_1(r_c) = \chi_2(r_c) \equiv \chi_0$ and $\chi_1'(r_c) \neq \chi_2'(r_c)$. Define $\tilde{\chi}(r)$ by

$$\tilde{\chi}(r) \equiv \begin{cases} \chi_1(r) & r < r_c \\ \chi_2(r) & r > r_c \end{cases},$$

which is a solution of Eq. (5.34) or Eq. (5.35) for both $r < r_c$ and $r > r_c$. The discontinuity of the derivative at $r = r_c$ indicates that $\tilde{\chi}(r)$ is a solution of a smooth potential plus a δ-barrier $\gamma\delta(r - r_c)$. Integrating the radial equation, the strength of the δ-barrier is obtained as

$$\gamma = \frac{\chi_2'(r_c) - \chi_1'(r_c)}{2\chi_0}.$$

To eliminate the discontinuity, one needs to add a counter δ-barrier $-\gamma\delta(r - r_c)$ to the potential. By using the perturbation theory, the correction δE is obtained as

$$\delta E = \gamma\chi_0^2. \tag{5.53}$$

As expected δE goes to zero when $\chi_1'(r_c) = \chi_2'(r_c)$.

So far we have discussed the algorithms for solving Eq. (5.34) or Eq. (5.35) for different types of boundary condition. In Eq. (3.104), one not only needs the radial wave function $\phi(r)$ but also its energy derivatives $\dot{\phi}(r)$ and $\ddot{\phi}(r)$ for the valence orbitals. In principle, one can calculate $\phi(r)$ at both E and $E \pm \Delta E$ and evaluate the energy derivatives numerically. An even better algorithm is to derive the radial equations of $\dot{\chi}(r)$ and $\ddot{\chi}(r)$ and integrate them with the same algorithm for solving $\chi(r)$.

Applying the energy derivative to Eq. (3.102) and Eq. (3.103), one obtains the radial equations

$$\hat{O}\chi(r) = 0, \tag{5.54}$$

$$\hat{O}\dot{\chi}(r) = F_1(r), \tag{5.55}$$

$$\hat{O}\ddot{\chi}(r) = F_2(r), \tag{5.56}$$

where the operator $\hat{O}$ and the inhomogeneous terms $F_1(r)$ and $F_2(r)$ are

$$\hat{O} = -\frac{1}{2}\partial_r^2 + \frac{l(l+1)}{2r^2} + [V(r) - E], \tag{5.57}$$

$$F_1(r) = \chi(r), \tag{5.58}$$

$$F_2(r) = 2\dot{\chi}(r). \tag{5.59}$$

and

$$\hat{O} = -\frac{1}{2}\partial_r^2 + \frac{l\left(l+1\right)}{2r^2} + M\left(r\right)\left[V\left(r\right) - E\right] - \frac{\alpha_c^2}{4M\left(r\right)}V'\left(r\right)\left(\partial_r - \frac{1}{r}\right), \tag{5.60}$$

$$F_1\left(r\right) = \left[2M\left(r\right) - 1\right]\chi\left(r\right) - \frac{\alpha_c^4}{8M^2\left(r\right)}V'\left(r\right)\left[\chi'\left(r\right) - \frac{\chi\left(r\right)}{r}\right], \tag{5.61}$$

$$F_2\left(r\right) = \left[4M\left(r\right) - 2\right]\dot{\chi}\left(r\right) - \frac{\alpha_c^4}{4M^2\left(r\right)}V'\left(r\right)\left[\dot{\chi}'\left(r\right) - \frac{\dot{\chi}\left(r\right)}{r}\right] + \alpha_c^2\chi\left(r\right) + \frac{\alpha_c^6}{8M^3\left(r\right)}V'\left(r\right)\left[\chi'\left(r\right) - \frac{\chi\left(r\right)}{r}\right], \tag{5.62}$$

respectively. As a result, Eq. (5.34) and Eq. (5.35) are modified to

$$\begin{aligned}
\frac{d}{dr}\begin{pmatrix}\dot{\chi}\left(r\right)\\ \dot{\psi}\left(r\right)\end{pmatrix} &= \begin{pmatrix}0 & 1\\ 2\left[U_{eff}\left(r\right) - E\right] & 0\end{pmatrix}\begin{pmatrix}\dot{\chi}\left(r\right)\\ \dot{\psi}\left(r\right)\end{pmatrix} - \begin{pmatrix}0\\ 2F_1\left(r\right)\end{pmatrix},\\
\frac{d}{dr}\begin{pmatrix}\ddot{\chi}\left(r\right)\\ \ddot{\psi}\left(r\right)\end{pmatrix} &= \begin{pmatrix}0 & 1\\ 2\left[U_{eff}\left(r\right) - E\right] & 0\end{pmatrix}\begin{pmatrix}\ddot{\chi}\left(r\right)\\ \ddot{\psi}\left(r\right)\end{pmatrix} - \begin{pmatrix}0\\ 2F_2\left(r\right)\end{pmatrix};
\end{aligned} \tag{5.63}$$

and

$$\begin{aligned}
\frac{d}{dr}\begin{pmatrix}\dot{\chi}\left(r\right)\\ \dot{\psi}\left(r\right)\end{pmatrix} &= \begin{pmatrix}\frac{1}{r} & 2M\left(r\right)\\ W_{eff}\left(r\right) - E & -\frac{1}{r}\end{pmatrix}\begin{pmatrix}\dot{\chi}\left(r\right)\\ \dot{\psi}\left(r\right)\end{pmatrix} - \begin{pmatrix}0\\ \frac{F_1(r)}{M(r)}\end{pmatrix},\\
\frac{d}{dr}\begin{pmatrix}\ddot{\chi}\left(r\right)\\ \ddot{\psi}\left(r\right)\end{pmatrix} &= \begin{pmatrix}\frac{1}{r} & 2M\left(r\right)\\ W_{eff}\left(r\right) - E & -\frac{1}{r}\end{pmatrix}\begin{pmatrix}\ddot{\chi}\left(r\right)\\ \ddot{\psi}\left(r\right)\end{pmatrix} - \begin{pmatrix}0\\ \frac{F_2(r)}{M(r)}\end{pmatrix}.
\end{aligned} \tag{5.64}$$

Eq. (5.63) and Eq. (5.64) are first order inhomogeneous differential equations. A special inhomogeneous solution can be obtained by using the same integration algorithm for solving Eq. (5.34) and Eq. (5.35). Notice that a general inhomogeneous solution can be constructed by a special inhomogeneous solution plus an arbitrary homogeneous solution. Consequently the solutions to Eq. (5.63) and Eq. (5.64) are in the form

$$\begin{aligned}
\dot{\chi}\left(r\right) &= \dot{\chi}_1\left(r\right) + \alpha\chi\left(r\right),\\
\ddot{\chi}\left(r\right) &= \ddot{\chi}_1\left(r\right) + \beta\chi\left(r\right),
\end{aligned}$$

where $\dot{\chi}_1\left(r\right)$ and $\ddot{\chi}_1\left(r\right)$ are the special inhomogeneous solutions, $\chi\left(r\right)$ is the homogeneous solution, and α and β are arbitrary constants. To uniquely determine $\dot{\chi}\left(r\right)$ and $\ddot{\chi}\left(r\right)$, one needs to apply additional constraints

$$\int_0^R \chi\left(r\right)\dot{\chi}\left(r\right) = 0, \tag{5.65}$$

$$\int_0^R \chi\left(r\right)\ddot{\chi}\left(r\right) = -\int_0^R \dot{\chi}^2\left(r\right), \tag{5.66}$$

www.ebook3000.com

which are derived by applying the energy derivative to the normalization condition Eq. (5.45).

5.6 Complex contour integral

This section discusses the algorithms for evaluating the energy integral in Eq. (3.133). The discussion in this section applies to both bulk systems and equilibrium two-probe systems. By definition the energy integral is on the real axis and the integration limits are from $-\infty$ to $+\infty$ [6]. On the real axis, however, the integrand $\overline{G_{iq}^{<}}(E)\,E^m$ has many sharp features and requires a large number of energy points in the numerical integration. The goal of this section is to convert the integral path from the real axis to a complex contour in order to reduce the computational cost.

For simplicity of notation and without losing generality, the zeroth energy moment ($m = 0$) is adopted in the discussion and the subscript iq and the disorder average $\overline{(\cdots)}$ are omitted. The task is to calculate the energy integral

$$\rho \equiv -\mathrm{i}\int_{-\infty}^{+\infty}\frac{dE}{2\pi}G^{<}(E)\,. \tag{5.67}$$

Notice that bulk systems are always in equilibrium. Hence $G^{<}(E)$ can be simplified by using the fluctuation-dissipation theorem Eq. (2.64)

$$\begin{aligned}\rho &= -\mathrm{i}\int_{-\infty}^{+\infty}\frac{dE}{2\pi}f(E)\left[G^{a}(E)-G^{r}(E)\right]\\ &= -\mathrm{i}\int_{-\infty}^{+\infty}\frac{dE}{2\pi}f(E)\left\{\left[G^{r}(E)\right]^{\dagger}-G^{r}(E)\right\}\\ &= \mathrm{i}\left(J-J^{\dagger}\right),\end{aligned} \tag{5.68}$$

where $f(E)$ is the Fermi function

$$f(E)=\frac{1}{e^{\frac{E-\mu}{k_BT}}+1},$$

and J is defined by

$$J \equiv \int_{-\infty}^{+\infty}\frac{dE}{2\pi}f(E)\,G^{r}(E)\,. \tag{5.69}$$

Before working on the energy integral J, let us make a singularity analysis of the integrand. Although the energy integral is defined on the real axis, it will be advantageous to work with $f(z)$ and $G^{r}(z)$ which are the analytical continuations of $f(E)$ and $G^{r}(E)$ on the complex plane. $f(z)$

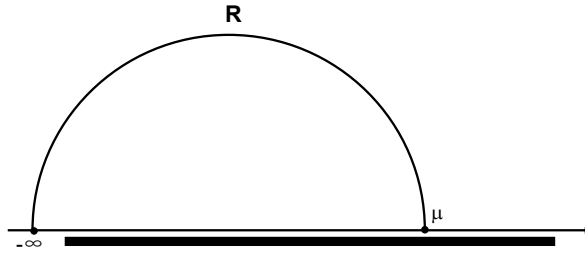


Fig. 5.5 The complex contour at the zero temperature. The complex contour is composed of a semicircle R from $z = -\infty$ to $z = \mu$. The thick line indicates the singularities of $G^r(z)$.

has a series of poles located on a vertical line, $z_k = \mu + \mathrm{i}(2k-1)\pi k_B T$, where k is an integer. z_k is referred to as Fermi poles and the residue of each Fermi pole is $-k_B T$. $G^r(z)$ has two types of singularities, branch cuts and poles, corresponding to continuous bands and discrete bound states. All singularities of $G^r(z)$ lie on a horizontal line slightly lower than the real axis, and the distance is determined by the infinitesimal imaginary part in the definition of the retarded Green's function. The singularities of the integrand have been sketched in Fig. 5.7. The difficulty of the numerical integration becomes clear: The integral path is along the real axis which is too close to the singularities of $G^r(z)$. In the vicinity of the singularities, $G^r(z)$ changes rapidly and requires a large number of energy points to capture the sharp features. Let's imagine those singularities as some dangerous land mines. To avoid running into them, we need to change the integral path from the real axis to a complex contour with the aid of the residue theorem.

We first investigate the energy integral J at zero temperature [7]. At zero temperature, the Fermi function $f(E) = \theta(\mu - E)$ is a step function, and J is reduced to

$$J = \int_{-\infty}^{\mu} \frac{dz}{2\pi} G^r(z). \tag{5.70}$$

Since $G^r(z)$ has no singularity on the upper half plane, the integral path can be changed from the real axis to a semicircle R as shown in Fig. 5.5

$$J = \int_{R} \frac{dz}{2\pi} G^r(z). \tag{5.71}$$

On the complex contour, z can be rewritten by a polar coordinate $z = z_c + Re^{\mathrm{i}\theta}$, in which z_c is the center of the semicircle and $\theta \in (0, \pi)$. Consequently

www.ebook3000.com

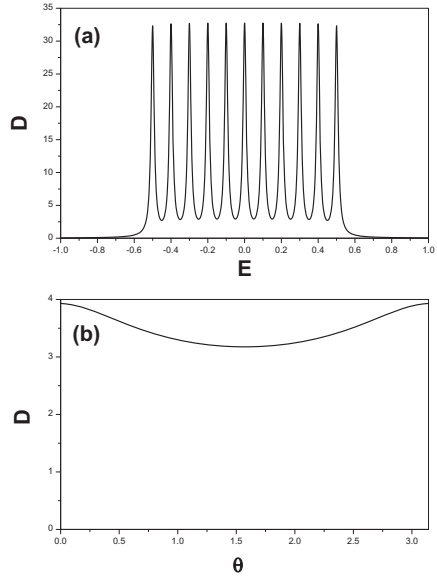


Fig. 5.6 Comparison of the integrand on two integral paths. (a) The integral is along the real axis, $-1 \leqslant E \leqslant 1$, and $D(E) \equiv -\frac{1}{\pi}\mathrm{Im}G^r(E)$. (b) The integral is along the complex contour $z = e^{i\theta}$, $0 \leqslant \theta \leqslant \pi$, and $D(\theta) \equiv -\frac{1}{\pi}\mathrm{Im}\left[-\mathrm{i}e^{\mathrm{i}\theta}G^r(\theta)\right]$. Notice that the area under the curves of (a) and (b) are exactly the same.

Eq. (5.71) is rewritten as

$$J = \int_{\pi}^{0} \frac{d\theta}{2\pi}\mathrm{i}Re^{\mathrm{i}\theta}G^r\left(z_c + Re^{\mathrm{i}\theta}\right) \equiv \int_0^{\pi} d\theta F(\theta). \tag{5.72}$$

The integrand on the complex contour is much smoother than on the real axis. As a illustration, consider the retarded Green's function

$$G^r(z) = \sum_p \frac{1}{z + \mathrm{i}\eta - E_p},$$

where $E_p = -0.5, -0.4, \cdots, 0.5$ and $\eta = 10^{-2}$. Fig. 5.6 compares the imaginary part of the integrand on the real axis and on the complex contour. It is clear that the integrand on the real axis has many sharp peaks while the integrand on the complex contour is featureless.

In the numerical calculation, the integral of $F(\theta)$ is discretized as

$$\int_0^{\pi} d\theta F(\theta) = \sum_i F(\theta_i)\Delta\theta_i,$$

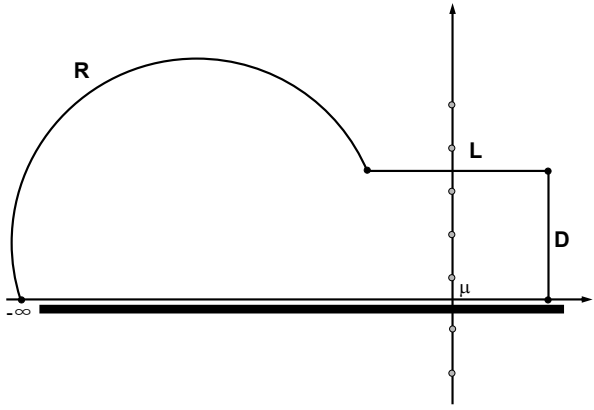


Fig. 5.7 The complex contour at finite temperature. The complex contour is composed of three pieces: a semicircle R, a horizontal line L, and a vertical line D. The singularities of $G^r(z)$ and $f(z)$ are also shown in the plot. The poles and branch cuts of $G^r(z)$ (thick blue line) are distributed on a horizontal line slightly lower than the real axis. The Fermi poles of $f(z)$ (green dots) are uniformly distributed on the vertical line of $\mathrm{Re}z = \mu$.

where θ_i is called the abscissa and $\Delta\theta_i$ the weight. Since $F(\theta)$ is a smooth function of θ, it can be well approximated by a polynomial. Suppose the degree of the polynomial is $2N-1$. Without knowing more details, we can evaluate the integral of the polynomial *exactly* by using N elaborately selected abscissas and weights. The algorithm for selecting the abscissas and weights is called Gaussian quadrature [8]. Usually 40 energy points are sufficient for the evaluation of the integral on the complex contour with Gaussian quadrature.

Next we investigate the energy integral J at finite temperature [9]. At finite temperature, we need to worry about the Fermi poles as well as the singularities of the Green's function. In this situation, the integral path is changed from the real axis to a complex contour shown in Fig. 5.7. The complex contour is composed of three pieces: a semicircle R, a horizontal line L, and a vertical line D. The semicircle R is from $z_1 = -\infty$ to $z_2 = \mu - Nk_BT + \mathrm{i}M2\pi k_BT$ to avoid the real axis as far as possible. The horizontal line L is from z_2 to $z_3 = \mu + Nk_BT + \mathrm{i}M2\pi k_BT$ and goes through the middle of two Fermi poles. The vertical line D is from z_3 to $z_4 = \mu + Nk_BT$ which comes back to the real axis. Here M and N are two positive integers controlling the shape of the complex contour. M is the number of Fermi poles enclosed by the complex contour. N is the

www.ebook3000.com

Fermi function cutoff defined by the condition that $f(E) \approx \theta(\mu - E)$ if $|E - \mu| > Nk_BT$. For $N = 20$, the accuracy of the Fermi function cutoff approximation is 2×10^{-9}.

According to the residue theorem, the real axis integral is equal to the complex contour integral plus the residues of the enclosed Fermi poles. The integral of R can be evaluated with the Gaussian quadrature similar to that of zero temperature

$$J_R = \int_R \frac{dz}{2\pi} G^r(z), \tag{5.73}$$

where $f(z) \equiv 1$ due to the Fermi function cutoff approximation. The integral of L can be reduced to

$$J_L = \int_{\mu - Nk_BT}^{\mu + Nk_BT} \frac{dE}{2\pi} f(E) G^r(E + \mathrm{i}M2\pi k_BT), \tag{5.74}$$

where $f(E + \mathrm{i}M2\pi k_BT) = f(E)$ is used in the derivation. Since the integral path is by a distance $M2\pi k_BT$ away from the real axis, the integrand $G^r(E + \mathrm{i}M2\pi k_BT)$ is also a smooth function of E. J_L can be evaluated efficiently by using a weighted Gaussian quadrature with Fermi function being a weight function [8]. The integral of D is negligible due to the Fermi function cutoff approximation. The residues of the Fermi poles can be evaluated as

$$J_P = 2\pi\mathrm{i} \sum_{m=1}^{M} (-k_BT) G^r[\mu + \mathrm{i}(2m-1)\pi k_BT]. \tag{5.75}$$

Finally the integral J is obtained as

$$J = J_R + J_L + J_P. \tag{5.76}$$

To sum up, we have investigated how to change the integral path from the real axis to a complex contour to evaluate the density matrix in equilibrium. It is worth mentioning that the shape of the complex contour is not unique. One may choose any reasonable contour shape as long as the path is far away from the singularities, e.g., the dumbbell shaped contour shown in Fig. 1 of Ref. [10]. Besides the complex contour integral, one may also calculate the energy integral by summing up all the residues of modified Fermi poles [11]. Interested readers are referred to Appendix A.21 for details.

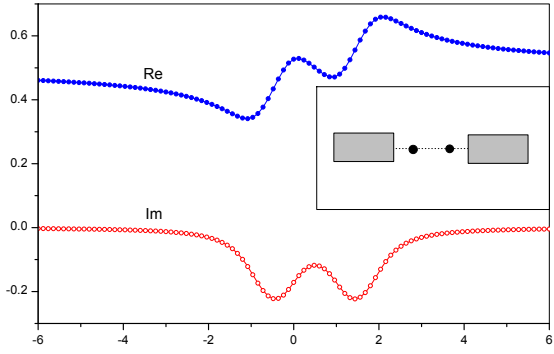


Fig. 5.8 Comparison of the numerical solution and analytical solution of the coherent potential $\tilde{\varepsilon}^r$ in the two-site model shown in the inset. The solid dot and hollow dot are for the real part and imaginary part of the numerical solution, while the solid lines are for the analytical solution. The error between the two solutions is less than 2×10^{-7}. Other parameters are: $x_A = x_B = 0.5$, $\varepsilon_A = 0$, $\varepsilon_B = 1$, $t_0 = 1$, $\Gamma = 1$.

5.7 CPA equations

This section discusses the algorithms for solving the CPA-LMTO equations (3.123). The discussion in this section applies to both bulk systems and equilibrium two-probe systems.

Before studying algorithms and implementation details, we first do a warm up exercise by studying an analytically solvable model. Although the model does not have too much physical meaning, it illustrates some essential features of the CPA equations. The model is composed of two identical disorder sites (see the inset of Fig. 5.8), each of which has a random on-site energy

$$\varepsilon_i = \begin{cases} \varepsilon_A & P = x_A \\ \varepsilon_B & P = x_B \end{cases}, \tag{5.77}$$

where $i = 1, 2$ is the site index, P is the probability, and $x_A + x_B = 1$. The coupling between the two sites is t_0, and each site couples to an external reservoir with a self-energy $-\frac{\mathrm{i}}{2}\Gamma$. By using Eqs. (2.133), the CPA equations

www.ebook3000.com

for the model are obtained as

$$\overline{G_i^r} = x_A \overline{G_{iA}^r} + x_B \overline{G_{iB}^r}, \tag{5.78}$$

$$\overline{G^r} = \left[E - \begin{pmatrix} 0 & t_0 \\ t_0 & 0 \end{pmatrix} - \begin{pmatrix} \tilde{\varepsilon}^r & 0 \\ 0 & \tilde{\varepsilon}^r \end{pmatrix} - \begin{pmatrix} -\frac{\mathrm{i}}{2}\Gamma & 0 \\ 0 & -\frac{\mathrm{i}}{2}\Gamma \end{pmatrix} \right]^{-1}, \tag{5.79}$$

$$\overline{G_i^r} = \left[\overline{G^r} \right]_{ii}, \tag{5.80}$$

$$\overline{G_i^r} = \frac{1}{E - \tilde{\varepsilon}^r - \Omega^r}, \tag{5.81}$$

$$\overline{G_{iA}^r} = \frac{1}{E - \varepsilon_A - \Omega^r}, \tag{5.82}$$

$$\overline{G_{iB}^r} = \frac{1}{E - \varepsilon_B - \Omega^r}, \tag{5.83}$$

where $\tilde{\varepsilon}_1^r = \tilde{\varepsilon}_2^r \equiv \tilde{\varepsilon}^r$ and $\Omega_1^r = \Omega_2^r \equiv \Omega^r$ because the two disorder sites are identical.

The CPA equations (5.78–5.83) can be solved analytically as follows. By eliminating $\overline{G_i^r}$ in Eqs. (5.79,5.80,5.81), one obtains

$$\frac{1}{E - \tilde{\varepsilon}^r - \Omega^r} = \frac{E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma}{\left(E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma\right)^2 - t_0^2}, \tag{5.84}$$

which can be further simplified as

$$\left(E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma\right)\left(\Omega^r + \frac{\mathrm{i}}{2}\Gamma\right) = t_0^2. \tag{5.85}$$

By inserting Eqs. (5.81,5.82,5.83) into Eq. (5.78), one obtains

$$\frac{1}{E - \tilde{\varepsilon}^r - \Omega^r} = \frac{x_A}{E - \varepsilon_A - \Omega^r} + \frac{x_B}{E - \varepsilon_B - \Omega^r}. \tag{5.86}$$

Ω^r can be solved from Eq. (5.86) as

$$\Omega^r = E + \frac{\varepsilon_A \varepsilon_B - \tilde{\varepsilon}^r \hat{\varepsilon}}{\tilde{\varepsilon}^r - \bar{\varepsilon}}, \tag{5.87}$$

where $\bar{\varepsilon} \equiv x_A \varepsilon_A + x_B \varepsilon_B$ and $\hat{\varepsilon} \equiv x_A \varepsilon_B + x_B \varepsilon_A$. By inserting Eq. (5.87) into Eq. (5.85), one obtains a quadratic equation of $\tilde{\varepsilon}^r$,

$$C_2 \left(\tilde{\varepsilon}^r\right)^2 + C_1 \tilde{\varepsilon}^r + C_0 = 0, \tag{5.88}$$

where the coefficients are

$$C_2 = E + \frac{\mathrm{i}}{2}\Gamma - \hat{\varepsilon}, \tag{5.89}$$

$$C_1 = -\left(E + \frac{\mathrm{i}}{2}\Gamma\right)^2 + \left(E + \frac{\mathrm{i}}{2}\Gamma\right)(\hat{\varepsilon} - \bar{\varepsilon}) + \varepsilon_A \varepsilon_B + t_0^2, \tag{5.90}$$

$$C_0 = \left(E + \frac{\mathrm{i}}{2}\Gamma\right)^2 \bar{\varepsilon} - \left(E + \frac{\mathrm{i}}{2}\Gamma\right)\varepsilon_A \varepsilon_B - t_0^2 \bar{\varepsilon}. \tag{5.91}$$

Eq. (5.88) has two roots, one with negative imaginary part and the other with positive imaginary part. The one with the negative imaginary part is the physical solution of $\tilde{\varepsilon}^r$. By substituting the solution to Eq. (5.79), one obtains the disorder averaged retarded Green's function and other physical quantities. It will be a good exercise to check the analytical solution in the low concentration limit where $x_A \approx 1$ and $x_B \approx 0$. Let $x_A = 1 - x$ and $x_B = x$. In the limit $x \to 0$, the analytical solution of Eq. (5.88) can be approximated up to $O(x)$. After some algebra [12], $\tilde{\varepsilon}^r$ is obtained as

$$\tilde{\varepsilon}^r = \varepsilon_A + xt^r, \tag{5.92}$$

$$t^r \equiv \left[\frac{1}{\varepsilon_B - \varepsilon_A} - \frac{E + \frac{\mathrm{i}}{2}\Gamma - \varepsilon_A}{\left(E + \frac{\mathrm{i}}{2}\Gamma - \varepsilon_A\right)^2 - t_0^2}\right]^{-1}, \tag{5.93}$$

which is consistent with Eq. (2.135).

So much for the analytical work. In practice, analytically solvable models are very rare, and most of the problems have to be solved numerically. The value of analytically solvable models is to provide some insights to the mathematical structure and to check the accuracy of numerical algorithms. Now let us solve the same problem numerically with the iterative method. The procedure is described as follows:

(1) Make an initial guess of $\tilde{\varepsilon}^r$,

$$\tilde{\varepsilon^r} = x_A\varepsilon_A + x_B\varepsilon_B;$$

(2) Calculate the diagonal element of $\overline{G^r}$ by using Eqs. (5.79,5.80)

$$\overline{G_i^r} = \left\{\left[E - \begin{pmatrix} 0 & t_0 \\ t_0 & 0 \end{pmatrix} - \begin{pmatrix} \tilde{\varepsilon}^r & 0 \\ 0 & \tilde{\varepsilon}^r \end{pmatrix} - \begin{pmatrix} -\frac{\mathrm{i}}{2}\Gamma & 0 \\ 0 & -\frac{\mathrm{i}}{2}\Gamma \end{pmatrix}\right]^{-1}\right\}_{ii}$$
$$= \frac{E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma}{\left(E - \tilde{\varepsilon}^r + \frac{\mathrm{i}}{2}\Gamma\right)^2 - t_0^2};$$

(3) Update Ω^r by using Eq. (5.81)

$$\Omega^r = E - \tilde{\varepsilon}^r - \left(\overline{G_i^r}\right)^{-1};$$

(4) Update $\tilde{\varepsilon}^r$ by using Eqs. (5.78,5.81,5.82,5.83)

$$\tilde{\varepsilon}^r = E - \Omega^r - \left[\frac{x_A}{E - \varepsilon_A - \Omega^r} + \frac{x_B}{E - \varepsilon_B - \Omega^r}\right]^{-1};$$

(5) Go back to the step (2) to repeat the process until $\tilde{\varepsilon}^r$ is fully converged.

The analytical solution and numerical solution of $\tilde{\varepsilon}^r$ are plotted in Fig. 5.8. As expected, the two solutions agree with each other precisely. It

www.ebook3000.com

is interesting to observe that the non-physical solution of Eq. (5.88) does not show up in the iterative method. In other words, it had been rejected automatically by the iterative method.

With the experience of solving the simple model, we move on to solve the general CPA-LMTO equations (3.123). Analogously to the simple model, we can adopt the iterative method to solve the equations. Here the major differences are (i) $E-H$ is replaced by $P(E)-S(k)$ in the LMTO method and (ii) an integral over k is needed to take into account the periodicity of bulk systems. The procedure is described as follows:

(1) Make an initial guess of $\tilde{P}_i^r(E)$,

$$\tilde{P}_i^r(E) = \sum_q x_{iq} P_{iq}(E); \tag{5.94}$$

(2) Calculate $\overline{\mathcal{G}}_i^r(E)$ by using the second, third, and fourth line of Eqs. (3.123)

$$\begin{aligned}
\overline{\mathcal{G}}_i^r(E) &= \int_{BZ} \frac{d^3k}{(2\pi)^3} \overline{\mathcal{G}}_i^r(E,k), \qquad (5.95)\\
\overline{\mathcal{G}}_i^r(E,k) &= \left[\overline{\mathcal{G}}^r(E,k)\right]_{ii},\\
\overline{\mathcal{G}}^r(E,k) &= \left[\tilde{P}^r(E) - S(k)\right]^{-1},\\
\left[\tilde{P}^r(E)\right]_{ij} &= \delta_{ij}\tilde{P}_i^r(E);
\end{aligned}$$

(3) Update $\Omega_i^r(E)$ by using the fifth line of Eqs. (3.123)

$$\Omega_i^r(E) = \tilde{P}_i^r(E) - \left[\overline{\mathcal{G}}_i^r(E)\right]^{-1}; \tag{5.96}$$

(4) Update $\tilde{P}_i^r(E)$ by using the first, fifth, and sixth line of Eqs. (3.123)

$$\tilde{P}_i^r(E) = \Omega_i^r(E) + \left\{\sum_q x_{iq}\left[P_{iq}(E) - \Omega_i^r(E)\right]^{-1}\right\}^{-1}; \tag{5.97}$$

(5) Go back to the step (2) to repeat the process until $\tilde{P}_i^r(E)$ is fully converged.

Finally we would like to discuss some implementation details. The above iterative procedure has been implemented in the method *CPA_solution* of @class_cpaBulk (@class_necpaTwoProbe). To be precise, the method *CPA_solution* calls the methods *calcCPA_gr*, *calcCPA_Omega_r*, and *calcCPA_tiltP_r* of @class_cpaAtom (@class_necpaAtom) to carry out the step (2), (3), (4), respectively. Notice that the k-integral is implemented in the

private function *integrateBZ* which is the bottleneck of the whole CPA iteration. To evaluate the k-integral, one needs to select proper k-points in the Brillouin zone, and the k-sampling methods will be discussed in Section 6.6.

In post-analysis calculations, the CPA equations need to be solved accurately at every energy point (see e.g., Section 5.10). In self-consistent calculations, however, it is unnecessary to solve the CPA equations accurately in every self-consistent iteration. In the early stages of a self-consistent calculation, the potential is far away from the correct answer, so it does not make too much sense to solve the CPA equations to high precision. Instead the CPA loop and the self-consistent loop can be merged into one big loop, and the atomic potential and coherent potential are converged simultaneously, see Fig. 5.9. This idea has been implemented in the method *CPA_iteration* of @class_cpaBulk (@class_necpaTwoProbe). The flow of *CPA_iteration* is quite similar to that of *CPA_solution* except that the CPA iteration is carried out only once. To use the method *CPA_iteration* in a self-consistent calculation, it is necessary to construct a reasonable initial guess of the coherent potential. The "warm up" is done by calling *CPA_solution* prior to the self-consistent loop.

In some circumstances, the disorder concentration is very low and it is unnecessary to solve the CPA equations iteratively. An approximate analytical expression is available in Eq. (3.124). To implement the formula, one needs to first calculate the unperturbed Green's function $\mathcal{G}_0^r(E)$ by using Eqs. (3.127,3.129) ($\Sigma^r = 0$ in bulk systems). Afterward one can proceed to calculate the coherent potential $\tilde{P}_i^r(E)$ by using Eqs. (3.124,3.126). Once the $\tilde{P}_i^r(E)$ is available, the disorder averaged Green's function $\overline{\mathcal{G}^r}(E)$ can be evaluated by using the CPA-LMTO equations (3.123). So the computational cost is nearly twice that of the clean systems, but is still much lower than the iterative method. The low concentration limit has been implemented in the method *CPA_solution_lowX* of @class_cpaBulk (@class_necpaTwoProbe) which calls the method *calcCPA_tiltP_r_lowX* of @class_cpaAtom (@class_necpaAtom).

To sum up, the CPA-LMTO equations can be solved with the iterative procedure Eqs. (5.94,5.95,5.96,5.97). The procedure has been implemented in the methods *CPA_solution*, *CPA_iteration*, and *CPA_solution_lowX*. The method *CPA_solution* is to solve the CPA equations for a given LMTO Hamiltonian; The method *CPA_iteration* is to iterate the CPA equations once for a given LMTO Hamiltonian; The method *CPA_solution_lowX* is to evaluate an approximate solution of the CPA equations in the low concentration limit.

www.ebook3000.com

5.8 Fermi level

This section discusses the algorithms for solving the Fermi level in bulk systems. In bulk systems, the Fermi level μ is determined by the charge neutrality condition Eq. (3.178). Given an atomic potential, we need to find a proper μ so that the number of electrons is equal to the nuclear charge. Consequently we have double loops in a self-consistent calculation: The outer loop is to solve the atomic potential and the inner loop is to solve the Fermi level. It will be shown below that the double loops can be merged into one big loop analogous to what we did for solving the CPA equations.

Notice that calculating the electron occupation numbers is very costly in the Green's function approach. The computational cost dominates all other steps of the self-consistent calculation. On the other hand, it is unnecessary to solve the Fermi level accurately in every self-consistent iteration. We only need the accurate Fermi level for the final converged atomic potential. The situation is quite similar to the coherent potentials discussed in Section 5.7. Here we shall adopt the same solution strategy, namely, to merge the double loops into one big loop and converge the atomic potential and the Fermi level simultaneously (see also Fig. 5.9).

The new challenge is that charge neutrality must be imposed strictly in every self-consistent step. The reason is that a minor violation of the charge neutrality in a single unit cell can be magnified to infinitely large due to the periodicity of bulk systems. Suppose the Fermi level is at μ_0 and the net charge is Q_0. We need to make a correction to μ_0 so that the resulting net charge is equal to zero exactly. By using Eqs. (3.141,3.134,3.133), the correction $\delta\mu$ is obtained as

$$\delta\mu = -\frac{Q_0}{D_0}, \tag{5.98}$$

where the minus sign is to compensate the net charge Q_0. Here D_0 is the density of states at $E = \mu_0$

$$D_0 = \mathrm{Tr}\; A\left(\mu_0\right), \tag{5.99}$$

where $A\left(z\right)$ is the spectrum function defined by

$$A\left(z\right) \equiv \frac{\mathrm{i}}{2\pi}\left[\overline{G^r}\left(z\right) - \overline{G^a}\left(z\right)\right]. \tag{5.100}$$

In practice, $A\left(\mu_0\right)$ is evaluated at the energy point that is closest to μ_0 in the contour integral calculation. At zero temperature, the energy point

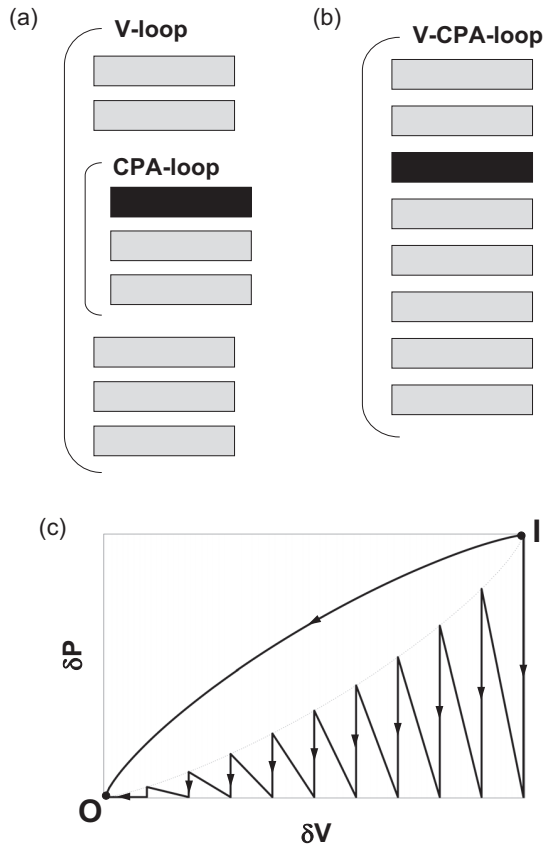


Fig. 5.9 A schematic plot of two iteration schemes for solving the CPA equations. (a) Nested double loops for solving the atomic potentials and the coherent potentials. (b) Merged loop for solving the atomic potentials and the coherent potentials simultaneously. In (a) and (b), the gray box and black box represent light and heavy calculation steps respectively. (c) The convergence paths of (a) and (b). The initial guess is I and the final solution is O. δP and δV are the error of the coherent potential and the atomic potential respectively.

is the last point of the semicircle (see Fig. 5.5); At finite temperature, the energy point is the lowest Fermi pole (see Fig. 5.7). By shifting the Fermi level from μ_0 to $\mu_0 + \delta\mu$, the total charge is corrected from Q_0 to 0. Consequently the charge of each atomic site needs to be corrected by the energy moment correction

$$\delta M^{(k)} = (\mu_0)^k \, A\,(\mu_0)\, \delta\mu, \tag{5.101}$$

www.ebook3000.com

so that the total charge is consistent with the summation of individual atomic charges.

Another implementation detail is the initial guess of Fermi level. To start the merged self-consistent loop, one needs to make an initial guess of the Fermi level in the field *System.InitialFermiLevel* of the input file. Generally the initial guess can be very poor and the bulk system is made far from charge neutral. To stabilize the self-consistent iteration, in the first a few steps, one needs to freeze the atomic potential and iterate the Fermi level to a reasonable accuracy. Afterward the Fermi level may evolve adiabatically with the atomic potential.

5.9 Bulk calculator: band structure

This section discusses the algorithms for calculating the band structure of bulk systems. For clean bulk systems, the band structure is calculated by Eqs. (3.170,3.171) and implemented in @calculator_band_EIG. For disordered bulk systems, the CPA band structure is calculated by Eq. (3.173,3.174,3.175,3.176) and implemented in @calculator_band_CPA. Below we shall discuss some implementation details.

To calculate the band structure of clean bulk systems, one needs to solve the eigenvalues at different k-points. The eigen problem is solved by the method *calcEigen* of @class_cpaBulk. For the LMTO potential parameters, H_{orth} is independent on E, and Eq. (3.170) defines a linear eigenvalue problem which can be easily solved with linear algebra. For the MTO potential parameters, H_{orth} is dependent on E, and Eq. (3.170) defines a nonlinear eigenvalue problem which is much more difficult to solve. In practice one can solve the linear eigenvalue problem of $H_{orth}(\varepsilon)$ at a certain energy ε to obtain the eigenvalues $\{\lambda_i(\varepsilon)\}$. By varying ε, one can find the solution of $\varepsilon = \lambda_i(\varepsilon)$ to obtain the MTO band structure.

The k-points are defined along the lines connecting high symmetry points in the Brillouin zone. For example, in the FCC primitive cell, the cell vectors are

$$\mathbf{a}_1 = a\left[0, \frac{1}{2}, \frac{1}{2}\right],$$
$$\mathbf{a}_2 = a\left[\frac{1}{2}, 0, \frac{1}{2}\right],$$
$$\mathbf{a}_3 = a\left[\frac{1}{2}, \frac{1}{2}, 0\right];$$

and the high symmetry points are

$$\mathbf{k}(\Gamma) \equiv \frac{4\pi}{a}[0,0,0],$$
$$\mathbf{k}(X) \equiv \frac{4\pi}{a}\left[\frac{1}{2},0,0\right],$$
$$\mathbf{k}(W) \equiv \frac{4\pi}{a}\left[\frac{1}{2},\frac{1}{4},0\right],$$
$$\mathbf{k}(L) \equiv \frac{4\pi}{a}\left[\frac{1}{4},\frac{1}{4},\frac{1}{4}\right],$$
$$\mathbf{k}(X') \equiv \frac{4\pi}{a}\left[\frac{1}{2},\frac{1}{2},0\right],$$

where a is the lattice constant. It is more convenient to work with the (dimensionless) fractional coordinates $q_i \equiv \mathbf{k}\cdot\mathbf{a}_i$, where $\mathbf{k} = \frac{q_1}{2\pi}\mathbf{b}_1 + \frac{q_2}{2\pi}\mathbf{b}_2 + \frac{q_3}{2\pi}\mathbf{b}_3$ and $\{\mathbf{b}_i\}$ are the reciprocal unit cell vectors. In the fractional coordinates, the above high symmetry points are reduced to

$$\mathbf{q}(\Gamma) \equiv [0,0,0],$$
$$\mathbf{q}(X) \equiv 2\pi\left[0,\frac{1}{2},\frac{1}{2}\right],$$
$$\mathbf{q}(W) \equiv 2\pi\left[\frac{1}{4},\frac{1}{2},\frac{3}{4}\right],$$
$$\mathbf{q}(L) \equiv 2\pi\left[\frac{1}{2},\frac{1}{2},\frac{1}{2}\right],$$
$$\mathbf{q}(X') \equiv 2\pi\left[\frac{1}{2},\frac{1}{2},1\right].$$

Some symmetry points have been defined in @LatticeType_* for frequently-used lattice types such as BCC, FCC, HEX, and SC. One may also define his or her own symmetry points for any lattice type in the field *Calculator.SymmetryPointList* of the input file.

Notice that the symmetry points are defined for the standard primitive cell. Sometimes one may prefer to work with a supercell or a non-standard unit cell. It is necessary to re-construct the Hamiltonian of the standard primitive cell before calculating the band structure. The procedure is called unfolding. The idea of unfolding is to first expand the Hamiltonian of the unit cell periodically and then pick up the Hamiltonian of the standard primitive cell from the crystal. As an example, the unfolding algorithm for the FCC crystal is presented in Appendix A.19. The unfolding algorithms have been implemented in @CrystalType_*, where * represents the crystal

www.ebook3000.com

type FCC, BCC, HCP, SC, Diamond, ZnS, Wurtzite, CsCl, NaCl, and General [13].

To calculate the CPA band structure of disordered bulk systems, one needs to calculate the k-resolved density of states $\overline{D}(E,k)$. In clean bulk systems, $\overline{D}(E,k)$ is composed of a series of δ-functions $\sum_i \delta[E - E_i(k)]$ where $E_i(k)$ are the eigenvalues of H_{orth}. In disordered bulk systems, the δ-functions are shifted and broadened by the coherent potential. To calculate $\overline{D}(E,k)$ in disordered systems, one needs to solve the coherent potential from the CPA-LMTO equations (3.123). The solution algorithms have been discussed in Section 5.7 and implemented in the method *CPA_solution* and *CPA_solution_lowX* of @class_cpaBulk (@class_necpaTwoProbe). Once the coherent potential is available, one can proceed to calculate $\overline{D}(E,k)$ with Eqs. (3.173,3.174,3.175,3.176) and plot the CPA band structure with the MATLAB function *surface*.

5.10 Bulk calculator: density of states

This section discusses the algorithms for calculating the density of states. For both bulk and two-probe systems, the density of states can be calculated by using the Green's function approach which is implemented in @calculator_dos_CPA. For clean bulk systems, the density of states can also be calculated by using the wave function approach which is implemented in @calculator_dos_THD. Below we shall discuss some implementation details and compare the two approaches.

In the Green's function approach, one needs to first calculate the disorder-averaged physical Green's function $\overline{G^r_{iq}}(E,k)$. $\overline{G^r_{iq}}(E,k)$ can be calculated with the conditional auxiliary Green's function $\overline{\mathcal{G}^r_{iq}}(E,k)$ by using Eq. (3.130). $\overline{\mathcal{G}^r_{iq}}(E,k)$ can be solved from the CPA-LMTO equations (3.123) for bulk systems or the NECPA-LMTO equations (3.121) for two-probe systems. The solution algorithms have been discussed in Section 5.7 and implemented in the method *CPA_solution* and *CPA_solution_lowX* of @class_cpaBulk (@class_necpaTwoProbe). Once the disorder-averaged Green's function is available, one can proceed to calculate the disorder-averaged density of states with Eq. (3.150).

In the wave function approach, one needs to first solve the eigenvalues of the Hamiltonian $H_{orth}(k)$ on a uniform k-grid (N_1-by-N_2-by-N_3) in the Brillouin zone. Afterward each small cube in the k-grid is divided into 6 tetrahedrons (see Fig. 5 of Ref. [14]). The eigenvalues inside each tetrahedron is approximated by a linear interpolation of the eigenvalues on the

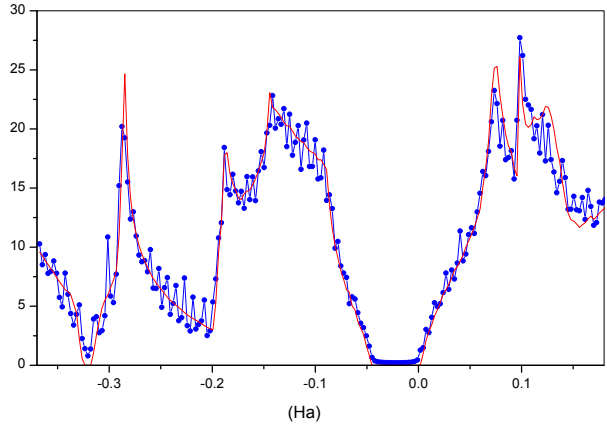


Fig. 5.10 Density of states in clean bulk Si. The smooth curve is calculated by the wave function approach (tetrahedron interpolation) and the noisy curve is calculated by the Green's function approach. Both approaches used a $20 \times 20 \times 20$ uniform k-grid in the Brillouin zone.

four corners. Thus one can calculate the number of states in each tetrahedron below a given energy E to high precision. The density of states is just the derivative of the number of states as a function of E. The contribution of each tetrahedron to the density of states has been explicitly derived by Eqs. (C1,C2,C3,C4) of Ref. [14]. The density of states is a sum of the contributions from the $6N_1N_2N_3$ tetrahedrons.

The Green's function approach is quite general, and is applicable to both bulk and two-probe systems, both clean and disordered systems. The wave function approach, in contrast, is only applicable to clean bulk systems. The reason for implementing the wave function approach is that it can produce much smoother density of states thanks to the tetrahedron interpolation. This is very useful for a quick check of the electronic structure of a lead which is always a clean bulk system. Fig. 5.10 compares the density of states of Si calculated by the two approaches. It is clear that the numerical accuracy of the wave function approach is much higher than the Green's function approach in clean bulk systems.

Bibliography

[1] D. Waldron, Ph.D. thesis, McGill University, 2007.

www.ebook3000.com

[2] J. C. Slater, *Insulators, Semiconductors, and Metals: Quantum Theory of Molecules and Solids*, Vol. **3** (McGraw-Hill, New York, 1967).

[3] At the classical turning point $r = r_c$, the total energy equals the potential energy and the kinetic energy is zero in the classical picture. Therefore a classical particle can only move in the $r < r_c$ region while a quantum particle is allowed to enter the $r > r_c$ region by tunneling.

[4] The energy derivative of the logarithmic derivative can be obtained as $\frac{\partial}{\partial E}\frac{\chi_1'(r)}{\chi_1(r)} = -\frac{2}{\chi_1^2(r)}\int_0^r \chi_1^2(r')\,dr' < 0$ where Eq. (A.175) is used in the derivation.

[5] The energy derivative of the logarithmic derivative can be obtained as $\frac{\partial}{\partial E}\frac{\chi_2'(r)}{\chi_2(r)} \approx 1 > 0$ where the $\partial_r V(r)$ term is ignored and the asymptotic expression $\chi_2(r) \sim e^{-\sqrt{2(V-E)}r}$ is used in the derivation.

[6] Here $-\infty$ and $+\infty$ represent the lower bound and the upper bound of the energy spectrum of the systems. Outside the energy spectrum the integrand is always zero.

[7] J. Taylor, H. Guo, J. Wang, Phys. Rev. B **63**, 245407 (2001).

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1987).

[9] M. Brandbyge, J.-L. Mozos, P. Ordejón, J. Taylor, and K. Stokbro, Phys. Rev. B **65**, 165401 (2002).

[10] L. Lin, M. Chen, C. Yang, and L. He, J. Phys.: Condens. Matter **25**, 295501 (2013).

[11] T. Ozaki, Phys. Rev. B **75**, 035123 (2007).

[12] To avoid the tedious work, one may use *mathematica* in the symbolic derivation. Notice that mathematica does not know the sign of the square root. One has to modify the terms $\sqrt{[\cdots]^2}$ to $[\cdots]$ so that the expression can be further simplified. The *mathematica* code for the theoretical derivation can be found in *ResearchCode*.

[13] A crystal type is composed of a lattice type and a group of atoms. Different crystal types may have the same lattice type, e.g, the crystal type FCC and Diamond belong to the same lattice type FCC.

[14] P. E. Blöchl, O. Jepsen, O. K. Andersen, Phys. Rev. B **49**, 16223 (1994).

Chapter 7

NanoDsim: optimization and parallelization

By using the algorithms of Chapter 5 and Chapter 6, one can develop a preliminary version of the NanoDsim package, which is capable of simulating two-probe systems with 10 to 10^2 atoms on a single PC. However, realistic device simulations often involve much more atoms. Nowadays supercomputers have become standard computing tools, and the power of parallel computing makes it possible to carry out quantum transport simulation of large atomic systems. The goal of this chapter is to parallelize and optimize the NanoDsim package so that it is capable of handling two-probe systems with 10^3 to 10^4 atoms on a moderate supercomputer.

In a large scale atomic simulation, memory and speed are the two most important issues. In Section 7.1, the memory usage and the speed bottlenecks of NanoDsim are thoroughly analyzed. Based on the analysis, various techniques for saving memory and accelerating speed are discussed in Section 7.2 and 7.3. In particular, the principal layer algorithm and the parallelization with MPI are discussed in Section 7.4, 7.5, and 7.6. A benchmark test is provided in Section 7.9 to demonstrate the capability of NanoDsim. Finally, convergence issues and error analysis are further addressed in Section 7.7 and 7.8.

7.1 Performance analysis

In this section, we shall make an analysis of the performance of the preliminary version of NanoDsim. To proceed, we first introduce a few analysis tools. The first tool is MATLAB profiler which can track the execution of an m-code and record the CPU time and the call number of each individual

www.ebook3000.com

function. The syntax of MATLAB profiler is as follows

```
profile on
[code to be analyzed]
profile viewer
```

The tool is extremely useful to identify speed bottlenecks of a MATLAB code. Very often 10% of the code may take 90% of the execution time. Instead of optimizing the code everywhere, it is more economical to work on the 10%. With the aid of MATLAB profiler, one is able to identify the speed bottlenecks and focus on them. After removing a bottleneck, other parts may appear as a new bottleneck. One needs to profile the code again and optimize the new bottleneck. The optimization continues until several bottlenecks coexist and further optimization can gain no more than a small fraction of the total execution time.

Similarly one can make a profile on the memory usage. This can be done by using the command *top* in Linux or the *task manager* in Windows. The problem is that the mapping between the memory usage and the execution of the MATLAB code is obscure. If an abrupt increase is observed in the memory usage, it is hard to accurately locate a specific line responsible for the memory increase. So it is necessary to develop a MATLAB-based memory monitor which is the second tool for the performance analysis. The memory monitor is implemented in *@MemoryMonitor* which not only allows user to dynamically monitor the memory usage but also supports user-defined marker points inside a MATLAB code. By turning on the memory monitor, one can measure the usage of physical memory as a function of time with several markers indicating the location in the MATLAB code. The syntax of the memory monitor is as follows

```
turnOn(MemoryMonitor, dt)
[code to be analyzed: part1]
marker(MemoryMonitor, color_string, type_string)
[code to be analyzed: part2]
marker(MemoryMonitor, color_string, type_string)
[code to be analyzed: part3]
...
turnOff(MemoryMonitor)
```

The memory monitor checks the size of used physical memory for every time interval *dt*. The command *marker* defines a marker point in an m-code which will be indicated on the memory usage curve by a marker. The color

and shape of the marker is defined by *color_string* ('r', 'b', 'g', 'k', 'y', 'm', 'c', 'w') and *type_string* ('+', 'o', '*', '.', 'x', 's', 'd', '^', 'v', '>', '<', 'p', 'h') respectively, and the meaning of the strings is exactly the same as that of MATLAB's *plot* function.

By using these analysis tools, one can identify memory-consuming variables and speed bottlenecks. To make a quantitative estimate of those variables and bottlenecks, one needs to further analyze the operations in the algorithms. Notice that the cost of addition and subtraction are negligible compared to multiplication and division. Hence one only needs to count the number of multiplications and divisions in an algorithm. As a result, the addition or substraction of two $N \times N$ matrices takes no time; the multiplication of two $N \times N$ matrices needs N^3 multiplications and hence takes time $O\left(N^3\right)$; the inverse of an $N \times N$ matrix can be viewed as an inverse operation of matrix multiplication and hence also takes time $O\left(N^3\right)$, etc.

Now we are ready to analyze the memory and computational cost of NanoDsim. Since self-consistent calculations are much more difficult than post-analysis calculations and two-probe systems are much more complicated than bulk systems, we shall focus on the analysis of self-consistent two-probe calculations. In the flowchart of Fig. 3.5, the calculations of six variables are identified to be either memory costly or computational costly. The six variables are: the structure constant S and its Fourier transform $S\left(k\right)$, the lead self-energy $\Sigma_\beta^r\left(E,k\right)$, the Madelung constant M_{CC}, the LMTO orbitals $\{\phi_{iq}\left(r\right)\}$, and the on-site quantities of the NECPA theory $\{X_{iq}\}$. The costs of these variables are estimated in Table 7.1.

variable	memory cost	computational cost	data type	number of calls	algorithm
S	NN_bD^2	$NN_b^3D^3$	real	1	Section 5.3
$S\left(k\right)$	$NN_bD^2N_k$	$NN_bD^2N_k$	complex	1	Section 5.3
$\Sigma_\beta^r\left(E,k\right)$	$N_\beta^2D^2N_EN_kN_\sigma$	$C_1\cdot N_\beta^3D^3N_EN_kN_\sigma$	complex	1	Section 6.4
M_{CC}	$4N^2$	$C_2\cdot 4N^2$	real	1	Section 6.3
$\{\phi_{iq}\left(r\right)\}$	$6NN_\phi N_rN_\sigma$	$C_3\cdot 6NN_\phi N_rN_\sigma$	real	∞	Section 5.5
$\{X_{iq}\}$	$6NN_EN_\sigma D^2$	$3N^3D^3N_EN_kN_\sigma$	complex	∞	full matrix operation

(7.1)

In the table, the memory cost is measured in the units of real / complex number in double precision, and the computational cost is measured by the number of real / complex multiplications.

The parameters of Table 7.1 are explained as follows: N is the number of atomic sites in the central region, and N_β is the number of atomic sites in the lead-β. N_k and N_E are the number of k-points and E-points on the integral path. N_σ is the number of spin species, where $N_\sigma = 1$ is for

www.ebook3000.com

the case of neutral spin and $N_\sigma = 2$ for the case of collinear spin. N_ϕ is the number of atomic orbitals and N_r is the number of radial mesh of each orbital. N_b is the number of neighbors of an atomic site. In the FCC structure, $N_b = 12$ within the nearest neighbors and $N_b = 18$ within the second nearest neighbors. $D = (l_{\max} + 1)^2$ is the size of an atomic site in the orbital space. For $l_{\max} = 2$, s,p,d orbitals are used as a basis set, and an atomic site is represented by a 9×9 matrix block in the orbital space.

The prefactors of table 7.1 are explained as follows: In the fourth row the factor 4 is to take into account one component of monopole and three components of dipole. In the fifth row, the factor 6 is to take into account the six radial functions $\left\{\phi_{iq}, \phi'_{iq}, \dot{\phi}_{iq}, \dot{\phi}'_{iq}, \ddot{\phi}_{iq}, \ddot{\phi}'_{iq}\right\}$. In the sixth row, the factor 6 is to take into account the on-site quantities $\left\{\mathcal{G}^r_{iq}, \mathcal{G}^<_{iq}, \Omega^r_{iq}, \Omega^<_{iq}, \tilde{P}^r_{iq}, \tilde{P}^<_{iq}\right\}$ and the factor 3 is the number of full matrix operations in the lesser Green's function calculation. Other factors are estimated as $C_1 \sim \left(10^2 - 10^3\right)$, $C_2 \sim \left(10 - 10^2\right)$, $C_3 \sim \left(10^2 - 10^3\right)$.

Finally, a few comments are in order. (1) The first four variables can be calculated outside the self-consistent loop only once and hence the number of calls is 1. The last two variables have to be updated in each self-consistent step and hence the number of calls is undetermined (represented by ∞). Therefore one needs to pay more attention to the latter. (2) The computational cost of $\{X_{iq}\}$ is estimated by assuming that full matrix operations are used in the Green's function calculations. It will be replaced by the principal layer algorithm and the cost estimate should also be updated by Table (7.46). (3) In the analysis of computational cost, the calculation of $\Sigma^r_\beta(E,k)$ is independent of the system size N; the calculations of variables S, $S(k)$, and $\{\phi_{iq}(r)\}$ are proportional to N; the calculation of M_{CC} is proportional to N^2; the calculation of $\{X_{iq}\}$ is proportional to N^3. It is clear that the calculation of $\{X_{iq}\}$ is the bottleneck for large N and will be the focus of the following sections.

7.2 Memory issues

This section is devoted to the memory issues in the simulation of large atomic systems. Suppose we have an access to 200 processors each of which has 3GB physical memory. The goal is to carry out a quantum transport simulation of 10^3 to 10^4 atoms. To accommodate thousands of atoms in the limited memory, four tricks are adopted and discussed below.

The first trick is to distribute the atomic sites to different processors.

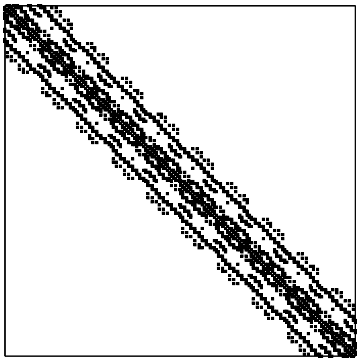


Fig. 7.1 Sparse pattern of the structure constant for a BCC lattice of $3 \times 3 \times 10$ cubic unit cell.

