Numerical Partial Differential Equations for Environmental Scientists and Engineers

A First Practical Course

Daniel R. Lynch

Springer

NUMERICAL PARTIAL DIFFERENTIAL EQUATIONS FOR ENVIRONMENTAL SCIENTISTS AND ENGINEERS

A First Practical Course

NUMERICAL PARTIAL DIFFERENTIAL EQUATIONS FOR ENVIRONMENTAL SCIENTISTS AND ENGINEERS

A First Practical Course

by

Daniel R. Lynch
Dartmouth College
Dartmouth, New Hampshire
USA

Springer

Library of Congress Cataloging-in-Publication Data

Lynch, Daniel R.
Numerical partial differential equations for environmental scientists and engineers : a first practical course / by Daniel R. Lynch.
p. cm.
Includes bibliographical references and index.
ISBN 0-387-23619-8 (alk. paper)
1. Differential equations, Partial—Numerical solutions. 2. Finite differences. 3. Finite element method. 4. Inverse problems (Differential equations) I. Title.

QA374.L96 2005
518'.64—dc22

2004059140

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11055716

springeronline.com

Contents

List of Tables

Preface

This book concerns the practical solution of Partial Differential Equations. We assume the reader knows what a PDE is – that he or she has derived some, and solved them with the limited but powerful arsenal of analytic techniques. We also assume that (s)he has gained some intuitive knowledge of their solution properties, either in the context of specific applications, or in the more abstract context of applied mathematics. We assume the reader now wants to solve PDE's for real, in the context of practical problems with all of their warts – awkward geometry, driven by real data, variable coefficients, nonlinearities – as they arise in real situations. The applications we envision span classical mathematical physics and the "engineering sciences": fluid mechanics, solid mechanics, electricity and magnetism, heat and mass transfer, wave propagation. Of course, these all share a joyous interdisciplinary unity in PDE's.

The material arises from lectures at Dartmouth College for first-year graduate students in science and engineering. That audience has shared the above motivations, and a mathematical background including: ordinary and partial differential equations; a first course in numerical analysis; linear algebra; complex numbers at least at the level of Fourier analysis; and an ability to program modern computers. Some working exposure to applications of PDE's in their research or practice has also been a common denominator. This classical undergraduate preparation sets the stage for our "First Practical Course".

Naturally, the "practical" aspect of the course involves computation. The bottom-line answer sought needs to be computed and that means an approximation to the PDE solution is inevitable. Accordingly we try to systematically expose the processes of discretization from continuum to algebra, and describe useful algorithms for assembling and solving the algebraic equations on finite machines. There is a standard triad of concerns: *accuracy* of the discrete solution; its *stability* relative to data and computational noise; and the *economy* of algorithms for computing it.

We have inherited a wonderful archive of knowledge about PDE's from the persons whose names are attached to them, all largely a pre-20^{th}C creation. In the first half of the 20^{th}C we have the addition of detailed developments in the Engineering Sciences mentioned above, where significant theoretical advances have resulted from the posing of specific natural phenomena in terms of PDE's. The significant recent development is the unprecedented growth in computing machinery – both in terms of availability and power. This is a capability which our predecessors could hardly begin to imagine; the modern frontier of their work is the generation of practical solutions to practical problems, and the understanding of the goodness of those solutions. That is the subject of our work together in this book.

So the practical aspect of this book is the infused focus on computation. We present two major discretization methods – Finite Difference and Finite Element. The blend of theory, analysis, and implementation practicality supports solving and understanding complicated problems at the

journeyman scientific level typical of early graduate students. Reading that material alone is boring and unrewarding without the culminating experience of computing with these methods. So these lectures need to be supplemented with computational exercises. At Dartmouth we have used a blend of Fortran (the Latin of the business) and Matlab; course software is normally constructed by students, with internet access to important libraries. Problems are drawn from students' particular disciplines; practical experience with in-depth examples is essential. The text material here is neutral with respect to computer language and operating system.