In the LMTO-ASA method, the real space is divided into many slightly overlapped atomic spheres. It is natural to assign a group of atoms to each processor to reduce the memory cost. For example, consider a system containing $N = 10000$ atoms and $N_\phi = 5$, $N_r = 500$, $N_\sigma = 2$. The memory cost of $\{\phi_{iq}(r)\}$ and M_{CC} are 2.4GB and 3.2GB respectively. By distributing the atomic sites to 100 processors, the memory cost will be reduced to 24MB and 32MB which are negligible compared to the 3GB physical memory.

The second trick is to use sparse matrices to represent the structure constant. Notice that most of the matrix elements of S are zero as long as the distance between the two atomic sites exceeds a cutoff length. The larger the atomic system, the more zeros there are in the S matrix. A typical sparse pattern of S matrix is shown in Fig. 7.1. To store S in the memory, it is much more efficient to work with the sparse matrix than the full matrix. MATLAB supports the sparse matrix type and corresponding matrix operations such as multiplication and inversion. Compared to the full matrix, the sparse matrix reduces the memory usage by $\frac{N_b}{N}$ where $N_b \sim 10$ and $N \sim 10^4$.

The third trick is to move unused variables from the memory to the hard disk. In the self-consistent iteration, not all memory-costly variables are needed simultaneously. One can move those unused variables to temporary files and re-load them to memory whenever needed. As long as the saving and loading time are negligible compared to the computing time, using

www.ebook3000.com

temporary files is an economical way to extend memory. In NanoDsim, the temporary files are managed by the class *@HDV*, and the usage of *@HDV* is illustrated by the following example

```
%initialize
StoragePath = './TempData';
initialize(HDV, StoragePath);
%set data
x1 = rand(100, 100);
x2 = rand(100, 100);
setData(HDV, 'x', x1, 'key1');
setData(HDV, 'x', x2, 'key2');
clear x1
clear x2
%get data
x1 = getData(HDV, 'x', 'key1');
x2 = getData(HDV, 'x', 'key2');
%clear data
clear(HDV, 'x')
```

In the example, temporary files are located in the folder *TempData*. The values of x1 and x2 are saved to temporary files with the method *setData*, and the values are loaded back to the memory with the method *getData*.

In NanoDsim, $S(k)$ and $\Sigma_\beta^r(E,k)$ are two typical variables to apply this trick to. The values of $S(k)$ and $\Sigma_\beta^r(E,k)$ are calculated outside the self-consistent loop and saved to temporary files. In the self-consistent iteration, the values are retrieved to calculate the Green's function. In practice, the situations are more complicated than described above: For very small systems or very large memory, it is desirable to keep everything in the memory to avoid the I/O overhead. For very large systems or very limited hard disk, it is desirable to calculate everything on the fly to relieve the hard disk's burden. In the implementation, the calculation of $S(k)$ is managed by the class *@SkBlock_** where * represents *memo*, *hdvp*, and *calc*, corresponding to the options of *save to the memory*, *save to temporary files*, and *calculate on the fly*, respectively. The calculation of $\Sigma_\beta^r(E,k)$ is managed by the class *@SigmaBlock_** where * represents *file*, *hdvp*, and *calc*, corresponding to the options of *save to temporary files (E resolved)*, *save to temporary files (E and k resolved)*, and *calculate on the fly*, respectively.

The last trick is to make use of the multithreading technique supported by MATLAB. Usually a supercomputer is organized in computing nodes.

Each node has several processors sharing one large memory (e.g., 8 processors sharing 24GB memory on a single node). For memory-consuming jobs, one can run fewer MATLAB processes than the number of processors. As a result, each MATLAB can access larger memory (e.g., running 4 MATLABs per node allows 6GB memory access to each MATLAB). A fewer number of MATLABs than processors does not necessarily mean that some processors are idling. By using the multithreading technique, a group of processors are organized to work together on one MATLAB. The technique is managed automatically by MATLAB and the complexity is hidden from the user. Generally MATLAB is able to determine the optimal number of threads in the multithreading. One may also appoint the number of threads manually with the MATLAB command *maxNumCompThreads*. Although not all calculations can be accelerated by the multithreading technique, most of the matrix operations used in NanoDsim can be sped up pretty well.

7.3 Speed issues

This section is devoted to the speed issues in the simulation of large atomic systems. The first five variables listed in Table 7.1 can be easily parallelized over atomic sites, E-points and k-points, which will be discussed in Section 7.5. After parallelization, those variables are no longer speed bottlenecks. The real challenge is in the calculation of nonequilibrium coherent potential which involves the energy integral of nonequilibrium Green's function. Even if the integral is fully parallelized over E-points and k-points, the calculation of a single Green's function is still very costly for large atomic systems. As indicated in the sixth row of Table 7.1, the computational cost is proportional to N^3 where N is the number of atomic sites. In the following subsections, we shall review various methods in the literature to improve the $O\left(N^3\right)$ scaling and discuss their applicability to nonequilibrium quantum transport [1].

7.3.1 *Order-N methods*

The order-N methods pursue an approximate solution of the density matrix whose computational cost is proportional to N rather than N^3. The order-N methods include the Fermi operator expansion, the Fermi operator projection, the divide-and-conquer method, the density-matrix minimization, the orbital minimization, the optimal basis density-matrix minimization, etc., which are reviewed in Ref. [2, 3]. The algorithms have been

www.ebook3000.com

implemented in several DFT packages, such as SIESTA [4], ONESTEP [5], BIGDFT [6], CONQUEST [7], and so on.

Although the algorithms are different in various order-N methods, all of them are based on the locality principle: The properties of a certain observation region comprising one or a few atoms are only weakly influenced by factors that are spatially far away from this observation region [2]. In other words, an atom in a condensed matter environment is "nearsighted" and cannot "see" atoms too far away [8]. In the language of statistical mechanics, the nearsightedness can be expressed as the statement that the elements of density matrix ρ decay quickly with increasing distance. For materials without an energy gap at the Fermi energy (metals), it can be shown that [9]

$$\rho(\mathbf{r}_1, \mathbf{r}_2) \propto \frac{\cos(k_F |\mathbf{r}_1 - \mathbf{r}_2|)}{|\mathbf{r}_1 - \mathbf{r}_2|^2} \exp\left[-c\frac{k_B T}{k_F} |\mathbf{r}_1 - \mathbf{r}_2|\right], \tag{7.2}$$

where k_F is the Fermi wave vector and c is a constant of $O(1)$. For materials with an energy gap Δ at the Fermi energy (insulators), it can be shown that [10]

$$\rho(\mathbf{r}_1, \mathbf{r}_2) \propto \exp[-ca\Delta |\mathbf{r}_1 - \mathbf{r}_2|] \tag{7.3}$$

in the weak-binding limit and

$$\rho(\mathbf{r}_1, \mathbf{r}_2) \propto \exp\left[-c\sqrt{\Delta} |\mathbf{r}_1 - \mathbf{r}_2|\right] \tag{7.4}$$

in the tight-binding limit where a is the lattice constant.

The locality principle was proposed and verified in equilibrium systems. In nonequilibrium, however, it is not applicable. Consider a central region connected to multiple leads with voltages $\{V_\beta\}$. Pick up an atom in the central region. If the atom is nearsighted and cannot see the leads, it won't be able to know the applied voltages and hence won't be able to reach the nonequilibrium steady state. Without the locality principle, the order-N methods cannot be applied to the nonequilibrium situation. Fortunately the locality principle can be "recovered" in two-probe systems described by localized atomic orbitals. Interested readers are referred to Appendix A.24 for more details.

7.3.2 *Iterative methods*

In localized atomic basis, the Hamiltonian elements between two atomic sites are zeros if the distance is beyond a cutoff length. Consequently the Hamiltonian matrix is highly sparse and contains a large number of

zeros (see e.g., Fig. 7.1). The question is how to make use of those zeros to reduce the cost of Green's function calculation. This subsection and the next subsection will investigate two different mathematical techniques, iterative methods and direct methods, to solve such large sparse matrices.

The iterative methods are highly efficient for solving eigenvalues or linear equations of large sparse matrices. The key idea is to construct a basis set with low cost and work within the subspace spanned by the basis set. Assume that A is an $N \times N$ sparse matrix and b is an $N \times 1$ vector. The subspace $span\left\{b, Ab, A^2b, \cdots, A^{m-1}b\right\}$ is called Krylov subspace where $m \ll N$. Since only matrix-vector multiplications are used in the construction of Krylov subspace, the cost is proportional to $nnz\left(A\right)Nm$ where $nnz\left(A\right)$ is the nonzero element number of A. The basis set will be a linear combination of $\left\{b, Ab, A^2b, \cdots, A^{m-1}b\right\}$, and hence the cost of the basis set construction is also proportional to $nnz\left(A\right)Nm$. Different choices of basis set result in different iterative methods, such as *cg*, *minres*, *bicg*, *gmres*, *qmr* and so on [11, 12]. As an illustration, the two-sided Lanczos algorithm and the *bicg* iterative method are discussed in Appendix A.25.

To apply the iterative methods to the density matrix calculation, one may combine the method with the idea of divide-and-conquer and work atom by atom [13]. For one particular atom, one needs to encapsulate the atom with a cluster, and solve the density matrix of the cluster with the iterative methods. Since only the columns of the central atom are needed, the vector b can be assigned as $[\mathbf{0}; \cdots; \mathbf{0}; \mathbf{1}; \mathbf{0}; \cdots; \mathbf{0}]$ where $\mathbf{1}$ is an identity matrix block for the central atom and $\mathbf{0}$ is a zero matrix block for other surrounding atoms. The cluster calculations are carried out for each individual atom in the system, and the total cost will be proportional to Nn^2m where N is the total atom number, n is the cluster atom number, and m is the size of Krylov space. The accuracy and efficiency are controlled by the parameters n and m [13]. The iterative methods have been applied successfully to large scale electronic structure calculations in bulk systems and implemented in the packages OPENMX [13] and ELSES [14, 15].

It is much more difficult to apply the iterative methods to two-probe systems than to bulk systems. Although the cost of each iteration is still proportional to Nn, the iteration number m will be a new concern in two-probe systems. It is found that m increases drastically if the energy z gets close to the singularities of Green's function [16]. Fig. 7.2 shows a color map of m in the complex plane of z. One can see that the iteration number is very large in the vicinity of the Green's function's branch cut $z \in (-2, 2)$. In bulk systems, $m \sim 30$ is sufficient because one can carry out the energy

www.ebook3000.com

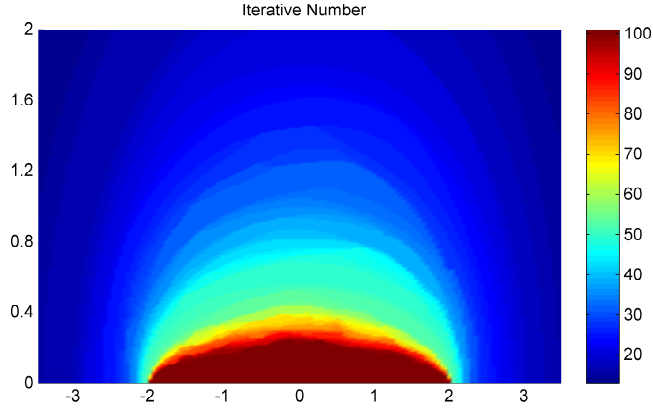


Fig. 7.2 The color map of the iteration number in the complex plane. By using the iterative algorithm *bicg*, the retarded Green's function of a 1d two-probe system is solved at z to the accuracy $\delta = 10^{-6}$. The iteration number is plotted as a function of z. The 1d two-probe system is shown in Fig. 2.7, containing 101 sites in the central region. The site in the center has on-site energy $\varepsilon_C = 0$ and nearest neighbor coupling $t_C = 0.1$. Other sites have on-site energy $\varepsilon_0 = 0$ and nearest coupling $t_0 = 1$.

integral on the complex contour or use the collinear residue theorem [14] to avoid the singularities on the real axis. In two-probe systems, however, one has to work on the real energies for the nonequilibrium part of density matrix and the transmission coefficient. So the dark region of Fig. 7.2 is inevitable.

To improve the convergence on the real energies, one may consider using preconditioned iterative methods. The idea is to apply a preconditioner to the left or right of A to improve the convergence behavior. For example, to solve the linear equation

$$Ax = b, \tag{7.5}$$

one can multiply Eq. (7.5) with $(A')^{-1}$ from the left

$$\left[(A')^{-1} A\right] x = (A')^{-1} b, \tag{7.6}$$

so that the original linear equation is converted to a new linear equation

$$\tilde{A}x = \tilde{b}, \tag{7.7}$$

where $\tilde{A} \equiv (A')^{-1} A$ and $\tilde{b} \equiv (A')^{-1} b$. As far as A' is a good approximation of A, $\tilde{A}$ is close to an identity matrix, and hence the convergence of Eq. (7.7) will be improved remarkably. On the other hand, the operation of $(A')^{-1}$

will bring a new cost to the iterative method. Instead of calculating $(A')^{-1}$ explicitly, one may calculate the matrix-vector product $(A')^{-1} v$ by solving another linear equation $A'y = v$. So the criteria for a good preconditioner are: (a) A' captures the essential feature of A and (b) $A'y = v$ can be solved with low cost. Although there are some general preconditioners (e.g., SOR, ILU) available in the literature [11, 12], a good preconditioner strongly depends on the mathematical properties of a specific problem. In Appendix A.26, a special preconditioner is designed for the Green's function calculation on the real energies in two-probe systems [17].

7.3.3 *Direct methods*

Direct methods can be viewed as variations of Gauss elimination and the solutions are always *exact*. To make use of the many zeros in a sparse matrix, one needs to analyze sparse pattern and reorder the matrix elements so that a large number operations involving the zeros can be avoided. The frontal method and the multifrontal method are two reordering techniques invented in the context of finite element analysis. The same techniques can be applied to general sparse matrices such as those encountered in atomic simulations. Interested readers are referred to the monograph [18] for a review of various direct methods. Here we attempt to illustrate the spirit of the frontal and multifrontal method by a simple example.

Consider an $L \times L$ atomic square lattice. In the localized atomic basis, an atom may have nonzero Hamiltonian elements with other atoms within a cutoff length d. The Green's function is defined by $G = h^{-1}$ where h is the reduced Hamiltonian. The definition of $h(z)$ depends on the basis type: In the orthogonal tight-binding basis,

$$h(z) \equiv z - H, \tag{7.8}$$

where H is the Hamiltonian matrix; In the LCAO basis,

$$h(z) \equiv zS - H, \tag{7.9}$$

where H and S are Hamiltonian matrix and overlap matrix respectively; In the LMTO basis,

$$h(z) \equiv P(z) - S(k), \tag{7.10}$$

where P and S are potential function and structure constant respectively. Here we don't care about the basis type and regard h as an abstract sparse matrix whose elements are nonzero if and only if two atoms are within

www.ebook3000.com

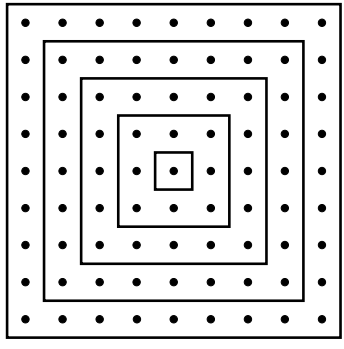


Fig. 7.3 The frontal method applied to the square lattice. The dots represent the atoms in the square lattice which are divided into rectangular shells.

the cutoff length d. The goal is to calculate the Green's function on the boundary of square lattice (see Fig. 7.3).

The spirit of the frontal method is to draw an interface (frontal) between the system and the environment. All the environmental effects are taken into account by a "self-energy" (see Section 2.5). Let 0 represent the system degrees of freedom and 1 the environment degrees of freedom. By definition the Green's function is the inverse a 2×2 block matrix

$$G = \begin{pmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{pmatrix}^{-1}.$$

By using Eq. (2.69), one obtains

$$G_{00} = (h_{00} - \Sigma_{11})^{-1}, \tag{7.11}$$

$$\Sigma_{11} = h_{01} h_{11}^{-1} h_{10}, \tag{7.12}$$

which are also referred to as Gauss elimination or Schur complement in the mathematical literature. To apply the frontal method to the square lattice, the lattice is divided into many rectangular shells with the thickness d (Fig. 7.3). Notice that each shell only has nonzero interaction with its nearest shells. The first elimination is to regard the smallest shell as the environment and the remaining as the system. The smallest shell can be eliminated by using Eqs. (7.11,7.12). Similarly, one can eliminate the second shell, the third shell, ..., all the way to the boundary.

The multifrontal method is a generalization of the frontal method. Instead of working on a single interface, one may work on many interfaces

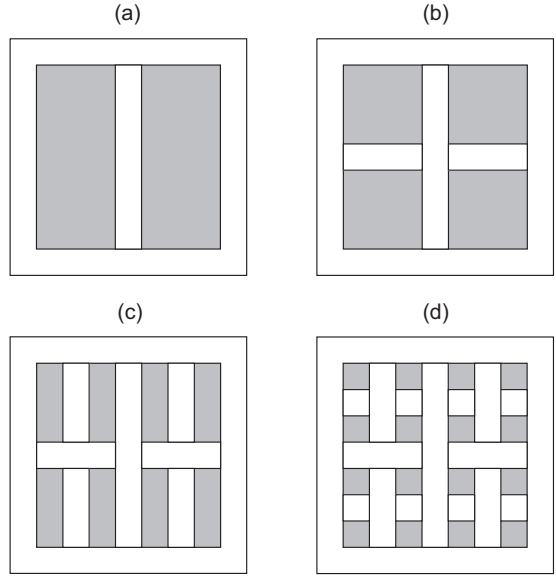


Fig. 7.4 The multifrontal method applied to the square lattice. From (a) to (d), the system is divided into sub-systems recursively with different levels of separators.

simultaneously. To apply the multifrontal method to the square lattice, the lattice is divided by multiple levels of separators with the thickness d. The first level of separator is to divide the lattice into two separate parts, which have no interaction with each other (Fig. 7.4a). The second level of separators is to further divide the two separate parts into four separate parts (Fig. 7.4b). The recursive partition continues until the remaining pieces are no larger than $d \times d$, and the pieces are called elementary blocks (Fig. 7.4d). After the partition, the multifrontal elimination proceeds in reverse: Firstly, the elementary blocks are eliminated from the lattice (Fig. 7.4d). Secondly, the highest level separators are eliminated from the lattice (Fig. 7.4c). The elimination continues until all levels of separators are eliminated and only the boundary shell remains.

It will be instructive to make a cost comparison of the two methods. Since full matrix multiplication and inversion are used in the elimination, the cost of each elimination is proportional the cube of the atom number involved. The atom number in turn is proportional to the area of shells or separators. In the frontal method, the outermost shell has the area $4Ld$,

www.ebook3000.com

the second outermost shell has the area $4\,(L-2d)\,d$, ..., etc. The cost of the frontal elimination can be estimated by

$$\begin{aligned} C_1 &\sim [4Ld]^3 + [4\,(L-2d)\,d]^3 + [4\,(L-4d)\,d]^3 + [4\,(L-6d)\,d]^3 + \cdots \\ &\sim d^6 \left[\left(\frac{L}{2d}\right)^3 + \left(\frac{L}{2d}-1\right)^3 + \left(\frac{L}{2d}-2\right)^3 + \left(\frac{L}{2d}-3\right)^3 + \cdots \right] \\ &\approx d^6 \frac{1}{4} \left(\frac{L}{2d}\right)^4 \sim L^4 d^2. \end{aligned} \tag{7.13}$$

In the multifrontal method, the first level has one separator with the area Ld, the second level has two separators with the area $\frac{1}{2}Ld$, the third level has four separators with the area $\frac{1}{2}Ld$, etc. The cost of the multifrontal method can be estimated by [19]

$$\begin{aligned} C_2 &\sim (Ld)^3 + 2 \times \left(\frac{1}{2}Ld\right)^3 + 4 \times \left(\frac{1}{2}Ld\right)^3 \\ &\quad + 8 \times \left(\frac{1}{4}Ld\right)^3 + 16 \times \left(\frac{1}{4}Ld\right)^3 + \cdots \\ &\approx \frac{5}{2}\,(Ld)^3 \sim L^3 d^3. \end{aligned} \tag{7.14}$$

In large systems, $L \gg d$, hence the multifrontal method has lower cost than the frontal method.

Finally we would like to mention that the direct methods for large sparse matrices have been implemented in several sparse matrix solvers, whose performances are evaluated in Ref. [20]. Meanwhile a special form of the frontal method, the principal layer algorithm, has been applied to atomic simulations for decades. The principal layer algorithm with emphasis on quantum transport will be discussed in Section 7.4. Recently a geometric version of the multifrontal methods called nested dissection algorithm [21] has also been applied to study quantum transport [22, 23, 27].

7.3.4 *Summary*

To sum up, we have briefly reviewed three types of methods in large scale atomic simulations. The order-N methods rely on the locality principle which is more physics-oriented. The iterative methods and direct methods are general mathematical techniques for large sparse matrices. Although order-N methods and iterative methods typically outperform the direct methods, the latter ones are exact and more robust in the self-consistent calculations. In the NanoDsim package, one of the direct methods, principal

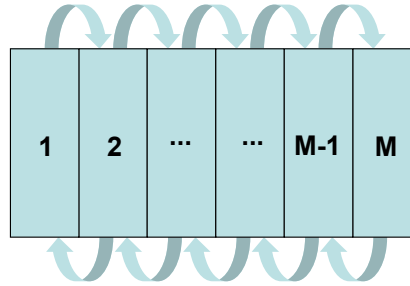


Fig. 7.5 The central region of a two-probe system is divided into M principal layers along the transport direction. Only neighboring layers have nonzero interaction with each other, resulting in a block tridiagonal Hamiltonian matrix.

layer algorithm, is used to speed up the calculation of Green's functions which will be the subject of the next section.

7.4 Principal layer algorithm

This section discusses the principal layer algorithm to accelerate the Green's function calculations. In localized atomic basis such as the LMTO method, the Hamiltonian of a two-probe system can be divided into many slices called principal layers along the transport direction. Each slice has nonzero interaction only with its neighboring slices (see Fig. 7.5), and hence the Hamiltonian is of the block tridiagonal form. The cost of matrix operations on such block tridiagonal matrices can be reduced substantially by using some mathematical tricks and the Langreth theorem.

7.4.1 *Retarded Green's function*

From the mathematical point of view, the calculation of retarded Green's function is equivalent to a matrix inverse

$$G^r = (h_0 - \Sigma^r)^{-1}, \tag{7.15}$$

where h_0 is the reduced Hamiltonian of the system and Σ^r is the retarded self-energy to take into account the environmental effect. Due to the partition of the principal layers, h_0 is block tridiagonal and Σ^r is block diagonal.

www.ebook3000.com

Define $h^r \equiv h_0 - \Sigma^r$

$$h^r = \begin{pmatrix} h^r_{1,1} & h^r_{1,2} & & & & & \\ h^r_{2,1} & h^r_{2,2} & h^r_{2,3} & & & & \\ & h^r_{3,2} & h^r_{3,3} & h^r_{3,4} & & & \\ & & h^r_{4,3} & \ddots & \ddots & & \\ & & & \ddots & h^r_{M-1,M-1} & h^r_{M-1,M} \\ & & & & h^r_{M,M-1} & h^r_{M,M} \end{pmatrix}_{M\times M}, \qquad (7.16)$$

where $h^r_{i,j}$ is the matrix block between the i^{th} principal layer and the j^{th} principal layer. The goal is to calculate selected block elements of G^r without the inverse of the whole h^r matrix [24–26].

First of all, we calculate the diagonal blocks $G^r_{1,1}$ and $G^r_{M,M}$. To calculate $G^r_{M,M}$, one can eliminate the principal layers one by one from the left to the right. Eliminating the first layer with Eqs. (7.11,7.12), one obtains

$$h^r \longrightarrow h^r_{eff} = \begin{pmatrix} \tilde{h}^r_{2,2} & h^r_{2,3} & & & & \\ h^r_{3,2} & h^r_{3,3} & h^r_{3,4} & & & \\ & h^r_{4,3} & \ddots & \ddots & & \\ & & \ddots & h^r_{M-1,M-1} & h^r_{M-1,M} \\ & & & h^r_{M,M-1} & h^r_{M,M} \end{pmatrix}_{(M-1)\times(M-1)},$$

where $\tilde{h}^r_{2,2} = h^r_{2,2} - h^r_{2,1}\left(h^r_{1,1}\right)^{-1} h^r_{1,2}$. Notice that h^r and h^r_{eff} are equivalent in calculating the retarded Green's function of the layers from 2 to M. Similarly one can eliminate the second layer, the third layer, ..., all the way to the last layer. As a result, one obtains

$$\begin{aligned} C^r_1 &= \left(h^r_{1,1}\right)^{-1}, \\ C^r_{i+1} &= \left(h^r_{i+1,i+1} - h^r_{i+1,i} C^r_i h^r_{i,i+1}\right)^{-1}, \end{aligned} \qquad (7.17)$$

and

$$G^r_{M,M} = C^r_M. \qquad (7.18)$$

Here C^r_i has the physical meaning of the right surface Green's function for the principal layers $\{1, 2, \ldots, i\}$. To calculate $G^r_{1,1}$, one can eliminate the principal layers one by one from the right to the left. As a result, one obtains

$$\begin{aligned} D^r_M &= \left(h^r_{M,M}\right)^{-1}, \\ D^r_{i-1} &= \left(h^r_{i-1,i-1} - h^r_{i-1,i} D^r_i h^r_{i-1,i}\right)^{-1}, \end{aligned} \qquad (7.19)$$

and

$$G^r_{1,1} = D^r_1. \tag{7.20}$$

Here D^r_i has the physical meaning of the left surface Green's function for the principal layers $\{i, i+1, \cdots, M\}$.

Secondly, we calculate all the diagonal blocks of G^r, i.e., $\left\{G^r_{i,i}\right\}$. Since $G^r_{M,M}$ has been solved from Eq. (7.18), we need to derive a relation between the diagonal blocks $G^r_{i-1,i-1}$ and $G^r_{i,i}$. Divide the principal layers $\{1, 2, \cdots, M\}$ into two groups, $\{1, 2, \cdots, i\}$ and $\{i+1, i+2, \cdots, M\}$. h^r can be partitioned into a 2×2 super-block matrix

$$h^r = \left(\begin{array}{cc|c} h^r_{1,1} & h^r_{1,2} & \\ h^r_{2,1} & \ddots & \ddots \\ \ddots & h^r_{i-1,i-1} & h^r_{i-1,i} \\ & h^r_{i,i-1} & h^r_{i,i} \\ \hline & & h^r_{i+1,i} \end{array} \right. \left. \begin{array}{ccccc} h^r_{i,i+1} & & & & \\ h^r_{i+1,i+1} & h^r_{i+1,i+2} & & & \\ h^r_{i+2,i+1} & h^r_{i+2,i+2} & \ddots & & \\ & \ddots & \ddots & & h^r_{M-1,M} \\ & & h^r_{M,M-1} & & h^r_{M,M} \end{array} \right)$$

$$\equiv \left(\begin{array}{c|c} \tilde{h}^r_{1,1} & \tilde{h}^r_{1,2} \\ \hline \tilde{h}^r_{2,1} & \tilde{h}^r_{2,2} \end{array}\right). \tag{7.21}$$

Applying the lemma Eq. (2.70) to Eq. (7.21), one obtains

$$\tilde{G}^r_{1,1} = \left(\tilde{h}^r_{1,1}\right)^{-1} + \left(\tilde{h}^r_{1,1}\right)^{-1} \tilde{h}^r_{1,2} \tilde{G}^r_{2,2} \tilde{h}^r_{2,1} \left(\tilde{h}^r_{1,1}\right)^{-1}. \tag{7.22}$$

Notice that $\tilde{h}^r_{1,2}$ is nonzero only in the $(i,1)$ block and $\tilde{h}^r_{2,1}$ is nonzero only in the $(1,i)$ block. Taking the (i,i) block of $\tilde{G}^r_{1,1}$, one obtains

$$\left(\tilde{G}^r_{1,1}\right)_{i,i} = \left(\tilde{h}^r_{1,1}\right)^{-1}_{i,i} + \left(\tilde{h}^r_{1,1}\right)^{-1}_{i,i} \left(\tilde{h}^r_{1,2}\right)_{i,1} \left(\tilde{G}^r_{2,2}\right)_{1,1} \left(\tilde{h}^r_{2,1}\right)_{1,i} \left(\tilde{h}^r_{1,1}\right)^{-1}_{i,i},$$

leading to

$$G^r_{i,i} = C^r_i + C^r_i h^r_{i,i+1} G^r_{i+1,i+1} h^r_{i+1,i} C^r_i. \tag{7.23}$$

Here $\left(\tilde{h}^r_{1,1}\right)^{-1}_{i,i} = C^r_i$ because C^r_i is the right surface Green's function of the principal layers $\{1, 2, \cdots, i\}$. Thus the diagonal blocks $\left\{G^r_{i,i}\right\}$ can be calculated with $\{C^r_i\}$ by the recurrence

$$\begin{aligned} G^r_{M,M} &= C^r_M, \\ G^r_{i,i} &= C^r_i + C^r_i h^r_{i,i+1} G^r_{i+1,i+1} h^r_{i+1,i} C^r_i. \end{aligned} \tag{7.24}$$

www.ebook3000.com

Similarly one can also apply the lemma Eq. (2.73) to Eq. (7.21) and calculate $\left\{G^r_{i,i}\right\}$ with $\{D^r_i\}$

$$\begin{aligned} G^r_{1,1} &= D^r_1, \\ G^r_{i+1,i+1} &= D^r_{i+1} + D^r_{i+1} h^r_{i+1,i} G^r_{i,i} h^r_{i,i+1} D^r_{i+1}. \end{aligned} \tag{7.25}$$

Thirdly, we calculate the l^{th} block row or block column of G^r, i.e., $\left\{G^r_{l,i}\right\}$ or $\left\{G^r_{i,l}\right\}$. Since the diagonal block $G^r_{l,l}$ has been calculated from Eq. (7.24) or Eq. (7.25), we need to derive a relation between $G^r_{l,i}$ and $G^r_{l,i\pm1}$ or $G^r_{i,l}$ and $G^r_{i\pm1,l}$. For $i \geqslant l$, applying the lemma Eq. (2.71) and Eq. (2.72) to Eq. (7.21), one obtains

$$\tilde{G}^r_{1,2} = -\tilde{G}^r_{1,1} \tilde{h}^r_{1,2} \left(\tilde{h}^r_{2,2}\right)^{-1}, \tag{7.26}$$

$$\tilde{G}^r_{2,1} = -\left(\tilde{h}^r_{2,2}\right)^{-1} \tilde{h}^r_{2,1} \tilde{G}^r_{1,1}. \tag{7.27}$$

Notice that $\tilde{h}^r_{1,2}$ is nonzero only in the $(i,1)$ block and $\tilde{h}^r_{2,1}$ is nonzero only in the $(1,i)$ block. Taking the $(l,1)$ block of Eq. (7.26) and $(1,l)$ block of Eq. (7.27), one obtains

$$\left(\tilde{G}^r_{1,2}\right)_{l,1} = -\left(\tilde{G}^r_{1,1}\right)_{l,i} \left(\tilde{h}^r_{1,2}\right)_{i,1} \left(\tilde{h}^r_{2,2}\right)^{-1}_{1,1}, \tag{7.28}$$

$$\left(\tilde{G}^r_{2,1}\right)_{1,l} = -\left(\tilde{h}^r_{2,2}\right)^{-1}_{1,1} \left(\tilde{h}^r_{2,1}\right)_{1,i} \left(\tilde{G}^r_{1,1}\right)_{i,l}, \tag{7.29}$$

leading to

$$G^r_{l,i+1} = -G^r_{l,i} h^r_{i,i+1} D^r_{i+1}, \tag{7.30}$$

$$G^r_{i+1,l} = -D^r_{i+1} h^r_{i+1,i} G^r_{i,l}. \tag{7.31}$$

For $i \leqslant l-1$, applying the lemma Eq. (2.76) and Eq. (2.75) to Eq. (7.21), one obtains

$$\tilde{G}^r_{2,1} = -\tilde{G}^r_{2,2} \tilde{h}^r_{2,1} \left(\tilde{h}^r_{1,1}\right)^{-1}, \tag{7.32}$$

$$\tilde{G}^r_{1,2} = -\left(\tilde{h}^r_{1,1}\right)^{-1} \tilde{h}^r_{1,2} \tilde{G}^r_{2,2}, \tag{7.33}$$

Notice that $\tilde{h}^r_{1,2}$ is nonzero only in the $(i,1)$ block and $\tilde{h}^r_{2,1}$ is nonzero only in the $(1,i)$ block. Taking the $(l-i,i)$ block of Eq. (7.32) and $(i,l-i)$ block of Eq. (7.33), one obtains

$$\left(\tilde{G}^r_{2,1}\right)_{l-i,i} = -\left(\tilde{G}^r_{2,2}\right)_{l-i,1} \left(\tilde{h}^r_{2,1}\right)_{1,i} \left(\tilde{h}^r_{1,1}\right)^{-1}_{i,i}, \tag{7.34}$$

$$\left(\tilde{G}^r_{1,2}\right)_{i,l-i} = -\left(\tilde{h}^r_{1,1}\right)^{-1}_{i,i} \left(\tilde{h}^r_{1,2}\right)_{i,1} \left(\tilde{G}^r_{2,2}\right)_{1,l-i}, \tag{7.35}$$

leading to

$$G^r_{l,i} = -G^r_{l,i+1} h^r_{i+1,i} C^r_i, \tag{7.36}$$
$$G^r_{i,l} = -C^r_i h^r_{i,i+1} G^r_{i+1,l}. \tag{7.37}$$

In short, the l^{th} block row or block column of G^r can be deduced from $C^r_1, C^r_2, \cdots C^r_{l-1}$, $G^r_{l,l}$, $D^r_{l+1}, \cdots, D^r_{M-1}, D^r_M$ by using Eqs. (7.30,7.31) for $i \geqslant l$ and Eqs. (7.36,7.37) for $i \leqslant l-1$.

Thus all the blocks of G^r can be evaluated without the direct inverse of h^r. The procedure is to first obtain the C^r and D^r series by using Eqs. (7.17,7.19), and then obtain the diagonal blocks by using Eq. (7.24) or Eq. (7.25). Afterward one can go up and right from a diagonal block to construct the upper triangular part by using Eqs. (7.30,7.31); or go down and left from a diagonal block to construct the lower triangular part by using Eqs. (7.36,7.37).The computational cost will be further analyzed in Section 7.4.5.

7.4.2 *Lesser Green's function*

Once the blocks of retarded Green's function are available, the blocks of lesser Green's function can be derived with the Langreth theorem discussed in Section 2.3. Notice that the complex-time Green's function G_τ satisfies the same equation of motion as that of G^r. Therefore the principal layer algorithm of G^r can be generalized to G_τ by removing all the superscript r. Afterward one can apply the Langreth theorem to the principal layer algorithm of G_τ to derive the matrix blocks of G^r and $G^<$.

As an illustration we shall derive the recurrence of the diagonal blocks of lesser Green's function, i.e., $\left\{G^<_{i,i}\right\}$. Define the complex-time Green's function $G = h^{-1} = (h_0 - \Sigma)^{-1}$ where h_0 is block tridiagonal and Σ is block diagonal. Notice that $h^r_0 = h^a_0 = h_0$, $h^<_0 = 0$, $(\Sigma^r)^\dagger = \Sigma^a$, $(\Sigma^<)^\dagger = -\Sigma^<$. The recurrence of C^r_i, Eq. (7.17), can be generalized to the recurrence of complex-time C_i by removing all the superscripts r. Applying the Langreth theorem Eqs. (2.29) to the new recurrence, one obtains

$$\begin{aligned} C^<_1 &= C^r_1 \Sigma^<_{1,1} C^a_1, \\ C^<_{i+1} &= C^r_{i+1} \left(\Sigma^<_{i+1,i+1} + h^r_{i+1,i} C^<_i h^a_{i,i+1}\right) C^a_{i+1}, \end{aligned} \tag{7.38}$$

where $h^<_{i+1,i+1} = -\Sigma^<_{i+1,i+1}$ and $h^<_{i,i+1} = h^<_{i+1,i} = 0$ are used in the derivation. The recurrence of $G^r_{i,i}$, Eq. (7.23), can be generalized to the recurrence of complex-time $G_{i,i}$ by removing all the superscripts r. Applying

www.ebook3000.com

the Langreth theorem Eqs. (2.29) to the new recurrence, one obtains

$$\begin{aligned} G^<_{M,M} &= C^<_M, \\ G^<_{i,i} &= C^<_i + C^r_i h^r_{i,i+1} G^<_{i+1,i+1} h^a_{i+1,i} C^a_i \\ &\quad + \left(C^r_i h^r_{i,i+1} G^r_{i+1,i+1} h^r_{i+1,i} C^<_i - H.c. \right), \end{aligned} \tag{7.39}$$

where $h^<_{i,i+1} = h^<_{i+1,i} = 0$ and

$$C^r_i h^r_{i,i+1} G^r_{i+1,i+1} h^r_{i+1,i} C^<_i = -\left(C^<_i h^a_{i,i+1} G^a_{i+1,i+1} h^a_{i+1,i} C^a_i \right)^\dagger,$$

are used in the derivation. $\{C^r_i\}$ and $\left\{G^r_{i,i}\right\}$ are defined by Eq. (7.17) and Eq. (7.23), and $\{C^a_i\}$ can be obtained by $C^a_i = (C^r_i)^\dagger$.

It is worth mentioning that one can work on the Keldysh equation $G^< = G^r \Sigma^< G^a$ directly to obtain the recurrence of $\left\{G^<_{i,i}\right\}$ [27]. It turns out that the derived recurrence is equivalent to Eqs. (7.38,7.39) after some lengthy algebra.

7.4.3 *Transmission coefficient*

Transmission coefficient calculation can be reduced to lesser Green's function. One can apply the principal layer algorithm of $G^<$ to the transmission coefficient calculation. Suppose that the left (right) lead self-energy is nonzero only in the first (last) principal layer. The transmission coefficient can be simplified as

$$\begin{aligned} T &= \mathrm{Tr}\ G^r \Gamma_L G^a \Gamma_R \\ &= \mathrm{Tr}\ \left(G^r \Gamma_L G^a \right)_{M,M} (\Gamma_R)_{M,M} \\ &= \mathrm{Tr}\ \left(-\mathrm{i} G^<_L \right)_{M,M} (\Gamma_R)_{M,M}, \end{aligned} \tag{7.40}$$

where $G^<_L$ is defined by $G^<_L = G^r \tilde{\Sigma}^< G^a$ with $\tilde{\Sigma}^< = \mathrm{i}\Gamma_L$. The mathematical trick can be applied to both clean two-probe systems and disordered two-probe systems. In disordered two-probe systems, the transmission coefficient is

$$\begin{aligned} \overline{T} &= \mathrm{Tr}\ \overline{G^r \Gamma_L G^a} \Gamma_R \\ &= \mathrm{Tr}\ \left(\overline{G^r \Gamma_L G^a} \right)_{M,M} (\Gamma_R)_{M,M} \\ &= \mathrm{Tr}\ \left(-\mathrm{i} \overline{G^<_L} \right)_{M,M} (\Gamma_R)_{M,M}, \end{aligned} \tag{7.41}$$

where $\overline{G^<_L}$ is defined by $\overline{G^<_L} = \overline{G^r \left(\mathrm{i}\Gamma_L\right) G^a} = \overline{G^r} \tilde{\Sigma}^< \overline{G^a}$ with $\tilde{\Sigma}^< = \mathrm{i}\Gamma_L + \tilde{P}^<_L$ (see Eq. (3.154)).

In clean two-probe systems, the transmission coefficient can be calculated with an alternative principal layer algorithm [28] which is more efficient than Eq. (7.40). The algorithm is based on the observation that the transmission coefficient is independent of the partition of the central region and the leads in clean two-probe systems. One can eliminate the principal layers from the left and right until only one principal layer is left in the central region. The transmission coefficient will be calculated in the "irreducible" central region.

Assume that the reduced Hamiltonian of the central region is h_0 and $(h_0)^\dagger = h_0$. The self-energy is $\Sigma^r = \Sigma^r_L + \Sigma^r_R$ where Σ^r_L is nonzero only in the $(1,1)$ block and Σ^r_R is nonzero only in the (M,M) block. To calculate the transmission coefficient in the I^{th} layer, one needs to eliminate the $\{1,2,\cdots,I-1\}$ layers from the left and the $\{I+1,I+2,\cdots,M\}$ layers from the right. The recurrence is as follows: For $i = 1,2,\cdots,I-1$

$$\begin{aligned}\tilde{\Sigma}^r_{L,1} &= (\Sigma^r_L)_{1,1}\,,\\ \tilde{\Sigma}^r_{L,i+1} &= (h_0)_{i+1,i}\left[(h_0)_{i,i} - \tilde{\Sigma}^r_{L,i}\right]^{-1}(h_0)_{i,i+1}\,;\end{aligned} \tag{7.42}$$

For $i = M, M-1,\cdots,I+1$

$$\begin{aligned}\tilde{\Sigma}^r_{R,M} &= (\Sigma^r_R)_{M,M}\,,\\ \tilde{\Sigma}^r_{R,i-1} &= (h_0)_{i-1,i}\left[(h_0)_{i,i} - \tilde{\Sigma}^r_{R,i}\right]^{-1}(h_0)_{i,i-1}\,.\end{aligned} \tag{7.43}$$

After the elimination, the transmission coefficient can be calculated by

$$\begin{aligned}T &= \mathrm{Tr}\ \tilde{G}^r\tilde{\Gamma}_L\tilde{G}^a\tilde{\Gamma}_R,\\ \tilde{G}^r &= \left[(h_0)_{I,I} - \tilde{\Sigma}^r_{L,I} - \tilde{\Sigma}^r_{R,I}\right]^{-1},\\ \tilde{\Gamma}_L &= -\mathrm{i}\left(\tilde{\Sigma}^a_{L,I} - \tilde{\Sigma}^r_{L,I}\right),\\ \tilde{\Gamma}_R &= -\mathrm{i}\left(\tilde{\Sigma}^a_{R,I} - \tilde{\Sigma}^r_{R,I}\right),\end{aligned} \tag{7.44}$$

in which the advanced Green's function and self-energies are the Hermitian conjugates of their retarded counterparts. It is emphasized that the principal layer algorithm Eqs. (7.42,7.43,7.44) only apply to the transmission coefficient calculation of clean two-probe systems.

7.4.4 *Cost estimate*

With the principal layer algorithm, the cost of Green's function calculations is drastically reduced as compared to the full matrix operations. The

www.ebook3000.com

computational cost and memory cost are estimated in Table (7.45) for the diagonal blocks of Green's functions and the transmission coefficient.

variable	memory cost	computational cost	algorithm
$\left\{G^r_{i,i}\right\}$	$2MN_0^2D^2$	$7MN_0^3D^3$	Eqs. (7.17,7.24)
$\left\{G^<_{i,i}\right\}$	$3MN_0^2D^2$	$14MN_0^3D^3$	Eqs. (7.17,7.38,7.24,7.39)
$\overline{T}$	$\tilde{}N_0^2D^2$	$7MN_0^3D^3$	Eq. (7.40)
T	$\tilde{}N_0^2D^2$	$3MN_0^3D^3$	Eqs. (7.42,7.43,7.44)

(7.45)

Here M is the number of principal layers, N_0 is the atom number in each principal layer, and D is the orbital number of each atomic site. Notice that both memory cost and computational cost are proportional to the number of principal layers.

The cost estimate is explained as follows. The first row of the table is for the diagonal blocks of G^r which are needed in the equilibrium self-consistent calculation. In the upward sweeping to calculate $\{C^r_i\}$, the memory cost is $MN_0^2D^2$ in order to save $\{C^r_i\}$, while the computational cost is $3MN_0^3D^3$ in order to carry out one inverse and two multiplications in Eq. (7.17). In the downward sweeping to calculate $\left\{G^r_{i,i}\right\}$, the memory cost is $MN_0^2D^2$ in order to save $\left\{G^r_{i,i}\right\}$, while the computational cost is $4MN_0^3D^3$ in order to carry out four multiplications in Eq. (7.24). The second row of the table is for the diagonal blocks of $G^<$ which are needed in the nonequilibrium self-consistent calculation. In the upward sweeping to calculate $\{C^r_i\}$ and $\{C^<_i\}$, the memory cost is $2MN_0^2D^2$ in order to save $\{C^r_i\}$ and $\{C^<_i\}$, while the computational cost is $7MN_0^3D^3$ in order to carry out one inverse and six multiplications in Eq. (7.17) and Eq. (7.38). In the downward sweeping to calculate $\left\{G^<_{i,i}\right\}$, the memory cost is $MN_0^2D^2$ in order to save $\left\{G^<_{i,i}\right\}$, while the computational cost is $7MN_0^3D^3$ in order to carry out seven multiplications in Eq. (7.24) and Eq. (7.39). Notice that the common factors $C^r_i h^r_{i,i+1}$ and $C^r_i h^r_{i,i+1} G^r_{i+1,i+1} h^r_{i+1,i}$ can be reused and $h^a_{i+1,i} C^a_i = \left(C^r_i h^r_{i,i+1}\right)^\dagger$. The third row of the table is for the transmission coefficient in disordered two-probe systems. In the upward sweeping to calculate $\{C^r_i\}$ and $\{C^<_i\}$, the memory cost is negligible since it is unnecessary to save $\{C^r_i\}$ and $\{C^<_i\}$, while the computational cost is $7MN_0^3D^3$ in order to carry out one inverse and six multiplications in Eq. (7.17) and Eq. (7.38). Downward sweeping is unnecessary since only $G^<_{N,N} = C^<_N$ is used in the transmission coefficient calculation. The fourth row of the table is for the transmission coefficient of clean two-probe systems. In the elimination of principal layers, the memory cost is negligible, while the computational cost

is $3MN_0^3D^3$ in order to carry out one inverse and two multiplications in Eq. (7.42) and Eq. (7.43).

Accordingly the sixth row of Table (7.1) needs to be updated as

variable	memory cost	computational cost	data type	number of calls	algorithm
$\{X_{iq}\}$	$6NN_EN_\sigma D^2$	$14MN_0^3D^3N_EN_kN_\sigma$	complex	∞	principal layer

(7.46)

where the computational cost is estimated by the calculation of $\left\{G_{i,i}^<\right\}$. Here the memory cost of the principal layer algorithm is neglected due to the use of temporary files which will be discussed in the next subsection.

7.4.5 *Implementation details*

In the above discussion, it is assumed that the central region has been divided into M principal layers as illustrated in Fig. 7.5. In practice, the partition of principal layers is done automatically by the function *makePartition* of *Library07/LayerPartition*. The partition algorithm is as follows: In the central region, those atoms having nonzero interaction with the left lead are defined as the first layer, those atoms having nonzero interaction with the first layer are defined as the second layer, etc.; Similarly, those atoms having nonzero interaction with the right lead are defined as the M^{th} layer; those atoms having nonzero interaction with the M^{th} layer are defined as the $(M-1)^{\text{th}}$ layer, etc. The two sequences of principal layers keep on growing from both sides until they meet in the middle.

The principal layer algorithms listed in Table (7.45) have been implemented in the class *@solver_tridiagonal* of *Library07/PrincipalLayer*. Notice that most of the memory is spent on saving $\{C_i^r\}$ and $\{C_i^<\}$ during the upward sweeping and $\left\{G_{i,i}^r\right\}$ and $\left\{G_{i,i}^<\right\}$ during the downward sweeping. To reduce the memory cost of $\{C_i^r\}$ and $\{C_i^<\}$, the segments of $\{C_i^r\}$ and $\{C_i^<\}$ are moved to temporary files in the upward sweeping if the size of the data exceeds the value defined by the environment variable *TridiagBufferSize*. In the downward sweeping, the segments of $\{C_i^r\}$ and $\{C_i^<\}$ are loaded to memory whenever needed in the recurrence. To reduce the memory cost of $\left\{G_{i,i}^r\right\}$ and $\left\{G_{i,i}^<\right\}$, instead of saving $\left\{G_{i,i}^r\right\}$ and $\left\{G_{i,i}^<\right\}$ for the whole principal layer, the blocks of each atomic site are extracted and saved. The memory cost of the latter is only $\frac{1}{N_0}$ of the former and hence is negligible.

7.5 MATLAB interface to MPI

In the previous section, we reduce the computational cost by improving the algorithms. In this section, we further accelerate the execution of a

www.ebook3000.com

given algorithm by exploiting the power of parallel computing. Basically parallel computing is to organize many processors to carry out a challenging job. One needs to break the job into many sub-jobs and assign them to the processors. Once all the sub-jobs are done, the results are collected and put together. During the team work, it is necessary to have communications among the processors. The most popular communication language is called message passing interface (MPI), which have been implemented in libraries such as MPICH (www.mpich.org) and OpenMPI (www.open-mpi.org).

To use MPI in MATLAB, one needs to install an MPI library first, and then write a MATLAB interface to call the MPI library. Such a MATLAB-MPI interface was originally developed in Ref. [29] and later rewritten to the class *@MPI* of *Library12/InterfaceMPI* in the NanoDsim package. Below we shall explain a few important methods of the class @MPI: Firstly, one needs to initialize MPI at the beginning of parallelization and finalize MPI at the end of parallelization, and the methods are *initialize* and *finalize*. Secondly, to distinguish different processors, each of them is assigned a unique ID number called MPI rank. The MPI ranks are integers $0, 1, \cdots, N_p - 1$ where N_p is the total number of processors. N_p is also known as MPI size. The methods to get MPI rank and MPI size are *getRank* and *getSize*. Thirdly, two processors may communicate by sending and receiving data through messages. The methods to send and receive messages are *send* and *receive*. Fourthly, the whole group of processors may exchange data collectively. Typical methods for group communication are *broadcast* and *allreduce*.

The methods and syntax of @MPI are summarized in Table 7.47

method	**syntax**	**comment**
initialize	*initialize(MPI, isParallel)*	isParallel = true, false
finalize	*finalize(MPI, isExitMatlab)*	isExitMatlab = true, false
getRank	*mpirank = getRank(MPI)*	get MPI rank of this node
getSize	*mpisize = getSize(MPI)*	get MPI size
send	*send(MPI, x, ReceiverRank, MsgTag)*	send x from this node to receiver
receive	*x = receive(MPI, SenderRank, MsgTag)*	receive x from sender to this node
broadcase	*x = broadcast(MPI, x, RootRank)*	broadcast x from the root node
allreduce	*x = allreduce(MPI, x, operation)*	see comment in the text
barrier	*barrier(MPI)*	synchronize the execution
isInitialized	*tf = isInitialize(MPI)*	check if MPI is initialized
isParallel	*tf = isParallel(MPI)*	check if in the parallel mode
isMasterNode	*tf = isMasterNode*	check if this node is master

(7.47)

A few comments are in order. (1) *allreduce* is equivalent to organizing a loop in which the root node (rank = 0) exchanges data with all other nodes (rank $\neq$ 0). The data exchange is achieved by x_0 = *feval(operation,* x_0*,* x_i*)* where $i = 1, 2, \cdots, N_p - 1$ and *operation* can be 'plus', 'max', 'min', 'vertcat', 'horzcat', etc. Afterward the data of the root node is broadcast to all other nodes. (2) One needs to take special care of print and save operations in a parallelization. If the same message is printed by all the processors, the screen can be a big mess. If the same file is saved by all the processors, I/O errors may occur. So it is necessary to print and save only on the root node with the aid of *isMasterNode*. (3) In the method *send*, *receive*, *broadcast*, and *allreduce*, x is assumed to be a scalar or matrix (either real or complex). The value of x is converted to MPI-supported data type by the private function *pack* and converted back to the MATLAB data type by the private function *unpack*. To support more data types, one needs to extend the private function *pack* and *unpack*. (4) The private folder of @MPI contains a c-code, *mpi_proxy.c*, which is an interface between MATLAB and c-version of MPI implementation (also see Section 4.3). Before using @MPI, one needs to compile the c-code with the command, *mex CC=mpicc mpi_proxy.c*.

The usage of the class @MPI is illustrated by a simple example. The goal is to calculate the Gauss sum $1 + 2 + \cdots + 100$, and the parallelized code *GaussSum_parallel.m* is as follows

```
%% initialize
isParallel = true;
initialize(MPI, isParallel)
mpisize = getSize(MPI);
mpirank = getRank(MPI);
%% calculate
result = 0;
mpijob = 0;
for ii = 1:100
    mpijob = mpijob + 1;
    if mod(mpijob, mpisize) ~= mpirank  %check if it is my job
       continue
    end
    result = result + ii;
end %ii
result = allreduce(MPI, result, 'plus');
```

www.ebook3000.com

```
if isMasterNode(MPI)
    fprintf('result = %d \n', result)
end
%% finalize
finalize(MPI)
```

Issuing the command *mpirun -n 4 matlab GaussSum_parallel*, the code will run on four processors in parallel. Although all the processors are running the same code, the MPI rank k makes the execution different: The condition *mod(mpijob, mpisize) ~= mpirank* assigns those jobs indexed by the k^{th} residue class of N_p to the processor k. Specifically, the first processor ($k = 0$) works on $y_1 = 4 + 8 + \cdots + 100$; The second processor ($k = 1$) works on $y_2 = 1 + 5 + \cdots + 97$; The third processor ($k = 2$) works on $y_3 = 2 + 6 + \cdots + 98$; The fourth processor ($k = 3$) works on $y_4 = 3 + 7 + \cdots + 99$. Afterward the results on the four processors are summed up by *allreduce* where $y = y_1 + y_2 + y_3 + y_4$ is carried out. Finally comes the output *result = 5050*. Voila!

7.6 Parallelization

The example of Gauss sum provides a prototype of simple parallelization. Here simple parallelization means to divide a demanding job into many small jobs which can be carried out *independently* on different processors. Although not all computational problems can be parallelized in such a simple manner, the algorithms of NanoDsim do fit the simple parallelization model. This section discusses some details of the parallelization in NanoDsim package.

There are two main types of parallelization in NanoDsim: The parallelization over atomic sites in the real space and the parallelization over E-points and k-points in the orbital space. The first type of parallelization is organized by one parallel job manager which assigns the atomic sites to different processors to solve atomic orbitals. The second type of parallelization is organized by another parallel job manager which assigns E-points and k-points to different processors to calculate Green's functions.

To achieve a good parallelization efficiency, the rule of thumb is to make the workload as equal as possible among the processors. The parallelization over atomic sites is rather simple. One can imagine each atom as an integer and make the parallelization as we did in the Gauss sum example. The parallelization over E-points and k-points is more complicated. The

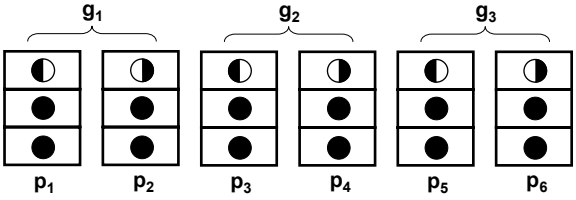


Fig. 7.6 The sketch of the full-time and part-time scheduler for the parallelization over E-points and k-points. Here 15 jobs are distributed to 6 processors, each of which is in charge of two full-time jobs and one part-time job.

optimal job distribution scheme depends on the E-point number N_E, the k-point number N_k, and the processor number N_p. Consider the following four scenarios: (1) $N_p = 100$, $N_E = 200$, $N_k = 20$. Since N_E can be divided by N_p, it is natural to distribute the E-points evenly to the N_p processors. Namely each processor is in charge of two E-points. (2) $N_p = 100$, $N_E = 175$, $N_k = 20$. Although N_E is not a multiple of N_p, it is still not bad to parallelize over E-points. As a result 75 processors are in charge of two E-points and 25 processors in charge of one E-point. (3) $N_p = 100$, $N_E = 5$, $N_k = 20$. Since $N_E \ll N_p$, it is inefficient to parallelize over E-points. Instead the 100 processors can be divided into 5 working groups, each of which takes care of one E-point. Because a single E-point is associated with 20 k-points, each processor in the working group is assigned to work on one k-point. (4) $N_p = 100$, $N_E = 205$, $N_k = 20$. The first 200 E-points can be distributed evenly to the 100 processors like the scenario (1). The remaining 5 E-points can be handled by the working groups like the scenario (4).

Taking into account the above four scenarios, a full-time and part-time scheduler is designed as follows (see Fig. 7.6). Assume that $N_E = N_p \cdot n + \tilde{N}_E$ where n and $\tilde{N}_E$ are the quotient and remainder of $\frac{N_E}{N_p}$ respectively. If $\tilde{N}_E > \frac{1}{2}N_p$, all the E-points will be distributed to the N_p processors, in which $\tilde{N}_E$ processors are assigned to $(n+1)$ E-points and $\left(N_p - \tilde{N}_E\right)$ processors assigned to n E-points. The job is referred to as a full-time job, implying that all the k-points associated with the E-point are calculated on the same processor. If $\tilde{N}_E \leqslant \frac{1}{2}N_p$, the first $N_p \cdot n$ E-points will be distributed to the N_p processors as full-time jobs. For the remaining $\tilde{N}_E$ E-points, the N_p processors are divided into $\tilde{N}_E$ working groups, each of which will take care of one E-point. The job is referred to as a part-time

www.ebook3000.com

job, implying that the k-points associated with the E-point are parallelized within the working group.

Another consideration in the parallelization is to reduce the communication among processors as much as possible. Especially in the self-consistent loop, to exchange data of 10^4 atoms in every iteration may take considerable time simply for the communication. So the strategy is to allreduce the variables unless absolutely necessary and collect the atomic data after the self-consistent loop. The strategy is not only helpful to reduce the network burden but also important to save the memory cost. For two-probe self-consistent calculations (*@SCFsolver_TwoProbe/SCF_f.m*), the atomic data and their parallel distribution are summarized in Table 7.48.

quantity	**variable**	**method**	**status**
atomic orbital	$\left\{\phi_{iq}(r), \dot{\phi}_{iq}(r)\right\}$	*calculatePsi*	distributed
potential parameter	$\{C_{iq}, \Delta_{iq}, \gamma_{iq}\}$	*calcualteCDG*	allreduced
coherent potential	$\left\{\tilde{P}_i^r, \tilde{P}_i^<, \tilde{\Omega}_i^r, \tilde{\Omega}_i^<\right\}$	*CPA_iteration*	distributed
energy moment	$\left\{\tilde{M}_{iq}^k\right\}$	*calculateEM*	allreduced
charge density	$\{\rho_{iq}(r)\}$	*calculateRho*	distributed
charge density derivative	$\left\{\rho'_{iq}(r), \rho''_{iq}(r)\right\}$	*calculateDrho*	distributed
kinetic energy density	$\{t_{iq}(r)\}$	*calculateKED*	distributed
atomic charge	$\{Q_i\}$	*calculateCharge*	not parallelized
atomic dipole	$\{\mathbf{P}_i\}$	*calcualteDipole*	distributed
linearization center	$\left\{E_{iq}^0\right\}$	*calcualteE0*	distributed
atomic potential	$\{V_{iq}(r)\}$	*calculateV*	distributed

(7.48)

One can see that most of the variables are distributed to local processors. There are only two allreduce operations for potential parameters and energy moments, where the parallelization of real space and the parallelization of orbital space intersect with each other.

Finally it is worth mentioning that the Green's function calculation is still the bottleneck even after the parallelization over E-points and k-points. If sufficient processors are available (e.g., 10^4 processors), one may consider implementing multiple level parallelization. The higher level parallelization is over E-points and k-points, and the lower level parallelization is to implement a parallelized principal layer algorithm [26, 27].

7.7 Benchmark

By implementing the principal layer algorithm (Section 7.5), the parallelization in real space and orbital space (Section 7.6), and the various

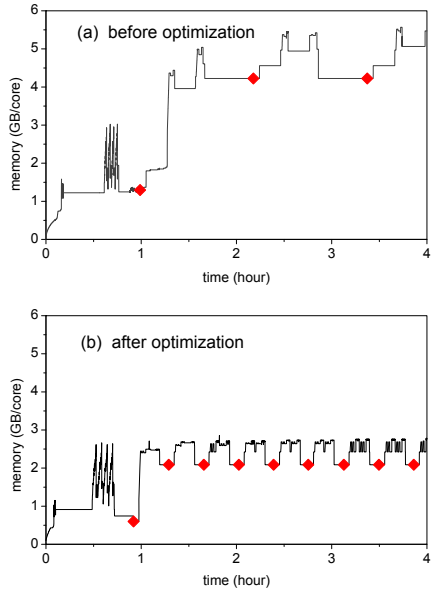


Fig. 7.7 Memory consumption versus computation time (a) before and (b) after the optimization. The test is carried out for the simulation of a two-probe Si device with 12,800 atomic spheres. See the text for details.

memory-saving techniques (Section 7.2), NanoDsim is capable of simulating quantum transport in large atomic systems with up to 10^4 atomic sites on a supercomputer by using 10^2 processors. In this section, the performance of NanoDsim is evaluated with a benchmark test [1, 30].

Fig. 7.7 shows the memory cost as a function of computing time before and after the optimization, for simulating an Si device having 12,800 atomic spheres. The memory usage is recorded by the MATLAB-based memory monitor (see Section 7.1). The red markers indicate the start of each self-consistent iteration. Before the first marker is the initialization stage of the self-consistent calculation. Between the first and second marker is the first self-consistent step where some memory-costly variables are created and stay in memory. The curve becomes periodic after the second marker indicating that the memory usage of the simulation has stabilized. Upon optimization, the peak memory consumption is reduced from 5.6 GB/core to 2.9 GB/core; and the computation time of each self-consistent step is reduced from 72 minutes to 20 minutes.

www.ebook3000.com

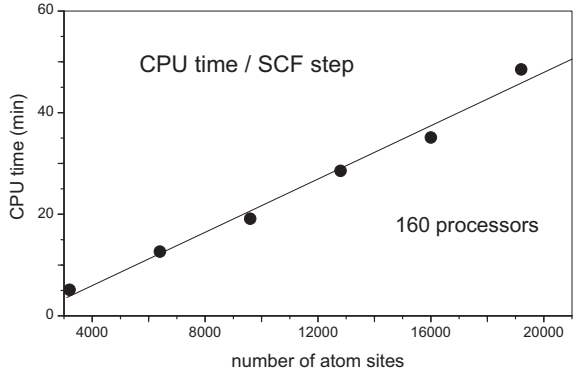


Fig. 7.8 Computing time of one self-consistent step versus the number of atomic spheres. 160 processors is used in the test. See the text for details.

Fig. 7.8 shows a benchmark test of NanoDsim for a two-probe Si device, using 160 processors. The two-probe Si device is a thin film-like structure, which is periodic in the x-direction ($L_x = 0.543$ nm), finite in the y-direction ($L_y = 10.86$ nm), and has a channel length L_z in the z-direction. Six different channel lengths $L_z = 5.43$ nm, 10.86 nm, 16.29 nm, 21.72 nm, 27.15 nm, 32.58 nm are calculated, where the number of atomic spheres in the scattering region increases from $3,200$ to $19,200$. In the Green's function calculation, 20 E-points are used on the complex contour and 10 k-points are used in the x-direction. For a device having $L_z = 21.72$ nm ($12,800$ atomic sites), it takes 28.5 minutes to complete a single self-consistent step. As shown in Fig. 7.8, the computing time grows almost linearly with the channel length which is consistent with the estimate of the principal layer algorithm.

7.8 Convergence issues

The benchmark test of Fig. 7.8 has demonstrated that NanoDsim is capable of simulating quantum transport in large atomic systems. The test is carried out for a single self-consistent iteration and the unspoken assumption is that the self-consistent iteration should converge within reasonable steps (e.g., 10^2 steps). The assumption, however, is far from obvious. On the contrary, the larger the system, the more variables involved, and the more difficult the convergence. Especially in nonequilibrium two-probe systems

the convergence becomes extremely difficult due to the sharp features in the real axis integral. To achieve convergence in nonequilibrium is like tightrope walking, which requires special tricks and experience. Here are some useful tips from our own research.

Tip 1: Making a good initial guess. If no initial guess is provided, a self-consistent calculation will start from isolated atoms. The self-consistent solver will first spend a large number of iterations to develop local chemical bonds among those isolated atoms. Afterward more iterations are needed to fine tune small charge transfers due to disorder scattering and external field. This is obviously not economical. One could have made a good initial guess to set up the local chemical environment properly.

A strategy to make good initial guesses is based on the following observations: Small systems are always easier to converge than large systems; Bulk systems are always easier to converge than two-probe systems; Equilibrium systems are always easier to converge than nonequilibrium systems; Clean systems are always easier to converge than disordered systems. Hence one can use the easy-to-converge systems as the initial guess to solve difficult-to-converge systems. For example, to simulate a large two-probe system, one may adopt the following strategy to achieve a smooth convergence: (1) Divide the central region into many small pieces, each of which can be easily converged with a bulk calculation; (2) Carry out a supercell calculation of the central region by using the solutions of small pieces as the initial guess [31]; (3) Carry out an equilibrium two-probe calculation by using the supercell solution as the initial guess; (4) Carry out nonequilibrium two-probe calculation by using the equilibrium two-probe solution as the initial guess.

Tip 2: Using proper k-sampling and E-sampling. As illustrated in Fig. 6.8, discrete uniform k-sampling has the physical meaning of constructing a cyclic supercell. The finer the k-sampling, the larger the supercell, and the smoother the density of states. Therefore using finer k-sampling may help to improve the convergence. Generally nonequilibrium self-consistent calculations need more k-points than equilibrium ones because the E-points on the integral path are closer to the real axis. Also keep in mind that the uniform k-sampling is more stable than the symmetric k-sampling in self-consistent calculations.

On a particular k-point, the 3d two-probe system is reduced to a 1d two-probe system. As demonstrated in Appendix A.2, the 1d two-probe system may have three types of sharp features in the density of states: van Hove singularities, interface resonances, and quasi-bound states. In equilibrium,

www.ebook3000.com

one can avoid the sharp features by evaluating the energy integral on the complex contour. In nonequilibrium, however, one has to work on the real axis in the energy window of bias voltages (see Fig. 6.6). So the energy mesh of the real axis integral must be sufficiently dense to capture the shape features. Otherwise the energy integral may contain a random numerical error leading to instability of the self-consistent calculation. In addition to dense E-sampling, one may also include a small imaginary energy $i\eta$ to smear the sharp features. Here $i\eta$ is a phenomenological term to simulate the dephasing mechanisms.

Tip 3: Adopting an effective mixing scheme. Firstly, one needs to tune the mixing parameters to make the system converge smoothly. The linear mixer is fairly stable. One can always expect to achieve convergence by decreasing the mixing rate. The Anderson mixer is highly efficient. By using the Anderson mixer, the error of solution decays exponentially in the vicinity of the final solution. A hybrid mixing scheme may take the advantage of both: One may first run N steps of linear mixer with a small mixing rate β to reduce the error, and then switch to the Anderson mixer to accelerate the convergence. The optimal values of N and β depend on the system size and material properties.

Secondly, one needs to manage the mixers of different variables to achieve the convergence *adiabatically*. In the presence of disorder, both atomic potential and coherent potential are to be solved. To avoid the interference of the two quantities, one may first iterate the coherent potential for a few steps and keep the atomic potential frozen at this stage. Once the coherent potential is reasonably good, one may iterate the atomic potential and the coherent potential simultaneously. The key is to make the variables catch up to each other and converge simultaneously. The trick also applies to the convergence of atomic potential and Fermi level in bulk systems.

7.9 Error analysis

Error analysis is important to evaluate the validity and applicability of the obtained results. In NanoDsim, the main sources of error are: (1) the error due to the XC-functional, (2) the error due to the incomplete basis set, (3) the error due to the discretization, (4) the error due to the atomic sphere approximation, (5) the error due to the linearization, and (6) the error due to the nonequilibrium coherent potential approximation. Below we shall analyze these sources of error one by one.

The error due to the XC-functional is common to all DFT calculations.

Although the Hohenberg–Kohn theorem guarantees the existence of an exact XC-functional, it does not give us any details about it. All available XC-functionals are extracted from the results of uniform electron gas and turn out to be pretty good for a large number of materials. Nevertheless the conventional XC-functionals (LDA, GGA) are not always satisfactory for the purpose of quantum transport. For example, both LDA and GGA seriously underestimate the band gap of Silicon. To solve the problem, one may try other types of XC-functional. For the case of Si, MBJ can obtain accurate band gap as well as good effective mass by tuning one phenomenological parameter (see Section 8.3).

The error due to the incomplete basis set is common to all atomic orbital methods. In the plane wave method or the real space method, one can reduce the error by increasing the grid density in the k-space or the real space. In the atomic orbital methods, the basis set is far from complete and there is no *systematical* way to approach the completeness. Generally the larger the basis set, the more accurate the result. The proper basis size depends on the shell structure of elements. For Cu ($3d^{10}4s^1$) and Co ($3d^74s^2$), including *spd*-basis in the valence orbitals is good enough for the purpose of quantum transport simulation [32].

The error due to the discretization is common to all numerical simulations. Due to limited computing resources, one has to use a finite number of E-points, k-points, and radial mesh points. Theoretically it is hard to estimate the effects of these parameters on the accuracy of final results. Numerically one can always increase one of the parameters to check its influence on the final results. Here are some typical values of the parameters: On the complex contour, 20 to 40 E-points are good enough; On the real axis, E-point number needs to be comparable to $\frac{V}{\eta}$ where V is the applied voltage and η is the small imaginary part of the energy. In self-consistent calculations, the k-point number multiplied by the unit cell size is about $5nm$ for insulators and $10nm$ for metals. In post-analysis calculations, k-point number depends on the material properties, ranging from 40^2 (Si pn-junction) to 1000^2 (Fe/MgO/Fe junction). For LDA and GGA calculations, 400 radial mesh points are good enough; For MBJ calculations, 800 radial mesh points are necessary to evaluate the kinetic energy density accurately.

The error due to the atomic sphere approximation is specific to the muffin-tin orbitals. For close-packaged structures such as FCC, BCC, HCP, the approximation is rather good and the error is negligible compared to other sources of error. For regular non-close-packed structures such as

www.ebook3000.com

diamond, zincblende, Wurtzite, and graphene, the approximation is still acceptable as long as the vacuum is filled with empty spheres. For irregular structures such as surfaces and interfaces, the approximation must be used with care. Atomic spheres need to be arranged properly to reproduce the correct electronic structure. To make it easier, NanoDsim provides an ASA-optimizer to tune the sphere positions and radiuses automatically to match a target band structure. Details of ASA schemes for regular structures and the ASA-optimizer for irregular structures are presented in Appendix A.17.

The error due to the linearization is specific to the LMTO method. For metals, quantum transport is dominated by electrons near the Fermi level, and hence the linearization is a good approximation. For semiconductors and insulators, due to finite energy gap, the energies involved in quantum transport can be out of the linear regime. Sometimes it is necessary to analyze quantum transport with the MTO method to avoid the error arising from the linearization. Alternatively one can also adopt multi-panel technique to expand the energy-dependent muffin-tin orbital with multiple linearization centers. The technique can be viewed as a compromise between the LMTO and the MTO method.

The error due to the nonequilibrium coherent potential approximation is specific to disordered systems. The approximation is derived by assuming that the correlation of the scattering on different disorder sites is negligible. The approximation can be applied to a *full range* of disorder concentration from $x = 0$ to $x = 1$. Therefore it is superior to the virtual crystal approximation and the supercell method. As a verification of the accuracy, the nonequilibrium coherent approximation is compared to the supercell method at some intermediate concentrations. It is found that the results of the two methods agree very well with each other, and the details are given in Section 8.5.

Finally we would like to mention that the errors are not independent and may cancel with each other. We don't have to examine the errors one by one. Instead we can "calibrate" the electronic structure by comparing to the results of other methods or experimental data in bulk systems. The key point is that the same atomic spheres and system parameters will be used in two-probe systems. It is reasonable to expect that the results of a quantum transport simulation are reliable as long as the electronic structures have been "calibrated" properly.

Bibliography

[1] Y. Zhu, L. Liu, and H. Guo, technical report of the industrial research assistance program *NAQEDA: a software tool for nanoelectronics modeling and design*, NRC-IRAP Project #700796 (2012).

[2] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).

[3] S. Y. Wu, C. S. Jayanthi, Physics reports, **358**, 1 (2002).

[4] J. M. Soler, E. Artacho, J. D Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, J. Phys.: Condens. Matter **14** 2745, (2002).

[5] P. D. Haynes, C.-K. Skylaris, A. A. Mostofi, and M. C. Payne, Phys. Stat. Sol. (b) **243**, 2489 (2006).

[6] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider, J. Chem. Phys. **129** 014109 (2008).

[7] D. R. Bowler and T. Miyazaki, J. Phys.: Condens. Matter **22**, 074207 (2010).

[8] W. Kohn, Phys. Rev. Lett. 76, 3168 (1996).

[9] S. Goedecker, Phys. Rev. B **58**, 3501 (1998).

[10] S. Ismail-Beigi and T. Arias, Phys. Rev. Lett. **82**, 2127 (1999).

[11] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[12] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.

[13] T. Ozaki, Phys. Rev. B **74**, 245101 (2006).

[14] R. Takayama, T. Hoshi, T. Sogabe, S.-L. Zhang, and T. Fujiwara, Phys. Rev. B **73**, 165108 (2006).

[15] T. Hoshi and T. Fujiwara, J. Phys.: Condens. Matter **21**, 064233 (2009).

[16] Y. Zhu, L. Liu, and H. Guo, unpublished (2009).

[17] Y. Zhu, L. Liu. and H. Guo, unpublished (2010).

[18] I. S. Duff, A. M. Erisman, J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, New York, 1986.

[19] To be precise, the eliminations of odd level separators and even level separators have different prefactors. Nevertheless the scaling of the cost estimate remains unchanged.

[20] N. I. M. Gould, J. A. Scott, and Y. Hu, ACM Transactions on Mathematical Software, Vol. 33, No. 2, Article 10, (2007).

[21] A. George, SIAM J. Numer. Anal. **10**, 345 (1973).

[22] S. Li, S. Ahmed, G. Klimeck, and E. Darve, J. Comp. Phys. **227**, 9408 (2008).

[23] L. Lin, J. Lu, L. Ying, R. Car, and W. E, Commun. Math. Sci. **7**, 755 (2009).

[24] S. Y. Wu, J. Cocks, C. S. Jayanthi, Phys. Rev. B **49**, 7957 (1994).

[25] D. Waldron, L. Liu, and H. Guo, Nanotechnology **18**, 424026 (2007).

[26] S. Cauley, J. Jain, C.-K. Koh, and V. Balakrishnand, J. Appl. Phys. **101**, 123715 (2007).

[27] D. E. Petersen, S. Li, K. Stokbro, H. H. B. Sørensen, P. C. Hansen, S. Skelboe, E. Darve, J. Comp. Phys. **228**, 5020 (2009).

[28] H. H. B. Sørensen, P. C. Hansen, D. E. Petersen, S. Skelboe, and K. Stokbro, Phys. Rev. B **77**, 155301 (2008).

www.ebook3000.com

[29] D. Waldron, Ph.D. thesis, McGill University, 2007.
[30] J. Maassen, M. Harb, V. Michaud-Rioux, Y. Zhu, and H. Guo, Proc. IEEE **101**, 518 (2013).
[31] If the left and right leads are made of the same material, one can repeat the central region periodically along the transport direction to define the supercell. If the left and right leads are made of different materials, one can put the central region and its mirror image together as a supercell to avoid an abrupt interface between the left and right leads.
[32] K. Xia, M. Zwierzycki, M. Talanana, P. J. Kelly, and G. E. W. Gauer, Phys. Rev. B **73**, 064420 (2006).

Chapter 8

Kaleidoscope of the physics in disordered systems

Confucius said "it is a great pleasure to learn something and apply it to solve realistic problems". In previous chapters, we have learned the theory and the algorithms of the NanoDsim package. Now it is time to apply the package to study the quantum transport in nanoelectronic devices. We are especially interested in how disorder scattering at the surfaces, interfaces, dopant sites, and structural defects affect the electronic structures and transport properties.

In this chapter, we shall investigate four bulk systems and five two-probe systems to illustrate the colorful device physics. Although the device materials involve metals, semiconductors and insulators, the governing principles are unified and implemented in one software package. This is the art of first principles quantum transport simulation.

8.1 Simple examples: bulk Cu, Fe, Co, Ni

To get started, we study a few simple clean bulk systems in this section. To carry out a simulation with NanoDsim, the first step is to prepare an input file (e.g., input.txt) containing the atom positions and control parameters. The format of the input file is defined by the input file templates located in *Manual/InputTemplate*. Once the input file is available, one simply issues the command *nanodsim input.txt* in MATLAB and everything goes automatically.

The very first example is bulk Cu which has the FCC structure with the unit cell vectors

$$\mathbf{v}_1 = a\left(0, \frac{1}{2}, \frac{1}{2}\right),$$

www.ebook3000.com

$$\mathbf{v}_2 = a\left(\frac{1}{2}, 0, \frac{1}{2}\right),$$
$$\mathbf{v}_3 = a\left(\frac{1}{2}, \frac{1}{2}, 0\right),$$

where $a = 6.8219$ $(3.61\mathring{A})$ is the lattice constant. Due to the ASA, the volume of atomic spheres equals the volume of unit cell

$$\frac{4\pi}{3}R^3 = a^3 \det\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix},$$

and the Wigner–Seitz radius of Cu atom is obtained as $R = 2.6660$. In addition to the geometric information, one also needs to set up various control parameters, such as the cutoff length of the structure constant, the basis size and radial mesh in atomic spheres, the number of energy points on the integral contour, and the mixing scheme in the self-consistent iteration. These details are documented in the user's manual.

The simulation will be carried out in two phases, the self-consistent calculation and the post-analysis calculation. The goal of the self-consistent calculation is to solve a Hamiltonian which contains all the dynamic information of the system. The goal of the post-analysis calculation is to compute physical quantities based on the obtained Hamiltonian. Generally the self-consistent calculation is more difficult than the post-analysis calculation due to unpredictable convergence issues.

In the bulk Cu example, the self-consistent calculation proceeds as follows

```
-------------------------------------------------------------------------------
step       charge d(Potential)  d(Evalence)       d(Ef)    d(Omega_r)   time
   1     6.761427     0.592638     0.440519    0.076157         NaN 17:34:19
   2     9.843039     0.323121     0.022252    0.014551         NaN 17:34:20
   3    10.749818     0.255508     0.016553    0.016853         NaN 17:34:22
  ..    .........     ........     ........    ........         ... ........
  25    11.000610     0.000136     0.000086    0.000047         NaN 17:34:57
  26    11.000023     0.000059     0.000020    0.000002         NaN 17:34:58
-------------------------------------------------------------------------------
```

One can see that the errors of charge, potential, valence energy, and Fermi level decrease with the iterations, and reach the prescribed accuracy after 26 steps. After the self-consistent calculation, a mat file, *NanoData.mat*, is generated in the working directory. It contains all the necessary data (potential parameters, structure constant, etc.) to construct the LMTO Hamiltonian.

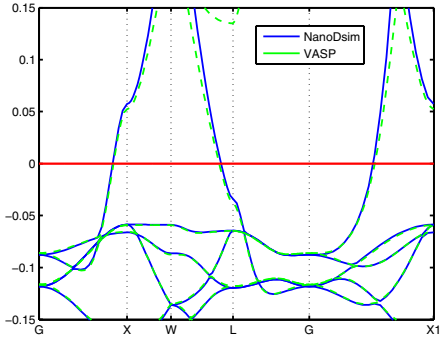


Fig. 8.1 Band structure of bulk Cu calculated with NanoDsim (solid line) and VASP (dashed line). The Fermi level is indicated by a horizontal line.

To read the physics out of the Hamiltonian, one needs to carry out a post-analysis calculation for the band structure (see Fig. 8.1). As a comparison, the band structure is also calculated independently with VASP [1] and the results are shown on the same plot. One can see that the two band structures are very close to each other, verifying the implementation of NanoDsim.

In bulk Cu, the spin-↑ and spin-↓ bands are degenerate. In ferromagnetic metals, however, the spin-↑ and spin-↓ bands are split. Let us now reveal the magic of ferromagnetism in bulk Fe, Co, Ni. Fig. 8.2 shows the spin-dependent density of states in bulk Fe (BCC), Co (HCP), and Ni (FCC) calculated with NanoDsim. It is observed that the density of states for spin-↑ and spin-↓ are split as if there were an "internal molecular field". The origin of the "internal molecular field" remained a mystery until Heisenberg introduced the concept of the exchange interaction. Due to the exchange interaction, the symmetry between the spin-↑ and spin-↓ is broken spontaneously, leading to ferromagnetism. In numerical simulations, one needs to make an initial "kick" to break the symmetry: An initial spin polarization is set up in the input file, and the final converged magnetization does not rely on the initial value. Quantitatively, the obtained magnetic moments are summarized as follows

atom	**structure**	**a**	**NanoDsim**	**VASP**
Fe	BCC	$2.87\mathring{A}$	$2.22\mu_B$	$2.19\mu_B$
Co	HCP	$2.51\mathring{A}$	$1.55\mu_B$	$1.56\mu_B$
Ni	FCC	$3.52\mathring{A}$	$0.55\mu_B$	$0.58\mu_B$

www.ebook3000.com

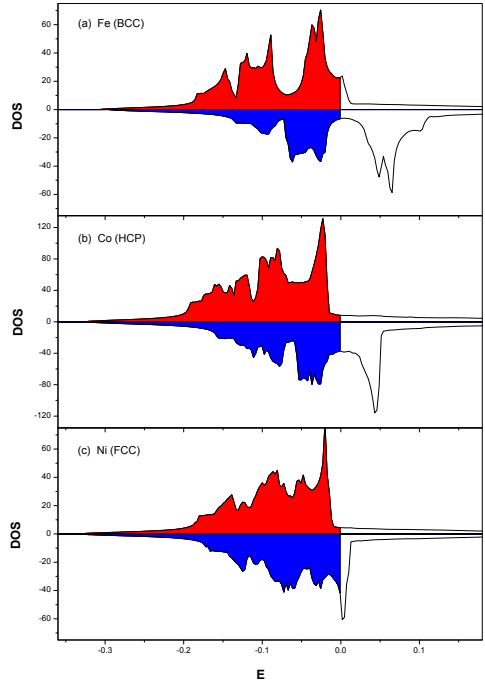


Fig. 8.2 Spin-dependent density of states for bulk Fe (BCC), Co (HCP), and Ni (FCC). The area below the Fermi level is filled with red for spin-↑ and blue for spin-↓.

Again the results agree very well between NanoDsim and VASP, verifying the implementation of NanoDsim in the presence of spin-polarization.

To sum up, by using simple examples of bulk Cu, Fe, Co, Ni, we have demonstrated that NanoDsim can obtain the correct electronic structure of bulk systems with or without spin-polarization.

8.2 CPA vs supercell: Cu/Co alloy

One of the unique features of NanoDsim is the capability to simulate disordered systems. We have studied clean bulk Cu and clean bulk Co in Section 8.1, now let us investigate the Cu/Co alloy in which Cu atoms and Co atoms are distributed randomly on an FCC lattice (see Fig. 8.3).

The first step is to solve the Hamiltonian of the Cu/Co alloy self-consistently. The calculation is carried out in a primitive cell having a disorder site Cu_xCo_{1-x}. By using the CPA technique (see Section 2.8), the

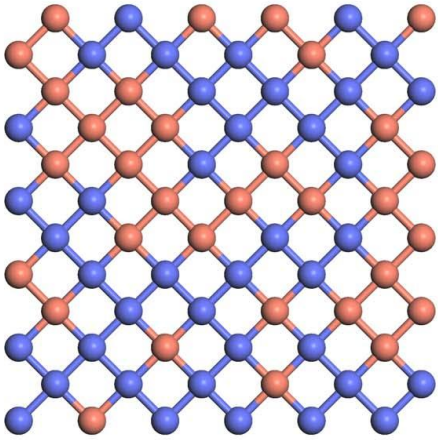


Fig. 8.3 The atomic view of the Cu/Co alloy. The Cu/Co alloy has the FCC structure with lattice constant $a = 3.614\mathring{A}$.

potential parameters of Cu atom and Co atom are solved in an environment of randomly distributed Cu/Co atoms.

Next we proceed to calculate the density of states of the Cu/Co alloy. As a comparison, the calculation is carried out with three methods: the supercell method, the CPA method, and the VCA method. The supercell method is to construct a large supercell and put Cu and Co atoms randomly on the lattice sites according to the probability. The CPA method is to calculate the disorder-averaged density of states by solving the coherent potential. The VCA method is to replace the disorder site by an averaged atom. Both the CPA method and the VCA method work with the primitive cell and hence the computational cost is much lower than that of the supercell method. The accuracy of the three methods are investigated in Fig. 8.4. One can see that the CPA method provides a good approximation to the supercell method, while the VCA method deviates remarkably from the supercell method due to the neglect of disorder scattering.

Finally we study the band structure of the Cu/Co alloy. In principle, the concept of band structure is only applicable to clean bulk systems which have translational symmetry. In disordered bulk systems, such as Cu_xCo_{1-x}, the periodicity is destroyed by the randomness. On the other hand, the band structure still makes sense in the limit of $x \ll 1$ where the system is composed of one type of host atoms with small amounts of impurity atoms. In that situation, the band structure closely mimics that of the host material. The effect of impurity atoms is to modify the shape of the

www.ebook3000.com

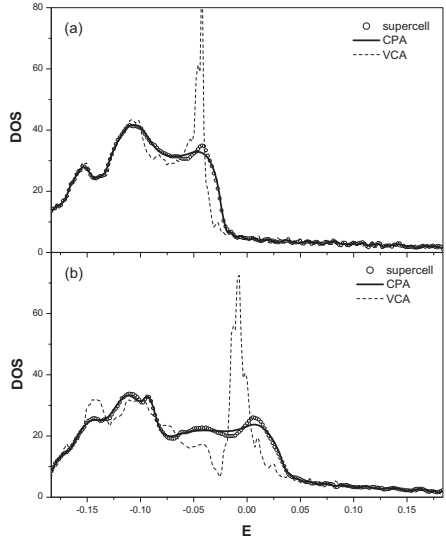


Fig. 8.4 The density of states of $Cu_{0.5}Co_{0.5}$ alloy for (a) spin-↑ and (b) spin-↓. Three methods are used in the calculation: the supercell method, the CPA method, and the VCA method. In the supercell method, a cubic unit cell containing 500 atoms is used to simulate a random alloy.

bands and induce an effective broadening to each band. With increasing x, the bands are smeared more and more severely, and the conventional band structure needs to be generalized to the CPA band structure (see Section 3.12). Fig. 8.5 shows the CPA band structure of $Cu_{0.5}Co_{0.5}$ alloy. Even at such a high concentration, the spin-↑ bands are still visible, while some spin-↓ bands are barely resolved. It indicates that spin-↓ electrons have much stronger disorder scattering than spin-↑ electrons in the Cu/Co alloy. Quantitatively, the lifetime τ of each Bloch state can be deduced from the broadening δ of the CPA band by the uncertainty relation $\tau = \frac{1}{\delta}$.

To sum up, the CPA-LMTO method has been verified in the Cu/Co alloy. It is the basis for the simulations of equilibrium disordered systems. In Section 8.5, the CPA-LMTO method will be generalized to the nonequilibrium situation and verified in a Cu/Co interface.

8.3 Si with uniaxial strain

In Section 8.1 and 8.2, we have studied the electronic structures of metals; In this section and the next section, we shall study the electronic structures

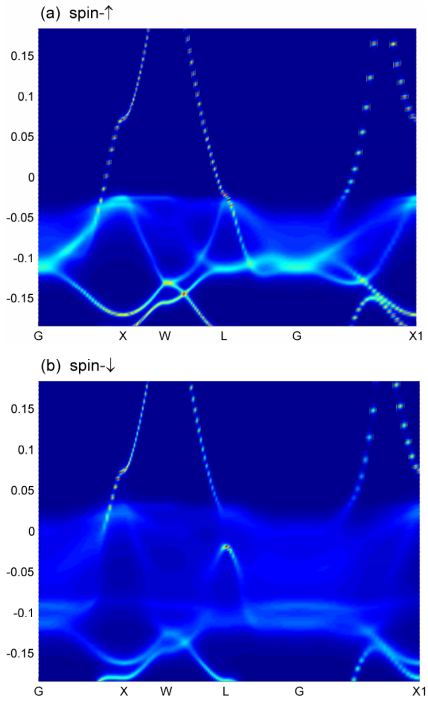


Fig. 8.5 The CPA band structure of the $Cu_{0.5}Co_{0.5}$ alloy for (a) spin-↑ and (b) spin-↓.

of semiconductors. This section will focus on bulk Si which is the basis of the whole semiconductor industry.

There are two obstacles to applying the LMTO method to simulate semiconductors. First, the LMTO method was originally developed for metals having close-packed structures such as FCC, BCC and HCP. Semiconductors usually have non-close-packed structures such as diamond, zincblende, and Wurtzite. One needs to fill up the vacuum with empty spheres properly to apply the LMTO method. The ASA-schemes for regular non-close-packed structures of semiconductors are available in Appendix A.17. Second, it is well known that either LDA or GGA seriously underestimates the band gap of semiconductors. Advanced techniques such as GW or hybrid functional are too expensive to study nanostructures with even though they can produce the correct band gap for most semiconductors. Alternatively NanoDsim implements a recently proposed modified Becke-Johnson (MBJ) semilocal exchange potential which requires a similar computational cost to that of LDA [2].

www.ebook3000.com

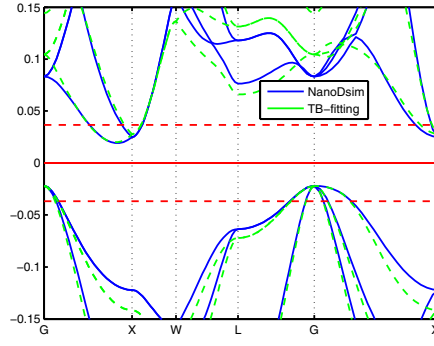


Fig. 8.6 Band structure of bulk Si calculated with NanoDsim (blue) and the tight-binding *spds** model (green). The two band structures are in a perfect agreement within the energy range of $(-1V, 1V)$ with respect to the Fermi level, as guided by the red horizontal lines.

MBJ contains one parameter C which can be calculated from an empirical formula [2]. In the LMTO method, however, it is more convenient to assign the parameter C directly to each atomic sphere. Since the MBJ band gap is a monotonic function of C, one can tune the C parameter to fit the experimental band gap. Different from other parameter fitting methods, LMTO-MBJ can produce the correct band gap as well as effective masses by tuning only *one* empirical parameter. Fig. 8.6 shows the band structures of bulk Si calculated with NanoDsim and the tight-binding (TB) *spds** model. The tight-binding model contains 19 parameters to fit the experimental data of band edges at Γ, X, L, $X_{\min}$ and various effective masses within an accuracy of 5% [3]. One can see that the two band structures accurately agree with each other in the energy range of $(-1V, 1V)$ with respect to the Fermi level. Since the transport in semiconductors are dominated by the charge carriers in this energy range, the accuracy of LMTO-MBJ is well acceptable as compared to the experimental data.

Although both the TB parameter fitting approach and the first principles approach are capable of producing satisfactory electronic structure, the latter has the advantage of making predictions for various situations within a *unified* formalism. As a demonstration, let us consider the influences of uniaxial strain on the electronic structure of bulk Si. In the TB parameter fitting approach, one needs to model how the TB parameters change with the structural distortion and induce more phenomenological parameters to fit the experimental data [5]. In the first principles approach, one simply

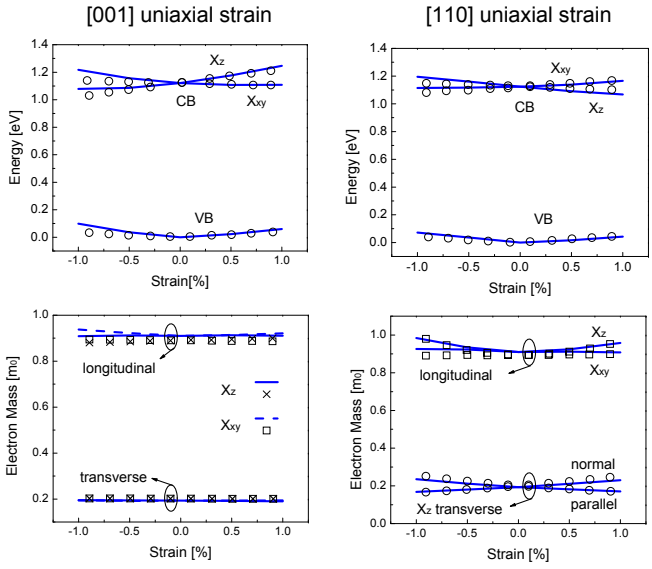


Fig. 8.7 (Adapted from Ref. [4] with permission) The [001] and [110] uniaxial strain effects on the band edges and the effective masses of bulk Si. The band edges are calculated with NanoDsim (lines) and tight-binding model (symbols). The effective masses are calculated with NanoDsim (lines) and VASP (symbols). Here X_z and X_{xy} refer to the X-valleys along and perpendicular to the strain direction.

needs to study how the atoms move in response to the uniaxial strain and leave other complexities to the computer [4]. Fig. 8.7 shows the strain effects on the band edges and effective masses calculated with NanoDsim. For the strain-induced band splitting, NanoDsim's results are in a good agreement with those of TB model which fits the experimental data. For the strain effects on the effective masses, NanoDsim's results also agree very well with those of VASP. By using the first principles simulation, Ref. [4] re-discovered that the [110] strain is much more efficient to modulate the effective masses of Si which had been reported in Ref. [6].

To sum up, we have demonstrated that NanoDsim can reach similar accuracy to other well-established atomic models in bulk systems. The real power of NanoDsim is in the simulation of nanostructures and quantum transport. Further study on Si transistors with channel length 10.8 nm indicates that the uniaxial strain may have significant impact on the device performance. Interested readers are referred to Ref. [4] for more details.

www.ebook3000.com

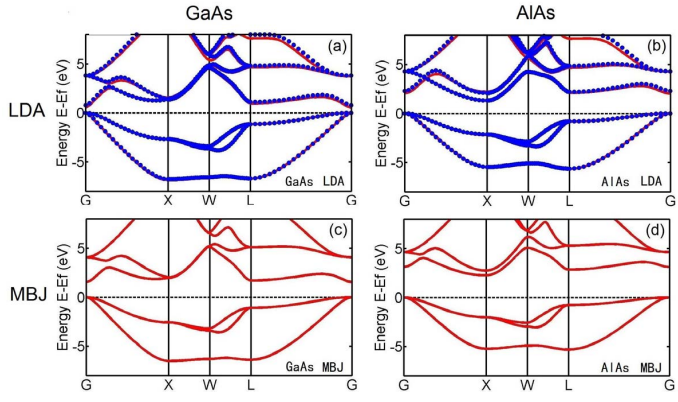


Fig. 8.8 (Adapted from Ref. [7] with permission) Band structures of GaAs (left column) and AlAs (right column). (a) and (b) are calculated with LDA using NanoDsim (line) and VASP (symbol). (c) and (d) are calculated with MBJ using NanoDsim.

8.4 Band offset of GaAs/$Al_xGa_{1-x}As$ heterojunctions

In addition to Si and Ge, III-V compound semiconductors are also widely used in electronic and optoelectronic devices. In this Section, we shall investigate the material properties of GaAs, AlAs, $Al_xGa_{1-x}As$, and the band offset in the heterojunctions of GaAs/$Al_xGa_{1-x}As$ [7].

We first study the band structures of pure GaAs and AlAs. Fig. 8.8 shows the results calculated with NanoDsim and VASP. As pointed out in Section 8.3, LDA seriously underestimates the band gap of GaAs and AlAs (see Fig. 8.8a and 8.8b), while MBJ can produce the correct band gap as well as dispersion relation (see Fig. 8.8c and 8.8d). Quantitatively the energies (in eV) of the conduction band minima at Γ, X, and L with respect to the valence band maximum at Γ are summarized below (adapted from Ref. [7] with permission)

system	band gap	VASP (LDA)	NanoDsim (LDA)	NanoDsim (MBJ)	Experiment
GaAs	$E_c(\Gamma) - E_v(\Gamma)$	0.493	0.761	1.518	1.519
GaAs	$E_c(X) - E_v(\Gamma)$	1.334	1.346	1.960	1.981
GaAs	$E_c(L) - E_v(\Gamma)$	0.948	1.100	1.691	1.815
AlAs	$E_c(\Gamma) - E_v(\Gamma)$	2.014	2.300	3.099	3.099
AlAs	$E_c(X) - E_v(\Gamma)$	1.312	1.307	2.258	2.24
AlAs	$E_c(L) - E_v(\Gamma)$	2.086	2.191	2.835	2.46

where the experimental data are from Ref. [8]. One can see that the MBJ band structure remarkably improves the LDA's and agrees very well with the experimental data.

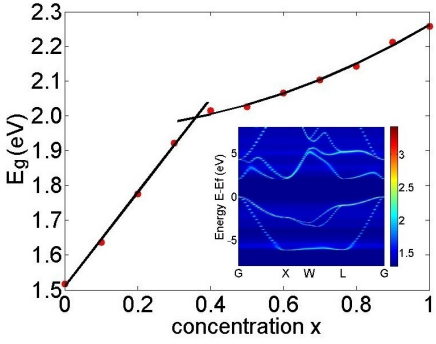


Fig. 8.9 (Reproduced from Ref. [7] with permission) Band gap of the alloy $Al_xGa_{1-x}As$ as a function of x calculated with NanoDsim (symbols). The solid lines are the fittings to numerical data in the direct-gap regime and the indirect-gap regime. Inset: the CPA band structure for the alloy $Al_xGa_{1-x}As$ at $x = 0.36$, showing the crossover from the direct-gap to the indirect-gap.

Next we study the band gap of doped $Al_xGa_{1-x}As$. By changing the doping concentration x from 0 to 1, the electronic structure evolves continuously from GaAs to AlAs. Notice that GaAs is a direct-gap semiconductor whose Γ valley is lower than the X valley in the conduction band, while AlAs is an indirect-gap semiconductor whose X valley is lower than the Γ valley in the conduction band. Consequently a transition from direct-gap semiconductor to indirect-gap semiconductor is expected at some intermediate doping concentration. Indeed it was found experimentally that the transition occurs around $x = x_c = 0.38$. For $x < x_c$, the direct band gap scales linearly with x; For $x > x_c$, the indirect band gap scales quadratically with x. Fig. 8.9 shows the calculated band gap of $Al_xGa_{1-x}As$ using NanoDsim. The results are highly consistent with the experimental findings: The band gap has a linear scaling for $x < \tilde{x}_c$ and a quadratic scaling for $x > \tilde{x}_c$, where $\tilde{x}_c$ is determined to be 0.36 from the fittings of NanoDsim data.

Finally we study the band offset of the GaAs/$Al_xGa_{1-x}As$ heterojunctions. The band offset describes the relative alignment of the electronic bands across the semiconductor interface. An accurate determination of the band offset is critical for the analysis of quantum transport as well as the optoelectronic properties. The theoretical calculation of the band offset represents a serious challenge because it involves a small amount of charge transfer between two semiconductors and requires the capability of simu-

www.ebook3000.com

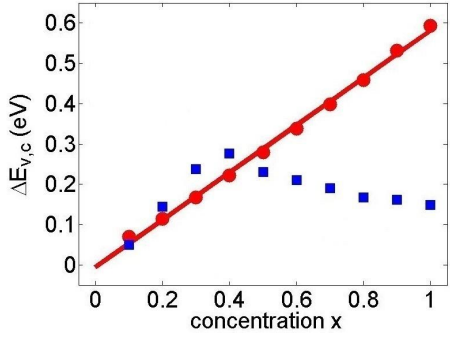


Fig. 8.10 (Adapted from Ref. [7] with permission) VBO (dot) and CBO (square) as a function of x in the GaAs/$Al_xGa_{1-x}As$ heterojunctions. The line is a linear fitting to the VBO data. The calculation is carried out in a superlattice composed of 9 layers of GaAs and 9 layers of $Al_xGa_{1-x}As$.

lating disordered systems. Fig. 8.10 shows the valence band offset (VBO) and the conduction band offset (CBO) in GaAs/$Al_xGa_{1-x}As$ calculated with NanoDsim. The obtained VBO data can be fit by VBO$\approx 0.587x$ eV, which approximately recovers the experimental result VBO$\approx 0.55x$ eV [9]. CBO can be deduced from VBO and $Al_xGa_{1-x}As$ band gap (see Fig. 8.9), which also agrees very well with the experimental data [9]. Although some other DFT codes may also obtain good VBO, most of them fail to produce good CBO due to the incorrect band gap. With the aid of MBJ, NanoDsim is capable of producing good VBO and CBO simultaneously which is important for the nonequilibrium quantum transport.

To sum up, we have investigated the electronic structure of GaAs, AlAs, $Al_xGa_{1-x}As$, and the band offset of GaAs/$Al_xGa_{1-x}As$ heterojunctions. The obtained results are in an excellent agreement with the experimental data, establishing an accuracy benchmark for simulating III-V semiconductors with NanoDsim.

8.5 NECPA vs supercell: Cu/Co interface

In Section 8.2, we have established the accuracy of CPA by comparing it to the supercell method in the Cu/Co alloy; In this section, we shall establish the accuracy of NECPA by comparing it to the supercell method in the Cu/Co interface. The atomic structures and the supercell calculations are adapted from Ref. [10] for the case of single disorder layer. As a byproduct,

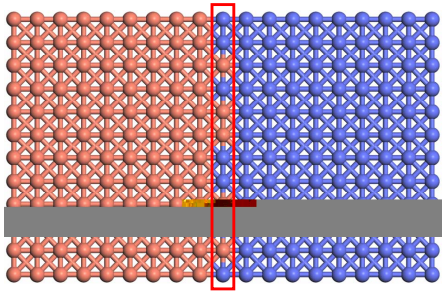


Fig. 8.11 Atomic view of the Cu/Co interface. A monolayer of Cu/Co disorder sites is indicated by the red box.

we shall also interpret the physical meaning of the transmission terms in the NECPA method by connecting them to the supercell method.

Fig. 8.11 shows the atomic structure of the Cu/Co interface, where the left side is FCC Cu and the right side is FCC Co having the same lattice constant $a = 3.614$ Å. To simulate the atomic roughness at the Cu/Co interface, a disorder layer of $Cu_{1-x}Co_x$ is inserted between the pure Cu layer and the Co layer. We are interested in how the roughness affects the conductance of the Cu/Co interface. The research is done with two methods, the supercell method and the NECPA method. In the supercell method [11], one constructs a large supercell in the lateral dimensions, and generates random disorder configurations of Cu and Co according to the concentration x. The conductance is calculated for each disorder configuration and averaged over the ensemble. In the NECPA method, one works with a primitive cell in the lateral dimensions, and takes into account the disorder effect by solving the nonequilibrium coherent potential. Fig. 8.12 shows the conductance calculated with the supercell method (symbols) and the NECPA method (lines) for both spin-↑ and spin-↓. To make a fair comparison, the number of k-points in the NECPA method are made equal to the size of supercell which is 20 by 20. One can see that the symbols are almost on top of the lines, indicating that the NECPA method is an excellent approximation to the supercell method. It is worth mentioning that the computational cost of the NECPA method is just a small fraction of the supercell method.

Next we would like to interpret the physical meaning of the NECPA method by making a connection to the supercell method. Notice that the system has translational symmetry in both the left and right regions where

www.ebook3000.com

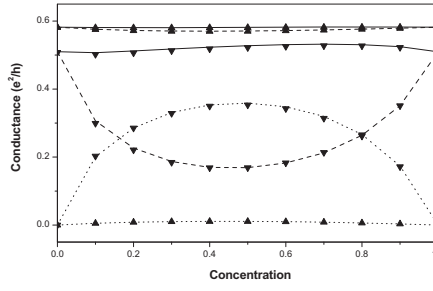


Fig. 8.12 Conductance per unit cell calculated with the supercell method (triangles) and the NECPA method (lines). The total conductance (solid line) is decomposed into the specular part (dashed line) and the diffusive part (dotted line). The spin-↑ and spin↓ conductances are marked by the upper and lower triangles respectively. This figure reproduces Fig. 9 of Ref. [10] with the NECPA method.

k_x and k_y are good quantum numbers. At the disorder layer, due to the randomness, k_x and k_y are no longer good quantum numbers. So an incoming wave with a momentum (k_x, k_y) will be scattered into a group of momentums by the disorder layer. The scattering from (k_x, k_y) to (k_x, k_y) itself is called specular scattering, and the scattering from (k_x, k_y) to $\left(k'_x, k'_y\right) \neq (k_x, k_y)$ is called diffusive scattering. It is straightforward to separate the two parts by solving the scattering states in the supercell method [12]. It is less transparent to recognize the two parts in the NECPA method. Nevertheless we can still find a hint by analyzing Eq. (3.156): The first term of Eq. (3.156) is a simple average and is momentum conserved; The second term of Eq. (3.156) is a vertex correction and is not momentum conserved. So a reasonable guess is that the first term corresponds to the specular part and the second term corresponds to the diffusive part (see Appendix A.16 for more details). The hypothesis is verified numerically in Fig. 8.12: The specular conductance calculated from the supercell method is identical to the simple average in the NECPA method; The diffusive conductance calculated from the supercell method is identical to the vertex correction in the NECPA method. Furthermore it is interesting to observe that disorder scattering has very different effects on the two spin species. For spin-↑ electrons, both specular and diffusive conductance have weak dependence on the disorder concentration. For spin-↓ electrons, both specular and diffusive conductance have strong dependence on the disorder concentration. The reason is that Cu and Co have similar Fermi surfaces for

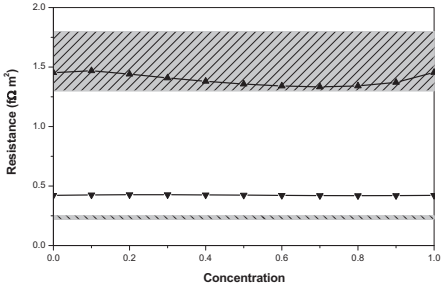


Fig. 8.13 Cu/Co interface resistance as a function of disorder concentration. The solid lines with upper and lower triangles are for the spin-↑ and spin↓ resistance respectively. The shaded area indicates the range of experimental data [13]. This figure reproduces Fig. 12 of Ref. [10] with the NECPA method.

spin-↑ electrons but very different Fermi surfaces for spin-↓ electrons (see Fig. 5 and Fig. 6 of Ref. [10]), resulting in different scattering behaviors.

Finally we would like to compare the theoretical calculations to the experimental measurements by following the analysis of Ref. [10]. In the experiment of Ref. [13], the magnetoresistance is measured for the multilayer structure composed of ferromagnetic metal (Co) and non-magnetic metal (Cu). The interface resistance between Cu and Co can be extracted by using the two current series resistor model [13]. On the theoretical side, we have calculated the conductance of Cu/Co two-probe structure as shown in Fig. 8.11. The resistance is just the reciprocal of the conductance. However the obtained resistance is the total resistance of three effective resistors: the resistor of semi-infinite Cu, the resistor of semi-infinite Co, and the resistor of the Cu/Co interface. To extract the resistance of the Cu/Co interface, we need to subtract the resistances of the semi-infinite Cu and the semi-infinite Co from the total. The resistance of a semi-infinite bulk is half of the resistance of an infinite bulk which is defined as the Sharvin resistance [14]. The calculation of the Sharvin resistance is equivalent to the calculation of conducting channel number, and the procedure has been well explained in Ref. [10]. Fig. 8.13 shows the comparison of the theoretical results and the experimental data. For spin-↓ electrons, the experimental data vary from 1.30 fΩm^2 to 1.80 fΩm^2, and the theoretical results just fit in this range. For spin-↑ electrons, the experimental data vary from 0.22 fΩm^2 to 0.25 fΩm^2, and the theoretical results agree within a factor of two. The discrepancy for the spin-↑ electrons has been analyzed further in Ref. [10].

www.ebook3000.com

To sum up, we demonstrated that the NECPA method and the supercell method are nearly equivalent in the quantum transport calculation of disordered systems. Although both methods can make quantitative predictions comparable to experimental measurements, the NECPA method has much lower computational cost and allows us to vary the disorder concentration continuously. Having established the accuracy of the NECPA-LMTO method, we are ready to study the device physics in the following sections.

8.6 Si transistors with localized doping

With the shrinking of channel length, Si transistors cannot be turned off completely due to the quantum tunneling. The small current in the off-state is called leakage current, leading to undesired heat dissipation and power consumption. Therefore the leakage current is an important issue in the design of nanotransistors. In this section, we shall investigate the possibility of suppressing leakage current and its variation by controlling the dopants' location in the channel region of Si transistors [15, 16].

In quantum tunneling, the current is extremely sensitive to the tunnel barrier height and width. So we first examine the potential profiles of the tunnel barrier in nanotransistors. The electrostatic potential can be calculated with the atomistic model as implemented in NanoDsim which has no phenomenological parameter. On the other hand, the electrostatic potential can also be calculated with the continuum model as implemented in the Sentaurus Device simulator [17] which relies on parameters extracted from the measurements. Fig. 8.14 shows the comparison of the potential profiles of Si transistors with the channel length 10.9 nm. Fig. 8.14a and Fig. 8.14b are for the uniform doping in the channel region. One can see that the results of NanoDsim and Sentaurus are nearly on top of each other, validating the NECPA-LMTO formalism and its implementation in NanoDsim. Fig. 8.14c is for the localized doping in the channel region where the dopants are distributed in a narrow region of 1.1 nm. The agreement between NanoDsim and Sentaurus is still acceptable. The small discrepancy can be attributed to the fact that Sentaurus parameters are extracted for uniform doping and their applicability to the localized doping as narrow as 1 nm is not fully justified. In an extreme case where the channel region only has one precisely located dopant [18], the material is completely "discrete" and one has to rely on the atomistic model to do the simulation.

Next we study the leakage current in Si nanotransistors by using NanoDsim. Although both NanoDsim and Sentaurus can produce cor-

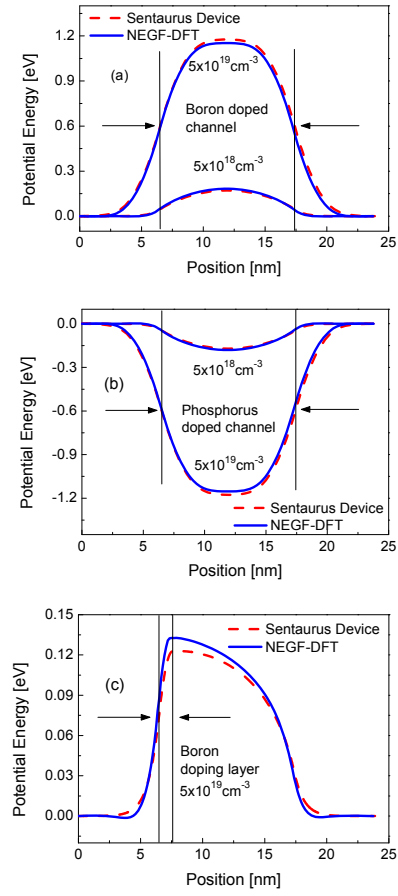


Fig. 8.14 (Adapted from Ref. [4] with permission) Potential profiles of Si transistors with the channel length 10.9 nm. (a) Uniform doping for n-p-n transistor. (b) Uniform doping for p-n-p transistor. (c) Localized doping for n-p-n transistor. The simulations are carried out with Sentaurus (dashed line) and NanoDsim (solid line). The doping concentration of the source and drain is 5×10^{19} cm^{-3}, and the doping concentration of the channel is indicated in the plot.

rect electrostatic potentials, NanoDsim is more relevant to studying the tunneling current which is a pure quantum effect. To simulate transistors with NanoDsim, we construct Si two-probe structures with n-p-n and p-n-p doping profile, whose source, channel, and drain region have the length 6.5 nm, 10.9 nm, and 6.5 nm respectively (see Fig. 8.15a). In the channel

www.ebook3000.com

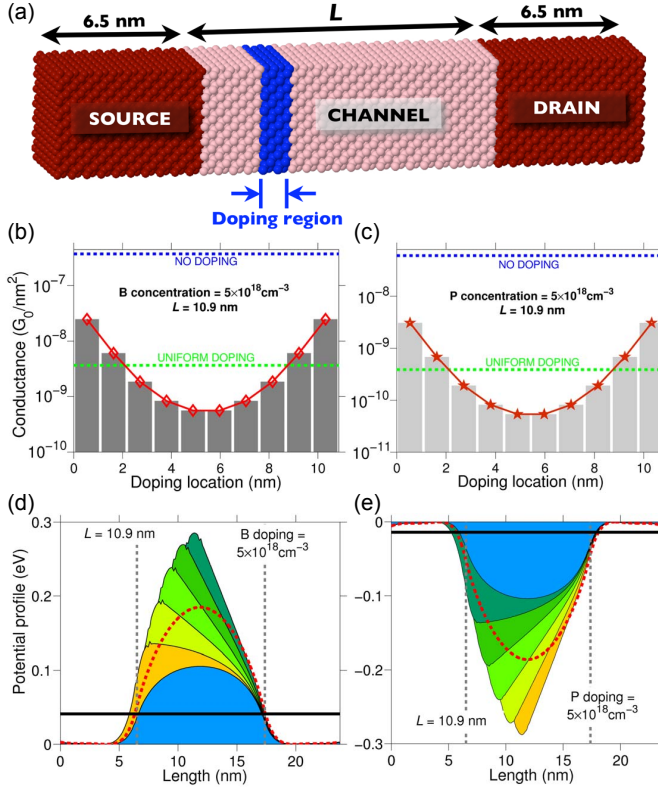


Fig. 8.15 (Adapted from Ref. [15] with permission) (a) Atomic view of the Si nanotransistor channel. The conducting channel has the length $L = 10.9$ nm, doped by a localized doping layer (blue region). The doping layer has the width 1.1 nm and the doping concentration 5×10^{18} cm^{-3}. The doping concentration in the source or drain region is 5×10^{19} cm^{-3}. (b, c) Conductance per area versus doping location for B-doped (left) and P-doped (right) channels. (d, e) Potential profile for B-doped (left) and P-doped (right) channels. The forefront blue area and the red dashed curve are for zero doping and uniform doping respectively. Other potential profiles from front to back are for localized doping whose center moves from the source side to the channel midpoint. The solid horizontal line indicates the position of the Fermi level.

region, B or P atoms are randomly doped to a narrow layer of 1.1 nm. The equilibrium conductance is calculated to characterize the leakage current in the off-state [19]. Fig. 8.15b and 8.15c show the off-state conductance as a function of doping layer location. It is observed that the off-state conductance strongly depends on the doping layer location: The maximum

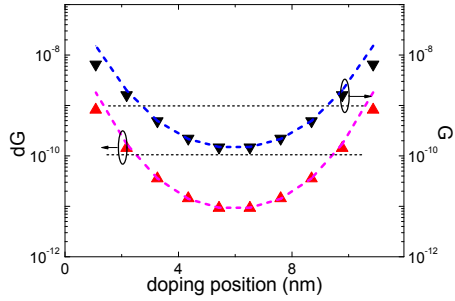


Fig. 8.16 (Adapted from Ref. [16] with permission) Disorder averaged off-state conductance (lower triangles) and its variation (upper triangles). The Si nano-transistor has the channel length $10.9nm$ doped with B atoms in a narrow layer of $1.1nm$. The triangles are the NanoDsim data and the dashed lines are the WKB fitting. As a comparison, the dotted lines show the results of uniform doping.

conductance ($G_{\max}$) is reached when the doping layer is in the vicinity of the source or drain region; The minimum conductance ($G_{\min}$) is reached when the doping layer is in the middle of the channel. As a comparison, the off-state conductance of uniform doping (G_{unif}) and zero doping (G_{zero}) are also calculated and shown in the same plot, and the trend is obtained as $G_{\min} < G_{\text{unif}} < G_{\max} < G_{\text{zero}}$. To understand the trend, Fig. 8.15d and 8.15e plot the corresponding electrostatic potentials. One can see that different doping profiles result in different tunnel barrier heights, and the trend of tunnel barrier heights coincides inversely with the trend of off-state conductances. Although n-p-n transistor and p-n-p transistor have similar doping effect, the off-state conductance of the former is one order of magnitude larger than that of the latter. Quantitative analysis indicates that the Fermi level in n-p-n transistor is further away from the tunnel barrier bottom (0.041 eV) than that of p-n-p transistor (0.014 eV). In the tunneling process, a slightly lower barrier height may result in much larger tunneling current as observed in 8.15b and 8.15c.

Finally we would like to analyze the device-to-device variation of the leakage current. When the channel length is below 10 nm, the device-to-device variability becomes significant because the number of dopants is very small and every individual dopant configuration defines a unique device. Especially the off-state current in nanotransistors may vary from device to device due to the random discrete dopants. Fig. 8.16 shows the average and the variation of the off-state conductance for localized doping and uniform

www.ebook3000.com

doping. One can see that localized doping away from the source or drain region not only suppresses the conductance but also suppresses its variation as compared to that of uniform doping. This is consistent with the intuition that dopants are less random in localized doping than uniform doping. On the other hand, localized doping near the source or drain region may even enhance the conductance variation as compared to that of uniform doping. The reason is that localized doping near the source or drain region results in larger variation of the tunneling barrier height and hence larger variation of the tunneling current. It is worth mentioning that the off-state conductance and its variation can be fitted very well by a Wentzel–Kramers–Brillouin (WKB) model [16], confirming the tunneling nature of the leakage current.