We interpret the *Environmental* target audience in the large. It is not Environmental technology which we address; that is a regulatory category. Rather we concentrate on natural *Environmental Phenomena*, occurring in the hydrosphere, atmosphere, cryosphere, lithosphere, biosphere, and ionosphere. Partial Differential Equations and the underlying Field Theory remain the descriptive language for phenomena occurring in these media. Mathematically, PDE's unite these; computational approaches should, also. Accordingly, this text reflects that fundamentally interdisciplinary approach to these problems.

There are three parts. Part I is a brief overview of Finite Difference ideas. I find that almost every student has learned much of this already, although not systematically, as FD methods are intuitive and now permeate undergraduate courses. So the material in this part is background, review, systemization. There are very good treatises on this subject, notably Smith [102], Morton and Mayers [87], and Ames [2].

The Part I material is introductory to the Finite Element Method, which involves more abstraction to get started. Here we provide in Part II a FEM tutorial. Many conventional expositions pose the FEM using the variational calculus; that is beyond the reach of many entry-level graduate students who want to solve PDE's. And, it often puts the method into a discipline-specific context, while our goal is fundamentally interdisciplinary. Accordingly, we pose the FEM as a solution method for general PDE's and utilize the Method of Weighted Residuals as a discretization principle. That is quite general, intuitive, and provides a simple link to Finite Differences. The final chapter of Part II summarizes some Numerical Analysis ideas focused on difference equations emanating from the FEM. With small adaptations, this chapter is applicable to FD discretizations also.

There are numerous works on the FEM, as there are for FDM. Segerlind [101] is a useful exposition which I have used. The works of Davies [29] and Johnson [49] provide interdisciplinary coverage, although they are largely variational in approach. The landmark volumes are from Zienkiewicz and colleagues, *e.g.* [120], [122]. An unusual synthesis of FD and FE methods is available in Lapidus and Pinder [52] which is recommended as a reference supplementing this exposition.

It is increasingly evident that practical problems are improperly posed, in the sense that one never really has completely unambiguous specifications of the necessary data. This is especially true in Environmental media. So Part III is an introduction to formal approaches for dealing with that. Students who know Linear Regression will recognize that approach as our baseline in the canonical problem of "fitting model to data" – the Inverse Problem. In this Part we rely on the notion of "model" as a FD or FE algebraic statement which is trustworthy if its data were known – the "Forward Problem" is solved, practically speaking. There is a tiny amount of probability/statistics involved here; but for our purposes it is not necessary to go beyond what a mature student should be comfortable with (*e.g.* moments of distributions). The solution of Inverse Problems is clearly at the forefront in a great many applications today and represents a major intellectual frontier.

Because of their centrality in the use of PDE's for research and design; because they pose massive increases in computer resources over the "forward" problems which they embed; and because of the massive computational power which is emerging – the time for Inverse Problems has come.

The full text as recorded here, accompanied by weekly in-depth computational exercises, could occupy a student for about 1.5 course-equivalents. Cutting the FD material based on students' prior exposure makes a 1-course offering possible for the balance of the text. An obvious addition might be a systematic coverage of modern linear algebra – we have used the excellent text by Golub and VanLoan [35] for this purpose along with LAPACK [3] exercises. Also useful in this area is Trefethen and Bau [105]. As always, the reader is recommended to have the Numerical Recipes [99] at hand; and to have recently acquired a good foundation in Numerical Analysis as in Burden and Faires [20].

Over the years I have benefited greatly from the help of colleagues and students, whose insights perfuse this work. Notable are F. Werner, J. Sullivan, K. Paulsen, and K. ONeill, who helped me start the Numerical Methods Laboratory at Dartmouth. Additional important colleagues in this work have included M. Foreman, D. Greenberg, D. McGillicuddy, C. Hannah, C. Naimie, J. Ip, W. Brown, A. Bilgili, K. Smith, J. Manning, S. Geimer, J. Waugh, M. George, K. Lunn, and N. Soni. The National Science Foundation generously supported all of the applications described.

Daniel R. Lynch
Hanover, NH August 2004.