To sum up, we have investigated the doping effects on the leakage current in Si nanotransistors. It is found that different doping profiles (localized doping, uniform doping, zero doping) result in very different tunnel barrier potentials. A localized doping in the middle of the channel region may produce the largest tunnel barrier, the lowest off-state conductance, and the smallest conductance variation.

8.7 Graphene transistors with disorder scattering

Traditional electronic devices are made of Si, for which abundant experimental data and theoretical models are available in the literature. For emerging electronic materials, it is rather difficult to build up device models due to lack of data or parameters. NanoDsim offers the possibility of predicting the transport properties without knowing any material-related parameters. As a demonstration, we shall investigate transport properties and device application of graphene in this section [20, 21].

Graphene is an excellent conductor of electric current. It has been reported that pristine graphene may have mobility as high as 10^5 $\mathrm{cm^2V^{-1}s^{-1}}$ [22]. However, the mobility in fabricated graphene devices is usually several orders lower than this value [23–26]. It is believed that the mobility is reduced mainly by random scattering due to substrate, adatoms, impurities and structural defects. Here we study the impact of impurity scattering on the mobility of N-doped and B-doped graphene [20]. The atomic model is shown in Fig. 8.17a: Both the leads and the central region are made of graphene which is periodic in the x-direction and zigzag along the z-direction. The graphene in the leads is "pure", while the graphene in the central region contains random disorder $\mathrm{C}_{1-x}\mathrm{B}_x$ or $\mathrm{C}_{1-x}\mathrm{N}_x$. The random disorder in the central region is handled by the NECPA theory, while the

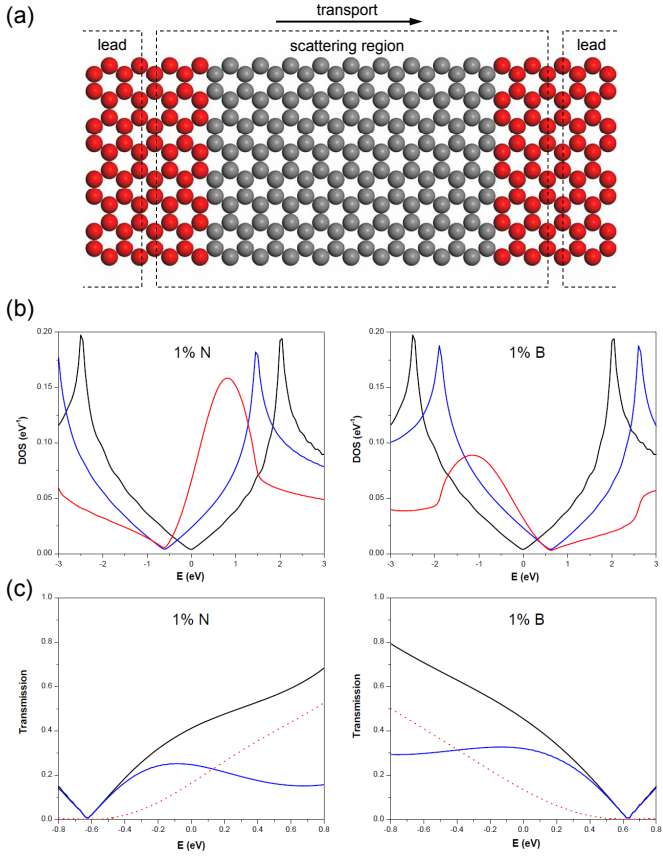


Fig. 8.17 (Reproduced according to Ref. [20]) (a) Schematic plot of a graphene two-probe system. The scattering region is randomly doped with B or N atoms. (b) Local density of states of N-doped (left) and B-doped (right) graphene bulk systems at the doping concentration 1%. Black curve: C atom in the pristine graphene; Blue curve: C atom in the doped graphene; Red curve: N or B atom in the doped graphene. (c) Transmission coefficient of N-doped (left) and B-doped (right) graphene two-probe systems at the doping concentration 1%. Black curve: the total transmission; Blue curve: the specular part; Red curve: the diffusive part. The length of the scattering region is $9.8nm$.

atomic sites in the leads are treated at the VCA level [27]. Fig. 8.17b and 8.17c show the local density of states and the transmission coefficient in N-doped and B-doped graphene. The effects of the dopant atoms are two-fold: First, the dopant atoms induce extra charge carriers in the system

www.ebook3000.com

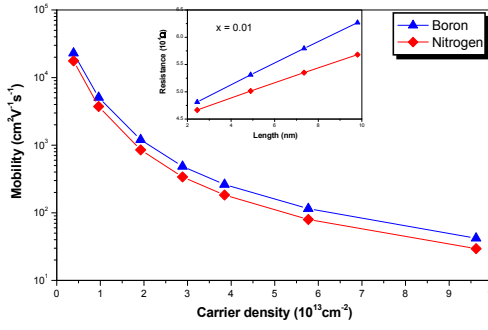


Fig. 8.18 (Reproduced according to Ref. [20]) Mobility as a function of charge carrier density in N-doped graphene (red diamond) and B-doped graphene (blue triangle). Inset: Resistance as a function of device length at the doping concentration $x = 1\%$.

and hence shift the Fermi level. One can see that the C atom in doped graphene has similar density of states to that of pristine graphene except that the Dirac cone center is shifted from $E = 0$ to $E = -0.6$ eV (N-doped) or $E = 0.6$ eV (B-doped). Consequently the Dirac cone center in the transmission coefficient is also shifted from $E = 0$ to $E = -0.6$ eV (N-doped) or $E = 0.6$ eV (B-doped). Second, the dopant atoms provide an effective scattering potential and make the transport no longer ballistic. One can see that there is a resonant peak in the density of states of the N atom and B atom in the energy regime (0 eV, 1 eV) and (−2 eV, 0 eV) respectively. As a result, the transmission coefficient in this energy regime is also suppressed due to disorder scattering.

Now we proceed to calculate the mobility of doped graphene. The procedure is as follows. From the transmission coefficient at the Fermi level ($E = 0$), one obtains the conductance or the resistance of the doped graphene. Notice that the doped graphene behaves like an Ohm resistor whose resistance R increases *linearly* with the length L of the scattering region (see the inset of Fig. 8.18). The slope of R as a function of L gives the resistivity ρ. The mobility μ is related to ρ by $\mu = \frac{1}{\rho n}$ where n is the carrier density. It is assumed that all dopant atoms are fully ionized and hence n is equal to the doping concentration. Fig. 8.18 shows the mobility as a function of carrier density in the doped graphene. One can see that the mobility μ decreases rapidly with increasing doping concentration x. μ is reduced by more than two orders of magnitude when x changes from 0.1% to 2.5%. Moreover, N-doped graphene has lower mobility than B-doped

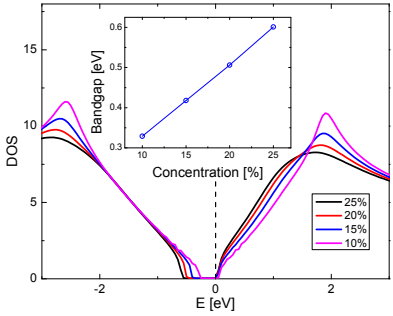


Fig. 8.19 (Reproduced from Ref. [21] with permission) Density of states of BN co-doped graphene at the doping concentration $x = x' = 10\%, 15\%, 20\%, 25\%$. Inset: band gap as a function of doping concentration extracted from the density of states.

graphene because the scattering potential of the N atom is stronger than that of the B atom (see Fig. 8.17b)

Band gap is another concern in device applications. Graphene is gapless around the Fermi level where charge carriers behave like massless relativistic particles. This can be a major obstacle in the device application because transistors made of graphene may have very low on/off ratio. Various methods have been developed to open a band gap in graphene. One of the most appealing methods is to dope graphene with N or B atoms so as to form n-type or p-type semiconductors [23–26]. Here we consider a BN co-doped graphene and investigate the doping effect on the band gap. The primitive cell of graphene has two atomic sites denoted by α and β. Assume that site α is doped with N atoms and the site becomes $C_{1-x}N_x$; site β is doped with B atoms and the site becomes $C_{1-x'}B_{x'}$. The doping concentration x and x' are not necessarily equal. For $x > x'$, the doped graphene will be an n-type semiconductor; For $x < x'$, the doped graphene will be a p-type semiconductor; For $x = x'$, the doped graphene will be an intrinsic semiconductor. Fig. 8.19 shows the density of states and the band gap of intrinsic doped graphene as a function of the doping concentration. One can see that the band gap increases linearly with the doping concentration. In particular, at the doping concentration $x = x' = 10\%$, the band gap Δ is 0.33 eV and the effective mass is $0.1m_e$, which are ideal for device applications.

Having studied the mobility and the band gap of doped graphene, we would like to investigate a realistic design of graphene transistor (see Fig. 8.20a). The source, drain, and channel region of the transistor are

www.ebook3000.com

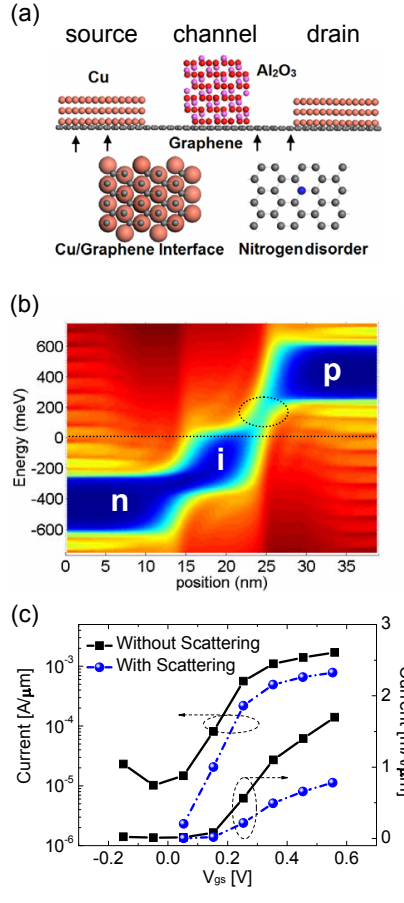


Fig. 8.20 (Adapted from Ref. [21] with permission) (a) Atomic view of the graphene transistor. (b) Local density of states in equilibrium. The Fermi level and the tunnel "straight" are indicated by the horizontal line and the circle. (c) Drain current as a function of gate voltage with or without disorder scattering. The bias voltage is fixed to 0.1 V.

composed of p-type, n-type, and intrinsic doped graphene, respectively. In the source and drain region, Cu is deposited on top of the graphene to serve as electrodes. In the channel region ($L_g = 10$ nm), aluminium-oxide layer is formed to separate graphene and metal gate (not shown). Fig. 8.20b shows the local density of states of the device in equilibrium. One can see that a band gap of 0.33 eV (blue region) is opened up due to the doping effect.

Meanwhile band offset appears at the source/channel and drain/channel interface because of the doping gradient. Applying a bias voltage, the device is driven to nonequilibrium. Nevertheless the current is exponentially small because of the wide tunnel barrier in the channel region. Applying a gate voltage, the band in the channel region is shifted downward, and the tunnel barrier is reduced from the whole channel to a narrow "straight" located at the drain/channel interface (see Fig. 8.20b). As a result, electrons in the drain's valence band are allowed to tunnel through the "straight" to the source's conduction band. Quantitatively the transport current is calculated as a function of gate voltage and the results are shown in Fig. 8.20c. Indeed the current can be modulated effectively by the gate field. The calculation also indicates that the current is reduced significantly by disorder scattering, implying a low mobility in doped graphene.

To sum up, we have studied the mobility and the band gap of N-doped and B-doped graphene, and investigated the feasibility of graphene-based tunneling field effect transistor. This is an illustration of using NanoDsim to study emerging materials and nanoelectronic devices.

8.8 Fe/MgO/Fe tunnel junctions

Electrons have spin as well as charge. Electronic devices exploiting electrons' spin degree of freedom are called spintronic devices. One of the most important spintronic devices is magnetic tunnel junction (MTJ) which may have applications in magnetic read heads, magnetic field sensors, magnetic random access memories, etc. In this section, we shall study the disorder effects on the magnetoresistance of Fe/MgO/Fe tunnel junctions [28].

MTJ is composed of an insulating layer sandwiched by two ferromagnetic layers. The insulating layer is extremely thin so that spin-polarized current can pass through by quantum tunneling. The resistance of the tunnel junction is low if the magnetization of the two ferromagnetic layers are in parallel configuration (PC) and high in anti-parallel configuration (APC). The phenomenon is called tunnel magnetoresistance (TMR), which is quantitatively measured by the ratio

$$TMR = \frac{R_{APC} - R_{PC}}{R_{PC}},$$

where R_{PC} and R_{APC} are the junction resistance in PC and APC respectively. The larger the TMR, the more sensitive the device. The TMR is largely determined by the insulating layer which is only a few nanometers thin. In the earlier generation of MTJ, amorphous AlO_x was used as

www.ebook3000.com

the insulator, and the TMR increased from only a few percent to 70% at room temperature (see Fig. 1 of Ref. [29]). A breakthrough was made in 2004 when crystalline MgO was used as the insulator, and the largest TMR reached 200% at room temperature [30, 31].

Theoretically the first TMR model was proposed by Jullière in 1975 [32], assuming that the tunneling current is proportional to the product of density of states in the ferromagnetic layers. The Jullière model does capture some essential features of MTJ and has been verified experimentally. However, the Jullière model is inadequate to explain the remarkable difference between AlO_x and MgO tunnel barriers since it does not contain any information about the insulating layer. It is necessary to take into account the quantum coherence of the scattering waves as well as the material properties. The first atomic simulation was done by Butler *et al* [33] for Fe/MgO/Fe tunnel junctions, providing a comprehensive understanding of the coherent spin-filtering effect: The Δ_1 band of Fe is fully spin-polarized and the Δ_1 complex band of MgO has the smallest decaying rate. It is the matching of the wave function symmetry that enhances the tunneling of majority spin in PC and results in a huge TMR (see Fig. 7 of Ref. [34]). Notice that the theoretical prediction of huge TMR [33] was made before the experimental observations [30, 31]. It is a good demonstration of the power of first principles quantum transport simulations.

Despite the success, the theoretically predicted TMR value of 10000% is several orders of magnitude larger than the experimental data. Understanding the discrepancy may help to improve the TMR further. Since Fe (100) surface and MgO (100) surface have a lattice mismatch of about 3%, it is generally believed that the atomic defects at the Fe/MgO interface lead to the degradation of TMR (see Fig. 1.4). Experiments were carried out to investigate oxygen vacancies (OV) inside the MgO tunnel barrier [35], and Ref. [36] provided direct experimental evidence of localized defect states inside the MgO energy gap which were due to the OV. Here we investigate the effect of OV on the TMR of Fe/MgO/Fe tunnel junctions. The model two-probe system is shown in Fig. 8.21a: The left and right leads are perfect Fe bulks, and the central region is an Fe/MgO/Fe sandwich structure. Notice that the square lattice of MgO is rotated $\frac{\pi}{4}$ with respect to the Fe square lattice so that O atoms are sitting on top of Fe atoms. The atomic structure is exactly the same as Ref. [33], except that some oxygen vacancies $O_{1-x}Vac_x$ are located either at the interface layer or the interior layer of MgO. Fig. 8.21b and 8.21c plot the TMR as a function of OV concentration for some typical OV distributions. A few observations

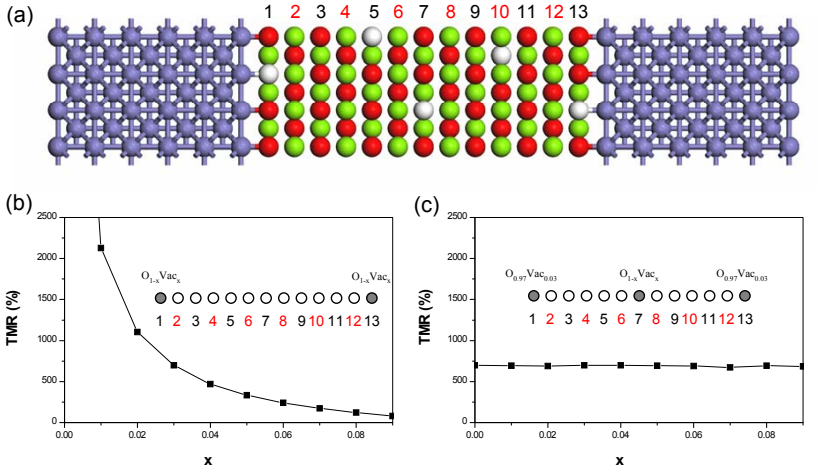


Fig. 8.21 (Reproduced according to Ref. [28]) (a) Atomic structure of Fe/MgO/Fe tunnel junction. Blue spheres: Fe atoms; Red spheres: O atoms; Green spheres: Mg atoms; White spheres: OV sites. (b, c) TMR as a function of OV concentration x for a 13-layer Fe/MgO/Fe tunnel junction. The OV sites (gray dots) are randomly distributed in the interfacial MgO layer and the interior MgO layer. The insets of (b,c) indicate layer components: Black dots represent disordered layers and white dots represent clean layers.

are in order: (1) Without the OV ($x = 0$), the TMR recovers the results of previous calculations for perfect Fe/MgO/Fe tunnel junction [33,37,38]. (2) A mere 4% of interfacial OV can reduce the TMR from the ideal limit of 8740% to 470%. (3) Interior OV has little effect on the TMR. These results indicate that disorder scattering induced by the interfacial OV can be a major mechanism to reduce the TMR.

To gain some insights to the disorder scattering, Fig. 8.22 plots the k-resolved transmission coefficient in the Brillouin zone. In the presence of disorder, transmission coefficient can be decomposed into two parts, the specular part and the diffusive part. The specular part corresponds to the transmission coefficient of disorder averaged transmission amplitude. The diffusive part is the fluctuation on top of the specular part due to disorder scattering (see Appendix A.16). The specular transmission in Fig. 8.22 is quite similar to that of perfect MTJ. The spin-↑ channel of PC has a sharp peak with circular symmetry at the Γ-point, while other channels have a pedal-like pattern with D_4 symmetry. Because of the coherent spin-filtering effect, the spin-↑ channel of PC dominates the specular transmission. The

www.ebook3000.com

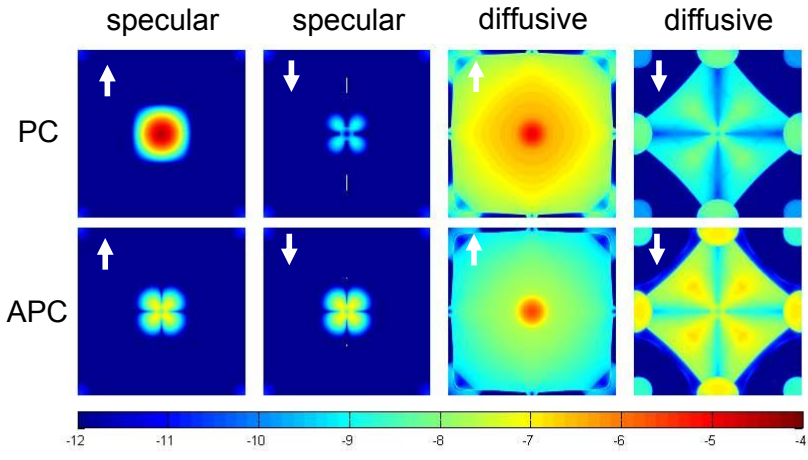


Fig. 8.22 (Reproduced according to Ref. [28]) k-resolved transmission coefficient (log-10 scale) in the Brillouin zone for 13-layer Fe/MgO/Fe tunnel junction. The MgO tunnel barrier has 3% OV in the first and the last layer.

diffusive transmission in Fig. 8.22 spreads much wider than the specular transmission because the transverse momentum is not conserved. Most importantly a new peak with circular symmetry emerges at the Γ-point in the spin-↑ channel. As a result, the conductance of the spin-↑ channel is enhanced considerably for both PC and APC, which effectively dilutes the coherent spin-filtering effect and suppresses the TMR.

The above results are for the equilibrium TMR. By applying a bias voltage, MTJ is driven to nonequilibrium and the TMR is suppressed drastically as shown in Fig. 8.23a. The trend is consistent with previous calculation of perfect MTJ [38]. Taking into account OV at the Fe/MgO interface, the TMR value is much lower but the TMR decay is less steep (see Fig. 8.23b). The voltage effect can be interpreted as follows: The applied voltage shifts the bands of the left and right leads to opposite directions. Consequently the tunneling between the Δ_1 bands are suppressed, resulting in a fast drop of TMR. On the other hand, disorder scattering assists the tunneling between the Δ_1 bands even if they are not at the same energy, resulting in a slow decay of TMR. Experimentally both fast drop [30, 31] and slow decay [39] of the TMR were observed at finite bias voltage.

To sum up, we have investigated the spin-polarized quantum transport in Fe/MgO/Fe tunnel junction in the presence of disorder. It is found that

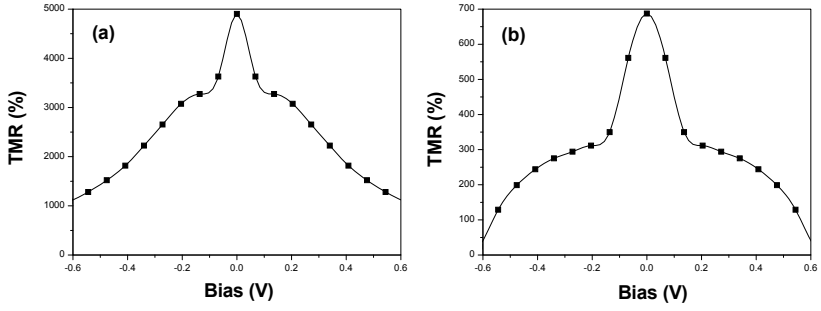


Fig. 8.23 TMR as a function of bias voltage for a 7-layer Fe/MgO/Fe tunnel junction. (a) Perfect MTJ without OV. (b) Disordered MTJ with 3% OV in the first and the last MgO layer. The current calculation is based on an extrapolation of potential profile at a small bias voltage.

a few percent of interfacial OV may result in significant TMR reduction due to disorder scattering. A possible treatment for OV is to dope N atoms in the MgO layer [40]. Further simulations indicate that doping N atoms to Fe/MgO/Fe tunnel junctions not only increases the TMR [28] but also reduces the device-to-device variability [41].

8.9 Cu films with surface scattering

In previous sections, we have studied the transport properties of nano-electronic devices. Those devices are connected by Cu wires in integrated circuits. On a modern chip, the total length of the Cu wires can be more than twenty miles [42]. With the shrinking of device size, the diameter of the Cu wires also shrinks from micrometers to nanometers. When the diameter of the Cu wires is comparable to the electron's mean free path $\lambda = 39$ nm, the resistivity will increase rapidly and result in serious heat dissipation and interconnect delay. Therefore the transport property of Cu wires is also part of the device physics. Among several electron scattering mechanisms, surface scattering has been identified as a major source of the size effect. In this section, we shall investigate the surface scattering effects on the resistivity of Cu films [43, 44].

Experimental growth of Cu films with perfectly flat surfaces has not been possible so far, as even annealed single-crystal Cu 001 layers still show a peak-to-valley roughness of more than 1 nm [45]. This geometrical disorder gives rise to a certain degree of diffusive scattering and increases

www.ebook3000.com

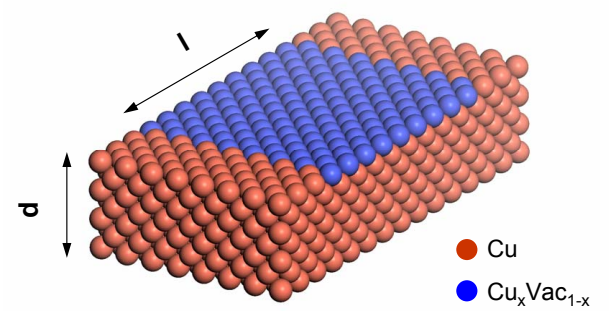


Fig. 8.24 (Reproduced according to Ref. [44]) Atomic structure of a Cu thin film modeled by a two-probe structure of length l, thickness d, and periodic in the width direction. The leads are perfect Cu films without disorder, and the central region contains Cu_xVac_{1-x} disorder sites.

the resistivity. To simulate the surface roughness, the Cu film is modeled by a two-probe system shown in Fig. 8.24: The left and right leads are perfect Cu films, and the central region has a diffusive area whose surface layer is composed of random sites Cu_xVac_{1-x} where Vac refers to vacancy.

Fig. 8.25a shows the resistance R of Cu film as a function of the diffusive region length l for several film thickness d. It is observed that R increases *linearly* with l which is typical behavior of an Ohmic resistor. At first glance, the results are rather surprising because a classical transport behavior is observed in a quantum transport simulation. Further study reveals that disorder scattering in the diffusive regime serves as a dephasing source which wipes out the phase of the scattering wave function. As a result, different parts of the conductor are connected incoherently, leading to the recovery of Ohm's law. Interested readers are referred to Section 2.9 for more discussions on the dephasing mechanism. The resistivity of the Cu film can be deduced from the slope of R as a function of l. Fig. 8.25b shows the resistivity ρ as a function of disorder concentration x for several film thicknesses d. As expected, ρ is zero in the limit $x = 0$ and $x = 1$ where the surface of the Cu film is completely flat. ρ reaches its maximum around $x = 0.5$ where the surface roughness is most severe. Also note that ρ increases rapidly with decreasing d, indicating the importance of surface scattering in thin films.

In the literature, surface scattering is conventionally described by a

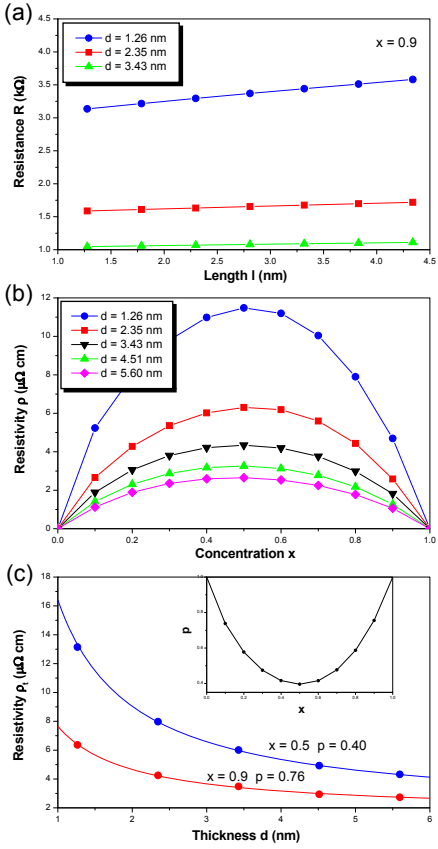


Fig. 8.25 (Reproduced according to Ref. [43]) Surface roughness effect on the resistivity of Cu thin film. (a) Resistance R as a function of length l and thickness d. (b) Resistivity ρ as a function of concentration x and thickness d. (c) Total resistivity $\rho_t = \rho + \rho_0$ as a function of concentration x and thickness d. The symbols are NanoDsim's data and the lines are fitting to Eq. (8.1). Inset: The mapping from the concentration x to the specularity p.

semi-classical model proposed by Fuchs [46] and Sondheimer [47]

$$\frac{\rho}{\rho_0} = 1 + \frac{3}{8}(1-p)\frac{\lambda}{d}, \tag{8.1}$$

where $\rho_0 = 1.67\ \mu\Omega\cdot$cm is the bulk resistivity of Cu and $\lambda = 39$ nm is the electron mean free path in Cu. p is the surface specularity which is 0 for diffusive surface and 1 for specular surface. The parameter p in the Fuchs–Sondheimer model plays a similar role to x in the atomic model.

www.ebook3000.com

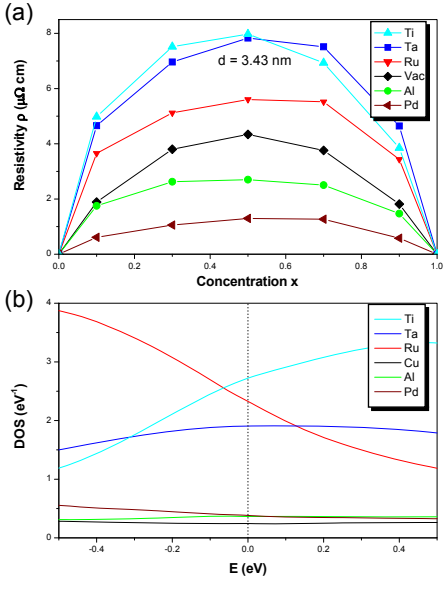


Fig. 8.26 (Reproduced according to Ref. [44]) Surface coating effect on the resistivity of Cu thin film. (a) Resistivity of Cu thin film coated with barrier metal Ti, Ta, Ru, Pd, Al in comparison to that of no coating (denoted by Vac). (b) Local density of states of barrier atom Ti, Ta, Ru, Pd, Al on a rough Cu surface. The line denoted by Cu is the local density of states of Cu atom on a perfect Cu surface.

Fig. 8.25c makes a connection between the two parameters: For a given x, p is extracted by fitting Eq. (8.1) from the calculated ρ as a function of d. One can see that the results of the first principles calculation fit very well to the Fuchs–Sondheimer formula, indicating that the NECPA-LMTO method can even be extended to the semi-classical regime by taking into account disorder scattering. The mapping from x to p is given in the inset of Fig. 8.25c: $x = 0$ or $x = 1$ corresponds to the specular limit and hence p reaches its maximum 1; $x = 0.5$ corresponds to the diffusive limit and hence p reaches its minimum 0.4. Here p can be viewed as an average of the specularity of the upper and lower surfaces.

Finally we would like to investigate the possibility of reducing surface scattering of Cu films. By coating barrier metal on a Cu film, the vacancies on the surface are filled with barrier atoms and the surface becomes flat in geometry. Experimentally barrier metals of Ti, Ta, Ru, Al and Pd [48–52] have been examined, and the results indicate that some

barrier metals actually increase the resistivity while others do reduce it. Theoretically the coating effect of barrier metals is investigated by using NanoDsim. To simulate the coating effect, the vacancies on a rough surface are replaced by barrier atoms. Namely the disorder site Cu_xVac_{1-x} in Fig. 8.24 is replaced by Cu_xM_{1-x} where M represents the barrier atom. The calculated results are shown in Fig. 8.26a which has the exact same trend as the experimental observation, namely, $\rho\,(\mathrm{Cu}+\mathrm{Ti}) \approx \rho\,(\mathrm{Cu}+\mathrm{Ta}) > \rho\,(\mathrm{Cu}+\mathrm{Ru}) > \rho\,(\mathrm{Cu}+\mathrm{Vac}) > \rho\,(\mathrm{Cu}+\mathrm{Al}) > \rho\,(\mathrm{Cu}+\mathrm{Pd})$, where $\rho\,(\mathrm{Cu}+\mathrm{Vac})$ is the resistivity of the rough Cu film without any coating. Superior to experimental studies, the theoretical approach allows us to go inside the atoms to understand the trend. Fig. 8.26b shows the local density of states of different barrier atoms. One can see that Al and Pd have very similar density of states as that of Cu and hence are good substitutes for missing Cu atoms. On the other hand, the density of states of Ti, Ta, or Ru is very different from that of Cu and the mismatch makes the surface scattering even stronger.

To sum up, the surface scattering of Cu films has been studied by using NanoDsim, and the calculated resistivity fits the well-known Fuchs–Sondheimer formula very well. The surface scattering can be reduced by coating the Cu film with some proper barrier metals having similar electronic structure to that of Cu.

8.10 Concluding remarks

This is almost the end of the monograph. But it can be also the beginning of a research career in first principles quantum transport. If you have never worked in this field before, you may take NanoDsim as your starting point and study various emerging materials and nanoelectronic devices as demonstrated in this chapter. If you are already an experienced researcher, you may take NanoDsim as a solid platform to add various new physics, e.g., electron-phonon interaction, electron-photon interaction, electron-magnon interaction, spin-orbit coupling, so on and so forth. Moreover, the formalisms and algorithms of NanoDsim are not only applicable to the LMTO method, but also extendable to other atomistic modeling methods, such as the KKR method and the tight-binding method. As long as the method is compatible with CPA, it can be integrated with the NECPA theory to study the quantum transport in disordered system.

With the continuous shrinking of device size and continuous increase of computing power, we are expecting that atomistic device simulation will be

www.ebook3000.com

an important approach in nanoelectronics. Eventually one should be able to build up a nanoelectronic device atom by atom on a computer and optimize the design before the real fabrication. In this monograph, we attempt to make a modest contribution to this growing field by sharing our experiences and software tools. Now it is your turn to make your own contribution!

Bibliography

[1] G. Kresse and J. Hafner, Phys. Rev. B **47**, R558 (1993); G. Kresse and J. Furthmuller, Phys. Rev. B **54**, 11169 (1996).
[2] F. Tran, P. Blaha, Phys. Rev. Lett. **102**, 226401 (2009).
[3] T. B. Boykin, G. Klimeck, F. Oyafuso, Phys. Rev. B **69** 115201 (2004).
[4] L. Zhang, F. Zahid, Y. Zhu, L. Liu, J. Wang, H. Guo, P. C. H. Chan, M. Chan, IEEE Transactions on Electron Devices, **60**, 3527(2013).
[5] T. B. Boykin, M. Luisier, M. Salmani-Jelodar, G. Klimeck, Phys. Rev. B **81** 125202 (2010).
[6] K. Uchida, T. Krishnamohan, K. C. Saraswat, Y. Nishi, IEEE IEDM Tech. Dig., 129 (2005).
[7] Y. Wang, F. Zahid, Y. Zhu, L. Liu, J. Wang, and H. Guo, Appl. Phys. Lett. **102**, 132109 (2013).
[8] I. Vurgaftman, J. R. Meyer, L. R. Ram-Mohan, J. Appl. Phys. **89**, 5815 (2001).
[9] J. Batey, S. L. Wright, J. Appl. Phys. **59**, 200 (1986).
[10] K. Xia, M. Zwierzycki, M. Talanana, P. J. Kelly, G. E. W. Bauer, Phys. Rev. B **73**, 064420 (2006).
[11] The supercell method has been described in Ref. [10]. Here a slight difference is that the supercell in Ref. [10] is constructed with the CPA bulk Hamiltonian while the supercell in this section is constructed with the NECPA two-probe Hamiltonian.
[12] By using the Γ point, the supercell is connected to itself in the lateral dimensions. In the lead region, the supercell becomes a cyclic structure characterized by discrete k-points due to the rotational symmetry (see Fig. 6.8). The scattering states can be classified by those discrete k-points to decompose the specular and diffusive part.
[13] J. Bass, W. P. Pratt, Jr., J. Magn. Magn. Mater. **200**, 274 (1999).
[14] One may wonder why a perfect metal has a nonzero resistance. In fact, it is more proper to think in terms of conductance instead of resistance. Metal's ability to conduct current is limited by the number of states available at the Fermi level. Therefore the conductance of perfect metal is not infinite and hence the resistance is not zero.
[15] J. Maassen, H. Guo, Phys. Rev. Lett. **109**, 266803 (2012).
[16] Q. Shi, H. Guo, Y. Zhu, L. Liu, Phys. Rev. Appl. **3**, 064008 (2015).
[17] TCAD Sentaurus Device Manual, ver. D-2010.03, Synopsys, Inc., Mountain View, CA, USA, 2010.

[18] G. P. Lansbergen, R. Rahman, C. J. Wellard, I. Woo, J. Caro, N. Collaert, S. Biesemans, G. Klimeck, L. C. L. Hollenberg, and S. Rogge, Nat. Phys. **4**, 656 (2008).
[19] In the off-state, the gate terminal is in equilibrium with the source terminal, and hence the gate is not included explicitly in the simulation.
[20] Z. Wang, Y. Ke, D. Liu, H. Guo, K. H. Bevan, Appl. Phys. Lett. **101**, 093102 (2012). The original work was done with a research code and the results presented in this section were reproduced with NanoDsim.
[21] Q. Shi, L. Zhang, Y. Zhu, L. Liu, M. Chan, H. Guo, *Atomic Disorder Scattering in Emerging Transistors by Parameter-Free First Principle Modeling*, 2014 IEEE International Electron Device Meeting (IEDM), Dec. 15–17, 2014, San Francisco, USA.
[22] K. Bolotin, K. Sikes, Z. Jiang, M. Klima, G. Fudenberg, J. Hone, P. Kim, H. Stormer, Solid State Commun. **146**, 351 (2008).
[23] D. Wei, Y. Liu, Y. Wang, H. Zhang, L. Huang, G. Yu, Nano Lett. **9**, 1752 (2009).
[24] K. Brenner, R. Murali, Appl. Phys. Lett. **98**, 113115 (2011).
[25] Z. Jin, J. Yao, C. Kittrell, and J. M. Tour, ACS Nano **5**, 4112 (2011).
[26] Y.-B. Tang, L.-C. Yin, Y. Yang, X.-H. Bo, Y.-L. Cao, H.-E. Wang, W.-J. Zhang, I. Bello, S.-T. Lee, H.-M. Cheng, and C.-S. Lee, ACS Nano **6**, 1970 (2012).
[27] The VCA-doping is tuned to align the Dirac cone centers of the lead and the central region.
[28] Y. Ke, K. Xia, H. Guo, Phys. Rev. Lett. **105**, 236801 (2010). The original work was done with a research code and the results presented in this section were reproduced with NanoDsim.
[29] M. Coey, Nature Mater. **4**, 9 (2005).
[30] S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, K. Ando, Nature Mater. **3**, 868 (2004).
[31] S. S. P. Parkin, C. Kaiser, A. Panchula, P. M. Rice, B. Hughes, M. Samant, S.-H. Yang, Nature Mater. **3**, 862 (2004).
[32] M. Jullière, Phys. Lett. **54A**, 225 (1975).
[33] W. H. Butler, X.-G. Zhang, T. C. Schulthess, J. M. MacLaren, Phys. Rev. B **63**, 054416 (2001).
[34] S. Yuasa, D. D. Djayaprawira, J. Phys. D **40**, R337 (2007).
[35] G. X. Miao, Y. J. Park, J. S. Moodera, M. Seibt, G. Eilers, M. Münzenberg, Phys. Rev. Lett. **100**, 246803 (2008).
[36] P. G. Mather, J. C. Read, and R. A. Buhrman, Phys. Rev. B **73**, 205412 (2006).
[37] J. Mathon, A. Umerski, Phys. Rev. B **63**, 220403 (2001).
[38] D. Waldron, V. Timoshevskii, Y. Hu, K. Xia, H. Guo, Phys. Rev. Lett. **97**, 226802 (2006).
[39] D. V. Dimitrov, Z. Gao, X. Wang, W. Jung, X. Lou, O. Heinonen, J. Appl. Phys. 105, 113905 (2009).
[40] C. H. Yang, M. Samant, and S. Parkin, in Proceedings of the American Physical Society 2009 March Meeting, Vol. **22**, p00004; M. Pesci, F. Gallino,

www.ebook3000.com

C. D. Valentin, G. Pacchioni, J. Phys. Chem. C **114**, 1350 (2010).
[41] J. Zhuang, Y. Wang, Y. Zhou, D. Liu, L. Zhang, Y. Zhu, L. Liu, J. Wang, H. Guo, unpublished (2014).
[42] A. Pratt, *Overview of the Use of Copper Interconnects in the Semiconductor Industry*, Advanced Energy Industries, Inc. (2004).
[43] Y. Ke, F. Zahid, V. Timoshevskii, K. Xia, D. Gall, H. Guo, Phys. Rev. B **79**, 155406 (2009). The original work was done with a research code and the results presented in this section were reproduced with NanoDsim.
[44] F. Zahid, Y. Ke, D. Gall, H. Guo, Phys. Rev. B **81**, 045406 (2010). The original work was done with a research code and the results presented in this section were reproduced with NanoDsim.
[45] J. M. Purswani and D. Gall, J. Appl. Phys. **104**, 044305 (2008).
[46] K. Fuchs, Proc. Cambridge Philos. Soc. **34**, 100 (1938).
[47] E. H. Sondheimer, Adv. Phys. **1**, 1 (1952).
[48] S. M. Rossnagel, T. S. Kuan, J. Vac. Sci. Technol. B **22**, 240 (2004).
[49] J. S. Chawla and D. Gall, Appl. Phys. Lett. **94**, 252101 (2009).
[50] K.-L. Ou, M.-S. Yu, R.-Q. Hsu, and M.-H. Lin, J. Vac. Sci. Technol. B **23**, 229 (2005).
[51] S. Tsukimoto, T. Onishi, K. Ito, M. Konno, T. Yaguchi, T. Kamino, M. Murakami, J. Electron. Mater. **36**, 1658 (2007).
[52] D.-Y. Shih, C.-A. Chang, J. Paraszczak, S. Nunes, J. Cataldo, J. Appl. Phys. **70**, 3052 (1991).

Appendix

This appendix is a collection of various "subroutines" and "functions". These "subroutines" and "functions" have no logical relation to each other. Instead they are "called" independently by previous chapters.

8.11 Atomic units

Atomic units are adopted throughout this monograph. In atomic units, the physical constants are redefined as

$$e = \hbar = m_e = \frac{1}{4\pi\varepsilon_0} = 1,$$

where e is the elementary charge, $\hbar$ is the reduced Planck constant, m_e is electron's mass, and ε_0 is the electric constant. Formulas are extremely simple in atomic units. For example, the Schrödinger equation of a hydrogen atom reads

$$\left(-\frac{1}{2}\nabla^2 + \frac{-1}{r}\right)\psi(\mathbf{r}) = E\psi(\mathbf{r}).$$

In atomic units, length and energy are measured by *Bohr* (a_B) and *Hartree* (Ha) respectively. Here a_B and Ha are the characteristic length and energy of a hydrogen $1s$ orbital which are defined by

$$Ha = \frac{1}{4\pi\varepsilon_0}\frac{e^2}{a_B} = \frac{\hbar^2}{m_e}\frac{1}{a_B^2}.$$

It is obtained that

$$a_B = \frac{\hbar^2}{m_e}\frac{4\pi\varepsilon_0}{e^2} \approx 0.5291772083 \text{ Å},$$

$$Ha = \frac{m_e e^4}{\hbar^2 (4\pi\varepsilon_0)^2} \approx 27.2113961 \text{ eV}.$$

www.ebook3000.com

For other physical quantities, atomic units can be derived from a_B and Ha. As a comparison, international system of units (SI) and atomic units (AU) are summarized as follows

	SI	**AU**
length	m	a_B
energy	J	Ha
time	s	$\frac{\hbar}{Ha}$
mass	kg	m_e
temperature	K	$\frac{Ha}{k_B}$
current	A	$\frac{eHa}{\hbar}$
voltage	V	$\frac{Ha}{e}$

where the physical constants are

$$\begin{aligned} e &\approx 1.60217733 \times 10^{-19}\ \mathrm{C}, \\ \hbar &\approx 1.05457266 \times 10^{-34}\ \mathrm{J \cdot s}, \\ m_e &\approx 9.1093897 \times 10^{-31}\ \mathrm{kg}, \\ k_B &\approx 1.380658 \times 10^{-23}\ \mathrm{J \cdot K^{-1}}, \\ \varepsilon_0 &\approx 8.854187817 \times 10^{-12}\ \mathrm{F \cdot m^{-1}}. \end{aligned}$$

To convert from SI to AU, the physical quantities need to be divided by proper atomic units. For example, electrons in Si have the mobility $\mu = 1430\ \mathrm{cm^2/(V \cdot s)}$. To convert the units, one first needs to construct atomic unit of mobility

$$\mu_0 = \frac{[length]^2}{[voltage \cdot time]} = \frac{a_B^2}{\frac{Ha}{e} \cdot \frac{\hbar}{Ha}} = \frac{a_B^2 e}{\hbar}.$$

Afterward the mobility in atomic units is obtained as (dimensionless)

$$\tilde{\mu} = \frac{\mu}{\mu_0} = \frac{1430\ \mathrm{cm^2/(V \cdot s)}}{\frac{a_B^2 e}{\hbar}} \approx 3.361 \times 10^4.$$

A.2 Phase diagram of the toy model

The 1d tight-binding toy model has been sketched in Fig. (2.7) and the Hamiltonian defined by Eq. (2.154). In this section, we analyze the singularities of its Green's function and identify different "phases" of its parameter space. We shall also study the behavior of the density of states and the transmission coefficient in these phases.

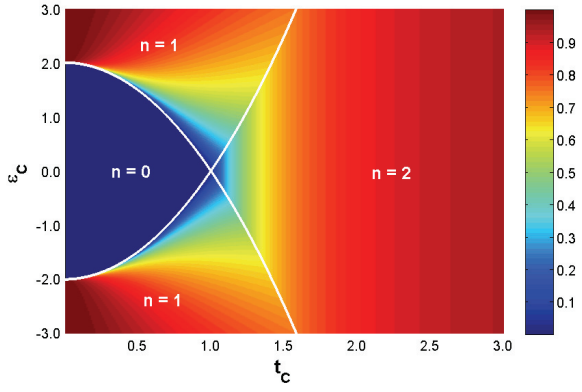


Fig. A.1 The color map of discrete spectrum weight W_d. The parameter space (t_C, ε_C) can be divided into four phases, and the boarder lines (white) are defined by $\varepsilon_C = \pm 2\left(1 - t_C^2\right)$.

In the 1d tight-binding toy model, let $t_0 = 1$ as the energy unit. Following Eq. (2.161), the retarded Green's function of the toy model is reduced to

$$G^r(z) = \frac{1}{z - \varepsilon_C - t_C^2\left(z - \mathrm{i}\sqrt{4 - z^2}\right)}, \tag{A.1}$$

where z is defined in the upper-half plane and the branch cut of $\sqrt{z}$ is chosen as $\mathrm{Re}\sqrt{z} > 0$. The complex function $G^r(z)$ has two types of singularities: branch cut and poles. The branch cut is due to $\sqrt{4 - z^2}$ and located on the real axis $-2 \leqslant z \leqslant 2$. The poles are the roots of the denominator

$$z - \varepsilon_C - t_C^2\left(z - \mathrm{i}\sqrt{4 - z^2}\right) = 0. \tag{A.2}$$

The equation can be reduced to

$$\left(1 - 2t_C^2\right) z^2 + 2\varepsilon_C\left(t_C^2 - 1\right) z + \left(4t_C^4 + \varepsilon_C^2\right) = 0, \tag{A.3}$$

which is a quadratic equation and generally has two roots. Notice that some roots of Eq. (A.3) might be extraneous solution to Eq. (A.2) and need to be eliminated. All roots of Eq. (A.2) are real and outside the branch cut, namely $|E_i| \geqslant 2$.

The behavior of the toy model can be classified into four phases in the parameter space (see Fig. A.1): (a) $\varepsilon_C < 2\left(1 - t_C^2\right)$ and $\varepsilon_C > -2\left(1 - t_C^2\right)$ where $G^r(z)$ has no pole. The system behaves like a good conductor with weak scattering in the central region. (b) $\varepsilon_C > 2\left(1 - t_C^2\right)$ and

www.ebook3000.com

$\varepsilon_C > -2\left(1-t_C^2\right)$ where $G^r(z)$ has one pole $E_+ > 2$. The system behaves like a localized state weakly coupled to the leads. (b') $\varepsilon_C < 2\left(1-t_C^2\right)$ and $\varepsilon_C < -2\left(1-t_C^2\right)$ where $G^r(z)$ has one pole $E_- < -2$. The system behaves similarly to the phase (b). (c) $\varepsilon_C > 2\left(1-t_C^2\right)$ and $\varepsilon_C < -2\left(1-t_C^2\right)$ where $G^r(z)$ has two poles $E_+ > 2$ and $E_- < -2$. Since $t_C > 1$, the central site strongly couples to the two nearest lead sites and they form a "molecule". Two eigenstates of the "molecule" become bound states and the other one becomes an extended state.

The singularities of $G^r(z)$ can be mapped to the density of states $D(E)$: The branch cut of $G^r(z)$ corresponds to the continuous density of states $D_c(E)$, and the poles correspond to the discrete density of states $D_d(E)$. To qualitatively measure the two parts, we define the weight of continuous spectrum and discrete spectrum

$$W_c \equiv \int D_c(E)\, dE,$$

$$W_d \equiv \int D_d(E)\, dE,$$

where $W_c + W_d = 1$ due to the normalization of $D(E)$. W_d can be evaluated by poles' residues

$$W_d = \sum_i res(E_i) = \sum_i \frac{1}{1-t_C^2\left(1-\frac{|E_i|}{\sqrt{E_i^2-4}}\right)},$$

and W_c is just a complement of W_d. Fig. A.1 shows the color map of W_d. One can see that continuous spectrum dominates phase (a) while both spectra coexist in other phases.

Finally we investigate the behavior of density of states and transmission coefficient in those phases. For $t_C = 1$, $D(E)$ and $T(E)$ are shown in Fig. (A.2a) and (A.2b), which is in the ballistic transport regime. At $\varepsilon_C = 0$, $D(E)$ has van Hove singularities at $E = \pm 2$, and $T(E)$ is in coincidence with the channel number. With increasing ε_C, van Hove singularities are replaced by bound states in $D(E)$, and the height of $T(E)$ is suppressed due to the scattering on the central site. For $t_C = 0.1$, $D(E)$ and $T(E)$ are shown in Fig. (A.2e) and (A.2f), which is in the resonant transport regime. At $\varepsilon_C < 2$, sharp resonances are observed around $E = \varepsilon_C$ in $D(E)$, and the maximum transmission 1 is reached in $T(E)$. At $\varepsilon_C > 2$, the discrete spectrum dominates $D(E)$ and the weight is over 99%. The transport is through the exponential tail of the continuous spectrum and hence $T(E)$ is hardly visible. For $t_C = 0.5$, $D(E)$ and $T(E)$ are shown in Fig. (A.2c) and

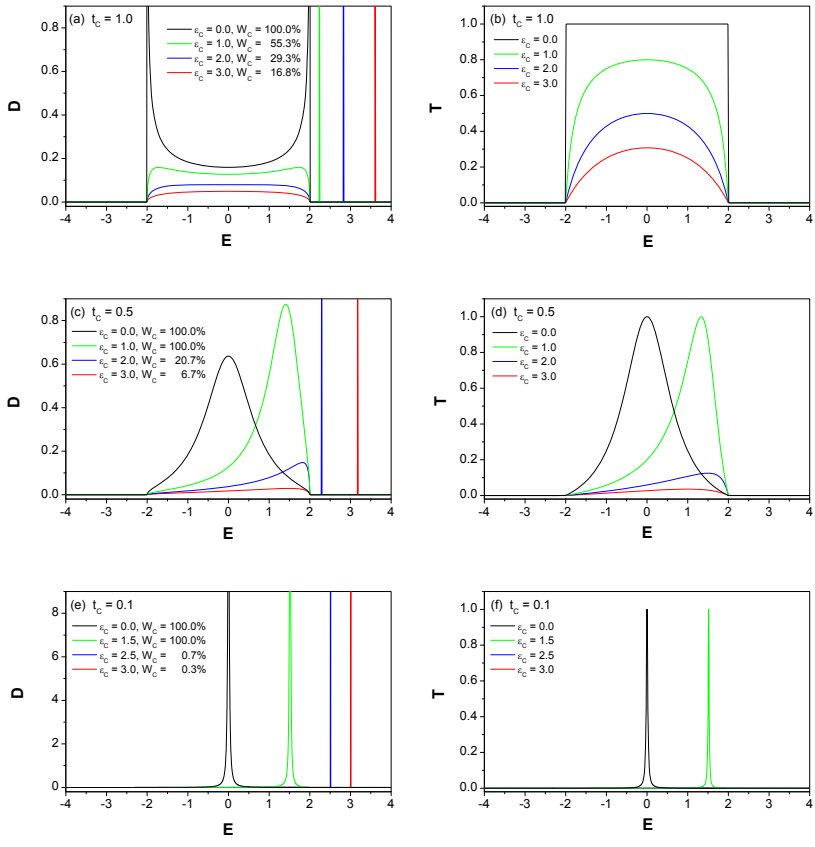


Fig. A.2 Density of states $D(E)$ (left panel) and transmission coefficient $T(E)$ (right panel) for some selected t_C and ε_C. Note that $D(E)$ has both continuous spectrum and discrete spectrum while $T(E)$ only continuous spectrum.

(A.2d), which can be viewed as a crossover from the ballistic transport to the resonant transport. For $t_C > 1$, the central site and the two nearest lead sites form a "molecule" with effective coupling $\tilde{t}_C = 1$. $D(E)$ and $T(E)$ have both von Hove singularities and bound states (not shown). Finally it is worth mentioning that although both resonances and bound states exhibit sharp peaks in $D(E)$, their wave functions are very different (see Fig. A.3). The resonances are extended states and hence are conductive in quantum transport. The bound states are localized states and hence have no contribution to quantum transport.

www.ebook3000.com

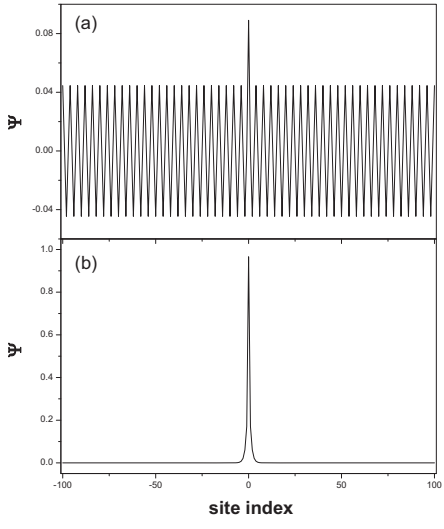


Fig. A.3 The wave function of (a) resonance and (b) bound state. The parameters are: (a) $t_C = 0.5$, $\varepsilon_C = 0$, $E = 0$; (b) $t_C = 0.5$, $\varepsilon_C = 3.0$, $E = 3.1721$.

To sum up, the 1d tight-binding toy model has been analyzed in detail. The simple model may exhibit very different behavior from resonant tunneling to ballistic transport, depending on the parameters t_C and ε_C. Due to its simplicity, the model is analytically solvable and hence can be an ideal testing case of various algorithms for two-probe systems.

A.3 Classical transport vs quantum transport

In this section, we investigate two transport models widely used in device simulation. The first one is the drift-diffusion model which is a classical transport model. The second one is the effective-mass model which is a quantum transport model. Comparison of the two models will elaborate the differences between the classical transport and the quantum transport.

To avoid the complexities related to real materials, we consider an artificial semiconductor with energy gap Δ, dielectric constant ϵ, and two types of charge carriers. One type of charge carrier are electrons having the dispersion relation

$$E = E_e + \frac{1}{2m_e}k^2; \tag{A.4}$$

the other type of charge carrier are holes having the dispersion relation

$$E = E_h + \frac{1}{2m_h}k^2. \tag{A.5}$$

Here E_α and m_α ($\alpha = e, h$) characterize the properties of the artificial semiconductor. E_α is the energy zero with $E_e = +\frac{\Delta}{2}$ and $E_h = -\frac{\Delta}{2}$. m_α is the effective-mass with $m_e > 0$ and $m_h < 0$. Other details such as spin degeneracy and valley degeneracy are ignored for simplicity. It is assumed that the system is uniform in the xy dimensions. The transport is along the z dimension where an external voltage V_b is applied. In addition, the system is doped to form p-regions and n-regions. Denote the dopant charge density by N_{dope}, with $N_{dope} > 0$ in p-regions and $N_{dope} < 0$ in n-regions.

A.3.1 *Drift-Diffusion model*

The key idea of the drift-diffusion model is that the transport current is composed of two parts, the drift current and the diffusion current. The drift current is proportional to the electric field while the diffusion current is proportional to the gradient of charge density. In short junctions, electron-hole recombination is negligible. Hence electrons and holes can be viewed as two types of independent particles. Denote the charge density and the current density of electrons and holes by n, j_n, p, j_p respectively. Since the system is uniform in the xy dimensions, n, j_n, p, j_p only depend on z.

The current density j_n and j_p are derived as [1]

$$j_n = Q_e \left[\nu_n n \left(-\partial_z U\right) + D_n \left(-\partial_z n\right)\right], \tag{A.6}$$

$$j_p = Q_e \left[\nu_p p \left(-\partial_z U\right) - D_p \left(-\partial_z p\right)\right], \tag{A.7}$$

where $Q_e = -1$ [2], $\nu_{n,p}$ is the mobility, and $D_{n,p}$ is the diffusion constant. Notice that $D_{n,p}$ is related to $\nu_{n,p}$ by $D_{n,p} = k_B T \nu_{n,p}$ where k_B is the Boltzmann constant and T is the temperature. In Eqs. (A.6,A.7), the first term is the drift current and the second term is the diffusion current. In a steady state, both j_n and j_p are conserved, leading to

$$\partial_z j_n = 0, \tag{A.8}$$

$$\partial_z j_p = 0. \tag{A.9}$$

U is the electrostatic potential, satisfying the Poisson equation

$$\partial_z^2 U = -\frac{4\pi}{\epsilon}\left(n - p + N_{dope}\right). \tag{A.10}$$

www.ebook3000.com

The boundary values of n, p, and U are determined by the charge neutrality condition. Suppose the transport region has a length L. At the boundaries of $z = 0$ and $z = L$, the system is charge neutral, resulting in

$$n(0) - p(0) + N_{dope}(0) = 0, \tag{A.11}$$

$$n(L) - p(L) + N_{dope}(L) = 0, \tag{A.12}$$

where U is hidden in the expressions of n and p (see below).

The expressions of n and p are derived as follows. For electrons, the local density of states is

$$D_e(E) = \begin{cases} \lambda_e \sqrt{E - E_e} & E > E_e \\ 0 & E < E_e \end{cases},$$

where $\lambda_e = \frac{4\pi\sqrt{2}}{(2\pi)^3} |m_e|^{\frac{3}{2}}$. In a potential U, the electron's charge density is calculated by a Boltzmann population of $D_e(E - U)$

$$n = \int D_e(E - U) e^{-\beta(E-\mu_e)} dE = N_e e^{-\beta(E_e + U - \mu_e)}, \tag{A.13}$$

where $N_e = \frac{\sqrt{\pi}}{2} (k_B T)^{\frac{3}{2}} \lambda_e$ and μ_e is the local chemical potential of electrons. For holes, the density of states is

$$D_h(E) = \begin{cases} 0 & E > E_h \\ \lambda_h \sqrt{E_h - E} & E < E_h \end{cases},$$

where $\lambda_h = \frac{4\pi\sqrt{2}}{(2\pi)^3} |m_h|^{\frac{3}{2}}$. In a potential U, the hole's charge density is calculated by a Boltzmann population of $D_h(E - U)$

$$p = \int D_h(E - U) e^{+\beta(E-\mu_h)} dE = N_h e^{+\beta(E_h + U - \mu_h)}, \tag{A.14}$$

where $N_h = \frac{\sqrt{\pi}}{2} (k_B T)^{\frac{3}{2}} \lambda_h$ and μ_h is the local chemical potential of holes. Notice that μ_e and μ_h are not necessarily equal because electrons and holes are not in equilibrium in the transport region. At the boundaries, however, μ_e and μ_h are identical because the charge carriers are assumed to be in equilibrium in the leads. The boundary values of μ_e and μ_h are determined by the applied voltage

$$\mu_e(0) = \mu_h(0) = +Q_e \frac{V_b}{2}, \tag{A.15}$$

$$\mu_e(L) = \mu_h(L) = -Q_e \frac{V_b}{2}. \tag{A.16}$$

Inserting Eqs. (A.13,A.14,A.15,A.16) into Eqs. (A.11,A.12), one can solve for the boundary values of n, p, and U.