Synopsis

Part I: The Finite Difference Method

1) Introduction

In this chapter we inspire the consideration of discrete approximants to PDE's as natural "engineering" approaches to describing common systems. The classical expositions in engineering science proceed from these to the continuum PDE's; we seek instead discrete representations of these systems which are arbitrarily close to the limiting continuum; and ways to demonstrate and quantify that. A classification of boundary conditions is given for the standard trio of initial and boundary conditions: Dirichlet, Neumann, Mixed. The chapter then introduces the "big three" canonical PDE's: Laplace, Diffusion, and Wave equations; and their classification in terms of characteristics. The discussion ends with the important practical question of necessary and sufficient boundary conditions and initial conditions for these equations.

2) Finite Difference Calculus

In this chapter we present and review standard finite difference approximations - how to generate them via Taylor Series and also by fitting and differentiating polynomials; and how to estimate their accuracy. General, nonuniform mesh expressions are obtained by example; the standard expressions and their error terms are derived for uniform meshes; higher order expressions are also derived.

3) Elliptic Equations

In this chapter we demonstrate the construction of discrete systems for elliptic PDE's. There is an emphasis on second-order, compact schemes and their Boundary Conditions. First a 1-D example is studied - a 2-point boundary value problem. Its matrix representation is displayed with various BC's. The standard BC's are exercised - Dirichlet, Neumann, Mixed. Then 2-D approximants are discussed. There is attention paid to banded (tridiagonal in the 1D case) matrix solution methods here, and the use of LU decomposition as a direct solution strategy. Conventional iterative solvers are briefly introduced and reviewed, with emphasis on review of the Jacobi, Gauss-Seidel, and SOR methods. Storage and run-time demands for direct and iterative approaches are compared.

The chapter concludes with a 1-D discussion of the steady-state advective-diffusion equation, with focus on the Peclet number as the determinant of quality. Exact solution of the difference equations for upstream, downstream, and centered approximations are studied.

4) Iterative Methods for Elliptic Equations

In this chapter we give an overview of iterative processes for solving linear systems of equations.

The concepts of spectral radius and iterative convergence rate are introduced. The familiar point and block iterative methods are reviewed in this context and the available relations among mesh size and spectral radius are summarized. The (Elliptic) Helmholtz problem is introduced as the useful Fourier Transform of the Hyperbolic wave equation. Its loss of diagonal dominance is demonstrated.

The chapter concludes with a discussion of gradient descent iterative methods, particularly in the context of 3-D systems wherein direct strategies lose their appeal. The familiar Jacobi and Gauss-Seidel methods are described in these terms. The Conjugate Gradient family is introduced; the reader is referred to more specific treatises for details.

5) Parabolic Equations

The 1-D diffusion equation (x,t) is used as the canonical equation here. A nomenclature is introduced which is used throughout the book:

- Distributed (Continuous) System: the PDE;
- Lumped System: the elliptic dimensions (x,y,z) discretized, the time dimension left continuous. This is a finite system of coupled ODE's, with a lot of structure;
- Discrete System: the time-discretization of the Lumped System.

The diffusion equation is Lumped by the finite difference method and its dispersion relation derived. Enforcement of boundary conditions is described, as a straightforward extension of the Elliptic material above. Three discrete systems are shown: Euler, Leapfrog, Backward Euler, and 2-level implicit. The standard trio of concerns – stability, consistency, convergence - is introduced. The stability and accuracy of these discrete systems are developed using Fourier (von Neumann) analysis. A Propagation Factor is introduced in the framework of this analysis; it is the recommended vehicle for examining accuracy - essentially, the accuracy implied by the fidelity of continuous and discrete dispersion relations. An example involving 3-time-level "leapfrog" Parabolic systems is studied, and the attendant parasitic solutions are discussed in detail. Conservation laws in the continuum and in the discrete system are developed; the analysis leads to prescriptions for estimating derivatives (Neumann data) at a Dirichlet boundary. A discussion of 2D (x,y,t) parabolic problems is given, as a dynamic version of the 2D elliptic problem.