In the numerical implementation, the central region is discretized into a uniform 1d mesh, $z_1 = 0$, $z_2 = a, \ldots, z_N = L$, where $a = \frac{L}{N-1}$. On the 1d mesh, Eqs. (A.6,A.7) are discretized as

$$j_{n,i+\frac{1}{2}} = \nu_n \frac{n_i + n_{i+1}}{2} \frac{U_{i+1} - U_i}{a} + D_n \frac{n_i - n_{i-1}}{a}, \tag{A.17}$$

$$j_{p,i+\frac{1}{2}} = \nu_p \frac{p_i + p_{i+1}}{2} \frac{U_{i+1} - U_i}{a} - D_p \frac{p_i - p_{i-1}}{a}. \tag{A.18}$$

Eqs. (A.8,A.9) are discretized as

$$j_{n,i-\frac{1}{2}} - j_{n,i+\frac{1}{2}} = 0, \tag{A.19}$$

$$j_{p,i-\frac{1}{2}} - j_{p,i+\frac{1}{2}} = 0. \tag{A.20}$$

Inserting Eqs. (A.19,A.20) to Eqs. (A.17,A.18), one obtains

$$\begin{aligned}(2U_i - U_{i-1} - U_{i+1} + 4k_BT)\, n_i + (U_i - U_{i-1} - 2k_BT)\, n_{i-1} \\ + (U_i - U_{i+1} - 2k_BT)\, n_{i+1} = 0,\end{aligned} \tag{A.21}$$

$$\begin{aligned}(2U_i - U_{i-1} - U_{i+1} - 4k_BT)\, p_i + (U_i - U_{i-1} + 2k_BT)\, p_{i-1} \\ + (U_i - U_{i+1} + 2k_BT)\, p_{i+1} = 0,\end{aligned} \tag{A.22}$$

where $i = 2, 3, \ldots, N-1$. Eq. (A.10) is discretized as

$$\frac{U_{i+1} + U_{i-1} - 2U_i}{a^2} = -\frac{4\pi}{\epsilon} \left(n_i - p_i + N_{dope,i}\right), \tag{A.23}$$

where $i = 2, 3, \ldots, N-1$. Given the boundary values n_1, p_1, U_1 and n_N, p_N, U_N, Eqs. (A.21,A.22,A.23) can be solved self-consistently. Since the equation array is nonlinear, Andersen mixer is adopted to improve the convergence (see Appendix A.20).

To sum up, we have derived the drift-diffusion equations (A.6) to (A.16) for 1d systems. We have also briefly discussed the algorithm for the numerical implementation. A 1d drift-diffusion solver is available in *appendix/ResearchCode/EM_vs_DD/DD_model/engine/@solver_DDmodel*.

A.3.2 *Effective-Mass model*

The key idea of the effective-mass model is that the transport current is due to a nonequilibrium occupation of scattering states. To solve the scattering states, the system is divided into three parts along the transport direction: the left lead ($z < 0$), the central region ($0 < z < L$), and the right lead ($z > L$). Analogous to the double δ-barrier problem solved in Section 1.1, we shall first solve the scattering states and then populate those states with some proper statistical weights. The major difference from Section 1.1 is

www.ebook3000.com

that the scattering potential in the central region is unknown, and we have to solve it self-consistently together with the Poisson equation.

The quantum system can be described by the Hamiltonian

$$H = \sum_{\alpha=e,h} H_\alpha,$$
$$H_\alpha = \frac{-1}{2m_\alpha}\left(\partial_x^2 + \partial_y^2 + \partial_z^2\right) + E_\alpha + U(z),$$

where α is the index of carrier species (electron or hole) and $U(z)$ is the electrostatic potential. It is assumed that $U(z)$ is flat in the left and right leads, namely, $U(z<0) = U(0) \equiv U_L$ and $U(z>L) = U(L) \equiv U_R$. Since the Hamiltonian of electrons and holes are separated, each of them can be treated independently. Also note that the potential only depends on z, so the wave function and the energy can be decomposed as

$$\psi(x,y,z) = \varphi(z)\, e^{\mathrm{i}k_x x} e^{\mathrm{i}k_y y},$$
$$E = E_z + \frac{1}{2m_\alpha}k_x^2 + \frac{1}{2m_\alpha}k_y^2. \tag{A.24}$$

Thus one can work on the z dimension and the xy dimensions separately.

As the first step, we solve the scattering problem in the z dimension for a given type of charge carrier. In the left lead, the traveling waves are

$$\varphi_L(z) = e^{\pm \mathrm{i}k_z z},$$

where k_z is defined by

$$E_z = \frac{1}{2m_\alpha}k_z^2 + E_\alpha + U_L.$$

In the right lead, the traveling waves are

$$\varphi_R(z) = e^{\pm \mathrm{i}q_z z},$$

where q_z is defined by

$$E_z = \frac{1}{2m_\alpha}q_z^2 + E_\alpha + U_R.$$

In the central region, the traveling waves are scattered by the potential $U(z)$. An incoming wave from the left lead can be either reflected back to the left lead or transmitted to the right lead (see Fig. 1.1a). For a given energy E_z, the wave functions in the left and right leads are

$$\varphi_L(z) = e^{\mathrm{i}k_z z} + re^{-\mathrm{i}k_z z}, \tag{A.25}$$
$$\varphi_R(z) = te^{\mathrm{i}q_z(z-L)}, \tag{A.26}$$

where $k_z > 0$ to represent an incoming traveling wave, $q_z > 0$ or $\mathrm{Im}\,(q_z) > 0$ to represent a transmitted traveling wave or decaying wave. In the central region, the wave function satisfies the Schrödinger equation

$$\left[-\frac{1}{2m_\alpha}\partial_z^2 + E_\alpha + U\,(z)\right]\varphi_C\,(z) = E_z\varphi_C\,(z)\,. \tag{A.27}$$

The wave functions $\varphi_L\,(z)$, $\varphi_C\,(z)$, $\varphi_R\,(z)$ are smoothly connected at the boundaries

$$\varphi_C\,(0) = \varphi_L\,(0)\,, \tag{A.28}$$
$$\varphi_C'\,(0) = \varphi_L'\,(0)\,, \tag{A.29}$$
$$\varphi_C\,(L) = \varphi_R\,(L)\,, \tag{A.30}$$
$$\varphi_C'\,(L) = \varphi_R'\,(L)\,. \tag{A.31}$$

Similarly an incoming wave from the right lead can be either reflected back to the right lead or transmitted to the left lead (see Fig. 1.1b). The wave functions $\tilde{\varphi}_L\,(z)$, $\tilde{\varphi}_C\,(z)$, $\tilde{\varphi}_R\,(z)$ are obtained as

$$\tilde{\varphi}_L\,(z) = \tilde{t}e^{-\mathrm{i}k_z z}, \tag{A.32}$$
$$\tilde{\varphi}_R\,(z) = e^{-\mathrm{i}q_z(z-L)} + \tilde{r}e^{\mathrm{i}q_z(z-L)}, \tag{A.33}$$

$$\left[-\frac{1}{2m_\alpha}\partial_z^2 + E_\alpha + U\,(z)\right]\tilde{\varphi}_C\,(z) = E_z\tilde{\varphi}_C\,(z)\,, \tag{A.34}$$

which are smoothly connected at the boundaries

$$\tilde{\varphi}_C\,(0) = \tilde{\varphi}_L\,(0)\,, \tag{A.35}$$
$$\tilde{\varphi}_C'\,(0) = \tilde{\varphi}_L'\,(0)\,, \tag{A.36}$$
$$\tilde{\varphi}_C\,(L) = \tilde{\varphi}_R\,(L)\,, \tag{A.37}$$
$$\tilde{\varphi}_C'\,(L) = \tilde{\varphi}_R'\,(L)\,. \tag{A.38}$$

Thus the scattering wave functions are uniquely determined by Eqs. (A.25–A.31) or Eqs. (A.32–A.38).

After solving the scattering problem in the z dimension, we can take into account the xy dimensions by defining an effective Fermi factor. In the xy dimensions, the wave functions are plane waves $e^{\mathrm{i}k_x x}e^{\mathrm{i}k_y y}$. The integral over k_x and k_y can be absorbed into the Fermi function to derive an effective Fermi factor $F_{\alpha\beta}\,(E_z)$

$$\begin{aligned} F_{\alpha\beta}\,(E_z) &\equiv q_\alpha \int \frac{dk_x}{2\pi}\int \frac{dk_y}{2\pi} f\left[q_\alpha\left(E_z + \frac{1}{2m_\alpha}k_x^2 + \frac{1}{2m_\alpha}k_y^2 - \mu_\beta\right)\frac{1}{k_B T}\right] \\ &= q_\alpha \frac{|m_\alpha|\,k_B T}{2\pi} F_0\left(q_\alpha\frac{\mu_\beta - E_z}{k_B T}\right), \end{aligned} \tag{A.39}$$

www.ebook3000.com

where $q_e = +1$ and $q_h = -1$, $f(x) \equiv \frac{1}{e^x+1}$, $F_0(x) = \ln(1+e^x)$, and μ_β is the chemical potential of the lead β

$$\mu_L = +Q_e \frac{V_b}{2}, \tag{A.40}$$

$$\mu_R = -Q_e \frac{V_b}{2}. \tag{A.41}$$

The charge density ρ and the current density j are determined by a nonequilibrium occupation of the scattering states

$$\rho(z) = \sum_\alpha \int \frac{dE_z}{2\pi} \left[|\varphi_C(z)|^2 2\pi D_{\alpha L}(E_z) F_{\alpha L}(E_z) + |\tilde{\varphi}_C(z)|^2 2\pi D_{\alpha R}(E_z) F_{\alpha R}(E_z) \right], \tag{A.42}$$

$$j = Q_e \sum_\alpha \int \frac{dE_z}{2\pi} T_\alpha(E_z) \left[F_{\alpha L}(E_z) - F_{\alpha R}(E_z) \right], \tag{A.43}$$

where $D_{\alpha\beta}(E_z)$ is the density of states of the incoming waves in the lead β

$$D_{\alpha L}(E_z) \equiv \frac{1}{2\pi} \frac{dk_z}{dE_z}, \tag{A.44}$$

$$D_{\alpha R}(E_z) \equiv \frac{1}{2\pi} \frac{dq_z}{dE_z}. \tag{A.45}$$

$T_\alpha(E_z)$ is the transmission coefficient solved from the scattering states equation

$$T_\alpha(E_z) = |t(E_z)|^2 = \left|\tilde{t}(E_z)\right|^2. \tag{A.46}$$

In the above derivation, it is assumed that the scattering potential $U(z)$ is known. Actually the shape of the scattering potential strongly depends on the charge density in the central region. Given the charge density $\rho(z)$, $U(z)$ can be solved from the Poisson equation

$$\partial_z^2 U(z) = -4\pi \left[\rho(z) + N_{dope}(z) \right], \tag{A.47}$$

with the boundary condition $U(z=0) = U_L$ and $U(z=L) = U_R$. Here the boundary values U_L and U_R are determined by the charge neutrality condition in the left and right leads

$$\sum_\alpha N_\alpha q_\alpha F_{\frac{1}{2}} \left(-q_\alpha \frac{E_\alpha + U_L - \mu_L}{k_B T} \right) + N_{dope}(0) = 0, \tag{A.48}$$

$$\sum_\alpha N_\alpha q_\alpha F_{\frac{1}{2}} \left(-q_\alpha \frac{E_\alpha + U_R - \mu_R}{k_B T} \right) + N_{dope}(L) = 0, \tag{A.49}$$

where $N_\alpha \equiv \frac{\sqrt{\pi}}{2}\left(k_B T\right)^{\frac{3}{2}} \lambda_\alpha$ with $\lambda_\alpha \equiv \frac{4\pi\sqrt{2}}{(2\pi)^3}\left|m_\alpha\right|^{\frac{3}{2}}$. $F_{\frac{1}{2}}(x)$ is the Fermi integral defined by

$$F_{\frac{1}{2}}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^\infty \sqrt{t} \frac{1}{e^{t-x}+1} dt,$$

which can be evaluated numerically [3]. The derivation of Eqs. (A.48,A.49) are analogous to Eqs. (A.13,A.14) except that the Boltzmann distribution is replaced by the Fermi–Dirac distribution.

In the numerical implementation, the central region is discretized into a uniform 1d mesh, $z_1 = 0$, $z_2 = a, \ldots, z_N = L$, where $a = \frac{L}{N-1}$. On the 1d mesh, Eq. (A.27) is discretized as

$$\frac{1}{2m_\alpha} \frac{\varphi_{C,i+1} + \varphi_{C,i-1} - 2\varphi_{C,i}}{a^2} + \left(E_z - E_\alpha - U_i\right)\varphi_{C,i} = 0,$$

where $i = 2, 3, \ldots, N-1$. Eqs. (A.28,A.29,A.30,A.31) are reduced to

$$\begin{aligned}
\varphi_{C,1} &= \varphi_L(0), \\
\varphi_{C,0} &= \varphi_L(-a), \\
\varphi_{C,N} &= \varphi_R(L), \\
\varphi_{C,N+1} &= \varphi_R(L+a),
\end{aligned}$$

Thus the differential equation (A.27) is converted to a linear equation array

$$\begin{pmatrix}
E_z - U_{\alpha 1} - \Sigma_L^r & t_\alpha & & & & & \\
t_\alpha & E_z - U_{\alpha 2} & t_\alpha & & & & \\
 & t_\alpha & E_z - U_{\alpha 3} & t_\alpha & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & t_\alpha & E_z - U_{\alpha,N-1} & t_\alpha & \\
 & & & & t_\alpha & E_z - U_{\alpha N} - \Sigma_R^r
\end{pmatrix}$$

$$\times \begin{pmatrix} r \\ \varphi_C(z_2) \\ \varphi_C(z_3) \\ \cdots \\ \varphi_C(z_{N-1}) \\ t \end{pmatrix} = \begin{pmatrix} -s_L \\ -t_\alpha \\ 0 \\ \cdots \\ 0 \\ 0 \end{pmatrix}, \tag{A.50}$$

where $t_\alpha \equiv \frac{1}{2m_\alpha a^2}$, $U_{\alpha i} \equiv E_\alpha + 2t_\alpha + U_i$, $\Sigma_L^r \equiv -t_\alpha e^{\mathrm{i}k_z a}$, $\Sigma_R^r \equiv -t_\alpha e^{\mathrm{i}q_z a}$,

www.ebook3000.com

$s_L \equiv E_z - U_{\alpha 1} + t_\alpha e^{-\mathrm{i}k_z a}$. Similarly Eq. (A.34) is converted to

$$\begin{pmatrix} E_z - U_{\alpha 1} - \Sigma_L^r & t_\alpha & & & & & \\ t_\alpha & E_z - U_{\alpha 2} & t_\alpha & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & t_\alpha & E_z - U_{\alpha,N-2} & t_\alpha & & \\ & & & t_\alpha & E_z - U_{\alpha,N-1} & t_\alpha & \\ & & & & t_\alpha & E_z - U_{\alpha N} - \Sigma_R^r \end{pmatrix}$$

$$\times \begin{pmatrix} \tilde{t} \\ \tilde{\varphi}_C(z_2) \\ \cdots \\ \tilde{\varphi}_C(z_{N-2}) \\ \tilde{\varphi}_C(z_{N-1}) \\ \tilde{r} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ -t_\alpha \\ -s_R \end{pmatrix}, \tag{A.51}$$

where $s_R \equiv E_z - U_{\alpha N} + t_\alpha e^{-\mathrm{i}qa}$.

On the 1d mesh, the dispersion relations of electron and hole

$$E_z = U_L + E_\alpha + \frac{1}{2m_\alpha} k_z^2, \tag{A.52}$$

$$E_z = U_R + E_\alpha + \frac{1}{2m_\alpha} q_z^2, \tag{A.53}$$

are adapted to

$$E_z = U_L + E_\alpha + 2t_\alpha (1 - \cos ka), \tag{A.54}$$

$$E_z = U_R + E_\alpha + 2t_\alpha (1 - \cos qa). \tag{A.55}$$

It is easy to verify that Eqs. (A.54,A.55) are reduced to Eqs. (A.52,A.53) in the limit $a \to 0$. As a consequence, the density of states are obtained as

$$D_{\alpha L}(E_z) = \frac{1}{2\pi} \frac{1}{2at_\alpha \sin ka}, \tag{A.56}$$

$$D_{\alpha R}(E_z) = \frac{1}{2\pi} \frac{1}{2at_\alpha \sin qa}. \tag{A.57}$$

Since the charge density ρ and the electrostatic potential U are coupled together by Eq. (A.42) and Eq. (A.47), they must be solved self-consistently. To accelerate the convergence, a nonlinear Poisson solver is adopted to provide an efficient mixing of the potential. Interested readers are referred to Appendix A.23 for details.

To sum up, we have derived the effective-mass equations (A.25–A.31, A.32–A.38,A.42,A.43,A.47) for 1d systems. We have also briefly discussed the algorithm for the numerical implementation. A 1d effective-mass solver is available in *appendix/ResearchCode/EM_vs_DD/EM_model/engine/@solver_EMmodel*.

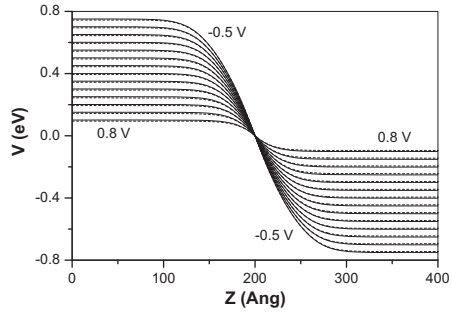


Fig. A.4 The electrostatic potential in the p-n junction simulated with the drift-diffusion model (dashed) and the effective-mass model (solid). The bias voltage $V_b = -0.5$ V, -0.4 V, ..., 0.8 V.

A.3.3 *Numerical results*

As a demonstration, the two transport models are applied to study the transport in a p-n junction. The doping charge density is assumed to be step-like

$$N_{dope}(z) = -N_0 sgn\left(z - \frac{L}{2}\right),$$

where $N_0 = 1 \times 10^{19}$ cm^{-3} and $L = 40$ nm. Here atomic units have been converted to SI units in order to be consistent with the electronic engineering literature. For simplicity, the effective-mass m_e and m_h are assumed to identical to the electron's mass m_0. Other parameters are: $\Delta = 1$ eV and $k_B T = 300$ K.

At equilibrium, a depletion layer is formed at the interface of p-region and n-region, resulting in a band bending around $0.99V$. By applying a negative bias voltage, the depletion layer grows thicker and thicker, and the band bending larger and larger. Essentially there is no transport current for a negative voltage. In contrast, by applying a positive bias voltage, the depletion region grows thinner and thinker, and the band bending smaller and smaller. Above a threshold voltage, the current increases exponentially with the bias voltage. Define the on-current slope as

$$s \equiv \frac{d\left(\log_{10} I\right)}{dV_b}.$$

www.ebook3000.com

At $V_b = 0.8V$, the on-current slope is obtained as $s = 16.5$ in the drift-diffusion model and $s = 16.7$ in the effective-mass model. We have seen that both transport models describe the rectification of p-n junctions very well.

A few comments are in order. (1) If the bias voltage is too large, both transport models will have problems. The drift-diffusion model predicts a non-physical charge accumulation at the boundaries due to the lack of electron-hole recombination. The effective-mass model is difficult to converge for nearly flat band bending due to the lack of inelastic scattering. (2) For simple structures such as the p-n junction, the results of the two models are quite similar, indicating that the physics is dominated by the electrostatics and the bulk material properties. That is why the drift-diffusion model can be "extrapolated" to much smaller scale where the classical assumptions are no longer valid [4]. (3) In nano-electronic devices, quantum effects and nonequilibrium statistics become more and more important, and the quantum transport model is more relevant than the classical transport model. If one insists on working with the classical transport model by tuning existing parameters or adding new parameters, the complexity of the model can be formidable in the near future. In fact, the number of parameters in compact models has seen an increase in the fashion of Moore's law [5]. It is time to upgrade from the classical transport model to the quantum transport model.

A.4 Lehmann spectrum of NEGF

In this section, we derive the Lehmann spectrum of NEGF to reveal its mathematical structure. We shall also discuss two important limits, the equilibrium limit and the weak coupling limit.

First of all, we derive the Lehmann spectrum of $G^r(E)$, $G^a(E)$, $G^<(E)$, $G^>(E)$. Denote the Hamiltonian operator with $\hat{H}$ and the density matrix operator with $\hat{\rho}$. Assume that $\hat{H}$ has eigenstates $\{E_n, |n\rangle\}$ where $\hat{H}|n\rangle = E_n|n\rangle$. $\hat{\rho}$ can be expanded in the basis set $\{|n\rangle\}$ as

$$\hat{\rho} = \sum_n \rho_n |n\rangle \langle n| ,$$

where $\sum_n \rho_n = 1$. It means that the system is a mixed state where the eigenstate $|n\rangle$ has the statistical weight ρ_n.

By definition, $G^{<}(E)$ can be expanded in the basis set $\{|n\rangle\}$ as

$$\begin{aligned} G^{<}(E) &= \int dt e^{\mathrm{i}Et} G^{<}(t) \\ &= \int dt e^{\mathrm{i}Et} \langle\langle a(t) | b(0) \rangle\rangle^{<} \\ &= \int dt e^{\mathrm{i}Et} \mathrm{i} \langle b(0) a(t) \rangle \\ &= \int dt e^{\mathrm{i}Et} \sum_{n} \mathrm{i} \langle n | \hat{\rho} b(0) a(t) | n \rangle \\ &= \int dt e^{\mathrm{i}Et} \sum_{nm} \mathrm{i} \langle n | \hat{\rho} b(0) | m \rangle \langle m | a(t) | n \rangle , \end{aligned}$$

where $\sum_m |m\rangle \langle m| = 1$ has been inserted into the last line. Notice that in the Heisenberg picture $c(t) = e^{\mathrm{i}Ht} c(0) e^{-\mathrm{i}Ht}$. $G^{<}(E)$ is further reduced to

$$\begin{aligned} G^{<}(E) &= \int dt e^{\mathrm{i}Et} \sum_{nm} \mathrm{i} \langle n | \hat{\rho} b | m \rangle \langle m | e^{\mathrm{i}Ht} a e^{-\mathrm{i}Ht} | n \rangle , \\ &= \int dt e^{\mathrm{i}Et} \sum_{nm} \mathrm{i}\rho_n \langle n | b | m \rangle e^{\mathrm{i}E_m t} \langle m | a | n \rangle e^{-\mathrm{i}E_n t}, \\ &= \sum_{nm} \int dt e^{\mathrm{i}(E - E_n + E_m)t} \mathrm{i}\rho_n \langle n | b | m \rangle \langle m | a | n \rangle . \end{aligned}$$

By using $\int dt e^{\mathrm{i}Et} = 2\pi\mathrm{i}\delta(E)$, the Lehmann spectrum of $G^{<}(E)$ is obtained as

$$G^{<}(E) = \sum_{nm} 2\pi\mathrm{i}\rho_n \delta(E - E_n + E_m) \langle m | a | n \rangle \langle n | b | m \rangle . \tag{A.58}$$

Similarly, the Lehmann spectrum of $G^{>}(E)$ is obtained as

$$G^{>}(E) = \sum_{nm} -2\pi\mathrm{i}\rho_m \delta(E - E_n + E_m) \langle m | a | n \rangle \langle n | b | m \rangle . \tag{A.59}$$

www.ebook3000.com

By definition, $G^r(E)$ can be expanded in the basis set $\{|n\rangle\}$ as

$$\begin{aligned}
G^r(E) &= \int dt e^{\mathrm{i}Et} G^r(t) \\
&= \int dt e^{\mathrm{i}Et} \langle\langle a(t) | b(0)\rangle\rangle^r \\
&= \int dt e^{\mathrm{i}Et} (-\mathrm{i})\,\theta(t) \langle\{a(t), b(0)\}\rangle \\
&= \int dt e^{\mathrm{i}Et} (-\mathrm{i})\,\theta(t) \sum_n \langle n|\hat{\rho}\{a(t), b(0)\}|n\rangle \\
&= \int dt e^{\mathrm{i}Et} (-\mathrm{i})\,\theta(t) \sum_{nm} [\langle n|\hat{\rho}a(t)|m\rangle\langle m|b(0)|n\rangle \\
&\quad + \langle n|\hat{\rho}b(0)|m\rangle\langle m|a(t)|n\rangle],
\end{aligned}$$

where $\sum_m |m\rangle\langle m| = 1$ has been inserted into the last line. Notice that in Heisenberg picture $c(t) = e^{\mathrm{i}Ht} c(0) e^{-\mathrm{i}Ht}$. $G^r(E)$ is further reduced to

$$\begin{aligned}
G^r(E) &= \int dt e^{\mathrm{i}Et} (-\mathrm{i})\,\theta(t) \sum_{nm} \left[\langle n|\hat{\rho}e^{\mathrm{i}Ht}ae^{-\mathrm{i}Ht}|m\rangle\langle m|b|n\rangle \right. \\
&\quad \left. + \langle n|\hat{\rho}b|m\rangle\langle m|e^{\mathrm{i}Ht}ae^{-\mathrm{i}Ht}|n\rangle\right], \\
&= \int dt e^{\mathrm{i}Et} (-\mathrm{i})\,\theta(t) \sum_{nm} \left[\langle m|\hat{\rho}e^{\mathrm{i}Ht}ae^{-\mathrm{i}Ht}|n\rangle\langle n|b|m\rangle \right. \\
&\quad \left. + \langle n|\hat{\rho}b|m\rangle\langle m|e^{\mathrm{i}Ht}ae^{-\mathrm{i}Ht}|n\rangle\right], \\
&= \int dt e^{\mathrm{i}Et} (-\mathrm{i})\,\theta(t) \sum_{nm} (\rho_m + \rho_n) e^{\mathrm{i}(E_m - E_n)t} \langle m|a|n\rangle\langle n|b|m\rangle \\
&= \sum_{nm} \int dt\, (-\mathrm{i})\,\theta(t) e^{\mathrm{i}(E - E_n + E_m)t} (\rho_m + \rho_n) \langle m|a|n\rangle\langle n|b|m\rangle.
\end{aligned}$$

By using $\int dt\,(-\mathrm{i})\,\theta(t) e^{\mathrm{i}Et} = \frac{1}{E+\mathrm{i}0^+}$, the Lehmann spectrum of $G^r(E)$ is obtained as

$$G^r(E) = \sum_{nm} (\rho_m + \rho_n) \frac{1}{E - E_n + E_m + \mathrm{i}0^+} \langle m|a|n\rangle\langle n|b|m\rangle. \quad \text{(A.60)}$$

Similarly the Lehmann spectrum of $G^a(E)$ is obtained as

$$G^a(E) = \sum_{nm} (\rho_m + \rho_n) \frac{1}{E - E_n + E_m - \mathrm{i}0^+} \langle m|a|n\rangle\langle n|b|m\rangle. \quad \text{(A.61)}$$

Eqs. (A.58,A.59,A.60,A.61) are the Lehmann spectrum of NEGF, which are the central results of this section.

Secondly we investigate the Lehmann spectrum in the equilibrium limit. In equilibrium, the statistical weight $\{\rho_n\}$ is known from the quantum statistics

$$\rho_n = \frac{1}{Z} e^{-\beta(E_n - \mu N_n)},$$

where $Z \equiv \sum_n e^{-\beta(E_n - \mu N_n)}$. Here μ is the chemical potential, $\beta = \frac{1}{k_B T}$ is proportional to the inverse of temperature, and E_n and N_n are the eigenenergy and the particle number of the state $|n\rangle$ respectively. It can be shown that $G^r(E)$, $G^a(E)$, $G^<(E)$, $G^>(E)$ are connected by the following fluctuation-dissipation theorem

$$\begin{aligned} G^<(E) &= f(E)\left[G^a(E) - G^r(E)\right], \\ G^>(E) &= -\overline{f}(E)\left[G^a(E) - G^r(E)\right], \end{aligned} \tag{A.62}$$

where

$$\begin{aligned} f(E) &\equiv \frac{1}{e^{\beta(E-\mu)} + 1}, \\ \overline{f}(E) &\equiv 1 - f(E) = \frac{1}{e^{-\beta(E-\mu)} + 1}. \end{aligned}$$

Proof: By using the Lehmann spectrum, the RHS of Eq. (A.62) can be reduced to

$$G^a(E) - G^r(E) = \sum_{nm} (\rho_m + \rho_n)\, 2\pi \mathrm{i}\delta(E - E_n + E_m) \langle m|a|n\rangle \langle n|b|m\rangle,$$

where the identity

$$\frac{1}{E - \mathrm{i}0^+} - \frac{1}{E + \mathrm{i}0^+} = 2\pi \mathrm{i}\delta(E)$$

is used in the derivation. Notice that $N_n - N_m = 1$ due to the factors $\langle m|a|n\rangle$ and $\langle n|b|m\rangle$, and $E_n - E_m = E$ due to the factor $\delta(E - E_n + E_m)$. Consequently one obtains

$$\rho_m + \rho_n = \frac{e^{-\beta(E_m - \mu N_m)} + e^{-\beta(E_n - \mu N_n)}}{Z} = \rho_n \frac{1}{f(E)} = \rho_m \frac{1}{\overline{f}(E)}.$$

Therefore the LHS is equal to the RHS in Eq. (A.62). QED.

Finally we investigate the Lehmann spectrum in the weak coupling limit. Consider a small interacting system which is weakly coupled to the environment. The quantum states can be approximated by the eigenstates of the small system, while the occupation of the states are determined by the coupling to the environment. Suppose the small system contains M sites with

www.ebook3000.com

many-body interaction, and the Hamiltonian is $H_{cent} = H_{cent}\left(\left\{c_i, c_i^\dagger\right\}\right)$ where $i = 1, 2, \ldots, M$. The M sites couple to different leads with different local chemical potentials $\{\mu_\beta\}$. The coupling strength between site i, site j and lead β is characterized by the linewidth function $\Gamma_{\beta,ij}$. The nonequilibrium statistical weight $\{\rho_n\}$ is determined by the steady state condition [6]

$$\sum_\beta \sum_{nm} \left[\rho_m f_\beta\left(E_n - E_m\right) - \rho_n \bar{f}_\beta\left(E_n - E_m\right)\right] \tilde{\Gamma}_{nm}^\beta Q_{nm}^l = 0, \qquad \text{(A.63)}$$

where $\tilde{\Gamma}_{nm}^\beta \equiv \sum_{ij} \langle m|c_i|n\rangle \left\langle n\left|c_j^\dagger\right|m\right\rangle \Gamma_{\beta,ij}$ and $Q_{nm}^l \equiv \delta_{nl} - \delta_{ml}$. Here $i, j = 1, 2, \ldots, M$ are the site indices and $l, m, n = 1, 2, \ldots, 2^M$ are the indices of the many-body eigenstates of H_{cent}. Notice that Eq. (A.63) defines 2^M equations for $l = 1, 2, \ldots, 2^M$. Of the 2^M equations, only $2^M - 1$ equations are independent. It is necessary to supplement them with the normalization condition $\sum_n \rho_n = 1$ to solve $\{\rho_n\}$. Once the nonequilibrium statistical weights are known, one can proceed to calculate NEGF in the weak coupling limit by using the Lehmann spectrum.

As an example, consider a model system containing two spins with the Coulomb interaction

$$H_{cent} = \varepsilon_\uparrow c_\uparrow^\dagger c_\uparrow + \varepsilon_\downarrow c_\downarrow^\dagger c_\downarrow + U c_\uparrow^\dagger c_\uparrow c_\downarrow^\dagger c_\downarrow.$$

The Hamiltonian has four eigenstates: $E_{00} = 0$, $|00\rangle = |0\rangle$; $E_{01} = \varepsilon_\uparrow$, $|01\rangle = c_\uparrow^\dagger|0\rangle$; $E_{10} = \varepsilon_\downarrow$, $|10\rangle = c_\downarrow^\dagger|0\rangle$; $E_{11} = \varepsilon_\uparrow + \varepsilon_\downarrow + U$, $|11\rangle = c_\downarrow^\dagger c_\uparrow^\dagger|0\rangle$. The statistical weights of these eigenstates can be written as $\rho_{00} = \langle(1 - n_\downarrow)(1 - n_\uparrow)\rangle$, $\rho_{01} = \langle(1 - n_\downarrow) n_\uparrow\rangle$, $\rho_{10} = \langle n_\downarrow (1 - n_\uparrow)\rangle$, $\rho_{11} = \langle n_\downarrow n_\uparrow\rangle$, where $n_\sigma \equiv c_\sigma^\dagger c_\sigma$ is the occupation number operator. Since spin is conserved in $\hat{H}$, $G_{\sigma\sigma'}^\lambda = \delta_{\sigma\sigma'} G_\sigma^\lambda$. By using Eqs. (A.58,A.59,A.60,A.61), the Green's functions are obtained as

$$\begin{aligned}
G_\sigma^r(E) &= \frac{\langle n_{\bar{\sigma}}\rangle}{E^+ - \varepsilon_\sigma - U} + \frac{1 - \langle n_{\bar{\sigma}}\rangle}{E^+ - \varepsilon_\sigma}, \\
G_\sigma^a(E) &= \frac{\langle n_{\bar{\sigma}}\rangle}{E^- - \varepsilon_\sigma - U} + \frac{1 - \langle n_{\bar{\sigma}}\rangle}{E^- - \varepsilon_\sigma}, \\
G_\sigma^<(E) &= 2\pi \mathrm{i} \langle n_{\bar{\sigma}} n_\sigma\rangle \delta\left(E - \varepsilon_\sigma - U\right) + 2\pi \mathrm{i} \langle(1 - n_{\bar{\sigma}}) n_\sigma\rangle \delta\left(E - \varepsilon_\sigma\right), \\
G_\sigma^>(E) &= -2\pi \mathrm{i} \langle n_{\bar{\sigma}}(1 - n_\sigma)\rangle \delta\left(E - \varepsilon_\sigma - U\right) \\
&\quad - 2\pi \mathrm{i} \langle(1 - n_{\bar{\sigma}})(1 - n_\sigma)\rangle \delta\left(E - \varepsilon_\sigma\right), \qquad \text{(A.64)}
\end{aligned}$$

where $E^\pm = E \pm \mathrm{i}0^+$. The statistical weights $\rho_{00}, \rho_{10}, \rho_{01}, \rho_{11}$ are to be determined by solving Eq. (A.63). In the limit $U \to 0$, the Hamiltonian

H_{cent} returns to the quadratic form and the Green's functions recover the single particle solution

$$\begin{aligned} G_\sigma^r(E) &= \frac{1}{E^+ - \varepsilon_\sigma}, \\ G_\sigma^a(E) &= \frac{1}{E^- - \varepsilon_\sigma}, \\ G_\sigma^<(E) &= 2\pi \mathrm{i} \langle n_\sigma \rangle \delta(E - \varepsilon_\sigma), \\ G_\sigma^>(E) &= -2\pi \mathrm{i} (1 - \langle n_\sigma \rangle) \delta(E - \varepsilon_\sigma). \end{aligned} \tag{A.65}$$

Comparing Eq. (A.64) to Eq. (A.65), one can see that the statistical weights appear in both $G^{r,a}$ and $G^{<,>}$ for non-quadratic Hamiltonian whereas they appear only in $G^{<,>}$ for quadratic Hamiltonian.

To sum up, the Lehmann spectrum of NEGF has been derived in Eqs. (A.58,A.59,A.60,A.61). NEGF in the equilibrium limit and the weak coupling limit are further investigated by using the Lehmann spectrum.

A.5 Low concentration approximation

In Chapter 2, the NECPA theory is derived by applying the Langreth theorem to the contour ordered CPA equation. The approach is more or less algebraic. In this section, we investigate an alternative approach which is more or less geometric. The new approach is based on the multiple scattering theory and the diagrammatic technique [7]. The two approaches are complementary but equivalent. In the low disorder concentration limit, we shall make a further approximation to derive analytical formulas for transmission coefficient and transmission variation.

A.5.1 *Multiple scattering theory*

Consider a two-probe system whose central scattering region contains some disorder sites (see Fig. 2.4). The on-site energies of the disorder sites are discrete random variables. Namely, on a disorder site i the on-site energy ε_i can take the value ε_{iq} with the probability x_{iq} where q indicates the possible atomic species on that site and normalization requires $\sum_q x_{iq} = 1$. In some applications, the disorder concentration is very low. For example, in semiconductor devices even for heavily doped Si, the impurity concentration 10^{20} cm^{-3} amounts to $x \sim 2 \times 10^{-3}$. The system is mainly composed of host atoms (e.g., Si), while the impurity atoms (e.g., P, N) shift the Fermi level and induce disorder scattering. Below we shall carry out a perturbation analysis of the disorder scattering.

www.ebook3000.com

The central region of the pure system is described by the Hamiltonian H_0 which is a definite variable. The central region of the doped system is described by the Hamiltonian H which is a random variable. The impurity atoms induce a random scattering potential $\hat{V}$ which can be viewed as a perturbation to H_0

$$H = H_0 + \hat{V}, \tag{A.66}$$

$$\hat{V} = \sum_i \hat{V}_i, \tag{A.67}$$

where $\hat{V}_i$ is the scattering potential on the disorder site i. $\hat{V}_i$ is a nearly all-zero matrix except for its i^{th} diagonal element

$$\hat{V}_i = diag\left[0, \ldots, 0, V_i, 0, \ldots, 0\right], \tag{A.68}$$

where V_i is a discrete random variable which can take the value V_{iq} with probability x_{iq}. V_{iq} is defined by the on-site energy difference between the impurity atom ($q = 0$) and the host atom ($q > 0$)

$$V_{iq} = \varepsilon_{iq} - \varepsilon_{i0}. \tag{A.69}$$

The retarded Green's function of the pure system and the doped system are G_0^r and G^r

$$G_0^r = \left(E - H_0 - \Sigma^r\right)^{-1},$$
$$G^r = \left(E - H - \Sigma^r\right)^{-1}.$$

Notice that G_0^r is a definite variable and G^r is a random variable. G^r can be expressed in terms of G_0^r with aid of T^r

$$G^r = G_0^r + G_0^r T^r G_0^r, \tag{A.70}$$

where T^r is defined by

$$T^r \equiv \hat{V}\left(1 - G_0^r \hat{V}\right)^{-1}. \tag{A.71}$$

T^r can be further expanded in terms of the scattering amplitude $\hat{t}_i^r$ [8]

$$T^r = \sum_i \hat{t}_i^r + \sum_i \sum_{j \neq i} \hat{t}_j^r G_0^r \hat{t}_i^r + \sum_i \sum_{j \neq i} \sum_{k \neq j} \hat{t}_k^r G_0^r \hat{t}_j^r G_0^r \hat{t}_i^r + \cdots, \tag{A.72}$$

where $\hat{t}_i^r$ represents multiple disorder scattering on the site i

$$\hat{t}_i^r \equiv \hat{V}_i + \hat{V}_i G_0^r \hat{V}_i + \hat{V}_i G_0^r \hat{V}_i G_0^r \hat{V}_i + \cdots$$
$$= \hat{V}_i \left(1 - G_0^r \hat{V}_i\right)^{-1}. \tag{A.73}$$

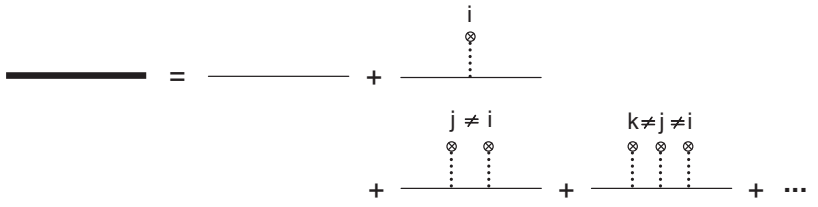


Fig. A.5 Diagram representation of Eq. (A.75).

Similar to $\hat{V}_i$, $\hat{t}_i^r$ is also a nearly all-zero matrix except for its i^{th} diagonal element

$$\hat{t}_i^r = diag\left[0, \ldots, 0, t_i^r, 0, \ldots, 0\right],$$

where t_i^r is a random variable which can take the value t_{iq}^r with probability x_{iq}. t_{iq}^r is obtained as

$$t_{iq}^r \equiv V_{iq}\left(1 - G_{0,ii}^r V_{iq}\right)^{-1}, \tag{A.74}$$

where $G_{0,ii}^r$ means to take the i^{th} diagonal element of G_0^r.

Inserting Eq. (A.72) into Eq. (A.70), G^r can be expanded into a series of scattering terms:

$$\begin{aligned} G^r = G_0^r + \sum_i G_0^r \hat{t}_i^r G_0^r + \sum_i \sum_{j\neq i} G_0^r \hat{t}_j^r G_0^r \hat{t}_i^r G_0^r \\ + \sum_i \sum_{j\neq i} \sum_{k\neq j} G_0^r \hat{t}_k^r G_0^r \hat{t}_j^r G_0^r \hat{t}_i^r G_0^r + \cdots . \end{aligned} \tag{A.75}$$

In the diagrammatic language, Eq. (A.75) can be represented by Fig. A.5 in which the thick line represents G^r, the thin line represents G_0^r, and the dotted line with a crossed dot represents $\hat{t}_i^r$ (random variable). It is required that adjacent $\hat{t}_i^r$ lines must have different site indices. Analogously a similar expansion can be carried out for the advanced Green's function G^a.

The t-matrix expansion in Eq. (A.75) is rigorous. In the low concentration limit, $x_{i,q=0} \gg x_{i,q>0}$, we make a further approximation to neglect higher order scattering terms and only keeps those terms proportional to the small parameter $x_{i,q>0}$. This is referred to as the low concentration approximation (LCA). By using the approximation, Eq. (A.75) is greatly simplified as

$$G^r \approx G_0^r + \sum_i G_0^r \hat{t}_i^r G_0^r, \tag{A.76}$$

which will be applied to studying disorder-averaged physical quantities in the next subsection.

www.ebook3000.com

A.5.2 *Transmission coefficient and transmission variation*

In this subsection, we shall derive analytical formulas of disorder-averaged transmission coefficient and transmission variation in LCA. We shall see that the two quantities correspond to the disorder average of two Green's functions product $\overline{G \cdot G}$ and four Green's functions product $\overline{G \cdot G \cdot G \cdot G}$ respectively.

Transmission coefficient can be expressed in terms of two Green's functions product

$$\overline{T} = \mathrm{Tr}\left\{\overline{G^r \Gamma_L G^a \Gamma_R}\right\}.$$

By using Eq. (A.76), $\overline{G^r \Gamma_L G^a \Gamma_R}$ is reduced to

$$\begin{aligned}
\overline{G^r \Gamma_L G^a \Gamma_R} &\approx \overline{\left(G_0^r + \sum_{i_1} G_0^r \hat{t}_{i_1}^r G_0^r\right) \Gamma_L \left(G_0^a + \sum_{i_2} G_0^a \hat{t}_{i_2}^a G_0^a\right) \Gamma_R} \\
&= G_0^r \Gamma_L G_0^a \Gamma_R + \sum_{i_2} \overline{G_0^r \Gamma_L \left(G_0^a \hat{t}_{i_2}^a G_0^a\right) \Gamma_R} + \sum_{i_1} \overline{\left(G_0^r \hat{t}_{i_1}^r G_0^r\right) \Gamma_L G_0^a \Gamma_R} \\
&\quad + \sum_{i_1} \sum_{i_2} \overline{\left(G_0^r \hat{t}_{i_1}^r G_0^r\right) \Gamma_L \left(G_0^a \hat{t}_{i_2}^a G_0^a\right) \Gamma_R}.
\end{aligned}$$

It is straightforward to evaluate the first three terms by carrying out the disorder average. The last term is a little bit tricky: For $i_1 \neq i_2$ or $q_1 \neq q_2$, the term is proportional to $x_{i_1 q_1} x_{i_2 q_2}$ and hence is negligible. For $i_1 = i_2 \equiv i$ and $q_1 = q_2 \equiv q$, $G_0^r \hat{t}_{i_1}^r G_0^r$ and $G_0^a \hat{t}_{i_2}^a G_0^a$ are statistically correlated. The term is proportional to x_{iq} and must be included. As a result, the disorder-averaged transmission is derived as

$$\overline{T} = T_0 + \sum_{i,q>0} x_{iq} Y_{iq}^{\alpha} + \sum_{i,q>0} x_{iq} Y_{iq}^{\beta} + \sum_{i,q>0} x_{iq} Y_{iq}^{\gamma}, \tag{A.77}$$

where T_0 is the transmission coefficient of pure system

$$T_0 \equiv \mathrm{Tr}\ G_0^r \Gamma_L G_0^a \Gamma_R, \tag{A.78}$$

and Y_{iq}^{α}, Y_{iq}^{β}, Y_{iq}^{γ} are defined by

$$Y_{iq}^{\alpha} = \mathrm{Tr}\ t_{iq}^a \left(G_0^a \Gamma_R G_0^r \Gamma_L G_0^a\right)_{ii}, \tag{A.79}$$

$$Y_{iq}^{\beta} = \mathrm{Tr}\ t_{iq}^r \left(G_0^r \Gamma_L G_0^a \Gamma_R G_0^r\right)_{ii}, \tag{A.80}$$

$$Y_{iq}^{\gamma} = \mathrm{Tr}\ t_{iq}^r \left(G_0^r \Gamma_L G_0^a\right)_{ii} t_{iq}^a \left(G_0^a \Gamma_R G_0^r\right)_{ii}. \tag{A.81}$$

Notice that $\left(Y_{iq}^{\alpha}\right)^* = Y_{iq}^{\beta}$ and $\left(Y_{iq}^{\gamma}\right)^* = Y_{iq}^{\gamma}$. The four terms of Eq. (A.77) are represented by the LCA diagrams in Fig. A.6.

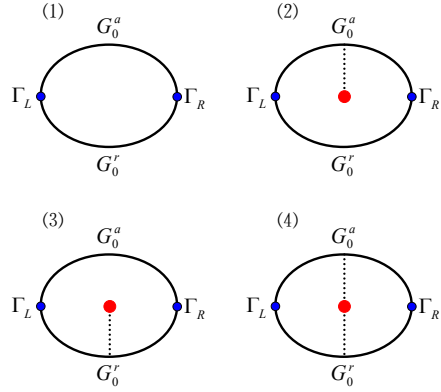


Fig. A.6 LCA diagrams of $\overline{T}$ corresponding to the four terms in Eq. (A.77).

The above procedure is from an algebraic deduction to a diagrammatic representation. Let's do it in reverse. We construct the LCA diagrams first and obtain the terms of Eq. (A.77) by reading the diagrams. In the LCA diagrams, the thick lines represent G_0^r or G_0^a; the blue dots represent Γ_L or Γ_R; the dotted line with a red dot represent t_{iq}^r or t_{iq}^a. Due to the cyclic property of trace operation, one can write the diagram elements clockwise from any point of the Green's function cycle. Notice that LCA diagrams are allowed to have a dangling impurity line or a contraction of impurity lines. Don't forget to add a statistical weight to the impurity line.

Eq. (A.77) provides the average of transmission coefficient in the presence of random disorder. One may also be interested in the variation of transmission coefficient due to disorder scattering. The variation of transmission coefficient is defined by

$$\delta T \equiv \sqrt{\overline{T^2} - \overline{T}^2}. \tag{A.82}$$

Since $\overline{T}$ has been calculated by Eq. (A.77), the remaining task is to evaluate $\overline{T^2}$ which involves a product of four Green's functions. This time we work on the diagrams directly. The LCA diagrams of $\overline{T^2}$ are constructed in Fig. A.7

It is observed that $\overline{T^2}$ has 7 disconnected diagrams which will cancel with the 7 low order terms in $\overline{T}^2$. The remaining 9 terms of $\overline{T^2} - \overline{T}^2$ can

www.ebook3000.com

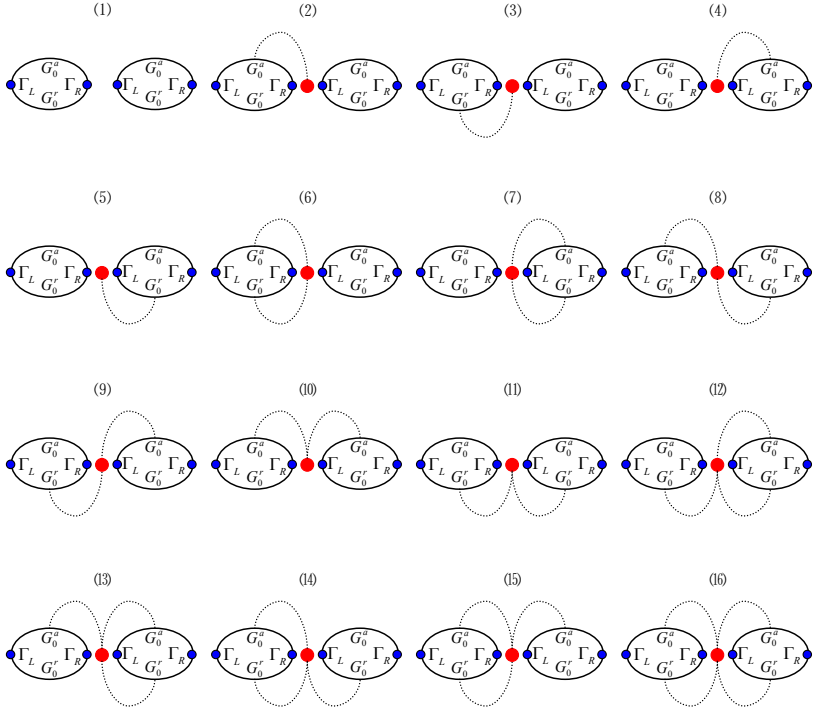


Fig. A.7 LCA diagrams of $\overline{T^2}$ corresponding to the 16 terms in the expansion of $\overline{T^2}$ to the lowest order of disorder concentration.

be reorganized to a compact form

$$\delta T^2 = \sum_{i,q>0} x_{iq} \left(Y_{iq}^{\alpha} + Y_{iq}^{\beta} + Y_{iq}^{\gamma} \right)^2, \tag{A.83}$$

where Y_{iq}^{α}, Y_{iq}^{β}, Y_{iq}^{γ} have been defined by Eqs. (A.79,A.80,A.81). It is inferred that $\delta T^2 > 0$ which is consistent with the physical meaning of the quantity. Note that the summation over i and q in Eq. (A.83) clearly identifies the contribution of each impurity species and disorder site to the total transmission variation.

To sum up, Eqs. (A.75,A.76,A.77,A.83) are the central results of this section. Interested readers are referred to Ref. [7] for more details on the diagrammatic technique and the transmission variation beyond LCA.

A.6 Scattering states approach

In this section, we present the scattering states approach to solve the nonequilibrium quantum transport problem. To elaborate the essential feature of the approach, let us make a comparison between equilibrium bulk calculations and nonequilibrium two-probe calculations. In equilibrium bulk calculations, we solve Bloch states and populate the states with Fermi–Dirac statistics. In nonequilibrium two-probe calculations, we solve scattering states and populate the states with local Fermi functions of the leads. As an illustration, the scattering states approach has been applied to solve the effective-mass model in Appendix A.3. Here we shall discuss a general formalism which is applicable to different models.

For the sake of generality, Schrödinger equation is rewritten as $h(E)\Psi = 0$ where Ψ is the wave function and $h(E)$ is the reduced Hamiltonian. The form of $h(E)$ depends on the type of basis set: In orthogonal atomic basis set $h(E) = E - H$ where H is the Hamiltonian matrix; In nonorthogonal atomic basis set, $h(E) = ES - H$ where H is the Hamiltonian matrix and S is the overlap matrix; In the LMTO method, $h(E) = P(E) - S(k)$ where $P(E)$ is the potential function and $S(k)$ is the Fourier transformed structure constant. In this section, all the formulas are based on $h(E)$ and hence are applicable to all types of basis set.

A.6.1 *Bulk states*

To study the wave scattering, we first solve the eigenstates of the leads. Those eigenstates will be the incoming and outgoing waves. Consider a 1d periodic system composed of repeated unit cells in the transport direction. Assume that only nearest neighbor unit cells have nonzero Hamiltonian elements. Consequently h is a block tridiagonal matrix, and the eigenstate Ψ satisfies

$$\begin{pmatrix} \ddots & \ddots & & & \\ \ddots & h_0 & h_+ & & \\ & h_- & h_0 & h_+ & \\ & & h_- & h_0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix} \Psi = 0, \tag{A.84}$$

www.ebook3000.com

where n is the unit cell index, h_0 and $h_\pm$ are $N \times N$ matrix blocks, and Ψ is defined by

$$\Psi \equiv \begin{pmatrix} \vdots \\ \phi_{n-1} \\ \phi_n \\ \phi_{n+1} \\ \vdots \end{pmatrix}.$$

Due to the periodicity, the eigenstates can be solved with the ansatz $\phi_n = \lambda^n \varphi_\lambda$. Eq. (A.84) is reduced to a quadratic eigenvalue problem

$$\left(h_- \lambda^{-1} + h_0 + h_+ \lambda\right) \varphi_\lambda = 0. \tag{A.85}$$

By introducing an auxiliary variable $\omega_\lambda \equiv \lambda \varphi_\lambda$, Eq. (A.85) is converted to a linear eigenvalue problem

$$\begin{pmatrix} 0 & 1 \\ h_- & h_0 \end{pmatrix} \begin{pmatrix} \varphi_\lambda \\ \omega_\lambda \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 \\ 0 & -h_+ \end{pmatrix} \begin{pmatrix} \varphi_\lambda \\ \omega_\lambda \end{pmatrix}. \tag{A.86}$$

Eq. (A.86) has $2N$ eigensolutions with eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_{2N}\}$ and eigenvectors $\{\varphi_{\lambda_1}, \varphi_{\lambda_2}, \ldots, \varphi_{\lambda_{2N}}\}$.

These $2N$ eigenstates can be classified into the following four categories: left traveling mode, left decaying mode, right traveling mode, right decaying mode, which are summarized as follows

mode	**symbol**	**direction**	**eigenvalue**	**group velocity**
left traveling	$\leftarrow$	$-$	$\lvert\lambda\rvert = 1$	$V_g < 0$
left decaying	$\uparrow$	$-$	$\lvert\lambda\rvert > 1$	N/A
right traveling	$\rightarrow$	$+$	$\lvert\lambda\rvert = 1$	$V_g > 0$
right decaying	$\downarrow$	$+$	$\lvert\lambda\rvert < 1$	N/A

(A.87)

In the last column, V_g is the group velocity defined by $V_g \equiv \frac{dE}{dk}$ where $\lambda = e^{ik}$ and $k \in (-\pi, \pi)$. Due to the time-reversal symmetry (see Section A.6.5), the number of left traveling modes is equal to the number of right traveling modes; the number of left decaying modes is equal to the number of right decaying modes. The left traveling and left decaying modes together are called left moving modes whose direction sign in Eq. (A.87) is $+$. The right traveling and right decaying modes together are called right moving modes whose direction sign in Eq. (A.87) is $-$. The eigenvalues and eigenvectors of the left moving modes are represented by

$$\begin{aligned} \Lambda_- &= diag\left(\left[\lambda_1^-, \lambda_2^-, \ldots, \lambda_N^-\right]\right), \\ \varphi_- &\equiv \left(\varphi_{\lambda_1^-}, \varphi_{\lambda_2^-}, \ldots, \varphi_{\lambda_N^-}\right). \end{aligned} \tag{A.88}$$

The eigenvalues and eigenvectors of the right moving modes are represented by

$$\Lambda_+ = diag\left(\left[\lambda_1^+, \lambda_2^+, \ldots, \lambda_N^+\right]\right), \tag{A.89}$$
$$\varphi_+ \equiv \left(\varphi_{\lambda_1^+}, \varphi_{\lambda_2^+}, \ldots, \varphi_{\lambda_N^+}\right).$$

Here $\Lambda_\pm$ and $\varphi_\pm$ are $N \times N$ matrices.

There are two methods to calculate the group velocity of traveling modes. The first method is to add a small imaginary part to the real energy and calculate V_g numerically. Suppose E is shifted to $\tilde{E} = E + \mathrm{i}\epsilon$. By solving the eigenvalue problem Eq. (A.86), the eigenvalue $\lambda = e^{\mathrm{i}k}$ is shifted to $\tilde{\lambda} = e^{\mathrm{i}\tilde{k}}$ where $\tilde{k} = k + i\eta$. V_g can be obtained as

$$V_g = \frac{\Delta E}{\Delta k} = \frac{\epsilon}{\eta}, \tag{A.90}$$

where $\eta = -\ln\left|\tilde{\lambda}\right|$. A consequence is that the left moving and right moving modes can be classified by the unified criteria $\left|\tilde{\lambda}\right| > 1$ and $\left|\tilde{\lambda}\right| < 1$ respectively. The second method is to calculate V_g analytically by using

$$V_g = \frac{dE}{dk} = -\mathrm{i}\frac{\varphi_\lambda^\dagger\left(h_+\lambda - h_-\lambda^{-1}\right)\varphi_\lambda}{\varphi_\lambda^\dagger\left(\dot{h}_-\lambda^{-1} + \dot{h}_0 + \dot{h}_+\lambda\right)\varphi_\lambda}, \tag{A.91}$$

where $\dot{h}_0$ and $\dot{h}_\pm$ are the energy derivatives of h_0 and $h_\pm$. The derivation of Eq. (A.91) is given in Section A.6.4.

A.6.2 *Wave scattering*

In the previous subsection, we have solved the eigenstates in the leads and classified them into four categories. In this subsection, we shall investigate how those eigenstates are scattered in the central region. The basic idea is quite similar to the scattering problem of a 1d δ-barrier (see Section 1.1).

Consider a traveling wave incoming from the left lead. The incoming wave can be scattered into either left traveling and decaying wave in the left lead or right traveling and decaying wave in the right lead (see Fig. A.8). The scattering wave function Ψ can be written as:

$$\Psi = \begin{cases} \Psi_{L\rightarrow} + \Psi_{L-} \cdot r_{L-,L\rightarrow} & \text{left lead} \\ \Psi_C & \text{central region} \\ \Psi_{R+} \cdot t_{R+,L\rightarrow} & \text{right lead} \end{cases}, \tag{A.92}$$

www.ebook3000.com

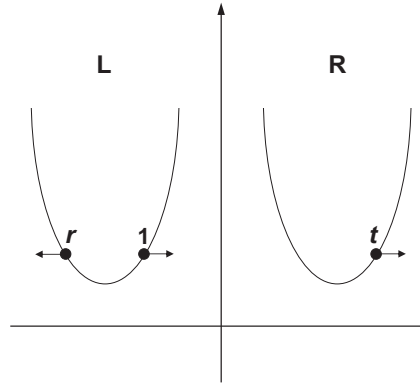


Fig. A.8 Schematic plot of wave scattering. An incoming wave from the left lead (marked by 1) is scattered by the central region. Part of the wave is transmitted to the right lead (marked by t) and part of the wave is reflected back to the left lead (marked by r).

where r is the matrix of reflection amplitude and t is the matrix of transmission amplitude. Ψ satisfies the Schrödinger equation $h\Psi = 0$, leading to

$$\begin{pmatrix} \ddots & \ddots & & & & & & \\ \ddots & h^0_{LL} & h^+_{LL} & & & & & \\ & h^-_{LL} & h^0_{LL} & h_{LC} & & & & \\ & & h_{CL} & h_{CC} & h_{CR} & & & \\ & & & h_{RC} & h^0_{RR} & h^+_{RR} & & \\ & & & & h^-_{RR} & h^0_{RR} & \ddots & \\ & & & & & \ddots & \ddots & \end{pmatrix} \Psi = 0. \tag{A.93}$$

Here Ψ is defined by

$$\Psi \equiv \begin{pmatrix} \vdots \\ \phi_{-2} \\ \phi_{-1} \\ \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \end{pmatrix},$$

and the components of Ψ are obtained as

$$\begin{aligned}
&\cdots \\
&\phi_{-2} = \varphi_{L\rightarrow}\Lambda_{L\rightarrow}^{-1} + \varphi_{L-}\Lambda_{L-}^{-1}\cdot r_{L-,L\rightarrow} \\
&\phi_{-1} = \varphi_{L\rightarrow} + \varphi_{L-}\cdot r_{L-,L\rightarrow} \\
&\phi_{0} = \varphi_{C} \\
&\phi_{1} = \varphi_{R+}\cdot t_{R+,L\rightarrow} \\
&\phi_{2} = \varphi_{R+}\Lambda_{R+}\cdot t_{R+,L\rightarrow} \\
&\cdots
\end{aligned} \quad , \tag{A.94}$$

according to Eq. (A.92).

Notice that the lead eigenstates $\Psi_{L\rightarrow}, \Psi_{L-}, \Psi_{R+}$ satisfy the Schrödinger equation of the lead. Hence most lines of Eq. (A.93) have already been satisfied. One only needs to solve the three rows in the middle

$$\begin{pmatrix} h_{LL}^{-} & h_{LL}^{0} & h_{LC} & 0 & 0 \\ 0 & h_{CL} & h_{CC} & h_{CR} & 0 \\ 0 & 0 & h_{RC} & h_{RR}^{0} & h_{RR}^{+} \end{pmatrix} \begin{pmatrix} \phi_{-2} \\ \phi_{-1} \\ \phi_{0} \\ \phi_{1} \\ \phi_{2} \end{pmatrix} = 0, \tag{A.95}$$

which can be simplified as

$$\begin{aligned}
&h_{LL}^{-}\left(\varphi_{L\rightarrow}\Lambda_{L\rightarrow}^{-1} + \varphi_{L-}\Lambda_{L-}^{-1}\cdot r_{L-,L\rightarrow}\right) \\
&\quad + h_{LL}^{0}\left(\varphi_{L\rightarrow} + \varphi_{L-}\cdot r_{L-,L\rightarrow}\right) + h_{LC}\left(\varphi_{C}\right) = 0, \qquad \text{(A.96)} \\
&h_{CL}\left(\varphi_{L\rightarrow} + \varphi_{L-}\cdot r_{L-,L\rightarrow}\right) \\
&\quad + h_{CC}\left(\varphi_{C}\right) + h_{CR}\left(\varphi_{R+}\cdot t_{R+,L\rightarrow}\right) = 0, \qquad \text{(A.97)} \\
&h_{RC}\left(\varphi_{C}\right) + h_{RR}^{0}\left(\varphi_{R+}\cdot t_{R+,L\rightarrow}\right) \\
&\quad + h_{RR}^{+}\left(\varphi_{R+}\Lambda_{R+}\cdot t_{R+,L\rightarrow}\right) = 0. \qquad \text{(A.98)}
\end{aligned}$$

Similarly one can construct the scattering wave function with the incoming traveling wave from the right lead:

$$\Psi = \begin{cases} \Psi_{L-}\cdot t_{L-,R\leftarrow} & \text{left lead} \\ \tilde{\Psi}_{C} & \text{central region} \\ \Psi_{R\leftarrow} + \Psi_{R+}\cdot r_{R+,R\leftarrow} & \text{right lead} \end{cases} , \tag{A.99}$$

and derive a similar equation array to Eqs. (A.96,A.97,A.98). It turns out that the two equation arrays have the same coefficient matrix and can be

www.ebook3000.com

written in a unified form

$$\begin{pmatrix} h^0_{LL}\varphi_{L-} + h^-_{LL}\varphi_{L-}\Lambda^{-1}_{L-} & h_{LC} & 0 \\ h_{CL}\varphi_{L-} & h_{CC} & h_{CR}\varphi_{R+} \\ 0 & h_{RC} & h^0_{RR}\varphi_{R+} + h^+_{RR}\varphi_{R+}\Lambda_{R+} \end{pmatrix} \begin{pmatrix} r_{L-,L\rightarrow} & t_{L-,R\leftarrow} \\ \varphi_C & \tilde{\varphi}_C \\ t_{R+,L\rightarrow} & r_{R+,R\leftarrow} \end{pmatrix}$$

$$= (-) \begin{pmatrix} h^0_{LL}\varphi_{L\rightarrow} + h^-_{LL}\varphi_{L\rightarrow}\Lambda^{-1}_{L\rightarrow} & 0 \\ h_{CL}\varphi_{L\rightarrow} & h_{CR}\varphi_{R\leftarrow} \\ 0 & h^0_{RR}\varphi_{R\leftarrow} + h^+_{RR}\varphi_{R\leftarrow}\Lambda_{R\leftarrow} \end{pmatrix}, \qquad \text{(A.100)}$$

which is the central result of this subsection.

Eq. (A.100) can be rewritten into a symmetric form

$$\begin{pmatrix} h^0_{LL} - \Sigma^r_L & h_{LC} & 0 \\ h_{CL} & h_{CC} & h_{CR} \\ 0 & h_{RC} & h^0_{RR} - \Sigma^r_R \end{pmatrix} \begin{pmatrix} \varphi_{L-} & & \\ & 1 & \\ & & \varphi_{R+} \end{pmatrix} \begin{pmatrix} r_{L-,L\rightarrow} & t_{L-,R\leftarrow} \\ \varphi_C & \tilde{\varphi}_C \\ t_{R+,L\rightarrow} & r_{R+,R\leftarrow} \end{pmatrix}$$

$$= (-) \begin{pmatrix} h^0_{LL}\varphi_{L\rightarrow} + h^-_{LL}\varphi_{L\rightarrow}\Lambda^{-1}_{L\rightarrow} & 0 \\ h_{CL}\varphi_{L\rightarrow} & h_{CR}\varphi_{R\leftarrow} \\ 0 & h^0_{RR}\varphi_{R\leftarrow} + h^+_{RR}\varphi_{R\leftarrow}\Lambda_{R\leftarrow} \end{pmatrix}, \qquad \text{(A.101)}$$

where Σ^r_L and Σ^r_R are the lead self-energies (see Section 6.4.3)

$$\Sigma^r_L = -h^-_{LL}\varphi_{L-}\Lambda^{-1}_{L-}\varphi^{-1}_{L-} , \qquad \text{(A.102)}$$

$$\Sigma^r_R = -h^+_{RR}\varphi_{R+}\Lambda_{R+}\varphi^{-1}_{R+} . \qquad \text{(A.103)}$$

One may have noticed that the first matrix in the LHS of Eq. (A.101) is nothing but $(G^r)^{-1}$ which connects the Green's function approach and the wave function approach.

A.6.3 *Transmission coefficient*

The transmission coefficient can be calculated by using the scattering matrix. The scattering matrix can be constructed by using the reflection amplitude r and the transmission amplitude t defined in the scattering wave function Eqs. (A.92,A.99). Notice that r and t have been determined up to an arbitrary normalization factor. To make the scattering matrix a unitary matrix, one needs to properly normalize the incoming waves and outgoing waves so that they carry the same amount of current. Here the normalization of scattering states is more complicated than that of discrete eigenstates because the former ones are continuous in energy.

Consider a small energy interval $(E_0, E_0 + \Delta E)$ which contains a group of traveling modes. The normalization of $\varphi_\lambda(E_0)$ requires that the total probability of traveling states is equal to the total number of traveling

states. In the energy interval, $h_\lambda(E) \equiv h_-(E)\lambda^{-1} + h_0(E) + h_+(E)\lambda$ can be linearly expanded as

$$h_\lambda(E) \approx h_\lambda(E_0) + \dot{h}_\lambda(E_0)(E - E_0) \equiv ES_\lambda - H_\lambda, \tag{A.104}$$

where S_λ and H_λ are defined by

$$S_\lambda \equiv \dot{h}_\lambda(E_0) = \dot{h}_-(E_0)\lambda^{-1} + \dot{h}_0(E_0) + \dot{h}_+(E_0)\lambda, \tag{A.105}$$

$$H_\lambda = -h_\lambda(E_0) + E_0 S_\lambda. \tag{A.106}$$

The total probability of traveling states is obtained as

$$\varphi_\lambda^\dagger(E_0) S_\lambda(E_0) \varphi_\lambda(E_0) \Delta E.$$

The total number of traveling states is obtained as

$$D(E_0)\Delta E = \frac{dk}{dE}\Delta E = \frac{1}{V_g}\Delta E, \tag{A.107}$$

where V_g is derived in Eq. (A.91) (see Section A.6.4). The normalization condition leads to

$$\varphi_\lambda^\dagger(E_0) S_\lambda(E_0) \varphi_\lambda(E_0) \Delta E = D(E_0)\Delta E, \tag{A.108}$$

which can be simplified as (E_0 is omitted)

$$\varphi_\lambda^\dagger(-\mathrm{i}h_+\lambda + H.c.)\varphi_\lambda = 1, \tag{A.109}$$

where Eqs. (A.91,A.105,A.107) have been used in the derivation.

Once the traveling modes are properly normalized, it is straightforward to construct the scattering matrix $\mathbf{S}$. By definition the elements of the scattering matrix are the scattering amplitudes of traveling waves. One simply removes the decaying modes in the r and t matrices and obtain

$$\mathbf{S} = \begin{pmatrix} \mathbf{r}_{LL} = r_{L\leftarrow,L\rightarrow} & \mathbf{t}_{LR} = t_{L\leftarrow,R\leftarrow} \\ \mathbf{t}_{RL} = t_{R\rightarrow,L\rightarrow} & \mathbf{r}_{RR} = r_{R\rightarrow,R\leftarrow} \end{pmatrix}, \tag{A.110}$$

where the column indices are incoming traveling waves and the row indices are outgoing waves. Due to probability conservation, the scattering matrix is unitary $\mathbf{S}\mathbf{S}^\dagger = 1$.

The traveling modes in the leads are also called conducting channels. Transmission coefficient is a sum over the transmission probability of all conducting channels

$$T(E) = \sum_{ij} \left|(\mathbf{t}_{RL})_{ij}\right|^2 = \mathrm{Tr}\ \mathbf{t}_{RL}\mathbf{t}_{RL}^\dagger. \tag{A.111}$$

www.ebook3000.com

A.6.4 *Further discussion: group velocity*

In this subsection, we derive the analytical expression of the group velocity analogous to the Hellmann–Feynman theorem. Suppose φ_λ is a traveling state satisfying

$$\left(h_-\lambda^{-1} + h_0 + h_+\lambda\right)\varphi_\lambda = 0. \tag{A.112}$$

For a traveling state, the eigenvalue $|\lambda| = 1$ and hence $\lambda^* = \lambda^{-1}$. The Hermitian conjugate of Eq. (A.112) leads to

$$\varphi_\lambda^\dagger\left(h_-\lambda^{-1} + h_0 + h_+\lambda\right) = 0, \tag{A.113}$$

where $h_0^\dagger = h_0$, $h_+^\dagger = h_-$, and $\lambda^* = \lambda^{-1}$ are used in the derivation.

Notice that λ, φ_λ, and h are dependent on the energy E. Applying the energy derivative $\frac{d}{dE}$ to Eq. (A.113) and multiplying φ_λ to the right, one obtains

$$\dot{\lambda} = \frac{\varphi_\lambda^\dagger\left(\dot{h}_-\lambda^{-1} + \dot{h}_0 + \dot{h}_+\lambda\right)\varphi_\lambda}{\varphi_\lambda^\dagger\left(h_-\lambda^{-2} - h_+\right)\varphi_\lambda}, \tag{A.114}$$

By using $\lambda = e^{\mathrm{i}k}$, the group velocity V_g is obtained as

$$\begin{aligned} V_g &= \frac{\partial E}{\partial k} \\ &= \mathrm{i}\lambda\frac{\partial E}{\partial \lambda} \\ &= -\mathrm{i}\frac{\varphi_\lambda^\dagger\left(h_+\lambda - h_-\lambda^{-1}\right)\varphi_\lambda}{\varphi_\lambda^\dagger\left(\dot{h}_-\lambda^{-1} + \dot{h}_0 + \dot{h}_+\lambda\right)\varphi_\lambda}. \end{aligned} \tag{A.115}$$

A.6.5 *Further discussion: number of modes*

In this subsection, we prove that the number of left decaying / traveling modes is equal to the number of right decaying / traveling modes.

Firstly we study the decaying modes. Applying Hermitian conjugate to Eq. (A.112), one obtains

$$\varphi_\lambda^\dagger\left(h_-\tilde{\lambda}^{-1} + h_0 + h_+\tilde{\lambda}\right) = 0, \tag{A.116}$$

where $\tilde{\lambda} \equiv \frac{1}{\lambda^*}$, $h_0^\dagger = h_0$ and $h_+^\dagger = h_-$ are used in the derivation. It indicates that if λ is an eigenvalue, $\frac{1}{\lambda^*}$ is also an eigenvalue. Therefore each left decaying mode whose $|\lambda| > 1$ can be mapped to a right decaying mode whose $|\lambda| < 1$ and vice versa. Consequently the number of left and right decaying modes are equal.

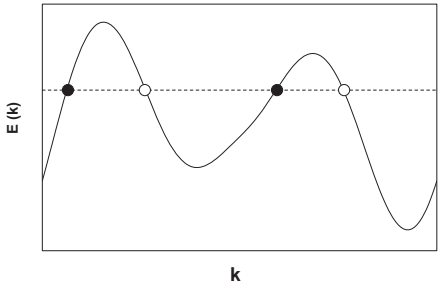


Fig. A.9 Schematic plot of a 1d band structure. At a given energy E, there are four eigen states. Two eigen states have positive group velocity (black dots), and the other two have negative group velocity (white dots).

Next we study the traveling modes. Although Eq. (A.116) is still valid, the eigenvalue will be mapped to itself for traveling modes ($\lambda = \frac{1}{\lambda^*}$). So we have to change tactics. Eq. (A.112) defines an eigenvalue problem of λ where the eigenvalues are solved for a given E. From an alternative point of view, Eq. (A.112) can be viewed as an eigenvalue problem of E where the eigenvalues are solved for a given $\lambda = e^{ik}$. For $k \in (-\pi, \pi)$, $|\lambda| = 1$, and all the eigenstates of E constitute a 1d band structure. The 1d band structure has N bands $\{E_n(k)\}$, each of which is a periodic function of k. At a given E, the traveling states are the intersections of $z = E$ and $z = E_n(k)$, see Fig. A.9. Due to the periodicity of $E_n(k)$, there are always an even number of intersections at which the slopes change signs alternately. Therefore half the eigenstates have positive group velocity and the other half has negative group velocity.

It is worth mentioning that the above argument is quite general and does not rely on the time-reversal symmetry.

A.6.6 *Further discussion: numerical issues*

In this subsection, we discuss some tricks in the implementation to avoid numerical instabilities and accelerate the calculation.

First of all, the group velocity V_g can be zero at some special energies where the left-traveling mode and the right-traveling mode are degenerate. One may have problems classifying the bulk states and normalizing the wave functions. As a treatment, one can add a small imaginary part to E so that the degeneracy between the left-traveling mode and the right-traveling mode are lifted slightly. As a consequence the eigenvalues of the traveling mode change from $|\lambda_\rightarrow| = |\lambda_\leftarrow| = 1$ to $|\lambda_\rightarrow| = 1 - \delta$ and

www.ebook3000.com

$|\lambda_{\leftarrow}| = 1 + \delta$. Therefore one needs to set up a tolerance $\delta_0 > 0$ to separate traveling modes from decaying modes.

Secondly, the eigenvalues of bulk states may have several zeros and infinities. The eigenvectors of infinities are problematic and make the coefficient matrix of Eq. (A.100) close to singular. Further investigation indicates that these zero and infinity states are induced by the rank deficiency of $h_{LL}^{\pm}$ and $h_{RR}^{\pm}$ and hence are not physical states. To eliminate these states, one may set up a tolerance $0 < \lambda_0 < 1$ and discard the decaying modes with eigenvalues $|\lambda| < \lambda_0$ or $|\lambda| > \frac{1}{\lambda_0}$. As a consequence φ_{L-} and φ_{R+} are no longer rectangular and Eq. (A.100) becomes an over-determined linear equation array. One needs to make a linear combination of the over-determined equations with random coefficients to remove the redundancy. In addition, it is recommended to include an extra lead unit cell in the central region to suppress the influence of the neglected decaying modes.

Finally, the solution of Eq. (A.100) can be accelerated by using the block tridiagonal technique. Suppose the two-probe system has been partitioned into $M-2$ principal layers along the transport direction (see Fig. 7.5), Eq. (A.100) can be written in a block tridiagonal form,

$$\begin{pmatrix} A_{1,1} & A_{1,2} & & & & \\ A_{2,1} & A_{2,2} & A_{2,3} & & & \\ & A_{3,2} & A_{3,3} & A_{3,4} & & \\ & & A_{4,3} & \ddots & \ddots & \\ & & & \ddots & A_{M-1,M-1} & A_{M-1,M} \\ & & & & A_{M,M-1} & A_{M,M} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{M-1} \\ x_M \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{M-1} \\ b_M \end{pmatrix}. \tag{A.117}$$

By using Gaussian elimination, the solution of Eq. (A.117) can be obtained as

$$\begin{aligned} x_M &= \tilde{A}_{M,M}^{-1} \tilde{b}_M, \\ x_i &= \tilde{A}_{i,i}^{-1} \left(\tilde{b}_i - A_{i,i+1} x_{i+1} \right), \end{aligned} \tag{A.118}$$

with $i = M-1, M-2, \ldots, 1$. Here $\tilde{A}_{i,i}$ and $\tilde{b}_i$ are defined by

$$\tilde{A}_{1,1} = A_{1,1},$$

$$\tilde{A}_{i,i} = A_{i,i} - A_{i,i-1} A_{i-1,i-1}^{-1} A_{i-1,i}, \tag{A.119}$$

$$\tilde{b}_1 = b_1,$$

$$\tilde{b}_i = b_i - A_{i,i-1} A_{i-1,i-1}^{-1} b_{i-1}, \tag{A.120}$$

with $i = 2, 3, \ldots, M$.

A.6.7 *Summary*

To sum up, the quantum transport is formulated in the scattering states approach. The scattering states satisfy Eq. (A.100) where lead eigenstates are defined by Eqs. (A.86,A.87) and normalized by Eq. (A.109). The transmission coefficient can be calculated with the scattering matrix Eqs. (A.110,A.111). Some numerical issues are also discussed for the implementation. We would like to emphasize that the scattering states approach is totally equivalent to the Green's function approach in clean two-probe systems. However it is much easier to adopt the Green's function approach in disordered two-probe systems.

A.7 Density matrix in clean bulk systems

In clean bulk systems, the density matrix and other physical quantities can be expressed in terms of the Bloch states. In this section, we make a connection from the Green's function approach to the wave function approach.

In the wave function approach, the Bloch states are solved from the eigenvalue problem

$$H_{orth}\left(k\right)\Psi_{i}\left(k\right)=\lambda_{i}\left(k\right)\Psi_{i}\left(k\right), \tag{A.121}$$

where $H_{orth}\left(k\right)$ is defined by Eq. (3.36). The density matrix can be expressed in terms of the Block states

$$\rho=\int_{BZ}\frac{d^{3}k}{\left(2\pi\right)^{3}}\sum_{i}f\left[\lambda_{i}\left(k\right)\right]\Psi_{i}\left(k\right)\Psi_{i}^{\dagger}\left(k\right), \tag{A.122}$$

where the Bloch states are orthonormalized to $\Psi_{i}^{\dagger}\left(k\right)\Psi_{j}\left(k\right)=\delta_{ij}$. Here $f\left(E\right)$ is the Fermi function defined by

$$f\left(E\right)\equiv\frac{1}{e^{\frac{E-\mu}{k_{B}T}}+1}. \tag{A.123}$$

The chemical potential μ is determined by the charge neutrality condition

$$\int_{BZ}\frac{d^{3}k}{\left(2\pi\right)^{3}}\sum_{i}f\left[\lambda_{i}\left(k\right)\right]=\sum_{i}Z_{i}^{val}, \tag{A.124}$$

where Z_{i}^{val} is the valence electron number of the i^{th} atom. Notice that Eq. (A.124) can be solved easily by using the bisection algorithm.

In the Green's function approach, the density matrix ρ is related to $G^{<}$ by

$$\rho=\int\frac{dE}{2\pi}\int_{BZ}\frac{d^{3}k}{\left(2\pi\right)^{3}}\left(-\mathrm{i}\right)G^{<}\left(E,k\right). \tag{A.125}$$

www.ebook3000.com

By using the fluctuation-dissipation theorem, Eq. (A.125) can be simplified as

$$\rho = \int \frac{dE}{2\pi} \int_{BZ} \frac{d^3k}{(2\pi)^3} (-\mathrm{i}) \left[G^a(E,k) - G^r(E,k)\right] f(E), \tag{A.126}$$

where G^r and G^a are defined by

$$G^r(E,k) = \left[E^+ - H_{orth}(k)\right]^{-1}, \tag{A.127}$$

$$G^a(E,k) = \left[E^- - H_{orth}(k)\right]^{-1}. \tag{A.128}$$

To connect the Green's function approach to the wave function approach, $H_{orth}(k)$ is diagonalized with the Bloch states

$$H_{orth}(k) = \Psi(k) \Lambda(k) \Psi^\dagger(k), \tag{A.129}$$

where $\Psi(k) = [\Psi_1(k), \Psi_2(k), \cdots]$ is the eigenvector matrix and $\Lambda(k) = diag([\lambda_1(k), \lambda_2(k), \cdots])$ is the eigenvalue matrix. Notice that $\Psi^\dagger(k)\Psi(k) = 1$ due to the orthonormalization. Inserting Eq. (A.129) into Eqs. (A.127,A.128), one obtains

$$(-\mathrm{i})\left[G^a(E,k) - G^r(E,k)\right] = 2\pi\Psi(k)\,\delta\left[E - \Lambda(k)\right]\Psi^\dagger(k),$$

where Eq. (2.53) is used in the derivation. As a result, Eq. (A.126) is reduced to

$$\rho = \int_{BZ} \frac{d^3k}{(2\pi)^3} \Psi(k) f\left[\Lambda(k)\right] \Psi^\dagger(k), \tag{A.130}$$

which is equivalent to Eq. (A.122). Thus the density matrix can be calculated either with the energy integral of lesser Green's function or the product of Bloch states and the statistical weights. Similarly other physical quantities can also be expressed in terms of the Bloch states. For example, the energy moment $\tilde{M}_i^l$ defined by Eq. (3.133) can be reduced to

$$\tilde{M}_i^l = \int_{BZ} \frac{d^3k}{(2\pi)^3} \left\{\Psi(k) f\left[\Lambda(k)\right] \Lambda^l(k) \Psi^\dagger(k)\right\}_{ii}. \tag{A.131}$$

To sum up, the Green's function approach is equivalent to the wave function approach in clean bulk systems. Consequently step-6 and step-7 in the flowchart Fig. 3.5 can be replaced by solving the Bloch states with Eq. (A.121), solving the Fermi level with Eq. (A.124), calculating the density matrix with Eq. (A.130), and calculating the energy moment with Eq. (A.131).

A.8 Connection to the CPA-NVC theory

This section proves the equivalence of Eq. (6.145) and Eq. (6.146), which makes a connection from the NECPA theory to the CPA-NVC theory. Hereafter the argument E is omitted for simplicity.

Proof: (1) Simplify Eq. (6.145). By using Eq. (6.144), one can eliminate $\Omega_i^<$ in Eq. (6.145),

$$\begin{aligned}\tilde{P}_i^< = & \left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_i^<\left(\overline{\mathcal{G}}_i^a\right)^{-1} \\ & -\sum_q x_{iq}\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_{iq}^r\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_i^<\left(\overline{\mathcal{G}}_i^a\right)^{-1}\overline{\mathcal{G}}_{iq}^a\left(\overline{\mathcal{G}}_i^a\right)^{-1} \\ & +\tilde{P}_i^< - \sum_q x_{iq}\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_{iq}^r\tilde{P}_i^<\overline{\mathcal{G}}_{iq}^a\left(\overline{\mathcal{G}}_i^a\right)^{-1}. \end{aligned} \tag{A.132}$$

(2) Derive a useful relation of $t_{iq}^r, \overline{\mathcal{G}}_i^r, \overline{\mathcal{G}}_{iq}^r$. Eliminating Ω_i^r from the following expressions of $\overline{\mathcal{G}}_i^r$ and $\overline{\mathcal{G}}_{iq}^r$

$$\begin{aligned}\overline{\mathcal{G}}_i^r &= \left(\tilde{P}_i^r - \Omega_i^r\right)^{-1}, \\ \overline{\mathcal{G}}_{iq}^r &= \left(P_{iq} - \Omega_i^r\right)^{-1},\end{aligned}$$

one obtains

$$\tilde{P}_i^r - P_{iq} = \left(\overline{\mathcal{G}}_i^r\right)^{-1} - \left(\overline{\mathcal{G}}_{iq}^r\right)^{-1}. \tag{A.133}$$

Inserting Eq. (A.133) into the definition of t_{iq}^r Eqs. (6.147,6.148), one obtains

$$t_{iq}^r = \left(\overline{\mathcal{G}}_i^r\right)^{-1}\left(\overline{\mathcal{G}}_{iq}^r - \overline{\mathcal{G}}_i^r\right)\left(\overline{\mathcal{G}}_i^r\right)^{-1}. \tag{A.134}$$

(3) Simplify Eq. (6.146). The RHS of Eq. (6.146) has two terms. By using Eq. (A.134), the first term can be simplified as

$$\begin{aligned}\sum_q x_{iq}t_{iq}^r\overline{\mathcal{G}}_i^<t_{iq}^a = & \sum_q x_{iq}\left(\overline{\mathcal{G}}_i^r\right)^{-1}\left(\overline{\mathcal{G}}_{iq}^r - \overline{\mathcal{G}}_i^r\right)\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_i^<\left(\overline{\mathcal{G}}_i^a\right)^{-1} \\ & \times\left(\overline{\mathcal{G}}_{iq}^a - \overline{\mathcal{G}}_i^a\right)\left(\overline{\mathcal{G}}_i^a\right)^{-1} \\ = & \sum_q x_{iq}\left[\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_{iq}^r\left(\overline{\mathcal{G}}_i^r\right)^{-1} - \left(\overline{\mathcal{G}}_i^r\right)^{-1}\right]\overline{\mathcal{G}}_i^< \\ & \times\left[\left(\overline{\mathcal{G}}_i^a\right)^{-1}\overline{\mathcal{G}}_{iq}^a\left(\overline{\mathcal{G}}_i^a\right)^{-1} - \left(\overline{\mathcal{G}}_i^a\right)^{-1}\right] \\ = & \sum_q x_{iq}\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_{iq}^r\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_i^<\left(\overline{\mathcal{G}}_i^a\right)^{-1}\overline{\mathcal{G}}_{iq}^a\left(\overline{\mathcal{G}}_i^a\right)^{-1} \\ & -\left(\overline{\mathcal{G}}_i^r\right)^{-1}\overline{\mathcal{G}}_i^<\left(\overline{\mathcal{G}}_i^a\right)^{-1}, \end{aligned} \tag{A.135}$$

www.ebook3000.com

where $\sum_q x_{iq} = 1$, $\sum_q x_{iq}\overline{\mathcal{G}}^r_{iq} = \overline{\mathcal{G}}^r_i$ and $\sum_q x_{iq}\overline{\mathcal{G}}^a_{iq} = \overline{\mathcal{G}}^a_i$ are used in the derivation. By using Eq. (A.134), the second term can be simplified as

$$\begin{aligned}
-\sum_q x_{iq} t^r_{iq}\overline{\mathcal{G}}^r_i \Lambda_i \overline{\mathcal{G}}^a_i t^a_{iq} &= \sum_q x_{iq} t^r_{iq}\overline{\mathcal{G}}^r_i \tilde{P}^<_i \overline{\mathcal{G}}^a_i t^a_{iq} \\
&= \sum_q x_{iq} \left(\overline{\mathcal{G}}^r_i\right)^{-1} \left(\overline{\mathcal{G}}^r_{iq} - \overline{\mathcal{G}}^r_i\right) \tilde{P}^<_i \left(\overline{\mathcal{G}}^a_{iq} - \overline{\mathcal{G}}^a_i\right) \left(\overline{\mathcal{G}}^a_i\right)^{-1} \\
&= \sum_q x_{iq} \left(\overline{\mathcal{G}}^r_i\right)^{-1} \overline{\mathcal{G}}^r_{iq} \tilde{P}^<_i \overline{\mathcal{G}}^a_{iq} \left(\overline{\mathcal{G}}^a_i\right)^{-1} - \tilde{P}^<_i, \qquad \text{(A.136)}
\end{aligned}$$

where $\sum_q x_{iq} = 1$, $\sum_q x_{iq}\overline{\mathcal{G}}^r_{iq} = \overline{\mathcal{G}}^r_i$ and $\sum_q x_{iq}\overline{\mathcal{G}}^a_{iq} = \overline{\mathcal{G}}^a_i$ are used in the derivation.

(4) Prove that Eq. (6.146) is equivalent to Eq. (A.132). By inserting Eqs. (A.135,A.136) to the RHS of Eq. (6.146), one obtains

$$\begin{aligned}
-\tilde{P}^<_i = &\sum_q x_{iq} \left(\overline{\mathcal{G}}^r_i\right)^{-1} \overline{\mathcal{G}}^r_{iq} \left(\overline{\mathcal{G}}^r_i\right)^{-1} \overline{\mathcal{G}}^<_i \left(\overline{\mathcal{G}}^a_i\right)^{-1} \overline{\mathcal{G}}^a_{iq} \left(\overline{\mathcal{G}}^a_i\right)^{-1} \\
&- \left(\overline{\mathcal{G}}^r_i\right)^{-1} \overline{\mathcal{G}}^<_i \left(\overline{\mathcal{G}}^a_i\right)^{-1} \\
&+ \sum_q x_{iq} \left(\overline{\mathcal{G}}^r_i\right)^{-1} \overline{\mathcal{G}}^r_{iq} \tilde{P}^<_i \overline{\mathcal{G}}^a_{iq} \left(\overline{\mathcal{G}}^a_i\right)^{-1} - \tilde{P}^<_i, \qquad \text{(A.137)}
\end{aligned}$$

which is equivalent to Eq. (A.132). QED.

To sum up, Eq. (6.145) is derived from the NECPA theory and Eq. (6.146) is derived from the CPA-NVC theory. Here we have demonstrated that the two theories are actually equivalent.

A.9 Explicit expressions of XC-functionals

In density functional theory, electrons are treated in the mean-field manner and the complexity of many-body interactions is hidden inside the XC-functionals. Although the existence of a universal XC-functional is guaranteed by the Hohenberg–Kohn theorem, nobody knows the exact form of the XC-functional. In practice, the XC-functional is constructed from known results of uniform electron gas with additional approximations. In this section, we provide the explicit expressions of some commonly used XC-functionals without any derivation.

A.9.1 *LDA: Perdew and Zunger (1981)*

In the local density approximation (LDA), the XC-functional and XC-potential have the following form [9]

$$E_{xc}^{LDA}[\rho] = \int d^3r \left[\rho \cdot \varepsilon_{xc}^{LDA}(\rho_\uparrow, \rho_\downarrow)\right], \tag{A.138}$$

$$V_{xc,\sigma}^{LDA}(\rho_\uparrow, \rho_\downarrow) \equiv \frac{\delta E_{xc}^{LDA}}{\delta \rho_\sigma} = \varepsilon_{xc}^{LDA}(\rho_\uparrow, \rho_\downarrow) + \rho \frac{\partial \varepsilon_{xc}^{LDA}}{\partial \rho_\sigma}, \tag{A.139}$$

where $\rho_\uparrow$ and $\rho_\downarrow$ are spin-↑ and spin-↓ charge density and $\rho = \rho_\uparrow + \rho_\downarrow$ is the total charge density. $\varepsilon_{xc}^{LDA}(\rho_\uparrow, \rho_\downarrow)$ has two terms, exchange term and correlation term,

$$\varepsilon_{xc}^{LDA}(\rho_\uparrow, \rho_\downarrow) = \varepsilon_{x}^{LDA}(\rho_\uparrow, \rho_\downarrow) + \varepsilon_{c}^{LDA}(\rho_\uparrow, \rho_\downarrow). \tag{A.140}$$

The expression of exchange term $\varepsilon_{x}^{LDA}(\rho_\uparrow, \rho_\downarrow)$ is

$$\varepsilon_{x}^{LDA}(\rho_\uparrow, \rho_\downarrow) = -\frac{3}{4\pi}\frac{1}{r}\left(\frac{9\pi}{4}\right)^{1/3} f_x(\xi), \tag{A.141}$$

where $r \equiv \left(\frac{4\pi}{3}\rho\right)^{-1/3}$, $\xi \equiv \frac{\rho_\uparrow - \rho_\downarrow}{\rho}$, and $f_x(\xi)$ is a spin-scaling factor

$$f_x(\xi) \equiv \frac{1}{2}\left[(1+\xi)^{4/3} + (1-\xi)^{4/3}\right].$$

The expression of correlation term $\varepsilon_{c}^{LDA}(\rho_\uparrow, \rho_\downarrow)$ is

$$\varepsilon_{c}^{LDA}(\rho_\uparrow, \rho_\downarrow) = u(r) + [p(r) - u(r)] f_c(\xi), \tag{A.142}$$

where $f_c(\xi)$ is a spin-scaling factor

$$f_c(\xi) \equiv \frac{1}{2^{4/3} - 2}\left[(1+\xi)^{4/3} + (1-\xi)^{4/3} - 2\right].$$

$u(r)$ and $p(r)$ are numerical functions

$$u(r) \equiv \begin{cases} \frac{\gamma_1}{1+\beta_1\sqrt{r}+\alpha_1 r} & r > 1 \\ a_1 \ln r + b_1 + c_1 r \ln r + d_1 r & r < 1 \end{cases};$$

$$p(r) \equiv \begin{cases} \frac{\gamma_2}{1+\beta_2\sqrt{r}+\alpha_2 r} & r > 1 \\ a_2 \ln r + b_2 + c_2 r \ln r + d_2 r & r < 1 \end{cases},$$

in which the coefficients are

α_1	β_1	γ_1	a_1	b_1	c_1	d_1
0.3334	1.0529	−0.1423	0.0311	−0.0480	0.0020	−0.0116
α_2	β_2	γ_2	a_2	b_2	c_2	d_2
0.2611	1.3981	−0.0843	0.01555	−0.0269	0.0007	−0.0048

www.ebook3000.com

A.9.2 *GGA: Perdew, Burke, and Ernzerhof (1996)*

In the generalized gradient approximation (GGA), the XC-functional and XC-potential have the following form [10]

$$E_{xc}^{GGA}[\rho] = \int d^3r \left[\rho \cdot \varepsilon_{xc}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow)\right], \quad \text{(A.143)}$$

$$V_{xc,\sigma}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow) \equiv \frac{\delta E_{xc}^{GGA}}{\delta\rho_\sigma} = \varepsilon_{xc}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow) + \rho\frac{\partial\varepsilon_{xc}^{GGA}}{\partial\rho_\sigma} - \nabla\cdot\left(\rho\frac{\partial\varepsilon_{xc}^{GGA}}{\partial\nabla\rho_\sigma}\right), \quad \text{(A.144)}$$

where $\rho_\uparrow$ and $\rho_\downarrow$ are spin-$\uparrow$ and spin-$\downarrow$ charge density and $\rho = \rho_\uparrow + \rho_\downarrow$ is the total charge density. $\varepsilon_{xc}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow)$ has two terms, exchange term and correlation term,

$$\varepsilon_{xc}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow) = \varepsilon_{x}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow) + \varepsilon_{c}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow). \quad \text{(A.145)}$$

The expression of exchange term $\varepsilon_{x}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow)$ is

$$\varepsilon_{x}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow) = \frac{1}{\rho}\sum_\sigma \rho_\sigma \varepsilon_x^0(2\rho_\sigma) F_x(2\rho_\sigma, 2|\nabla\rho_\sigma|), \quad \text{(A.146)}$$

where $\varepsilon_x^0(\rho_\sigma)$ is the exchange term of uniform electron gas and $F_x(\rho_\sigma, |\nabla\rho_\sigma|)$ is the factor due to charge density gradient. $\varepsilon_x^0(\rho_\sigma)$ is obtained as

$$\varepsilon_x^0(\rho_\sigma) = -\frac{3}{4\pi}\frac{1}{r_\sigma}\left(\frac{9\pi}{4}\right)^{1/3}, \quad \text{(A.147)}$$

where r_σ is defined by $r_\sigma \equiv \left(\frac{4\pi}{3}\rho_\sigma\right)^{-1/3}$. $F_x(\rho_\sigma, |\nabla\rho_\sigma|)$ is obtained as

$$F_x(\rho_\sigma, |\nabla\rho_\sigma|) = 1 + c_1 - \frac{c_1}{1 + \frac{c_2}{c_1}u_\sigma^2}, \quad \text{(A.148)}$$

where $c_1 = 0.804$ and $c_2 = \frac{\pi^2}{3} \times 0.066725$. $u_\sigma \equiv \frac{|\nabla\rho_\sigma|}{2k_\sigma\rho_\sigma}$ is a dimensionless charge density gradient with $k_\sigma = \left(3\pi^2\rho_\sigma\right)^{1/3}$.

The expression of correlation term $\varepsilon_{c}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow)$ is

$$\varepsilon_{c}^{GGA}(\rho_\uparrow, \rho_\downarrow, \nabla\rho_\uparrow, \nabla\rho_\downarrow) = \varepsilon_c^0(\rho_\uparrow, \rho_\downarrow) + \Delta_c(\rho_\uparrow, \rho_\downarrow, |\nabla\rho|), \quad \text{(A.149)}$$

where $\varepsilon_c^0(\rho_\uparrow, \rho_\downarrow)$ is the correlation term of uniform electron gas, and $\Delta_c(\rho_\uparrow, \rho_\downarrow, |\nabla\rho|)$ is the correction due to charge density gradient. $\varepsilon_c^0(\rho_\uparrow, \rho_\downarrow)$ is obtained as [11]

$$\varepsilon_c^0(\rho_\uparrow, \rho_\downarrow) = Y_1(r) - Y_3(r)\frac{f_c(\xi)}{f_c''(0)}\left(1-\xi^4\right) + \left[Y_2(r) - Y_1(r)\right] f_c(\xi)\,\xi^4, \quad \text{(A.150)}$$

where $r \equiv \left(\frac{4\pi}{3}\rho\right)^{-1/3}$, $\xi \equiv \frac{\rho_\uparrow - \rho_\downarrow}{\rho}$, and $f_c(\xi)$ is a spin-scaling factor

$$f_c(\xi) \equiv \frac{1}{2^{4/3}-2}\left[(1+\xi)^{4/3} + (1-\xi)^{4/3} - 2\right].$$

$Y_1(r)$, $Y_2(r)$, $Y_3(r)$ are numerical functions

$$Y_i(r) = -2c_{0i}(1+\lambda_{0i}r)\ln\left[1 + \frac{1}{2c_{0i}\left(\lambda_{1i}r^{1/2} + \lambda_{2i}r + \lambda_{3i}r^{3/2} + \lambda_{4i}r^2\right)}\right],$$

in which the coefficients are

	$i=1$	$i=2$	$i=3$
c_{0i}	0.031091	0.015545	0.016887
λ_{0i}	0.21370	0.20548	0.11125
λ_{1i}	7.5957	14.1189	10.357
λ_{2i}	3.5876	6.1977	3.6231
λ_{3i}	1.6382	3.3662	0.88026
λ_{4i}	0.49294	0.62517	0.49671

$\Delta_c(\rho_\uparrow, \rho_\downarrow, |\nabla\rho|)$ is obtained as

$$\Delta_c(\rho_\uparrow, \rho_\downarrow, |\nabla\rho|) = c_4\phi^3 \ln\left[1 + \frac{c_3}{c_4}v^2\frac{1+Av^2}{1+Av^2+A^2v^4}\right], \tag{A.151}$$

where $c_3 = 0.066725$ and $c_4 = (1-\ln 2)/\pi^2$. ϕ is a spin-scaling factor

$$\phi(\xi) \equiv \frac{1}{2}\left[(1+\xi)^{2/3} + (1-\xi)^{2/3}\right],$$

and A is defined by

$$A = \frac{c_3}{c_4}\frac{1}{\exp\left[-\frac{\varepsilon_c^0(\rho_\uparrow,\rho_\downarrow)}{c_4\phi^3}\right] - 1}.$$

$v \equiv \frac{|\nabla\rho|}{2\phi k_s \rho}$ is a dimensionless density gradient with $k_s = \sqrt{\frac{4}{\pi}(3\pi^2\rho)^{1/3}}$.

A.9.3 *MBJ: Tran and Blaha (2009)*

LDA and GGA can achieve moderate accuracy with low computational cost, and hence are widely used in solid-state physics. However, both LDA and GGA seriously underestimate the band gap of semiconductors and insulators, which may result in uncontrollable error in electronic device simulations. Hybrid XC-functionals or GW method may produce very accurate band gap. However, the computational cost is much higher than LDA or

www.ebook3000.com

GGA, and hence is not practical for device simulations. As a compromise, the modified Becke and Johnson (MBJ) XC-potential can achieve similar accuracy as hybrid functionals or GW method with similar computational cost as LDA or GGA.

The MBJ XC-potential is constructed as follows [12]

$$V_{xc,\sigma}^{MBJ}\left(\rho_{\uparrow},\rho_{\downarrow},t_{\sigma}\right)=V_{x,\sigma}^{MBJ}\left(\rho_{\sigma},t_{\sigma}\right)+V_{c,\sigma}^{0}\left(\rho_{\uparrow},\rho_{\downarrow}\right), \tag{A.152}$$

where $\rho_{\sigma}\equiv\sum_{i}n_{i\sigma}\psi_{i\sigma}^{*}\psi_{i\sigma}$ and $t_{\sigma}\equiv\frac{1}{2}\sum_{i}n_{i\sigma}\nabla\psi_{i\sigma}^{*}\cdot\nabla\psi_{i\sigma}$ are the charge density and the kinetic energy density of spin-σ electrons. $V_{c,\sigma}^{0}\left(\rho_{\uparrow},\rho_{\downarrow}\right)$ is the LDA correction potential

$$V_{c,\sigma}^{0}\left(\rho_{\uparrow},\rho_{\downarrow}\right)=\varepsilon_{c}^{0}\left(\rho_{\uparrow},\rho_{\downarrow}\right)+\rho\frac{\partial\varepsilon_{c}^{0}}{\partial\rho_{\sigma}}, \tag{A.153}$$

where $\varepsilon_{c}^{0}\left(\rho_{\uparrow},\rho_{\downarrow}\right)$ has been defined in Eq. (A.150). $V_{x}^{MBJ}\left(\rho_{\sigma},t_{\sigma}\right)$ is constructed as

$$V_{x,\sigma}^{MBJ}\left(\rho_{\sigma},t_{\sigma}\right)=cV_{x,\sigma}^{BR}\left(\rho_{\sigma},t_{\sigma}\right)+(3c-2)\frac{1}{\pi}\sqrt{\frac{5}{12}}\sqrt{\frac{2t_{\sigma}}{\rho_{\sigma}}}, \tag{A.154}$$

where $V_{x,\sigma}^{BR}\left(\rho_{\sigma},t_{\sigma}\right)$ is the Becke–Roussel potential [13]

$$V_{x,\sigma}^{BR}\left(\rho_{\sigma},t_{\sigma}\right)=-\frac{1}{b_{\sigma}}\left[1-\left(1+\frac{1}{2}x_{\sigma}\right)e^{-x_{\sigma}}\right]. \tag{A.155}$$

Here x_{σ} and b_{σ} are defined by

$$b_{\sigma}=\frac{x_{\sigma}e^{-x_{\sigma}/3}}{\left(8\pi\rho_{\sigma}\right)^{1/3}},$$
$$\frac{x_{\sigma}e^{-2x_{\sigma}/3}}{x_{\sigma}-2}=\frac{2}{3}\pi^{2/3}\frac{\rho_{\sigma}^{5/3}}{Q_{\sigma}},$$

in which $Q_{\sigma}=\frac{1}{6}\left(\nabla^{2}\rho_{\sigma}-2\gamma D_{\sigma}\right)$ with $D_{\sigma}=2t_{\sigma}-\frac{1}{4}\frac{\left(\nabla\rho_{\sigma}\right)^{2}}{\rho_{\sigma}}$ and $\gamma=0.8$.

In Eq. (A.154), c is a material dependent parameter within the range $(1,2)$. Although c can be calculated from an empirical formula [12], it is more convenient to optimize c directly in the LMTO method. For a given semiconductor or insulator, it is observed that the band gap increases smoothly and monotonically with c, see Fig. A.10. One can always find an optimal c to fit the experimental band gap. For systems having different materials, one can use different optimal c values in different atomic spheres.

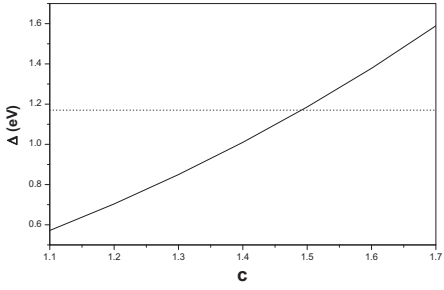


Fig. A.10 Si band gap Δ as a function of c. The horizontal dotted line indicates the experimental data $\Delta_0 = 1.17eV$, resulting in an optimal $c_0 = 1.49$.

A.9.4 *A few comments*

Here are a few comments to address some technical details. (1) In the LMTO method, the charge density ρ_σ and the kinetic energy density t_σ can be calculated with the formulas given in Section 3.11.9. (2) In heavy atoms, core electrons move very fast and hence relativistic effects may not be negligible. The relativistic correction of XC-functional have been derived in Ref. [14]. (3) Implementation of various XC-functionals is available in the XC-library written in C [15].

A.10 Complex-valued and real-valued spherical harmonics

There are two types of spherical harmonics: the complex-valued spherical harmonics and the real-valued spherical harmonics. The former ones are more useful in the theoretical derivation while the latter ones are more convenient in the numerical implementation. The two types of spherical harmonics are related by a unitary transformation.

The complex-valued spherical Harmonics are defined by

$$\tilde{Y}_{lm}(\theta,\phi) \equiv \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta)\, e^{im\phi}, \tag{A.156}$$

where $l \geqslant 0$ and $m = -l, -l+1, \ldots, l$. $P_l^m(x)$ is the associated Legendre polynomial

$$P_l^m(x) \equiv \begin{cases} (-1)^m \left(1-x^2\right)^{\frac{m}{2}} \frac{d^m}{dx^m} P_l(x) & m > 0 \\ P_l(x) & m = 0 \\ (-1)^m \frac{(l+m)!}{(l-m)!} P_l^{|m|}(x) & m < 0 \end{cases},$$

www.ebook3000.com

in which $P_l(x)$ is the Legendre polynomial

$$P_l(x) \equiv \frac{1}{2^l l!}\left[\frac{d^l}{dx^l}\left(x^2-1\right)^l\right].$$

The real-valued spherical harmonics are defined by the following unitary transform of the complex-valued spherical Harmonics

$$Y_{lm}(\theta,\phi) = \begin{cases} \frac{(-1)^m}{\sqrt{2}}\left[\tilde{Y}_{lm}(\theta,\phi) + (-1)^m \tilde{Y}_{l,-m}(\theta,\phi)\right] & m>0 \\ \tilde{Y}_{lm}(\theta,\phi) & m=0 \\ \frac{\mathrm{i}}{\sqrt{2}}\left[\tilde{Y}_{lm}(\theta,\phi) - (-1)^m \tilde{Y}_{l,-m}(\theta,\phi)\right] & m<0 \end{cases}. \quad (8.158)$$

By using the property

$$\tilde{Y}^*_{lm}(\theta,\phi) = (-)^m \tilde{Y}_{l,-m}(\theta,\phi),$$

Eq. (8.158) is reduced to an equivalent form

$$Y_{lm}(\theta,\phi) = \begin{cases} (-1)^m \sqrt{2}\mathrm{Re}\tilde{Y}_{lm}(\theta,\phi) & m>0 \\ \tilde{Y}_{lm}(\theta,\phi) & m=0 \\ -\sqrt{2}\mathrm{Im}\tilde{Y}_{lm}(\theta,\phi) & m<0 \end{cases}, \quad (8.159)$$

indicating that $Y_{lm}(\theta,\phi)$ always has real value.

For $l=0,1,2$, the expressions of $\tilde{Y}_{lm}$ and Y_{lm} are listed in the following table

$\tilde{Y}_{0,0} = \frac{1}{\sqrt{4\pi}}$	$\tilde{Y}_{1,+1} = -\sqrt{\frac{3}{8\pi}}\sin\theta e^{+\mathrm{i}\phi}$	$\tilde{Y}_{2,+2} = \sqrt{\frac{15}{32\pi}}\sin^2\theta e^{+\mathrm{i}2\phi}$
$\tilde{Y}_{1,-1} = \sqrt{\frac{3}{8\pi}}\sin\theta e^{-\mathrm{i}\phi}$	$\tilde{Y}_{1,0} = \sqrt{\frac{3}{4\pi}}\cos\theta$	$\tilde{Y}_{2,+1} = -\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta e^{+\mathrm{i}\phi}$
$\tilde{Y}_{2,-2} = \sqrt{\frac{15}{32\pi}}\sin^2\theta e^{-\mathrm{i}2\phi}$	$\tilde{Y}_{2,-1} = \sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta e^{-\mathrm{i}\phi}$	$\tilde{Y}_{2,0} = \sqrt{\frac{5}{16\pi}}\left(3\cos^2\theta-1\right)$

(8.160)

$Y_{0,0} = \frac{1}{\sqrt{4\pi}}$	$Y_{1,+1} = \sqrt{\frac{3}{4\pi}}\frac{x}{r}$	$Y_{2,+2} = \sqrt{\frac{15}{16\pi}}\frac{x^2-y^2}{r^2}$
$Y_{1,-1} = \sqrt{\frac{3}{4\pi}}\frac{y}{r}$	$Y_{1,0} = \sqrt{\frac{3}{4\pi}}\frac{z}{r}$	$Y_{2,+1} = \sqrt{\frac{15}{4\pi}}\frac{zx}{r^2}$
$Y_{2,-2} = \sqrt{\frac{15}{4\pi}}\frac{xy}{r^2}$	$Y_{2,-1} = \sqrt{\frac{15}{4\pi}}\frac{yz}{r^2}$	$Y_{2,0} = \sqrt{\frac{5}{16\pi}}\frac{3z^2-r^2}{r^2}$

(8.161)

Here the Cartesian coordinates (x,y,z) and the spherical coordinates (r,θ,ϕ) are related by

$$x = r\sin\theta\cos\phi,$$
$$y = r\sin\theta\sin\phi,$$
$$z = r\cos\theta,$$

with $r>0$, $\theta\in(0,\pi)$, and $\phi\in(0,2\pi)$. For $l\geqslant 3$, the expressions of $\tilde{Y}_{lm}$ and Y_{lm} are available in the website [16].

A.11 Gaunt coefficients

This section discusses the algorithm for calculating the Gaunt coefficient of complex-valued and real-valued spherical harmonics. Complex-valued and real-valued spherical harmonics have been defined in Appendix A.10.

The Gaunt coefficient of complex-valued spherical Harmonics is defined by

$$\tilde{C}_{L_1L_2L_3} \equiv \int d\Omega \tilde{Y}_{L_1}(\Omega)\, \tilde{Y}_{L_2}(\Omega)\, \tilde{Y}_{L_3}(\Omega), \tag{A.161}$$

where $\tilde{Y}_{L_1}(\Omega)$, $\tilde{Y}_{L_2}(\Omega)$, $\tilde{Y}_{L_3}(\Omega)$ are complex-valued spherical harmonics. $\tilde{C}_{L_1L_2L_3}$ can be expressed in terms of Wigner 3j-symbol [17]

$$\tilde{C}_{L_1L_2L_3} = \sqrt{\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}} \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} l_1 & l_2 & l_3 \\ 0 & 0 & 0 \end{pmatrix}, \tag{A.162}$$

and hence the task is reduced to calculating Wigner 3j-symbol.

Wigner 3j-symbol can be evaluated with the Racah formula [17].

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{j_1-j_2-m_3} F_1F_2F_3, \tag{A.163}$$

where F_1, F_2, F_3 are defined by

$$F_1 = \left[\frac{(j_1+j_2-j_3)!\,(j_2+j_3-j_1)!\,(j_3+j_1-j_2)!}{(j_1+j_2+j_3+1)!}\right]^{\frac{1}{2}},$$

$$F_2 = \left[(j_1+m_1)!\,(j_1-m_1)!\,(j_2+m_2)!\,(j_2-m_2)!\,(j_3+m_{31})!\,(j_3-m_3)!\right]^{\frac{1}{2}},$$

$$F_3 = \sum_t \frac{(-1)^t}{(t-t_1)!\,(t-t_2)!\,(t-t_3)!(t_4-t)!\,(t_5-t)!\,(t_6-t)!}. \tag{A.164}$$

In the expression of F_3, $t_1 = 0$, $t_2 = -(j_3 - j_2 + m_1)$, $t_3 = -(j_3 - j_1 - m_2)$, $t_4 = j_1 + j_2 - j_3$, $t_5 = j_1 - m_1$, $t_6 = j_2 + m_2$, and the summation over t is in the range

$$\max(t_1, t_2, t_3) \leqslant t \leqslant \min(t_4, t_5, t_6).$$

Wigner 3j-symbol is nonzero only if $|j_1 - j_2| \leqslant j_3 \leqslant j_1 + j_2$, $|m_1| \leqslant j_1$, $|m_2| \leqslant j_2$, $|m_3| \leqslant j_3$, and $m_1 + m_2 + m_3 = 0$, which are referred to as the selection rules. In the special case $l_3 = l_1 + l_2$, the calculation of Wigner

www.ebook3000.com

3j-symbol can be further simplified as [17]

$$\begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$
$$= (-1)^{l_1-l_2-m_3} \left[\frac{(2l_1)!\,(2l_2)!}{(2l_1+2l_2+1)!} \frac{(l_3+m_3)!(l_3-m_3)!}{(l_1+m_1)!\,(l_1-m_1)!\,(l_2+m_2)!\,(l_2-m_2)!} \right],$$

where $l_3 = l_1 + l_2$ and $m_3 = -(m_1 + m_2)$.

The Gaunt coefficient of real-valued spherical harmonics is defined by

$$C_{L_1L_2L_3} \equiv \int d\Omega Y_{L_1}(\Omega)\, Y_{L_2}(\Omega)\, Y_{L_3}(\Omega), \tag{A.165}$$

where $Y_{L_1}(\Omega)$, $Y_{L_2}(\Omega)$, $Y_{L_3}(\Omega)$ are real-valued spherical harmonics. By using Eq. (A.157), $Y_{lm}(\Omega)$ can be expressed as a linear combination of $\tilde{Y}_{lm}(\Omega)$ and $\tilde{Y}_{l,-m}(\Omega)$

$$Y_{lm}(\Omega) = \alpha_m^+ \tilde{Y}_{lm}(\Omega) + \alpha_m^- \tilde{Y}_{l,-m}(\Omega),$$

where the coefficients $\alpha_m^{\pm}$ are

$$\alpha_m^+ = \begin{cases} \frac{(-1)^m}{\sqrt{2}} & m > 0 \\ 1 & m = 0 \\ \frac{1}{\sqrt{2}}\mathrm{i} & m < 0 \end{cases},$$
$$\alpha_m^- = \begin{cases} \frac{1}{\sqrt{2}} & m > 0 \\ 0 & m = 0 \\ -\frac{(-1)^m}{\sqrt{2}}\mathrm{i} & m < 0 \end{cases}.$$

It is straightforward to evaluate $C_{L_1L_2L_3}$ by using the linear combination of $\tilde{C}_{L_1'L_2'L_3'}$

$$C_{L_1L_2L_3} = \sum_{s_1=\pm} \sum_{s_2=\pm} \sum_{s_3=\pm} \alpha_{m_1}^{s_1} \alpha_{m_2}^{s_2} \alpha_{m_3}^{s_3} \tilde{C}_{L_1'L_2'L_3'}, \tag{A.166}$$

where $L_1' \equiv (l_1, s_1 m_1)$, $L_2' \equiv (l_2, s_2 m_2)$, $L_3' \equiv (l_3, s_3 m_3)$.

To sum up, the Gaunt coefficient of complex-valued spherical harmonics can be calculated with Eqs. (A.162,A.163,A.164), and the Gaunt coefficient of real-valued spherical harmonics can be calculated with Eq. (A.166).

A.12 Eigensolutions of TST and TSC matrices

In this section, we present the exact eigensolutions of Toeplitz symmetric cyclic (TSC) and Toeplitz symmetric tridiagonal (TST) matrices.

TSC matrix has the form

$$H \equiv \begin{pmatrix} a & b & & & & b \\ b & a & b & & & \\ & b & a & \ddots & & \\ & & \ddots & \ddots & b & \\ & & & b & a & b \\ b & & & & b & a \end{pmatrix}_{N\times N}, \tag{A.167}$$

where a and b are real numbers. The normalized eigensolutions of Eq. (A.167) are

$$H\Psi = \Psi\Lambda,$$

where Ψ and Λ are obtained as

$$\begin{aligned} \Psi_{jk} &= \frac{1}{\sqrt{N}} \exp\left(\mathrm{i}\frac{2\pi}{N} jk\right), \\ \Lambda_{jk} &= \delta_{jk}\left(a + 2b\cos\frac{2\pi}{N}k\right), \end{aligned} \tag{A.168}$$

in which $j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, N$. The derivation is based on the fact that the Hamiltonian Eq. (A.167) can be mapped to a 1d tight-binding circle with on-site energy a and nearest neighbor coupling b. Due to the rotational symmetry, the eigensolutions can be obtained by using discrete Fourier transform.

TST matrix has the form

$$H \equiv \begin{pmatrix} a & b & & & & \\ b & a & b & & & \\ & b & a & \ddots & & \\ & & \ddots & \ddots & b & \\ & & & b & a & b \\ & & & & b & a \end{pmatrix}_{N\times N}, \tag{A.169}$$

where a and b are real numbers. The normalized eigensolutions of Eq. (A.167) are

$$H\Psi = \Psi\Lambda,$$

where Ψ and Λ are obtained as

$$\begin{aligned} \Psi_{jk} &= \sqrt{\frac{2}{N+1}} \sin\frac{jk\pi}{N+1}, \\ \Lambda_{jk} &= \delta_{jk}\left(a + 2b\cos\frac{k\pi}{N+1}\right), \end{aligned} \tag{A.170}$$

www.ebook3000.com

in which $j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, N$. The derivation is based on the fact that the Hamiltonian Eq. (A.169) can be mapped to a 1d tight-binding chain with on-site energy a and nearest neighbor coupling b. The eigensolutions are similar to the bound states in a flat potential well, and the detailed derivation can be found in page 130 of Ref. [18].

A.13 Proof of the Wronskian identity

This section proves the Wronskian identity Eq. (3.31). The proof proceeds in three steps.

The first step is to prove a lemma for the Wronskian of φ and $\dot{\varphi}$.

Lemma: Assume that φ satisfies the radial equation (see Section 3.2)

$$\left[-\frac{1}{2}\partial_r^2 + \frac{1}{r}\partial_r + \frac{l\left(l+1\right)}{2r^2} + V_{eff}\left(r\right) - E\right]\varphi\left(r, E\right) = 0, \tag{A.171}$$

and φ is normalized to $\int_0^R \varphi^2\left(r, E\right) r^2 dr = 1$. The Wronskian of φ and $\dot{\varphi}$ at $r = R$ is a constant

$$\left\{\varphi\left(r, E\right), \dot{\varphi}\left(r, E\right)\right\}_{r=R} = -2. \tag{A.172}$$

Proof: Define $\chi\left(r, E\right) \equiv r\varphi\left(r, E\right)$, Eq. (A.171) is reduced to

$$\left[-\frac{1}{2}\partial_r^2 + \frac{l\left(l+1\right)}{2r^2} + V_{eff}\left(r\right) - E\right]\chi\left(r, E\right) = 0, \tag{A.173}$$

where $\int_0^R \chi^2\left(r, E\right) dr = 1$. Differentiating Eq. (A.173) with respect to E, one obtains the differential equation of $\dot{\chi}$

$$\left[-\frac{1}{2}\partial_r^2 + \frac{l\left(l+1\right)}{2r^2} + V_{eff}\left(r\right) - E\right]\dot{\chi}\left(r, E\right) = \chi\left(r, E\right). \tag{A.174}$$

Multiplying Eq. (A.174) with χ and Eq. (A.173) with $\dot{\chi}$, and subtracting the two derived equations, one obtains

$$-\frac{1}{2}\int_0^R \left(\chi\dot{\chi}'' - \dot{\chi}\chi''\right) dr = \int_0^R \chi^2 dr.$$

The LHS can be evaluated analytically as $-\frac{1}{2}\left(\chi\dot{\chi}' - \dot{\chi}\chi'\right)_{r=R}$. The RHS is 1 due to the normalization of χ. Comparing the LHS and the RHS, one obtains

$$\left(\chi\dot{\chi}' - \dot{\chi}\chi'\right)_{r=R} = -2,$$

resulting in the Wronskian $\left\{\varphi, \dot{\varphi}\right\}_{r=R} = -2$. Two comments are in order: (1) The normalization of φ is essential to derive Eq. (A.172). (2) It is

inferred that

$$(\chi\dot{\chi}' - \dot{\chi}\chi')_{r=R} = -2\int_0^R \chi^2 dr < 0, \tag{A.175}$$

regardless of whether χ is normalized or not.

The second step is to check the identity

$$\{f_1, f_2\}\{f_3, f_4\} = \{f_1, f_3\}\{f_2, f_4\} - \{f_1, f_4\}\{f_2, f_3\}, \tag{A.176}$$

which is straightforward by using the Wronskian's definition.

The third step is to apply Eq. (A.176) to Eq. (3.31) and obtain

$$\begin{aligned}\{J_l, \phi_{il}\}\left\{K_l, \dot{\phi}_{il}\right\} - \left\{J_l, \dot{\phi}_{il}\right\}\{K_l, \phi_{il}\} &= \{J_l, K_l\}\left\{\phi_{il}, \dot{\phi}_{il}\right\} \\ &= \left(-\frac{\omega}{2}\right)(-2) = \omega,\end{aligned}$$

where Eq. (3.14) and Eq. (A.172) are used in the derivation. QED.

A.14 Numerical proof

In Section 3.6, we have proved the equivalence between Eq. (3.35) and Eq. (3.39) by some lengthy algebra. The proof there is rigorous in the eyes of mathematicians. In this section, we attempt to present a "numerical proof" which is in the style of physicists. The key idea is to do many numerical tests to check the equivalence of the two equations. The probability of violating the equivalence after so many positive results is further estimated. If the probability is so small that it is unlikely to happen within the lifetime of the universe, we can safely discard it and draw the conclusion.

The numerical tests proceed as follows. Let E be a random complex number, C, $\sqrt{\Delta}$, γ random real diagonal matrices, and S a random real full matrix. By generating many random arguments, $G^r(E)$ is calculated with both Eq. (3.35) and Eq. (3.39):

Numerical proof:

```
clc
clear

dim = 20;
N = 100;

rng(0)
epsilon = 0;
DeltaList = zeros(1, N);
for ii = 1:N
    E = 2 * (rand(1) + i * rand(1)) - 1;
    S = rand(dim, dim);
    ccccc = rand(1, dim);
```

www.ebook3000.com

```
    delta = rand(1, dim);
    gamma = rand(1, dim);
    sqrtd = sqrt(delta);
    D = delta + gamma .* (E - ccccc);
    lambda = diag(gamma ./ D);
    mu = diag(sqrtd ./ D);
    P = diag((E - ccccc) ./ D);
    Horth = diag(ccccc) + diag(sqrtd) * S * inv(eye(dim) - diag(gamma)*S) * diag(sqrtd);
    G1 = inv(E * eye(dim) - Horth);
    G2 = lambda + mu * inv(P - S) * mu;
    Delta0 = (trace(G1) + trace(G2)) / 2;
    epsilon0 = max(max(abs(G1 - G2)));
    DeltaList(ii) = Delta0;
    epsilon = max(epsilon, epsilon0);
end %ii

Delta = std(DeltaList);
log10_P = N * (log10(epsilon) - log10(Delta));
fprintf('Delta = %g espilon = %g \n', Delta, epsilon)
fprintf('The probability of equality violation is less than 10^%d \n', round(log10_P))
```

As expected the values of $G^r(E)$ calculated with the two equations are always equal within a tiny numerical error ε in all N tests. Meanwhile the values of $G^r(E)$ vary in the range Δ due to the random arguments. Suppose the two equations are unrelated and just happen to be equivalent in all tests, the lucky chance P is estimated as $\left(\frac{\varepsilon}{\Delta}\right)^N$. In our numerical tests, $\varepsilon = 5 \times 10^{-11}$, $\Delta = 15$, $N = 100$, and the upper limit of the probability is $P \leqslant 10^{-1145}$.

Is the probability sufficiently small? We need to do real experiments to verify. The shortest time required to carry out one experiment is estimated by the Plank time $\sqrt{\frac{\hbar G}{c^5}}$which is about $10^{-43}s$. The longest time to carry out all experiments is estimated by the lifetime of the universe which is about $10^{17}s$. The upper limit $P \leqslant 10^{-1145}$ means that the chance of violating the equivalence is no more than $10^{-1085} \ll 1$ which will never happen in reality. Therefore Eq. (3.35) and Eq. (3.39) are not equivalent by chance but equivalent intrinsically. This concludes the numerical proof.

A.15 Transmission coefficient in the LMTO method

In this section, we derive the transmission coefficient formula in terms of auxiliary Green's functions. By definition the transmission coefficient should be calculated by physical Green's functions

$$T = \mathrm{Tr}\ G^r_{CC}\Gamma^L_{CC}G^a_{CC}\Gamma^R_{CC}, \tag{A.177}$$

where G^r_{CC} and G^a_{CC} are the physical Green's functions of the central region, and Γ^L_{CC} and Γ^R_{CC} are the physical linewidth functions of the left and

right leads. Below we shall prove that Eq. (A.177) is still valid if the physical Green's functions and the physical linewidth functions are replaced by the auxiliary counterparts, namely

$$T = \mathrm{Tr}\ \tilde{G}^r_{CC}\tilde{\Gamma}^L_{CC}\tilde{G}^a_{CC}\tilde{\Gamma}^R_{CC}. \tag{A.178}$$

Note that the notation of auxiliary quantities is different from those of Chapter 3.

The proof will proceed in two steps: (1) Derive an alternative formula for the transmission coefficient; (2) Apply the new formula to both Eq. (A.177) and Eq. (A.178).

Step 1: Derive an alternative formula for the transmission coefficient. Notice that both physical and auxiliary Green's function can be written in a unified form $G^{r,a} = (h^{r,a})^{-1}$ where

$$h^{r,a} = E^{\pm} - H_{orth} \tag{A.179}$$

for the physical Green's function (see Eq. (3.35)) and

$$h^{r,a} = P\left(E^{\pm}\right) - S \tag{A.180}$$

for the auxiliary Green's function (see Eq. (3.37)). Assume that the central region is sufficiently long so that the left and right leads do not interact directly, i.e., $h^{r,a}_{LR} = h^{r,a}_{RL} = 0$. The transmission coefficient formula Eq. (A.177) or Eq. (A.178) can be rewritten in an equivalent form

$$\begin{aligned}&\mathrm{Tr}\ G^r_{CC}\Gamma^L_{CC}G^a_{CC}\Gamma^R_{CC}\\ &\quad = \mathrm{Tr}\ G^r_{RL}\mathrm{i}\left[(g^r_{LL})^{-1} - (g^a_{LL})^{-1}\right]G^a_{LR}\mathrm{i}\left[(g^r_{RR})^{-1} - (g^a_{RR})^{-1}\right],\end{aligned} \tag{A.181}$$

where $g^{r,a}_{\beta\beta} = \left(h^{r,a}_{\beta\beta}\right)^{-1}$ is the Green's function of lead β ($\beta = L, R$).

Proof: The plan is to replace the index C by the indices L and R in Eq. (A.177). Firstly we work on the Green's function $G^{r,a}_{CC}$. By definition

$$\begin{pmatrix} h^{r,a}_{LL} & h^{r,a}_{LC} & 0 \\ h^{r,a}_{CL} & h^{r,a}_{CC} & h^{r,a}_{CR} \\ 0 & h^{r,a}_{RC} & h^{r,a}_{RR} \end{pmatrix} \begin{pmatrix} G^{r,a}_{LL} & G^{r,a}_{LC} & G^{r,a}_{LR} \\ G^{r,a}_{CL} & G^{r,a}_{CC} & G^{r,a}_{CR} \\ G^{r,a}_{RL} & G^{r,a}_{RC} & G^{r,a}_{RR} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

one obtains

$$\begin{aligned} h^{r,a}_{RC}G^{r,a}_{CL} + h^{r,a}_{RR}G^{r,a}_{RL} &= 0, \\ G^{r,a}_{CC}h^{r,a}_{CL} + G^{r,a}_{CL}h^{r,a}_{LL} &= 0. \end{aligned} \tag{A.182}$$

Eliminating $G^{r,a}_{CL}$ in Eq. (A.182), one derives

$$h^{r,a}_{RC}G^{r,a}_{CC}h^{r,a}_{CL} = h^{r,a}_{RR}G^{r,a}_{RL}h^{r,a}_{LL}, \tag{A.183}$$

www.ebook3000.com

where the CC indices are replaced by RL. Similarly one derives

$$h^{r,a}_{LC}G^{r,a}_{CC}h^{r,a}_{CR} = h^{r,a}_{LL}G^{r,a}_{LR}h^{r,a}_{RR}, \tag{A.184}$$

where the CC indices are replaced by LR.

Secondly we work on the linewidth function Γ^{β}_{CC}. Notice that $E^{\pm}$ does not appear in the off-diagonal elements of $h^{r,a}$ and hence $h^{r}_{C\beta} = h^{a}_{C\beta}$ and $h^{r}_{\beta C} = h^{a}_{\beta C}$. By definition

$$\Gamma^{\beta}_{CC} = \mathrm{i}\left(h^{r}_{C\beta}g^{r}_{\beta\beta}h^{r}_{\beta C} - h^{a}_{C\beta}g^{a}_{\beta\beta}h^{a}_{\beta C}\right) = \mathrm{i}h^{r}_{C\beta}\left(g^{r}_{\beta\beta} - g^{a}_{\beta\beta}\right)h^{a}_{\beta C}, \tag{A.185}$$

where the indices CC are replaced by $\beta\beta$.

Substituting Eq. (A.185) and Eqs. (A.183,A.184) into the LHS of Eq. (A.181), one derives

$$\begin{aligned}
LHS &= \mathrm{Tr}\ G^{r}_{CC}\Gamma^{L}_{CC}G^{a}_{CC}\Gamma^{R}_{CC} \\
&= -\mathrm{Tr}\ G^{r}_{CC}\times h^{r}_{CL}\left(g^{r}_{LL} - g^{a}_{LL}\right)h^{a}_{LC}\times G^{a}_{CC}\times h^{a}_{CR}\left(g^{r}_{RR} - g^{a}_{RR}\right)h^{r}_{RC} \\
&= -\mathrm{Tr}\ \left(h^{r}_{RC}G^{r}_{CC}h^{r}_{CL}\right)\left(g^{r}_{LL} - g^{a}_{LL}\right)\left(h^{a}_{LC}G^{a}_{CC}h^{a}_{CR}\right)\left(g^{r}_{RR} - g^{a}_{RR}\right) \\
&= -\mathrm{Tr}\ \left(h^{r}_{RR}G^{r}_{RL}h^{r}_{LL}\right)\left(g^{r}_{LL} - g^{a}_{LL}\right)\left(h^{a}_{LL}G^{a}_{LR}h^{a}_{RR}\right)\left(g^{r}_{RR} - g^{a}_{RR}\right) \\
&= -\mathrm{Tr}\ G^{r}_{RL}\times h^{r}_{LL}\left(g^{r}_{LL} - g^{a}_{LL}\right)h^{a}_{LL}\times G^{a}_{LR}\times h^{a}_{RR}\left(g^{r}_{RR} - g^{a}_{RR}\right)h^{r}_{RR} \\
&= -\mathrm{Tr}\ G^{r}_{RL}\left[\left(g^{r}_{LL}\right)^{-1} - \left(g^{a}_{LL}\right)^{-1}\right]G^{a}_{LR}\left[\left(g^{r}_{RR}\right)^{-1} - \left(g^{a}_{RR}\right)^{-1}\right] \\
&= RHS,
\end{aligned}$$

where $h^{r}_{\beta\beta}\left(g^{r}_{\beta\beta} - g^{a}_{\beta\beta}\right)h^{a}_{\beta\beta} = \left(g^{a}_{\beta\beta}\right)^{-1} - \left(g^{r}_{\beta\beta}\right)^{-1}$ is used in the derivation.

Step 2: Apply the new transmission formula Eq. (A.181) to both Eq. (A.177) and Eq. (A.178). The problem is reduced to proving

$$\begin{aligned}
&\mathrm{Tr}\ G^{r}_{RL}\mathrm{i}\left[\left(g^{r}_{LL}\right)^{-1} - \left(g^{a}_{LL}\right)^{-1}\right]G^{a}_{LR}\mathrm{i}\left[\left(g^{r}_{RR}\right)^{-1} - \left(g^{a}_{RR}\right)^{-1}\right] \\
&= \mathrm{Tr}\ \tilde{G}^{r}_{RL}\mathrm{i}\left[\left(\tilde{g}^{r}_{LL}\right)^{-1} - \left(\tilde{g}^{a}_{LL}\right)^{-1}\right]\tilde{G}^{a}_{LR}\mathrm{i}\left[\left(\tilde{g}^{r}_{RR}\right)^{-1} - \left(\tilde{g}^{a}_{RR}\right)^{-1}\right].
\end{aligned} \tag{A.186}$$

By using Eq. (A.179) and Eq. (A.180), one obtains

$$\begin{aligned}
\left(g^{r}_{\beta\beta}\right)^{-1} - \left(g^{a}_{\beta\beta}\right)^{-1} &= E^{+} - E^{-}, \\
\left(\tilde{g}^{r}_{\beta\beta}\right)^{-1} - \left(\tilde{g}^{a}_{\beta\beta}\right)^{-1} &= P_{\beta}\left(E^{+}\right) - P_{\beta}\left(E^{-}\right);
\end{aligned} \tag{A.187}$$

By using Eq. (3.39), one obtains

$$\begin{aligned}
G^{r}_{RL} &= \mu^{+}_{R}\tilde{G}^{r}_{RL}\mu^{+}_{L}, \\
G^{a}_{LR} &= \mu^{-}_{L}\tilde{G}^{a}_{LR}\mu^{-}_{R}.
\end{aligned} \tag{A.188}$$

Inserting Eq. (A.187) and Eq. (A.188) into Eq. (A.186), the problem is reduced to proving

$$\left(E^{+}-E^{-}\right)\mu_{\beta}^{+}\mu_{\beta}^{-}=P_{\beta}\left(E^{+}\right)-P_{\beta}\left(E^{-}\right). \tag{A.189}$$

By definition (see Eqs. (3.41,3.38))

$$\mu\equiv\frac{\sqrt{\Delta}}{\Delta+\gamma\left(E-C\right)},$$
$$P=\frac{E-C}{\Delta+\gamma\left(E-C\right)}.$$

It is straightforward to verify that Eq. (A.189) is actually an identity. QED.

To sum up, we have proved that it is equivalent to calculate the transmission coefficient with either physical Green's functions or auxiliary Green's functions.

A.16 Specular scattering vs diffusive scattering

The transmission coefficient in disordered two-probe systems has been derived in Section 2.8 by using the NECPA theory. Although the Green's function approach provides an efficient and systematic way to derive physical quantities, the physical meaning is less transparent in the derivation as well as the derived results. In this section, we interpret the physical meaning of the transmission coefficient in the presence of disorder by connecting it to the scattering states approach.

First of all, let us consider a general disordered two-probe system. The disorder-averaged transmission coefficient can be written as (the argument E is omitted)

$$\begin{aligned}\overline{T}&=\mathrm{Tr}\ \overline{G^{r}\Gamma_{L}G^{a}\Gamma_{R}}\\&=\mathrm{Tr}\ \overline{G^{r}\Gamma_{L}}\,\overline{G^{a}}\Gamma_{R}\\&=\mathrm{Tr}\ \overline{G^{r}}\Gamma_{L}\overline{G^{a}}\Gamma_{R}+\mathrm{Tr}\ \overline{G^{r}}\Lambda_{L}\overline{G^{a}}\Gamma_{R}\\&\equiv T_{1}+T_{2},\end{aligned} \tag{A.190}$$

where T_1 is a simple average and T_2 is the contribution from the lesser coherent potential (see the second line of Eq. (2.134)). The physical meaning of T_1 can be interpreted analogously to Eq. (2.112). Decompose Γ_L and Γ_R into outer products of lead eigenstates by using Eq. (2.111). As a result,

www.ebook3000.com

T_1 is reduced to

$$\begin{aligned} T_1 &= \sum_{ij} \mathrm{Tr}\ \overline{G^r}\,|L_i\rangle\langle L_i|\,\overline{G^a}\,|R_j\rangle\langle R_j| \\ &= \sum_{ij} \mathrm{Tr}\ \langle R_j|\,\overline{G^r}\,|L_i\rangle\langle L_i|\,\overline{G^a}\,|R_j\rangle \\ &= \sum_{ij} \mathrm{Tr}\ \overline{\langle R_j|\,G^r\,|L_i\rangle} \times \overline{\langle L_i|\,G^a\,|R_j\rangle} \\ &= \sum_{ij} \left|\overline{t_{ji}}\right|^2, \end{aligned} \tag{A.191}$$

where t_{ji} is the transmission amplitude from the incoming wave $|L_i\rangle$ of the left lead to the outgoing wave $|R_j\rangle$ of the right lead. Notice that the disorder average in T_1 is inside the modular square. On the other hand, $\overline{T}$ is defined by

$$\overline{T} = \sum_{ij} \overline{|t_{ji}|^2}, \tag{A.192}$$

where the disorder average is outside the modular square. By comparing Eq. (A.191) and Eq. (A.192), T_2 is obtained as

$$T_2 = \sum_{ij} \left(\overline{|t_{ji}|^2} - \left|\overline{t_{ji}}\right|^2\right). \tag{A.193}$$

Therefore the physical meaning of T_1 is the transmission coefficient of averaged transmission amplitude, and T_2 is the correction due to the transmission amplitude fluctuation.

Next we investigate a disordered two-probe system with lateral periodicity. Due to the periodicity, the disorder-averaged Green's function and the linewidth function can be Fourier transformed. After the Fourier transform, Eq. (A.190) is reduced to

$$\begin{aligned} \overline{T} &= \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\ \overline{G^r}\,(k)\,\Gamma_L\,(k)\,\overline{G^a}\,(k)\,\Gamma_R\,(k) \\ &\quad + \int_{BZ} \frac{d^2k}{(2\pi)^2} \mathrm{Tr}\ \overline{G^r}\,(k)\,\Lambda_L \overline{G^a}\,(k)\,\Gamma_R\,(k) \\ &\equiv T_s + T_d, \end{aligned} \tag{A.194}$$

where T_s and T_d are referred to as the specular part and the diffusive part respectively. Since the left and right leads are disorder free, k is a good quantum number in the leads. The central region contains random

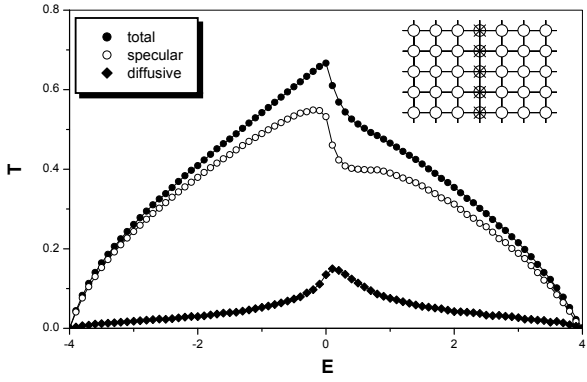


Fig. A.11 Specular and diffusive transmission coefficient calculated with Green's function approach (lines) and scattering states approach (symbols). The 2d tight-binding model is shown in the inset: The white dots represent clean sites having the on-site energy $\varepsilon_0 = 0$ and the nearest neighbor coupling $t_0 = 1$. The crossed dots represent disorder sites having the random on-site energy $\varepsilon_1 = 0$ with probability 50% and $\varepsilon_2 = 1.5$ with probability 50%.

disorder, and hence an incoming wave with transverse momentum k can be scattered to outgoing waves with $k' \neq k$. Eq. (A.194) indicates that T_s is the transmission coefficient of k-reserved scattering events and T_d is the transmission coefficient of non-k-reserved scattering events, namely,

$$T_s = \sum_k |t_{kk}|^2 , \tag{A.195}$$

$$T_d = \sum_{k \neq k'} |t_{k'k}|^2 . \tag{A.196}$$

As a numerical verification, we calculate the transmission coefficient of a 2d tight-binding model by using two different methods. The first method is the Green's function approach which does the disorder average with nonequilibrium coherent potential. The second method is the scattering states approach which does the disorder average with brute force. The second method allows us to explicitly distinguish the transmission coefficients of k-reserved and non-k-reserved scattering events. It is shown in Fig. A.11 that T_s and T_d of Eq. (A.194) agree very well with the results of the scattering states approach, verifying the physical meaning of specular part and diffusive part.

www.ebook3000.com

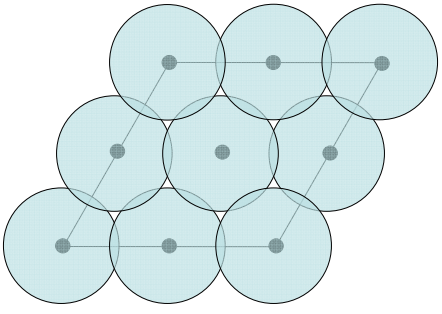


Fig. A.12 Schematic plot of the atomic sphere approximation.

A.17 Fill the space with atomic spheres

The LMTO method adopts atomic sphere approximation (ASA) to fill up the unit cell with slightly overlapped atomic spheres (see Fig. A.12). It is required that $V_{atom} = V_{cell}$ where V_{atom} is the total volume of atomic spheres and V_{cell} is the total volume of unit cell. For close-packed structures or nearly close-packed structures, one can assign atomic spheres only to atom sites. For non-close-packed structures, surfaces, and interfaces, one needs to assign atomic spheres to both atom sites and vacancy sites. This section provides some guidelines for filling the space with atomic spheres.

A.17.1 *Regular structures*

The ASA space filling schemes are known for the following regular structures: face centered cubic (FCC), body centered cubic (BCC), hexagonal close-packed (HCP), simple cubic (SC), diamond (DIA), rocksalt (NaCl), caesium chloride (CsCl), zincblende (ZnS), calcium fluoride (CaF_2), Wurtzite (WTZ), and graphene (GRN).

(1) FCC, BCC, HCP, CsCl are close-packed or nearly close-packed structures. An atomic sphere can be assigned to each atom site.

(2) SC structure can be converted to BCC structure by adding a vacancy site to the cubic center.

(3) NaCl structure is identical to SC structure if Na atom and Cl atom are not distinguished.

(4) DIA structure can be converted to BCC structure by adding vacancy sites which form another diamond. Consider a cubic unit cell of Diamond

structure whose unit cell vectors are

$$\mathbf{a}_1 = [1, 0, 0],$$
$$\mathbf{a}_2 = [0, 1, 0],$$
$$\mathbf{a}_3 = [0, 0, 1].$$

The positions of atom site (C) and vacancy site (Vac) are

site	C	C	Vac	Vac	C	C	Vac	Vac	C	C	Vac	Vac	C	C	Vac	Vac
x	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$
y	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$
z	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$

(5) ZnS structure is identical to Diamond structure if Zn atom and S atom are not distinguished.

(6) CaF_2 structure (e.g., HfO_2) can be converted to BCC structure by adding a vacancy site. Consider a primitive unit cell of CaF_2 whose unit cell vectors are

$$\mathbf{a}_1 = [0, \frac{1}{2}, \frac{1}{2}],$$
$$\mathbf{a}_2 = [\frac{1}{2}, 0, \frac{1}{2}],$$
$$\mathbf{a}_3 = [\frac{1}{2}, \frac{1}{2}, 0].$$

The positions of atom site (Ca, F) and vacancy site (Vac) are

site	Ca	F	F	Vac
x	0	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{2}$
y	0	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{2}$
z	0	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{2}$

(7) WTZ structure (e.g., ZnO) can be made nearly close-packed by adding two types of vacancy sites. Consider a primitive cell of an ideal Wurtzite structure whose unit cell vectors are

$$\mathbf{a}_1 = [1, 0, 0],$$
$$\mathbf{a}_2 = [-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0],$$
$$\mathbf{a}_3 = [0, 0, c],$$

www.ebook3000.com

where $c = \sqrt{\frac{8}{3}}$. The positions of atom site (Zn, O) and vacancy site (Vac_1, Vac_2) are

site	Zn	Vac_1	Vac_2	O	Zn	Vac_1	Vac_2	O
x	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$
y	0	$\frac{\sqrt{3}}{6}$	$\frac{\sqrt{3}}{3}$	0	$\frac{\sqrt{3}}{6}$	0	$\frac{\sqrt{3}}{3}$	$\frac{\sqrt{3}}{6}$
z	0	$\frac{1}{2}u \cdot c$	$\frac{1}{2}u \cdot c$	$u \cdot c$	$\frac{1}{2}c$	$\frac{1+u}{2}c$	$\frac{1+u}{2}c$	$\left(u + \frac{1}{2}\right) c$

where $u = \frac{3}{8}$. Notice that the two types of vacancy sites have different radiuses: $R\,(\text{Vac\_1}) < R\,(\text{Zn}) = R\,(\text{O}) < R\,(\text{Vac\_2})$.

(8) GRN structure can be made nearly close-packed by periodic expansion and adding vacancy sites. Firstly, a single graphene sheet is repeated periodically perpendicular to its plane to construct a graphene stack. As long as the distance between two graphene sheets is sufficiently large, each of the graphene sheets is nearly isolated. Secondly, two types of vacancy sites are filled into the vacuum between two graphene sheets. Consider a primitive cell of Graphene structure whose unit cell vectors are

$$\mathbf{a}_1 = [\frac{\sqrt{3}}{2}a, \frac{1}{2}a, 0],$$
$$\mathbf{a}_2 = [\frac{\sqrt{3}}{2}a, -\frac{1}{2}a, 0],$$
$$\mathbf{a}_3 = [0, 0, \frac{8}{6}b],$$

where $a = 2.45\mathring{A}$ and $b = 6.7\mathring{A}$. The positions of atom site (C) and vacancy site (Vac_1, Vac_2) are

site	C	C	Vac_1	Vac_2	Vac_2	Vac_1	Vac_2	Vac_2	Vac_1	Vac_2	Vac_2	Vac_1
u_1	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$
u_2	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$
u_3	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{1}{8}$	$\frac{2}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{4}{8}$	$\frac{4}{8}$	$\frac{5}{8}$	$\frac{6}{8}$	$\frac{6}{8}$	$\frac{7}{8}$

where fractional coordinates are used (i.e., $\mathbf{r} = u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + u_3\mathbf{a}_3$). Notice that the two types of vacancy sites have different radiuses: $R\,(\text{Vac\_1}) : R\,(\text{Vac\_2}) : R\,(\text{C}) = 2.20 : 1.59 : 1.59$.

8.27.2 *Irregular structures*

For irregular structures, there is no simple recipe to construct the ASA scheme. The general strategy is as follows: (1) Identify regular structures within an irregular structure and fill them with known ASA schemes. (2)

Fill the vacuum between regular structures with vacancy sites, and tune the centers and radiuses of vacancy spheres to optimize the spatial occupation. One can do it manually or use the accessory tool *ASA-volume optimizer*. (3) Check the obtained ASA scheme by comparing the band structure to a standard one which is obtained with an ASA-free method. If the band structure is satisfactory in the vicinity of Fermi energy, one can proceed to study the transport properties. Otherwise one needs to fine tune the ASA scheme to fit the standard band structure. One can do it manually or use the accessory tool *ASA-band optimizer*.

As an example, we shall investigate the ASA scheme of Fe/MgO/Fe sandwich structure [19]. Notice that bulk Fe and bulk MgO have good lattice match upon $\frac{\pi}{4}$ rotation around z-axis. It is assumed that both Fe and MgO adopt the same lattice constant $a = 5.4160$ in the transverse dimensions. By structural relaxation, the distance between Fe and MgO is determined as $d = 4.0820$.

Three regular structures can be identified in the sandwich structure: the left Fe region, the central MgO region, and the right Fe region. The left and right Fe regions have BCC structure, and each Fe atom is assigned to an atomic sphere with radius

$$R\,(\mathrm{Fe}) = 2.6667.$$

The central MgO region has NaCl structure (rotated by $\frac{\pi}{4}$ along z-axis) which can be converted to BCC structure by adding a vacancy layer (Vac_3). Further study on the band structure of MgO indicates that the optimal atomic sphere radiuses are

$$\begin{aligned} R\,(\mathrm{Mg}) &= 1.8141, \\ R\,(\mathrm{O}) &= 2.5577, \\ R\,(\mathrm{Vac\_3}) &= 1.2711, \end{aligned}$$

which is consistent with the fact that the radius of O^{2-} is larger than that of Mg^{2+}.

Having filled up regular structures, let's turn to the interface region (see Fig. A.13). Two vacancy layers need to be added to fill up the gap between Fe and MgO: Vac_1 layer is a replacement of Fe (a) layer from the Fe side, and Vac_2 layer is a replacement of Vac_3 layer from the MgO side. Further study on the band structure of Fe/MgO superlattice indicates that

www.ebook3000.com

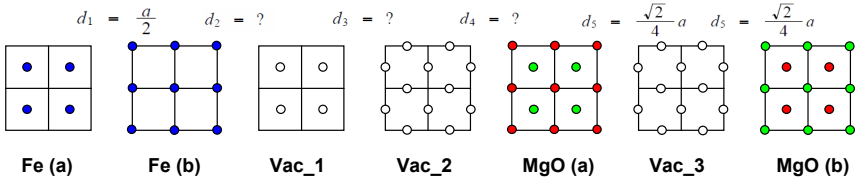


Fig. A.13 The ASA scheme of Fe/MgO interface. The blue, red, green, and white dots represent Fe, O, Mg, and Vac atomic sites respectively. The distances between the atomic layers are indicated by d_i, where $d_i =?$ means the value is to be optimized.

the optimal atomic sphere radiuses and layer distances are

$$\begin{aligned} R\,(\text{Vac\_1}) &= 1.5373, \\ R\,(\text{Vac\_2}) &= 1.2722, \\ d_2 &= 1.2998, \\ d_3 &= 0.8666, \\ d_4 &= 1.9156, \end{aligned}$$

where $d = d_2 + d_3 + d_4$.

A.18 Symmetric k-sampling

In this section, the symmetric k-samplings for the ten 2d point groups are summarized in Fig. A.14 to Fig. A.18. Each figure has three columns: The left column is the unit cell in real space; The middle column is the k-sampling in reciprocal space without time-reversal symmetry; The right column is the k-sampling in reciprocal space with time-reversal symmetry. Notice that the symmetric k-sampling is designed to be compatible with the uniform k-sampling which has a physical meaning as explained in Fig. 6.8.

The 2d point groups are C_n and D_n where $n = 1, 2, 3, 4, 6$. The group C_n only contains rotations while D_n contains both rotations and mirror reflections. In the left column of each figure, all the available symmetry operations are marked in the unit cell center. The rotation axis c_2, c_3, c_4, and c_6 are represented by oval, triangle, diamond, and hexagon, respectively. The reflection mirrors are represented by shaded bars.

In reciprocal space, the Brillouin zone is first sampled by a $N_1 \times N_2$ uniform mesh where the k-points are defined by

$$\mathbf{k} = \frac{n_1}{N_1}\mathbf{b}_1 + \frac{n_2}{N_2}\mathbf{b}_2,$$

with $n_1 = 0, 1, \ldots, N_1$ and $n_2 = 0, 1, \ldots, N_2$. Here $\mathbf{b}_1$ and $\mathbf{b}_2$ are the unit cell vectors of the reciprocal lattice. As discussed in Section 6.6.1, the uniform k-sampling has the physical meaning that the unit cell is repeated N_1 times in the first dimension and N_2 times in the second dimension and connected to itself to form a cyclic structure (see Fig. 6.8). The symmetric k-sampling is based on the 2d mesh of uniform k-sampling. Due to symmetry operations, only a portion of the Brillouin zone needs to be taken into account.

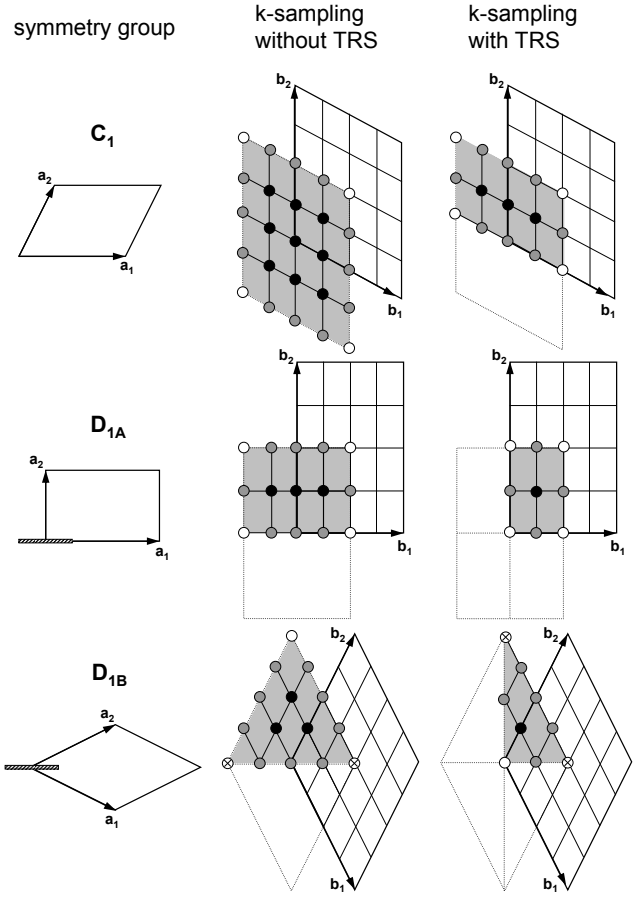


Fig. A.14 The symmetric k-sampling for the symmetry group C_1 and D_1. The center of the Brillouin zone is shifted to the Γ-point. There are two scenarios of D_1 which are referred to as D_{1A} and D_{1B}.

www.ebook3000.com

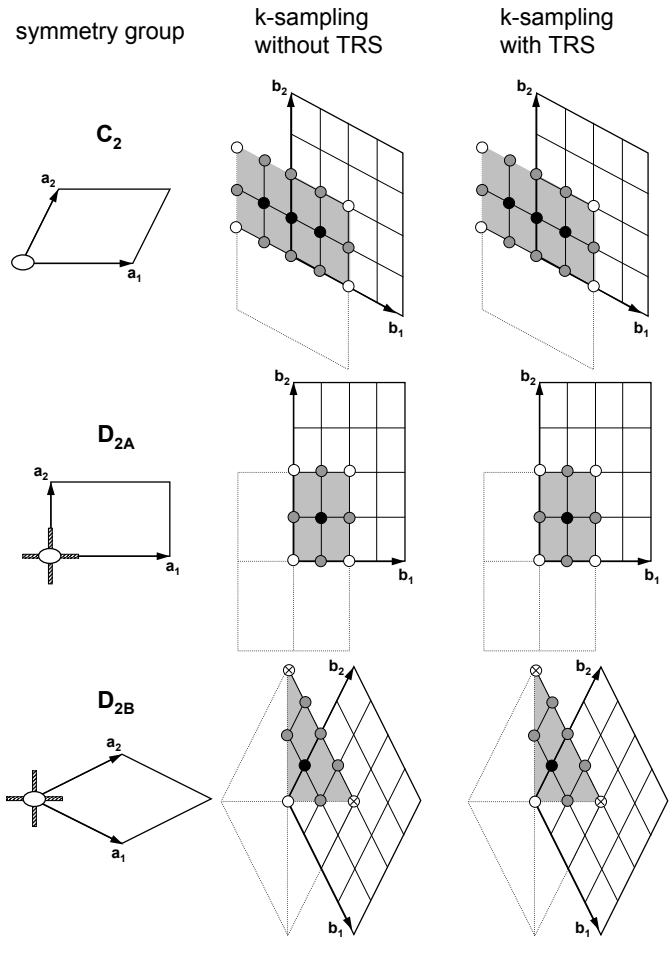


Fig. A.15 The symmetric k-sampling for the symmetry group C_2 and D_2. The center of the Brillouin zone is shifted to the Γ-point. There are two scenarios of D_2 which are referred to as D_{2A} and D_{2B}.

Besides the geometric symmetry, the time-reversal symmetry may also reduce the size of k-sampling. For those symmetry groups without c_2 operation, the time-reversal symmetry relates $+\mathbf{k}$ and $-\mathbf{k}$ together and hence reduces the number of k-points by half.

In Fig. A.14 to Fig. A.18, the k-points are represented by symbols: Black dot, gray dot, white dot, and crossed dot represent the k-points with

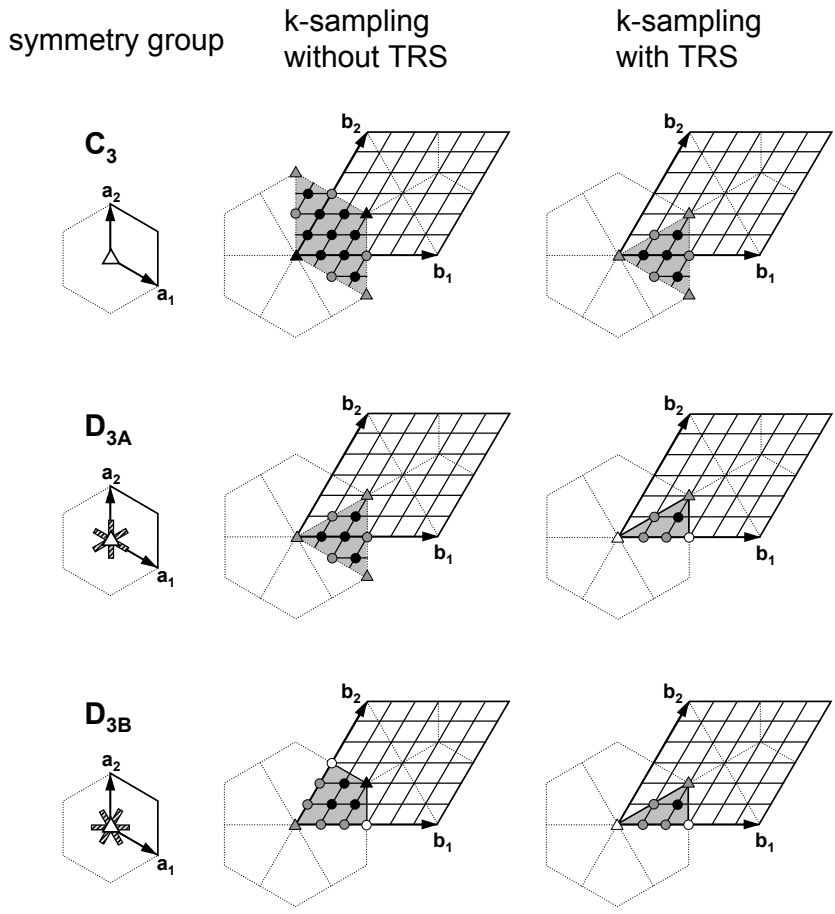


Fig. A.16 The symmetric k-sampling for the symmetry group C_3 and D_3. The first Brillouin zone is indicated by the hexagon with dotted line. There are two scenarios of D_3 which are referred to as D_{3A} and D_{3B}.

weight 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, respectively; Black triangle, gray triangle, and white triangle represent the k-points with weight $\frac{1}{3}$, $\frac{1}{6}$, $\frac{1}{12}$, respectively. The weights of the k-points are determined as follows: Assign each k-point a small territory (parallelogram, square, diamond, hexagon) to patch up the Brillouin zone, and the weight is defined as the fraction of the small territory inside the irreducible Brillouin zone (shaded area).

www.ebook3000.com

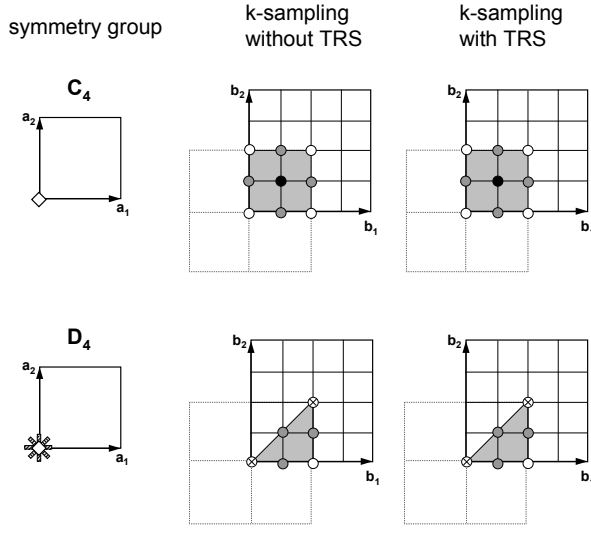


Fig. A.17 The symmetric k-sampling for the symmetry group C_4 and D_4. The first Brillouin zone is indicated by the square with dotted line.

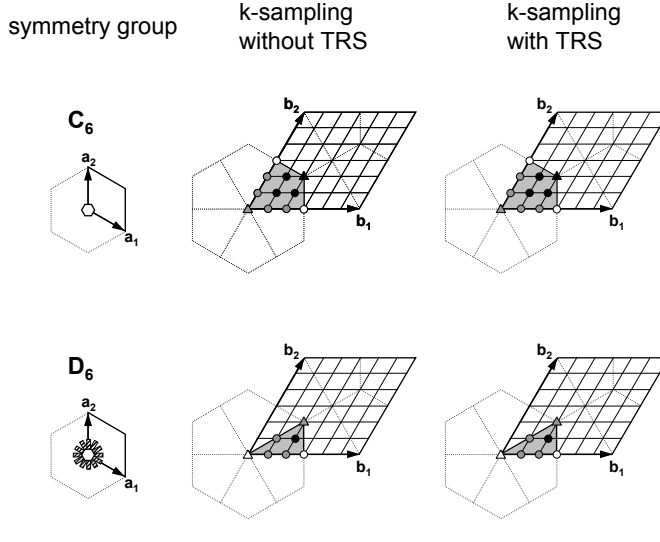


Fig. A.18 The symmetric k-sampling for the symmetry group C_6 and D_6. The first Brillouin zone is indicated by the hexagon with dotted line.

A.19 Unfolding algorithm

This section explains the unfolding algorithm, using the FCC crystal as an example. Suppose one unit cell of the FCC crystal is known, the goal is to find the primitive cell of the crystal.

The first step is to repeat the unit cell in all three dimensions so that the extended volume contains at least one primitive cell. The second step is to pick up the smallest tetrahedron in the extended volume (the algorithm will be explained later). The corners of the tetrahedron are denoted by R_0, R_1, R_2, R_3. The vectors $\mathbf{r}(R_1) - \mathbf{r}(R_0)$, $\mathbf{r}(R_2) - \mathbf{r}(R_0)$, $\mathbf{r}(R_3) - \mathbf{r}(R_0)$ consist of the unit cell vectors of a primitive cell.

The algorithm to pick up the smallest tetrahedron is described as follows:

Step 1: Pick up a single atom R_0.

Step 2: Find the 12 nearest neighbors of R_0, and the atoms are named as $A_1, A_2, \ldots, A_{12}$. The 12 atoms have equal distance $\frac{\sqrt{2}}{2}a$ to R_0, where a is the lattice constant.

Step 3: Pick up any atom from $A_1, A_2, \ldots, A_{12}$, and the atom is renamed as R_1. The remaining 11 atoms are renamed as $A'_1, A'_2, \ldots, A'_{11}$.

Step 4: Find the 4 nearest neighbors of R_1 from $A'_1, A'_2, \ldots, A'_{11}$, and the atoms are renamed as $A''_1, A''_2, A''_3, A''_4$. The 4 atoms have equal distance $\frac{\sqrt{2}}{2}a$ to R_1, where a is the lattice constant.

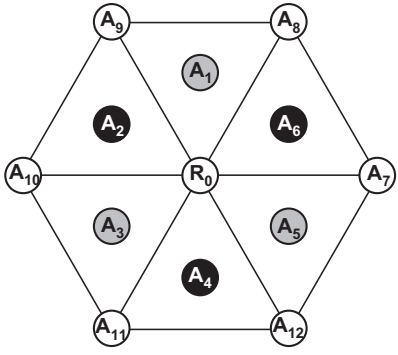


Fig. A.19 Algorithm for picking up the smallest tetrahedron in the FCC crystal. $A_1, A_2, \ldots, A_{12}$ are the nearest neighbors of R_0. The 13 atoms are distributed in three layers: The first layer contains A_1, A_3, A_5; The second layer contains $R_0, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}$; and the third layer contains A_2, A_4, A_6. The distance from A_i to R_0 is $\frac{\sqrt{2}}{2}a$ and the distance between the layers is $\frac{\sqrt{3}}{3}a$, where a is the lattice constant.

www.ebook3000.com

Step 5: Pick up any atom from $A_1'', A_2'', A_3'', A_4''$, and the atom is renamed as R_2. The remaining 3 atoms are renamed as A_1''', A_2''', A_3'''.

Step 6: Find the nearest neighbor of R_2 from A_1''', A_2''', A_3''', and the atom is renamed as R_3.

The four atoms R_0, R_1, R_2, R_3 are sitting on the corners of a tetrahedron which is exactly we are looking for.

The proof of the algorithm is as follows (see Fig. A.19): In step 2, $A_1, A_2, \ldots, A_{12}$ are distributed in three FCC close-packed layers. In step 3, due to symmetry consideration, it is sufficient to consider the case of $R_1 = A_1$ or $R_1 = A_7$. In step 4, if $R_1 = A_1$ then the 4 nearest neighbors are A_8, A_9, A_3, A_5; if $R_1 = A_7$ then the 4 nearest neighbors are A_8, A_6, A_5, A_{12}. In step 5, if $R_1 = A_1$ then R_2 is one of A_8, A_9, A_3, A_5; if $R_1 = A_7$ then R_2 is one of A_8, A_6, A_5, A_{12}. In step 6, no matter how R_1 and R_2 are selected, R_0, R_1, R_2, R_3 always define a tetrahedron. The various possibilities are summarized as follows:

$$
\begin{aligned}
&R_0 \to (A_1, A_2, \ldots, A_{12}) \\
&\to \begin{cases}
R_1 = A_1 \to (A_8, A_9, A_3, A_5) \to \begin{cases} R_2 = A_8 \to R_3 = A_9 \\ R_2 = A_9 \to R_3 = A_8 \\ R_2 = A_3 \to R_3 = A_5 \\ R_2 = A_5 \to R_3 = A_3 \end{cases} \\
R_1 = A_7 \to (A_8, A_6, A_5, A_{12}) \to \begin{cases} R_2 = A_8 \to R_3 = A_6 \\ R_2 = A_6 \to R_3 = A_8 \\ R_2 = A_5 \to R_3 = A_{12} \\ R_2 = A_{12} \to R_3 = A_5 \end{cases}
\end{cases}
\end{aligned}
$$

The algorithm is designed to find the primitive cell of an FCC crystal. For other crystal types, one can design similar algorithms based on the geometry deduction. Once the primitive cell is found, one can proceed to construct the Hamiltonian and calculate the band structure.

A.20 Mixing algorithms

In this section, we discuss two mixing algorithms, linear mixing and Anderson mixing. The efficiency of the two mixing algorithms will be compared by solving a model problem.

Consider a multidimensional nonlinear equation

$$X = F(X), \tag{A.197}$$

where X is an $n \times 1$ vector and F is an $n \times 1$ vector function. In principle one can make an initial guess X_0 and iterate Eq. (A.197) until the solution

is fully converged

$$X_{n+1} = F(X_n). \tag{A.198}$$

However the iteration may converge very slowly or even diverge. To improve the convergence, one needs to take into account the iteration history to make a better prediction of the next trial solution. This is called mixing algorithm.

The simplest mixing algorithm is linear mixing which only takes into account the previous step. In linear mixing, the iteration proceeds as follows

$$X_{n+1} = (1-\alpha) X_n + \alpha F(X_n), \tag{A.199}$$

where $\alpha \in (0,1)$ is the mixing rate.

Advanced mixing algorithms such as Anderson mixing take into account a longer iteration history. In Anderson mixing, the iteration proceeds as follows [20]

$$X_{n+1} = X_n + \beta D_n - \sum_{i=1}^{M} \gamma_i \left(\Delta X_{n-M+i} + \beta \Delta D_{n-M+i}\right), \tag{A.200}$$

where $\beta \in (0,1)$ is the mixing rate and D_i, ΔX_i, ΔD_i are defined by

$$\begin{aligned} D_i &= F(X_i) - X_i, \\ \Delta X_i &= X_i - X_{i-1}, \\ \Delta D_i &= D_i - D_{i-1}. \end{aligned}$$

The coefficients $\{\gamma_i\}$ are determined by the linear equation array

$$\sum_{j=1}^{M} O_{ij} \cdot \gamma_j = B_j,$$

where O and B are defined by

$$\begin{aligned} O_{ij} &\equiv \langle \Delta D_{n-M+i} | \Delta D_{n-M+j} \rangle \left(1 + \delta_{ij} w_0^2\right), \\ B_i &\equiv \langle \Delta D_{n-M+i} | D_n \rangle, \end{aligned}$$

$\langle \cdots | \cdots \rangle$ is the inner vector product, and $w_0 \approx 0.01$ is a stabilization factor. Despite the sophisticated form, Eq. (A.200) is nothing but a linear combination of $F(X_n)$, $F(X_{n-1})$, ..., $F(X_{n-M})$ and X_n, X_{n-1}, ..., X_{n-M}.

Next we examine the efficiency of linear mixing and Anderson mixing by solving a model problem [20]. The nonlinear equation is defined by

$$x_i = (1-d_i) x_i - c x_i^3, \tag{A.201}$$

www.ebook3000.com

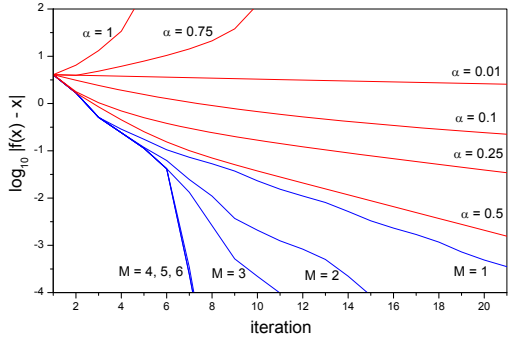


Fig. A.20 Solution error as a function of iteration step for solving Eq. (A.201). The initial guess is $X_0 = [1, 1, 1, 1, 1]$. The red lines are for linear mixing where the mixing rate α is indicated in the plot. The blue lines are for Anderson mixing where the mixing rate $\beta = 0.5$ and the history length M is indicated in the plot.

where $i = 1, 2, \ldots, 5$, $c = 0.01$, and $d = [3.0, 2.0, 1.5, 1.0, 0.5]$. Eq. (A.201) is solved with the above two mixing algorithms, and the solution error is plotted as a function of iteration step in Fig. A.20. The convergence of linear mixing strongly depends on the mixing rate α. If α is larger than a critical value, the iteration will diverge ($\alpha = 0.75$); If α is too small, the iteration will converge very slowly ($\alpha = 0.01$). There exists an optimal α to achieve fast convergence ($\alpha = 0.5$). The convergence of Anderson mixing not only depends on the mixing rate β but also on the history length M. For $M = 0$, Anderson mixing is reduced to linear mixing. With increasing M, the convergence gets faster and faster. For $M \geqslant 5$, the convergence ceases to improve. Generally speaking Anderson mixing outperforms linear mixing if the initial guess is not too far from the final solution. On the other hand, linear mixing is extremely stable as long as the mixing rate is sufficiently small. So a hybrid mixing strategy is to first reduce the error by applying linear mixer and then accelerate the convergence by switching to Anderson mixer.

To sum up, we have discussed two typical mixing algorithms, linear mixing in Eq. (A.199) and Anderson mixing in Eq. (A.200). In the literature, there are a few other advanced mixing algorithms including Broyden mixing [21], Pulay mixing [22], and multi-secant mixing [23], which have similar performance to Anderson mixing. In addition, it is not necessary to have the same mixing rate for all variables. One can assign larger mixing rates to slow-varying variables to accelerate the convergence [24, 25].

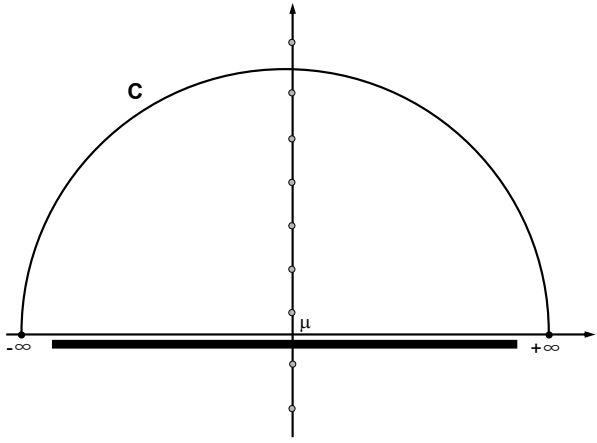


Fig. A.21 The auxiliary contour C in the Fermi pole summation. Since the integral on the semicircle C goes to zero with increasing radius, the real axis integral is equal to the summation of Fermi pole residues enclosed by C and the real axis.

A.21 Modified Fermi pole summation technique

To evaluate the energy integral J of Eq. (5.69), one can change the integral path from the real axis to a complex contour. Besides the complex contour integral, the energy integral can also be evaluated by a summation of Fermi pole residues [26], which will be the subject of this section.

In the complex plane, add a large contour C to the real axis integral path and let C go through the middle of two Fermi poles (see Fig. A.21). If the radius of C is sufficiently large, the integral of C is negligible because $G^r(z) \sim \frac{1}{z}$ and $|f(z)|$ is bounded. Therefore J is equal to the summation of Fermi pole residues

$$J = 2\pi \mathrm{i} \sum_{k=1}^{\infty} (-k_B T)\, G^r\left[\mu + \mathrm{i}\,(2k-1)\,\pi k_B T\right]. \tag{A.202}$$

However, the convergence of the series in Eq. (A.202) is rather slow because the Fermi poles are distributed uniformly along the imaginary axis. We wish to replace $f(z)$ by another complex function $\tilde{f}(z)$ which is a good approximation of $f(z)$ on the real axis but has improved pole distribution in the complex plane. Such a complex function $\tilde{f}(z)$ is constructed in Ref. [26] by using continued fraction

$$\tilde{f}(z) = \frac{1}{2} + F_M\left(\frac{z-\mu}{k_B T}\right), \tag{A.203}$$

www.ebook3000.com

where F_M is an odd complex function defined by the sum over $2M$ poles

$$F_M(t) = \sum_{k=1}^{2M} \frac{R_k}{t - \mathrm{i}\lambda_k}.$$

The pole $\mathrm{i}\lambda_k$ and the residue R_k are determined by the solutions of the generalized eigenvalue problem

$$Av_k = \lambda_k B v_k, \tag{A.204}$$

where A is a $2M$-by-$2M$ diagonal matrix and B is a $2M$-by-$2M$ tridiagonal matrix

$$A = -\begin{pmatrix} 1 & & & & & \\ & 3 & & & & \\ & & 5 & & & \\ & & & \ddots & & \\ & & & & 4M-1 \end{pmatrix}, \tag{A.205}$$

$$B = \frac{1}{2}\begin{pmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & \ddots & \ddots & & \\ & & \ddots & 0 & 1 & \\ & & & 1 & 0 & \end{pmatrix}. \tag{A.206}$$

Let λ_k and v_k denote the k^{th} eigenvalue and eigenvector. The residue R_k is obtained as

$$R_k = \frac{1}{4}\lambda_k V_{1k}\left[(BV)^{-1}\right]_{k1}, \tag{A.207}$$

where $V = [v_1, v_2, \ldots, v_{2M}]$ is the eigenvector matrix. Notice that both λ_k and R_k are real numbers.

The Fermi function $f(z)$ can be approximated by $\tilde{f}(z)$ to high precision on the real axis. The accuracy of $\tilde{f}(E)$ is listed in Table A.208 as a function of M in the energy range $\mu - 1000k_BT < E < \mu + 1000k_BT$ (At room temperature, the energy range amounts to ± 26 eV around the Fermi energy).

M	10	20	30	40	50	60
$\left\|\left\|f - \tilde{f}\right\|\right\|$	3×10^{-1}	4×10^{-2}	7×10^{-4}	2×10^{-6}	2×10^{-9}	3×10^{-13}

(A.208)

In the complex plane, $\tilde{f}(z)$ has $2M$ poles on the imaginary axis which are distributed symmetrically with respect to the real axis. The distribution of

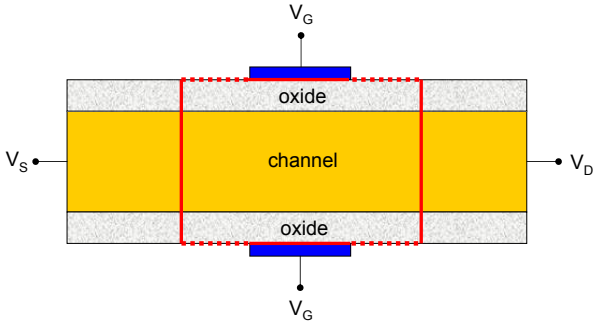


Fig. A.22 A sketch of double-gate FET having source, drain and gate terminals. The Poisson box is indicated by the thick rectangular box where solid line and broken line represent Dirichlet boundary and floating boundary respectively.

the poles in the upper plane is shown in Fig. 3 of Ref. [26]: The first 61% of poles are located near the original Fermi poles, but the remaining poles are further and further away from the original Fermi poles. To evaluate the energy integral J, one only needs to sum up the residues of the poles in the upper plane

$$J = 2\pi \mathrm{i} \sum_{k=1}^{M} R_k k_B T G^r \left(\mu + \mathrm{i}\lambda_k k_B T\right), \tag{A.209}$$

where $\lambda_k > 0$.

To sum up, Eq. (A.209) indicates that the energy integral in Eq. (5.69) can be evaluated efficiently by a summation of residues at modified Fermi poles defined in Eqs. (A.204–A.207).

A.22 Field effect transistor with gate terminals

Field effect transistors (FET) have more than two "probes". To simulate FETs, it is necessary to include the gate terminals in the two-probe model, which will be the subject of this Appendix [27].

The double-gate FET structure is sketched in Fig. A.22. Assume that oxide layers are sufficiently thick so that the leakage current through the gate terminal is negligible. Thus the effects of gate terminals can be taken into account through only electrostatics. The electrostatic potential V is determined by the Poisson equation

$$\nabla^2 V = -4\pi\rho, \tag{A.210}$$

www.ebook3000.com

where ρ is the charge density inside the Poisson box. V is subject to the following boundary condition

$$\begin{aligned} V|_{\text{source}} &= U_L + Q_e V_S, \\ V|_{\text{drain}} &= U_R + Q_e V_D, \\ V|_{\text{gate}} &= \Delta W + Q_e V_G, \\ \partial_n V|_{\text{other}} &= 0, \end{aligned} \tag{A.211}$$

where V_S, V_D, V_G are the source, drain, and gate voltages and $Q_e = -1$ is to take into account the negative sign of electron charge. U_L and U_R are the surface potentials of the left and right leads whose chemical potentials have been shifted to zero. ΔW is the work function difference between the gate material and the channel material, aligning the Fermi levels of the two parts. The source, drain and gate areas have Dirichlet boundary condition and other areas have floating boundary condition.

In the LMTO method, we don't solve the Poisson equation but calculate the Madelung potential instead. Notice that the Madelung potential V_M is also a solution of the Poisson equation except that V_M does not satisfy the boundary condition. Since the Poisson equation is a linear equation, one can make a correction δV to V_M such that

$$\nabla^2 \delta V = 0, \tag{A.212}$$

and $V = V_M + \delta V$ satisfies the boundary conditions Eq. (A.211). δV has to be solved on a real space grid. To use a coarse real space grid, one needs to make δV as smooth as possible. Notice that ρ in the RHS of Eq. (A.210) only involves the charge density inside the Poisson box. One has the freedom to manipulate the charge density outside the Poisson box so that V_M satisfies the boundary condition approximately and δV takes care of the small leftover.

For the double-gate FET structure, the charge density is constructed by mirror images as shown in Fig. A.23. The two-probe system of Fig. A.22 is denoted as image(0), and its images in the mirrors M_1 and M_2 consist of an infinite stack (± 1), (± 2), (± 3), $\cdots$. On the one hand, V_M of the stack automatically satisfies $\partial_n V = 0$ on both upper and lower boundaries due to the mirror symmetry. As a result, the boundary condition is automatically satisfied on the lead and buffer layer surfaces. On the other hand, the stack can be regrouped into double sized unit cells which are periodically expanded along the transverse dimension. Hence V_M can be readily calculated with the existing Ewald sum technique described in Section 5.4 and

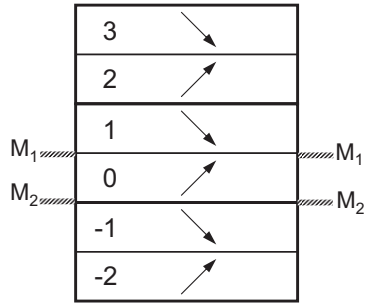


Fig. A.23 Mirror images of double-gate FET. The two mirrors M_1 and M_2 are placed on the upper and lower boundaries of the double-gate FET. The images of the double-gate FET can be regrouped into double sized unit cells, image(-2) and image(-1), image(0) and image(1), image(2) and image(3), etc.

6.3. It is worth mentioning that the mirror image construction works not only for an orthogonal but also for nonorthogonal Poisson box.

To sum up, the effects of gate terminals can be included in the two-probe model using electrostatics. In the LMTO method, the total electrostatic potential can be decomposed into the Madelung potential and the boundary correction term. For the double-gate FET structure, mirror images are constructed to reduce the computational cost of the boundary correction term.

A.23 Algorithms for solving the Poisson equation

In this section, we discuss algorithms for solving the Poisson equation. After the discretization of real space, the Poisson equation is converted from a differential equation to a linear equation array. Various algorithms are presented to solve the linear system in the 1d case and are later generalized to the 2d and 3d cases. The situations of nonorthogonal Poisson box and nonlinear Poisson equation are also discussed.

A.23.1 *Numerical discretization*

Consider a rectangular box of size $L_1 \times L_2 \times L_3$. Inside the box, the electrostatic potential V and the charge density ρ satisfies the Poisson equation

$$\left(\partial_x^2 + \partial_y^2 + \partial_z^2\right) V(x,y,z) = -4\pi\rho(x,y,z)\,. \tag{A.213}$$

www.ebook3000.com

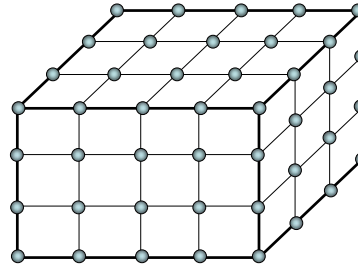


Fig. A.24 Discretization of Poisson box with uniform grid.

V is determined by ρ inside the box as well as the boundary condition. There are several types of boundary condition: (1) periodic boundary condition where the values of V on two opposite surfaces are equal; (2) Dirichlet boundary condition where the values of V are given on the surfaces; (3) Neumann boundary condition where the normal derivatives $\partial_n V$ are given on the surfaces; (4) floating boundary condition where the normal derivatives $\partial_n V$ are zero on the surfaces; (5) Robin boundary condition where the linear combinations of $\alpha V + \beta \partial_n V$ are given on the surfaces; and (6) mixed boundary condition where either the values of V or the normal derivatives $\partial_n V$ are given on the surfaces.

To solve the Poisson equation, the first step is to discretize the differential operator and the boundary condition on an $(N_1 + 1) \times (N_2 + 1) \times (N_3 + 1)$ uniform grid (see Fig. A.24). The grid point is numbered by the indices (i_1, i_2, i_3) where $i_k = 0, 1, \ldots, N_k$, corresponding to the coordinates $x = \frac{L_1}{N_1} i_1$, $y = \frac{L_2}{N_2} i_2$, $z = \frac{L_3}{N_3} i_3$. For the discretization of the RHS of Eq. (A.213), one can simply assign $\rho(x, y, z)$ to each grid point. For the discretization of the LHS of Eq. (A.213), one needs to replace the partial derivatives by finite differences on each grid point. To the second order of accuracy, the partial derivatives in x dimension at (i_1, i_2, i_3) are obtained as [28]

$$\partial_x V \approx \frac{V_{i_1+1,i_2,i_3} - V_{i_1-1,i_2,i_3}}{2\Delta x}, \tag{A.214}$$

$$\partial_x^2 V \approx \frac{V_{i_1+1,i_2,i_3} + V_{i_1-1,,i_2,i_3} - 2V_{i_1,i_2,i_3}}{(\Delta x)^2}, \tag{A.215}$$

which can be easily generalized to y and z dimensions. For the discretization of the boundary condition, one needs to obtain V or $\partial_n V$ at the boundary. V can be obtained by the boundary value, and $\partial_n V$ in x dimension at

(i_1, i_2, i_3) can be obtained by using the single sided finite difference [28]

$$\partial_x V \approx \frac{-3V_{i_1,i_2,i_3} + 4V_{i_1+1,i_2,i_3} - V_{i_1+2,i_2,i_3}}{2\Delta x}, \tag{A.216}$$

which can be easily generalized to y and z dimensions.

Once the discretization is done, the partial differential equation (A.213) is converted to a linear equation array. If we don't care about the computational cost, we can simply solve the linear system with existing linear algebra algorithms. However, the cost will grow rapidly with $(N_1 N_2 N_3)^3$ which is not affordable in many applications. Therefore it is necessary to develop specialized algorithms for solving linear systems derived from the Poisson equation, which will be the subject of the next subsection.

A.23.2 *Algorithms in 1d case*

In this subsection, we focus on the 1d Poisson equation, and the generalization to the 2d and 3d cases will be addressed in the next subsection.

In the 1d case, the Poisson equation is reduced to

$$\partial_x^2 V(x) = -4\pi\rho(x), \tag{A.217}$$

which can be discretized as

$$\begin{pmatrix} 1 & & & & & & & \\ \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & & & & \\ & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & \\ & & & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & \\ & & & & & & 1 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_{N-2} \\ V_{N-1} \\ V_N \end{pmatrix} = \begin{pmatrix} \tilde{V}_A \\ -4\pi\rho_1 \\ -4\pi\rho_2 \\ \vdots \\ -4\pi\rho_{N-2} \\ -4\pi\rho_{N-1} \\ \tilde{V}_B \end{pmatrix}, \tag{A.218}$$

where Dirichlet boundary condition is assumed and $\tilde{V}_A$ and $\tilde{V}_B$ are the boundary values. For other types of boundary condition, everything is the same except that the first and the last rows of Eq. (A.218) need to be modified accordingly. Below five algorithms for solving the linear system will be discussed.

The first algorithm is the direct method for solving sparse linear systems. Notice that the coefficient matrix of Eq. (A.218) is a highly sparse matrix which contains many zeros. To make use of those zeros, the direct method attempts to find a quasi-optimal way of eliminating the unknowns in order to reduce the number of arithmetic operations. Interested readers

www.ebook3000.com

are referred to Ref. [29] for technical details. Here we simply point out that the direct method has been implemented in MATLAB. One can call the linear equation solver with the syntax $x = A\backslash b$ where A is the coefficient matrix and b is the constant vector.

The second algorithm is the tridiagonal solver for tridiagonal linear systems. Notice that Eq. (A.218) can be rewritten into a tridiagonal form by eliminating V_0 and V_N

$$\begin{pmatrix} \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & & \\ \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} \\ & & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_{N-2} \\ V_{N-1} \end{pmatrix} = \begin{pmatrix} -4\pi\rho_1 - \frac{1}{(\Delta x)^2}\tilde{V}_A \\ -4\pi\rho_2 \\ \vdots \\ -4\pi\rho_{N-2} \\ -4\pi\rho_{N-1} - \frac{1}{(\Delta x)^2}\tilde{V}_B \end{pmatrix}. \tag{A.219}$$

Eq. (A.219) can be solved by using the recursive method for tridiagonal linear systems, which has been presented in Eqs. (A.117,A.118,A.119,A.120). It is worth mentioning the second algorithm is actually equivalent to the first one in the 1d case.

The third algorithm is the fast Poisson solver [30]. Notice that the coefficient matrix of Eq. (A.219) is a TST matrix which can be eigendecomposed by using Eq. (A.170)

$$A \equiv \begin{pmatrix} \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & & \\ \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} \\ & & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} \end{pmatrix} = \Psi\Lambda\Psi^\dagger,$$

where Ψ is an orthogonal matrix and Λ is a diagonal matrix

$$\Psi_{jk} = \sqrt{\frac{2}{N}} \sin\frac{jk\pi}{N},$$

$$\Lambda_{jk} = \delta_{jk}\left[\frac{-2}{(\Delta x)^2} + \frac{2}{(\Delta x)^2}\cos\frac{k\pi}{N}\right],$$

with $i, j = 1, 2, \ldots, N-1$. By using $\Psi = \Psi^\dagger = \Psi^{-1}$, V is formally obtained as

$$V = \Psi\Lambda^{-1}\Psi b, \tag{A.220}$$

where b is the RHS of Eq. (A.219). The key of the fast Poisson solver is that the multiplication of Ψ and a vector can be carried out efficiently with

fast Fourier transform. Therefore Eq. (A.220) can be evaluated by two subsequent operations of fast Fourier transform.

The fourth algorithm is the multigrid method. Instead of solving Eq. (A.217), the boundary value problem is converted to a relaxation problem

$$\partial_t \tilde{V}(x,t) = \partial_x^2 \tilde{V}(x,t) + 4\pi\rho(x). \tag{A.221}$$

At $t = 0$, one starts from an initial guess $\tilde{V}(x,0)$ and iterates Eq. (A.221) to obtain $\tilde{V}(x,\delta t)$, $\tilde{V}(x,2\delta t)$, $\tilde{V}(x,3\delta t)$, ..., where δt is a small time interval. The iteration proceeds until $\tilde{V}$ is fully relaxed and the solution is converged to $V(x) = \tilde{V}(x,\infty)$. The convergence of the simple iterative algorithm is rather slow. The multigrid method, in contrast, provides a new relaxation algorithm which makes a major improvement on the convergence. The idea is based on the observation that the error of $\tilde{V}$ is composed of different spacial frequencies. The relaxation will be most efficient if the length scale of the error is comparable to the grid size. Therefore real space is discretized into multiple grids with grid size Δx, $2\Delta x$, $4\Delta x$, etc., with grid level from low to high. The relaxation is always carried out on a fast converging grid. If the relaxation on a certain level of grid becomes slow, the algorithm will switch to a higher level of grid (restriction). If the relaxation on a certain level of grid is fully converged, the algorithm will switch to a lower level of grid (interpolation). The relaxation stops when the lowest level of grid (original grid) is fully converged. Interested readers are referred to [31] for more details.

The fifth algorithm is the fast Fourier transform which is applicable to periodic boundary condition. The periodic boundary condition assumes that $V(0) = V(L)$ and $\rho(0) = \rho(L)$, and hence $V(x)$ and $\rho(x)$ can be expanded with the Fourier series

$$V(x) = \sum_{n=-\infty}^{\infty} \tilde{V}_n e^{\mathrm{i}k_n x}, \tag{A.222}$$

$$\rho(x) = \sum_{n=-\infty}^{\infty} \tilde{\rho}_n e^{\mathrm{i}k_n x}, \tag{A.223}$$

where $k_n = \frac{2\pi}{L} n$. Inserting Eqs. (A.222,A.223) into Eq. (A.217), the differential equation is converted to an algebraic equation and the solution is obtained as

$$\tilde{V}_n = \frac{4\pi}{k_n^2} \tilde{\rho}_n. \tag{A.224}$$

www.ebook3000.com

Notice that $n = 0$ corresponds to a uniform background, and $\tilde{\rho}_0$ must be zero due to the charge neutrality. As a result, $\tilde{V}_0 = \frac{0}{0}$ is underdetermined, corresponding to an arbitrary addible constant. For convenience, $\tilde{V}_0$ is also assigned to zero so that the average of the electrostatic potential is zero. $\tilde{\rho}_n$ in the RHS of Eq. (A.224) can be obtained by using the Fourier transform

$$\tilde{\rho}_n = \frac{1}{L}\int_0^L e^{-\mathrm{i}k_n x}\rho(x) \approx \sum_{i=0}^{N-1} e^{-\mathrm{i}k_n x_i}\rho_i, \tag{A.225}$$

where $x_i = i\frac{L}{N}$ is the 1d grid point. Once $\tilde{V}_n$ is solved from Eq. (A.224), V_i can be obtained by using the inverse Fourier transform

$$V_i = \sum_{n=-M_1}^{M_2} \tilde{V}_n e^{\mathrm{i}k_n x_i}, \tag{A.226}$$

where $M_1 = \left[\frac{N-1}{2}\right]$ and $M_2 = N - 1 - M_1$ are the truncation limits of the Fourier series. The summation in Eq. (A.225) and Eq. (A.226) can be evaluated with fast Fourier transform which has been implemented in the MATLAB functions *fft* and *ifft*. Notice that $\tilde{V}_n$ in Eq. (A.226) needs to be reordered to adapt to the definition of the MATLAB function *fft*.

A.23.3 *Algorithms in 2d and 3d cases*

Although the algorithms in the previous subsection are presented for the 1d case and for Dirichlet or periodic boundary condition, scopes of their applicability are much wider.

First of all, the algorithms can be generalized to the 2d and 3d cases or other types of boundary condition as long as the x, y, z dimensions are separable. The applicability of the algorithms is summarized as follows

algorithm	**boundary condition**	**dimensions**
direct method	all	1d, 2d, 3d
tridiagonal solver	all	1d, 2d, 3d
fast Poisson solver	Dirichlet	1d, 2d, 3d
multigrid method	all	1d, 2d, 3d
fast Fourier transform	periodic	1d, 2d, 3d

Secondly, the algorithms can be combined to handle more sophisticated types of boundary condition. For example, suppose the boundary is periodic in xy dimensions and Dirichlet type in z dimension. One can apply the

fast Fourier transform to xy dimensions and apply the tridiagonal solver to z dimension. For another example, suppose the boundary is periodic in x dimension and Dirichlet type in yz dimensions. One can apply the fast Fourier transform to x dimension and apply the fast Poisson solver to yz dimensions.

Finally, each algorithm has its own advantage and disadvantage. The fast Fourier transform is highly efficient for periodic boundary condition. The tridiagonal solver is highly efficient for solving 1d problem. The fast Poisson solver is highly efficient for Dirichlet boundary condition. The multigrid method is a general algorithm but more sophisticated than others. The direct method is also a general algorithm but more costly than others.

A.23.4 *Nonorthogonal Poisson box*

In the previous discussion, we have assumed that the Poisson box is orthogonal. In this subsection, we shall investigate the situation of nonorthogonal Poisson box.

Consider a general Poisson box whose edges are in the direction

$$\mathbf{e}_1 = \begin{pmatrix} e_{11} \\ e_{21} \\ e_{31} \end{pmatrix}, \quad \mathbf{e}_2 = \begin{pmatrix} e_{12} \\ e_{22} \\ e_{32} \end{pmatrix}, \quad \mathbf{e}_3 = \begin{pmatrix} e_{13} \\ e_{23} \\ e_{33} \end{pmatrix}.$$

The vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ have been normalized but not necessarily orthogonal to each other. Using $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ as basis vectors, one can establish an affine frame where the affine coordinates are denoted by x', y', z'. The transformation from the affine coordinates to the Cartesian coordinates are

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}, \tag{A.227}$$

where R is defined by

$$R = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix}.$$

To solve the Poisson equation, we need to discretize the nonorthogonal Poisson box with a uniform grid according to the affine coordinates. The major obstacle is that the Poisson equation in the affine coordinates has a different shape from that of the Cartesian coordinates. The derivation of

www.ebook3000.com

the Poisson equation in the affine coordinates is as follows. The coordinate transformation Eq. (A.227) leads to the derivatives transformation

$$\left(\partial_{x'}\ \partial_{y'}\ \partial_{z'}\right) = \left(\partial_x\ \partial_y\ \partial_z\right) R. \tag{A.228}$$

As a result, the Laplacian is transformed by

$$\begin{aligned}\nabla^2 &= \partial_x^2 + \partial_y^2 + \partial_z^2 \\ &= \left(\partial_x\ \partial_y\ \partial_z\right)\begin{pmatrix}\partial_x \\ \partial_y \\ \partial_z\end{pmatrix} \\ &= \left(\partial_{x'}\ \partial_{y'}\ \partial_{z'}\right) R^{-1}\left(R^{-1}\right)^T\begin{pmatrix}\partial_{x'} \\ \partial_{y'} \\ \partial_{z'}\end{pmatrix}.\end{aligned}$$

Therefore the Poisson equation in the affine coordinates is derived as

$$\left(\partial_{x'}\ \partial_{y'}\ \partial_{z'}\right) R^{-1}\left(R^{-1}\right)^T\begin{pmatrix}\partial_{x'} \\ \partial_{y'} \\ \partial_{z'}\end{pmatrix} V\left(x', y', z'\right) = -4\pi\rho\left(x', y', z'\right). \tag{A.229}$$

Thus the algorithms of the orthogonal Poisson box can be transferred to the nonorthogonal situation with the aid of affine coordinates.

A.23.5 *Nonlinear Poisson equation*

In previous subsections, we have assumed that $\rho(x, y, z)$ in the RHS of Eq. (A.213) is known. On some occasions, $\rho(x, y, z)$ in turn may depend on $V(x, y, z)$, leading to the nonlinear Poisson equation

$$\left(\partial_x^2 + \partial_y^2 + \partial_z^2\right) V(x, y, z) = -4\pi\rho\left[V(x, y, z)\right]. \tag{A.230}$$

The nonlinear term in the RHS may significantly accelerate the convergence of self-consistent calculations in continuous transport models [32]. In this subsection, we discuss the algorithm for solving the 1d nonlinear Poisson equation which can be easily generalized to the 2d and 3d cases.

Consider the 1d nonlinear Poisson equation

$$\partial_x^2 V(x) = -4\pi\rho\left[V(x)\right], \tag{A.231}$$

with the boundary condition $V(0) = \tilde{V}_A$ and $V(L) = \tilde{V}_B$. The nonlinear term is defined by

$$\rho\left[V(x)\right] = \rho_0 e^{-\beta\left[V(x) - \tilde{V}(x)\right]}, \tag{A.232}$$

where ρ_0, β, and $\tilde{V}(x)$ are known.

To solve a nonlinear equation

$$F(X) = 0, \tag{A.233}$$

one can make an initial guess X_0 and update the trial solution by the iteration

$$X_{n+1} = X_n - \left[\frac{\partial F}{\partial X}(X_n)\right]^{-1} F(X_n), \tag{A.234}$$

which is referred to as Newton's method. Notice that F and X in Newton's method can be either scalars or vectors.

Now we apply Newton's method to solve the 1d nonlinear Poisson equation. After the discretization of Eq. (A.231), the nonlinear function $F(V)$ corresponding to the 1d nonlinear Poisson equation can be explicitly written as

$$F(V) = D \cdot V + 4\pi\rho(V) + V_{BC}, \tag{A.235}$$

where D is an $(N-1) \times (N-1)$ matrix

$$D = \begin{pmatrix} \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & & \\ \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} & \frac{1}{(\Delta x)^2} \\ & & & \frac{1}{(\Delta x)^2} & \frac{-2}{(\Delta x)^2} \end{pmatrix},$$

and V, $\rho(V)$, V_{BC} are $(N-1) \times 1$ vectors

$$V = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_{N-2} \\ V_{N-1} \end{pmatrix}, \quad \rho = \begin{pmatrix} \rho(V_1) \\ \rho(V_2) \\ \vdots \\ \rho(V_{N-2}) \\ \rho(V_{N-1}) \end{pmatrix}, \quad V_{BC} = \begin{pmatrix} \frac{1}{(\Delta x)^2}\tilde{V}_A \\ 0 \\ \vdots \\ 0 \\ \frac{1}{(\Delta x)^2}\tilde{V}_B \end{pmatrix}.$$

The Jacobian $\frac{\partial F}{\partial V}$ is obtained as

$$\frac{\partial F}{\partial V} = D + diag\left[4\pi\rho'(V)\right], \tag{A.236}$$

where $\rho'(V_i) = -\beta\rho(V_i)$ is derived from Eq. (A.232).

Thus one can apply Newton's method Eq. (A.234) to solve the 1d nonlinear Poisson equation Eq. (A.231), where $F(V)$ and $\frac{\partial F}{\partial V}$ have been derived by Eq. (A.235) and Eq. (A.236) respectively.

www.ebook3000.com

A.24 Locality in nonequilibrium

Generally the locality principle does not hold in nonequilibrium. However we show in this section that the locality principle can be recovered in clean two-probe systems along the transport direction.

The central region of a two-probe system can be divided into many slices along the transport direction. In localized atomic basis, if the slices are sufficiently thick, the Hamiltonian of each slice only has nonzero overlap with two neighboring slices. Consider one particular slice C_0. By using the NEGF formalism, the density matrix of C_0 is

$$\rho_0 = -\mathrm{i}\left[G^<\right]_{00}, \tag{A.237}$$

where $[\cdots]_{00}$ means to take the diagonal block corresponding to C_0 subspace. $G^<$ is the lesser Green's function of the whole central region

$$G^< = G^r \Sigma^< G^a, \tag{A.238}$$

$$G^r = \left(h_C - \Sigma^r\right)^{-1}, \tag{A.239}$$

$$G^a = \left(G^r\right)^\dagger. \tag{A.240}$$

Here h_C is the reduced Hamiltonian of the central region. The definition of the reduced Hamiltonian depends on the basis type: In the orthogonal TB basis, $h \equiv E - H$; In the LCAO, $h \equiv ES - H$; In the LMTO, $h \equiv P(E) - S(k)$. Notice that Eqs. (A.238,A.239) involve full matrix operations of the size of h_C which are too costly for large atomic simulations.

To reduce the computational cost, we derive an alternative expression of $[G^<]_{00}$ which only involves the quantities of the neighboring slices. Suppose C_0 has the neighboring slice C_1 on the left and C_2 on the right. Regard C_0 as a new central region and take into account the remaining parts of the two-probe system by self-energies from C_1 and C_2

$$G^r_{00} = \left(h_{00} - \Sigma^r_1 - \Sigma^r_2\right)^{-1}, \tag{A.241}$$

$$\Sigma^r_1 = h_{01} \mathcal{G}^r_{11} h_{10}, \tag{A.242}$$

$$\Sigma^r_2 = h_{02} \mathcal{G}^r_{22} h_{20}, \tag{A.243}$$

where h_{00} is the reduced Hamiltonian of C_0, and $h_{01}, h_{10}, h_{02}, h_{20}$ are the reduced Hamiltonians between C_0 and C_1, C_2. $\mathcal{G}^r_{11}$ and $\mathcal{G}^r_{22}$ are the surface Green's functions of C_1 and C_2. The retarded Green's function in Eqs. (A.241,A.242,A.243) can be generalized to the contour ordered Green's function. By using the Langreth theorem (see Section 2.3), one can obtain

the lesser Green's function

$$G_{00}^{<} = G_{00}^{r}\left(\Sigma_1^{<} + \Sigma_2^{<}\right) G_{00}^{a}, \tag{A.244}$$

$$\Sigma_1^{<} = h_{01}\mathcal{G}_{11}^{<}h_{10}, \tag{A.245}$$

$$\Sigma_2^{<} = h_{02}\mathcal{G}_{22}^{<}h_{20}. \tag{A.246}$$

Notice that C_1 will be in equilibrium with the left lead if the two-probe system is decoupled between C_1 and C_0. Due to the fluctuation-dissipation theorem,

$$\mathcal{G}_{11}^{<} = f_L\left(\mathcal{G}_{11}^{a} - \mathcal{G}_{11}^{r}\right); \tag{A.247}$$

and similarly

$$\mathcal{G}_{22}^{<} = f_R\left(\mathcal{G}_{22}^{a} - \mathcal{G}_{22}^{r}\right). \tag{A.248}$$

Until now everything is exact. We are going to make an approximation to the surface Green's function $\mathcal{G}_{11}^{r}$ and $\mathcal{G}_{22}^{r}$. As long as C_1 and C_2 are sufficiently thick, C_0 can only "see" the two neighboring slices due to the nearsightedness. Consequently the surface Green's function $\mathcal{G}_{11}^{r}$ and $\mathcal{G}_{22}^{r}$ can be approximated by

$$\mathcal{G}_{11}^{r} \approx g_{11}^{r} = h_{11}^{-1}, \tag{A.249}$$

$$\mathcal{G}_{22}^{r} \approx g_{22}^{r} = h_{22}^{-1}. \tag{A.250}$$

where h_{11} and h_{22} are the reduced Hamiltonians of C_1 and C_2.

To sum up, Eqs. (A.241–A.250) indicate that the density matrix of a slice can be approximately obtained by the local Hamiltonians of the slice and its neighboring slices. Thus the locality principle is recovered in nonequilibrium two-probe systems along the transport direction.

A.25 Lanczos algorithm

This section discusses the Lanczos algorithm which is one of the best known Krylov subspace methods. The original Lanczos algorithm was developed for Hermitian matrices. Here the algorithm is generalized to the two-sided Lanczos algorithm [18] which is applicable to general matrices.

Suppose A is an $N \times N$ sparse matrix. The Lanczos algorithm constructs two sets of basis $\{\langle p_j|\}$ and $\{|q_j\rangle\}$ $(j = 1, 2, \ldots, M)$ in the Krylov subspace in order to tridiagonalize A. The basis $\{\langle p_j|\}$ and $\{|q_j\rangle\}$ are constructed by the following recursion: $\langle p_1|$ and $|q_1\rangle$ are $1 \times N$ row vector and $N \times 1$

www.ebook3000.com

column vector with random elements. The two vectors are normalized such that $\langle p_1|q_1\rangle = 1$. The subsequent vectors are generated by

$$\alpha_j = \langle p_j|A|q_j\rangle\,, \tag{A.251}$$

$$|\tilde{q}_{j+1}\rangle = A\,|q_j\rangle - \alpha_j\,|q_j\rangle - \beta_{j-1}\,|q_{j-1}\rangle\,, \tag{A.252}$$

$$\langle\tilde{p}_{j+1}| = \langle p_j|\,A - \alpha_j\,\langle p_j| - \gamma_{j-1}\,\langle p_{j-1}|\,, \tag{A.253}$$

$$\beta_j\gamma_j = \langle\tilde{p}_{j+1}|\tilde{q}_{j+1}\rangle\,, \tag{A.254}$$

$$|q_{j+1}\rangle = \frac{1}{\gamma_j}\,|\tilde{q}_{j+1}\rangle\,, \tag{A.255}$$

$$\langle p_{j+1}| = \frac{1}{\beta_j}\,\langle\tilde{p}_{j+1}|\,, \tag{A.256}$$

where $\beta_0 = \gamma_0 = 0$ and $j = 1, 2, \ldots, M-1$.

Lemma: $\{\langle p_j|\}$ and $\{|q_j\rangle\}$ defined by Eqs. (A.251–A.256) are mutually orthonormal, i.e.,

$$\langle p_i|q_j\rangle = \delta_{ij}. \tag{A.257}$$

Proof: The proof is carried out by using mathematical induction. For $i = j = 1$, $\langle p_1|q_1\rangle = 1$ by construction. Assume $\langle p_i|q_j\rangle = \delta_{ij}$ is true for $i, j \le k$, one needs to proved that $\langle p_i|q_j\rangle = \delta_{ij}$ is also true for $i, j \le k+1$. Three cases are to be examined, (a) $i = k+1$, $j = k+1$; (b) $i = k+1$, $j \le k$; (c) $i \le k$, $j = k+1$.

In case (a), $\langle p_{k+1}|q_{k+1}\rangle = 1$ due to Eqs. (A.254,A.255,A.256).

In case (b), $\langle p_{k+1}|q_j\rangle \sim \langle\tilde{p}_{k+1}|q_j\rangle$ due to Eq. (A.256), and hence one needs to prove $\langle\tilde{p}_{k+1}|q_j\rangle = 0$. By using Eq. (A.253), $\langle\tilde{p}_{k+1}|q_j\rangle$ is reduced to

$$\begin{aligned}\langle\tilde{p}_{k+1}|q_j\rangle &= (\langle p_k|\,A - \alpha_k\,\langle p_k| - \gamma_{k-1}\,\langle p_{k-1}|)\cdot|q_j\rangle \\ &= \langle p_k|\,A\,|q_j\rangle - \alpha_k\,\langle p_k|q_j\rangle - \gamma_{k-1}\,\langle p_{k-1}|q_j\rangle \\ &= \langle p_k|\,A\,|q_j\rangle - \alpha_k\delta_{kj} - \gamma_{k-1}\delta_{k-1,j}\ ,\end{aligned} \tag{A.258}$$

where $\langle p_k|q_j\rangle = \delta_{kj}$ and $\langle p_{k-1}|q_j\rangle = \delta_{k-1,j}$ are used in the derivation. By using Eq. (A.252), $\langle p_k|\,A\,|q_j\rangle$ can be reduced to

$$\begin{aligned}\langle p_k|A|q_j\rangle &= \langle p_k|\cdot(|\tilde{q}_{j+1}\rangle + \alpha_j\,|q_j\rangle + \beta_{j-1}\,|q_{j-1}\rangle) \\ &= \langle p_k|\cdot(\gamma_j\,|q_{j+1}\rangle + \alpha_j\,|q_j\rangle + \beta_{j-1}\,|q_{j-1}\rangle) \\ &= \gamma_j\delta_{k,j+1} + \alpha_j\delta_{kj}\ ,\end{aligned} \tag{A.259}$$

where $\langle p_k|q_{j+1}\rangle = \delta_{k,j+1}$, $\langle p_k|q_j\rangle = \delta_{kj}$, and $\langle p_k|q_{j-1}\rangle = 0$ $(j \le k)$ are used in the derivation. Inserting Eq. (A.259) into Eq. (A.258), one obtains $\langle\tilde{p}_{k+1}|q_j\rangle = 0$.

In case (c), $\langle p_j|q_{k+1}\rangle \sim \langle p_j|\tilde{q}_{k+1}\rangle$ due to Eq. (A.255), and one can prove $\langle p_j|\tilde{q}_{k+1}\rangle = 0$ in analogous to case (b). QED.

Theorem: The matrix A is tridiagonalized in the orthonormal dual basis $\{\langle p_j|\}$ and $\{|q_j\rangle\}$, i.e.,

$$PQ = 1, \tag{A.260}$$

$$PAQ = T, \tag{A.261}$$

where P is an $M \times N$ full matrix, Q is an $N \times M$ full matrix, and T is a tridiagonal matrix

$$P \equiv \begin{pmatrix} \langle p_1| \\ \langle p_2| \\ \cdots \\ \langle p_M| \end{pmatrix},$$

$$Q \equiv \left(|q_1\rangle\ |q_2\rangle \cdots |q_M\rangle\right),$$

$$T \equiv \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{M-1} \\ & & \beta_{M-1} & \alpha_M \end{pmatrix}.$$

Proof: Eq. (A.260) is equivalent to Eq. (A.257).

Eq. (A.261) is derived by using Eqs. (A.252,A.255)

$$\begin{aligned} A\,|q_j\rangle &= |\tilde{q}_{j+1}\rangle + \alpha_j\,|q_j\rangle + \beta_{j-1}\,|q_{j-1}\rangle \\ &= \gamma_j\,|q_{j+1}\rangle + \alpha_j\,|q_j\rangle + \beta_{j-1}\,|q_{j-1}\rangle\,. \end{aligned} \tag{A.262}$$

The matrix element $\langle p_i|\,A\,|q_j\rangle$ is obtained by applying Eq. (A.257) to Eq. (A.262). QED.

A few comments are in order. (1) The coefficients $\{\beta_j\}$ and $\{\gamma_j\}$ are not uniquely determined by Eq. (A.254). A possible choice is $\beta_j = \gamma_j = \sqrt{\langle\tilde{p}_{j+1}|\tilde{q}_{j+1}\rangle}$. (2) If A is Hermitian, $A = A^\dagger$, the two basis sets are reduced to one basis set where $\langle p_j| = (|q_j\rangle)^\dagger$. If A is symmetric, $A = A^T$, the two basis sets are also reduced to one basis set where $\langle p_j| = (|q_j\rangle)^T$. (3) The Lanczos algorithm can be generalized to the block Lanczos algorithm in which $\langle p_j|$ and $|q_j\rangle$ are $l \times N$ and $N \times l$ block vectors and the coefficients $\alpha_j, \beta_j, \gamma_j$ are $l \times l$ matrix blocks [33, 34]. (4) The Lanczos algorithm may suffer from the round-off error which is called the Lanczos disease. As a treatment, it is necessary to re-orthogonalize the basis sets or restart the Lanczos recursion once the orthogonality deteriorates.

www.ebook3000.com

Finally we would like to discuss two applications of the Lanczos algorithm. The first application is to solve for a few of the largest eigenvalues of a sparse Hermitian matrix H. To proceed one needs to tridiagonalize H by using the Lanczos algorithm to obtain $\tilde{H} = U^\dagger H U$, where $\tilde{H}$ is an $m \times m$ tridiagonal matrix and $U^\dagger U = 1$. The first m largest eigenvalues (in magnitude) of H can be approximated by those of $\tilde{H}$. The second application is to solve the sparse linear equation $Ax = b$. To proceed one needs to tridiagonalize A by using the Lanczos algorithm to obtain $PAQ = T$. As a result, A^{-1} is approximated by $QT^{-1}P$, and the solution to the linear equation is obtained as

$$x = A^{-1}b \approx \left(QT^{-1}P\right) b. \tag{A.263}$$

The Lanczos algorithm together with Eq. (A.263) can be rewritten to an iterative algorithm for solving x (see Ref. [18])

$$\alpha_{j-1} = \frac{\langle \tilde{r}_{j-1} | r_{j-1} \rangle}{\langle \tilde{p}_{j-1} | A | p_{j-1} \rangle}, \tag{A.264}$$

$$|x_j\rangle = |x_{j-1}\rangle + \alpha_{j-1} |p_{j-1}\rangle, \tag{A.265}$$

$$|r_j\rangle = |r_{j-1}\rangle - \alpha_{j-1} A |p_{j-1}\rangle, \tag{A.266}$$

$$\langle \tilde{r}_j | = \langle \tilde{r}_{j-1} | - \alpha_{j-1} \langle \tilde{p}_{j-1} | A^\dagger, \tag{A.267}$$

$$\beta_{j-1} = \frac{\langle \tilde{r}_j | r_j \rangle}{\langle \tilde{r}_{j-1} | r_{j-1} \rangle}, \tag{A.268}$$

$$|p_j\rangle = |r_j\rangle + \beta_{j-1} |p_{j-1}\rangle, \tag{A.269}$$

$$\langle \tilde{p}_j | = \langle \tilde{r}_j | + \beta_{j-1} \langle \tilde{p}_{j-1} |, \tag{A.270}$$

where $j = 1, 2, \ldots, M$. The initial values are $|r_0\rangle = |p_0\rangle = |b\rangle - A|x_0\rangle$, $\langle \tilde{p}_0 | = \langle \tilde{r}_0 |$, where $|x_0\rangle$ and $\langle \tilde{r}_0 |$ are some random vectors with the constraint that $\langle \tilde{r}_0 | r_0 \rangle \neq 0$. It is worth mentioning that the iterative algorithm defined by Eqs. (A.264–A.270) is nothing but the biconjugate gradient (bicg) iterative algorithm.

To sum up, we have derived the two-sided Lanczos algorithm in Eqs. (A.251–A.256) to generate mutually orthonormal basis sets. With the mutually orthonormal basis sets, a sparse matrix can be tridiagonalized with low computational cost, leading to iterative algorithms for solving eigenvalues and linear equations.

A.26 Preconditioner designed for quantum transport

Although some preconditioners are available in mathematical textbooks on iterative methods, there is no universal preconditioner which works for all linear systems. The construction of a preconditioner highly depends on the properties of the sparse matrices involved in a specific problem. In this section, we attempt to design a preconditioner for quantum transport.

One can see in Fig. 7.2 and Fig. A.25 that Green's function converges smoothly at the complex energies far away from the real axis. It becomes more and more difficult when the energy gets closer to the singularities on the real axis. On the other hand, real energies are inevitable in the calculations of transmission coefficient as well as nonequilibrium density matrix. To solve the problem, we attempt to "rescue" the bad point located near the real axis with a series of good points off the real axis. At first the Green's function is solved far away from the real axis with the iterative method, and the solution is used as a preconditioner to solve the Green's function closer to the real axis. The new solution can be used as a new preconditioner to solve the Green's function even closer to the real axis. The multi-level preconditioning continues until the end point is sufficiently close to the real axis (see Fig. A.25). The preconditioning chain will be referred to as the rescue ladder of preconditioners (RLPC).

Next we analyze the efficiency of the RLPC. We shall show that the condition number $\kappa \leqslant \eta \equiv \frac{\alpha_2}{\alpha_1}$ if the Green's function at $z_2 = E+\mathrm{i}\alpha_2$ is used as a preconditioner to "rescue" the Green's function at $z_1 = E+\mathrm{i}\alpha_1$, where E is a real energy and $\alpha_2 > \alpha_1 > 0$. The discussion will proceed in two steps, the two-probe systems without lead self-energies and the two-probe systems with lead self-energies.

Firstly, we focus on the isolated central region and ignore the leads for a while. The retarded Green's function is defined by $G^r(z) = h^{-1}(z)$ with $\mathrm{Im}(z) > 0$, where $h(z)$ is the reduced Hamiltonian (see Eqs. (7.8,7.9,7.10)). The condition number of the preconditioned linear system is determined by the eigenvalues of $h^{-1}(z_2)\, h(z_1)$.

Lemma: All the eigenvalues of $h^{-1}(z_2)\, h(z_1)$ are distributed on the circle $|z - z_0| = r_0$ where z_0 and r_0 are defined by

$$z_0 = \frac{1+\eta^{-1}}{2}, \tag{A.271}$$

$$r_0 = \frac{1-\eta^{-1}}{2}. \tag{A.272}$$

www.ebook3000.com

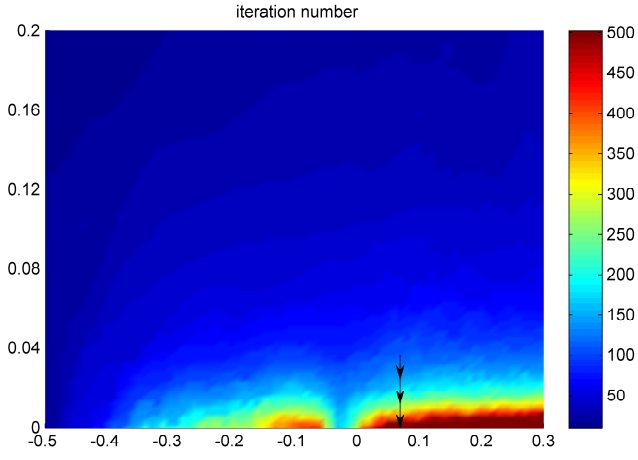


Fig. A.25 The color map of the iteration number on the complex energy plane. The retarded Green's function is solved with the iterative method for a Silicon cluster composed of 264 Si atoms and 238 empty spheres using the LMTO method. The arrows indicate the RLPC from the easy-to-converge region to the difficult-to-converge region.

Proof: (1) In the orthogonal TB basis, the reduced Hamiltonian is $h(z) = z - H$. The eigensolutions of H are obtained as

$$H\Psi = \Psi\Lambda, \tag{A.273}$$

or equivalently

$$H = \Psi\Lambda\Psi^{-1}, \tag{A.274}$$

where $\Lambda = diag([\lambda_1, \lambda_2, \ldots, \lambda_n])$ is the eigenvalue matrix and $\Psi = (|\phi_1\rangle, |\phi_2\rangle, \ldots, |\phi_n\rangle)$ is the eigenvector matrix. The reduced Hamiltonian $h(z)$ can be decomposed by the same eigenvectors

$$h(z) = \Psi(z - \Lambda)\Psi^{-1}. \tag{A.275}$$

Consequently $h^{-1}(z_2)h(z_1)$ is reduced to

$$h^{-1}(z_2)h(z_1) = \Psi\frac{z_1 - \Lambda}{z_2 - \Lambda}\Psi^{-1}, \tag{A.276}$$

indicating that the eigenvalues of $h^{-1}(z_2)h(z_1)$ are in the form of $\xi_i = \frac{z_1-\lambda_i}{z_2-\lambda_i}$. By using $z_1 = E+\mathrm{i}\alpha_1$ and $z_2 = E+\mathrm{i}\alpha_2$, it is straightforward to verify that $|\xi_i - z_0| = r_0$ where z_0 and r_0 are defined by Eqs. (A.271,A.272).

(2) In the LCAO basis, the reduced Hamiltonian is $h(z) = zS - H$. The eigensolutions of H and S are obtained as

$$H\Psi = S\Psi\Lambda, \tag{A.277}$$

or equivalently

$$H = S\Psi\Lambda\Psi^{-1}. \tag{A.278}$$

The reduced Hamiltonian $h(z)$ can be decomposed by the same eigenvectors

$$h(z) = S\Psi(z - \Lambda)\Psi^{-1}. \tag{A.279}$$

Consequently $h^{-1}(z_2)\, h(z_1)$ is reduced to

$$h^{-1}(z_2)\, h(z_1) = \Psi\frac{z_1 - \Lambda}{z_2 - \Lambda}\Psi^{-1}. \tag{A.280}$$

The remaining discussion is the same as case (1).

(3) In the LMTO basis, the reduced Hamiltonian is $h(z) = P(z) - S$. Notice that $h(z)$ is related to $z - H_{orth}$ by Eq. (3.34), where H_{orth} is the LMTO orthogonal Hamiltonian defined by Eq. (3.36). The eigensolutions of H_{orth} are obtained as

$$H_{orth}\Psi = \Psi\Lambda, \tag{A.281}$$

or equivalently

$$H_{orth} = \Psi\Lambda\Psi^{-1}. \tag{A.282}$$

By using Eq. (3.34), the reduced Hamiltonian $h(z)$ can be decomposed by the same eigenvectors

$$h(z) = \frac{1}{\Delta + \gamma(z - C)}\sqrt{\Delta}\Psi(z - \Lambda)\Psi^{-1}\frac{1}{\sqrt{\Delta}}(1 - \gamma S). \tag{A.283}$$

Consequently $h^{-1}(z_2)\, h(z_1)$ is reduced to

$$\begin{aligned} h^{-1}(z_2)\, h(z_1) &\approx (1 - \gamma S)^{-1}\sqrt{\Delta}\Psi\frac{z_1 - \Lambda}{z_2 - \Lambda}\Psi^{-1}\frac{1}{\sqrt{\Delta}}(1 - \gamma S), \\ &= \Phi\frac{z_1 - \Lambda}{z_2 - \Lambda}\Phi^{-1}. \end{aligned} \tag{A.284}$$

where $\Phi \equiv (1 - \gamma S)^{-1}\sqrt{\Delta}\Psi$. In the derivation, it is assumed that $\Delta + \gamma(z_1 - C) \approx \Delta + \gamma(z_2 - C)$ so that the two prefactors cancel with each other. The remaining discussion is the same as case (1).

QED.

www.ebook3000.com

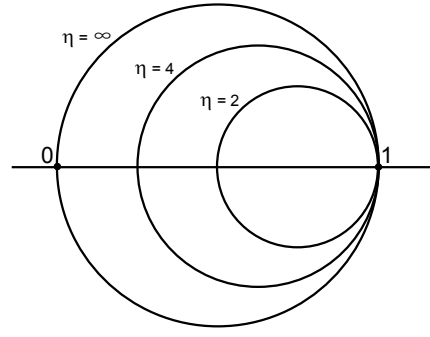


Fig. A.26 The eigenvalues of $h^{-1}(z_2)\,h(z_1)$ are distributed on or within the circle defined by $|z - z_0| = r_0$. The center and radius of the circle is determined by the ratio $\eta \equiv \frac{\alpha_2}{\alpha_1}$, where $\alpha_2 > \alpha_1 > 0$ are the imaginary parts of z_2 and z_1.

Secondly, we investigate the impact of the leads on the eigenvalues of $h(z)$. The lead effects can be taken into account by self-energies and will be treated as a perturbation. Assume that λ_i and $|\phi_i\rangle$ are the eigenvalue and eigenvector of the isolated central region respectively. By including the lead self-energies, the first order correction to the eigenvalue is

$$\delta = \frac{\langle\phi_i|\,\Sigma^r(E)\,|\phi_i\rangle}{\langle\phi_i|\phi_i\rangle}. \tag{A.285}$$

The imaginary part of δ is

$$\begin{aligned}
\mathrm{Im}\delta &= -\frac{\mathrm{i}}{2}(\delta - \delta^*) \\
&= -\frac{\mathrm{i}}{2}\frac{\langle\phi_i|\,[\Sigma^r(E) - \Sigma^a(E)]\,|\phi_i\rangle}{\langle\phi_i|\phi_i\rangle} \\
&= -\frac{1}{2}\frac{\langle\phi_i|\,\Gamma(E)\,|\phi_i\rangle}{\langle\phi_i|\phi_i\rangle},
\end{aligned} \tag{A.286}$$

where $\Gamma(E) = -\mathrm{i}[\Sigma^a(E) - \Sigma^r(E)]$ is the linewidth function. Since $\Gamma(E)$ is always positive-definite, it is concluded that $\mathrm{Im}\delta < 0$. As a consequence, the eigenvalues of $h^{-1}(z_2)\,h(z_1)$ are modified from $\xi_i = \frac{z_1-\lambda_i}{z_2-\lambda_i}$ to $\tilde{\xi}_i = \frac{z_1-(\lambda_i+\delta_i)}{z_2-(\lambda_i+\delta_i)}$ where δ_i is the correction induced by the lead self-energies. It is straightforward to verify that $\left|\tilde{\xi}_i - z_0\right| < r_0$ provided $\mathrm{Im}\delta_i < 0$. In other words, all the eigenvalues of $h^{-1}(z_2)\,h(z_1)$ are distributed within the circle $|z - z_0| = r_0$ in two-probe systems. The results are illustrated in Fig. A.26.

Finally, we estimate the condition number and the convergence rate. The condition number is determined by the ratio between the largest and the smallest eigenvalue module

$$\kappa \equiv \frac{|\xi_{\max}|}{|\xi_{\min}|} \leqslant \frac{1}{1-2r_0} = \eta. \tag{A.287}$$

The convergence rate at the m^{th} iteration step is estimated by $\left(1-\frac{2}{\kappa}\right)^m$ where $\kappa \gg 1$. These are the results of a single rescue step. The RLPC is composed of multiple rescue steps extending from the blue region all the way to the red region in Fig. A.25. What is the optimal arrangement of the preconditioners? If the ratio η is too small, many rescue steps will be needed to reach the real axis and the computational cost can be too high. If the ratio η is too large, many iterations will be needed in a single rescue step and the iterative method may suffer from round-off error. So the optimal arrangement of RLPC is to make the ratio η as large as possible before the outbreak of the Lanczos disease, which may depend on the numerical accuracy, the system size, and the material properties.

A.27 Content of the affiliated CD

The affiliated CD has two packages: the NanoDsim package and the ResearchCode package.

A.27.1 *NanoDsim package*

The NanoDsim package provides MATLAB source code of NanoDsim as well as 16 examples of atomistic simulations using NanoDsim. Please read the user's manual, *nanodsim_manual.pdf*, for the instruction of software installation. Note that the NanoDsim package is distributed under the terms of the GNU General Public License.

A.27.2 *ResearchCode package*

The ResearchCode package contains a number of research codes to produce some of the results in the monograph. The content of the ResearchCode package is summarized in the following table.

www.ebook3000.com

folder	section	description
chapter1/DoubleBarrier	Section 1.1	solve double-barrier model
chapter1/AtomSolver	Section 1.2	solve single atom with DFT
chapter1/Bulk1d_CPA	Section 1.3	calculate disorder average with CPA
chapter2/ABring_dephasing	Section 2.9	study dephasing in an AB-ring
chapter2/OhmLaw_dephasing	Section 2.9	study dephasing in a tight-binding chain
chapter2/Eigen_TwoProbe	Section 2.10	eigenvalues of a two-probe system
chapter2/NECPA	Section 2.10	NECPA method vs supercell method
chapter3/ScreeningTransform	Section 3.7	test screening transform in BCC
chapter4/matlab_examples	Section 4.1	ten examples of MATLAB usage
chapter4/vectorization	Section 4.2	two examples of vectorization
chapter4/hybrid_programming	Section 4.3	three examples of hybrid programming
chapter4/class_rational	Section 4.4	class of rational numbers
chapter4/class_shape	Section 4.4	class of colorful shapes
chapter5/Ewald_vs_Poisson	Section 5.4	Ewald sum vs Poisson equation
chapter5/LogDerivative	Section 5.5	study logarithmic derivative
chapter5/ContourIntegral	Section 5.6	complex contour vs real axis
chapter5/CPA_two_site	Section 5.7	study the two-site model with CPA
chapter6/Ewald_surface	Section 6.3	test jellium correction
chapter6/TRS_neqb	Section 6.6	check TRS in nonequilibrium
chapter6/TRS_trans	Section 6.6	check TRS in disordered system
chapter6/NECPA_two_site	Section 6.7	study the two-site model with NECPA
chapter6/TwoProbe_CuCuCu	Section 6.10	bulk system vs two-probe system
chapter6/TwoProbe_CuVacCu	Section 6.10	test nonequilibrium voltage drop
chapter7/IterationNumber_map	Section 7.3	plot map of iteration number
chapter7/GaussSum_parallel	Section 7.5	an example of parallelization
chapter8/BulkCu	Section 8.1	study bulk Cu
chapter8/BulkFe	Section 8.1	study bulk Fe
chapter8/BulkCo	Section 8.1	study bulk Co
chapter8/BulkNi	Section 8.1	study bulk Ni
chapter8/BulkCuCo	Section 8.2	study bulk Cu/Co alloy
chapter8/BulkSi	Section 8.3	study bulk Si
chapter8/TwoProbeCuCo	Section 8.5	study two-probe Cu/Co interface
appendix/ToyModel	Section A.2	study the phase diagram of the toy model
appendix/EM_vs_DD	Section A.3	effective mass model vs drift-diffusion model
appendix/NumericalProof	Section A.14	verify an identify numerically
appendix/specular_vs_diffusive	Section A.16	specular transmission vs diffusive transmission
appendix/MixingAlgorithm	Section A.20	Anderson mixer vs linear mixer
appendix/FermiPoleSum	Section A.21	study modified Fermi pole sum
appendix/Lanczos_and_CG	Section A.25	study Lanczos algorithm and CG algorithm

Bibliography

[1] Y. Taur, T. H. Ning, *Fundamentals of Modern VLSI Devices* (Cambridge University Press, New York, 2009).

[2] Throughout the monograph, electrons are assumed to have a positive charge for convenience. To make the physical observables consistent with the common convention of charge polarity, the factor $Q_e = -1$ is multiplied to current and voltage.

[3] J. S. Blakemore, Solid-State Electronics **25**, 1067 (1982).

[4] M. S. Lundstrom, D. A. Antoniadis, IEEE Trans. Electron Devices, **61**, 225 (2014).

[5] L. Zhang, M. Chan, *A Common Platform and Standard for Compact Model Development*, presentation at the SMEE meeting (The University of Hong Kong, Hong Kong, 2013).
[6] Y. Zhu, T. H. Lin, Q. F. Sun, Commun. Theor. Phys. **40**, 369 (2003).
[7] Y. Zhu, L. Liu, and H. Guo, Phys. Rev. B **88**, 085420 (2013).
[8] B. Velický, Phys. Rev. **184**, 614 (1969).
[9] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
[10] J. P. Perdew, K. Burke, andM. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[11] J. P. Perdew, Y. Wang, Phys. Rev. B **45**, 13244 (1992).
[12] F. Tran and P. Blaha, Phys. Rev. Lett. **102**, 226401 (2009).
[13] A. D. Becke, M. R. Roussel, Phys. Rev. A **39**, 3761 (1989).
[14] A. H. MacDonald, S. H. Vosko, J. Phys. C: Solid State Phys. **12**, 2977 (1979).
[15] http://tddft.org/programs/octopus/wiki/index.php/Libxc.
[16] http://en.wikipedia.org/wiki/Table_of_spherical_harmonics.
[17] http://mathworld.wolfram.com/Wigner3j-Symbol.html.
[18] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
[19] Y. Ke, K. Xia, H. Guo, Phys. Rev. Lett. **105**, 236801 (2010).
[20] V. Eyert, J. Comp. Phys. **124**, 271 (1996).
[21] G. P. Srivastava, J. Phys. A: Math. Gen. **17**, L317 (1984). Notice that the last "+" in the first line of Eq. (16) should be "−", see G. P. Srivastava, J. Phys. A: Math. Gen. **17**, 2737 (1984).
[22] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).
[23] L. D. Marks, D. R. Luke, Phys. Rev. B **75**, 075114 (2008).
[24] G. P. Kerker, Phys. Rev. B **23**, 3082 (1981).
[25] T. Ozaki, K. Nishio, H. Kino, Phys. Rev. B **81**, 035116 (2010).
[26] T. Ozaki, Phys. Rev. B **75**, 035123 (2007).
[27] Y. Zhu, L. Liu, and H. Guo, unpublished (2012).
[28] http://en.wikipedia.org/wiki/Finite_difference_coefficient.
[29] S. Duff, A. M. Erisman, J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, New York, 1986.
[30] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, London, 1996.
[31] A. Brandt, Mathematics of Computation, **31**, 333 (1977).
[32] Z. Ren, Ph. D. thesis, Purdue University (2001).
[33] J. Inoue and Y. Ohta, J. Phys. C: Solid State Phys. **20**, 1947 (1987).
[34] Z. Bai, D. Day, and Q. Ye, SIAM J. Matrix Anal. Appl. **20**, 1060 (1999).

www.ebook3000.com