6) Hyperbolic Equations

The 1-D wave equation (x,t) is used as the canonical equation here. Its equivalence to a coupled system of two first-order PDE's in two field variables is discussed. These distinctly different forms lead to two different Lumped Systems using conventional FD; and in turn to two families of Discrete Systems as one adds time-stepping methods to complete the discretization. There are a few Discrete Systems in 1D which have no counterpart in 2D and higher; these are studied insofar as they are useful in practice. But those that generalize at least to 2D (x,y,t) are given preference.

The harmonic or Helmholtz approach is introduced by Fourier-transforming the time-domain. This reduces the Lumped Systems to Elliptic Discrete Systems, which were studied above. A distinguishing feature is the loss of diagonal dominance in the Helmholtz case.

For conventional time-domain simulation, the two Lumped Systems generate two families of discrete systems. Each has a characteristic blend of stability, economy, and accuracy properties. In some cases, there are parasitic modes which are poorly conditioned and potentially solution-dominating; this is a fundamental liability in Hyperbolic problems and is given a fair amount of attention. All of these system properties are studied via Fourier dispersion analysis.

The chapter concludes with a discussion of 2D (x,y,t) systems. In 2D the staggered-grid ap-

proach presents new options and ambiguities over the 1D cases. In particular the Arakawa classification of staggered grids is presented.

Part II: The Finite Element Method

7) General Principles

We introduce the Method of Weighted Residuals here. This is the general approach to discretization used throughout the FEM discussion. We proceed through the concept of Weak-Form PDE as an integral equation; and then by introducing finite bases, to the Discrete System (in the sense introduced in Part 1). Various MWRs are discussed - Subdomain, Least Squares, Galerkin, etc. Boundary Condition enforcement is discussed, with a major distinction for Dirichlet and Neumann or "natural" boundaries. Variational Principles are discussed with two aims: first, to provide a conceptual linkage to the large FEM literature which begins with these; and second, to illustrate the identity of Galerkin and Variational approaches for a common class of problems. Finally, some global conservation properties of Weak-Form PDE's are introduced.

8) A 1D Tutorial

Polynomial bases are introduced in the context of Lagrangian interpolation. Local and Global interpolation is introduced; the element is introduced as the support unit for local interpolation. The degree of inter-element continuity is discussed and illustrated by the Hermite family of elements. Boundary condition enforcement is illustrated in the context of an example.

Standard FEM concepts are introduced: the Element Matrix; the Incidence List and the System Assembly process. The importance of prior knowledge of matrix structure is emphasized; banded and sparse storage methods are described. There are short discussions of treating variable coefficients and of numerical integration (a review). The assembly process is illustrated as commonly practiced, with Gauss (or other) Quadrature.

9) Multi-Dimensional Elements

Linear triangles are introduced as the simplest 2D elements. Interpolation, differentiation, and integration are described on these elements. The Helmholtz formulation is given in detail for these elements. Higher order triangular elements are described. Here we introduce a Local Coordinate System (area coordinates for triangles) and the transformation from local to global for interpolation, differentiation, and integration. Curved-sided triangular elements and the isoparametric coordinate transformation are introduced. Finally, quadrilateral elements are discussed in various forms (Lagrangian, Serendipity) along with the common local coordinates and the isoparametric transformation.

10) Time-Dependent Problems

For hyperbolic and parabolic problems, the general approach is: use FEM to produce the Lumped System *i.e.* to discretize the elliptic or spatial part of the PDE; then use any conventional time-stepping or time-integration method to discretize the temporal part. This is the general strategy in common use. Examples are given for the Diffusion, Advection-Diffusion, Wave, and Telegraph equations.

11) Vector Problems

In this chapter we turn to PDE's whose solutions are Vector Fields; the discussion up to now has been in terms of Scalar Fields. We first discuss the seemingly simple problem of determining the gradient of a scalar for which a FEM solution is already available (as form the methods described above). We discuss the Galerkin form of the problem; rotation to and from the natural local coordinate system; and the handling of Dirichlet and Neumann boundary conditions.

We then turn to several example applications in common use: Elasticity, E&M, Fluid Mechanics, Oceanic Tides. Each has some peculiarities and presents some important choices about bases, weights, time-domain versus Harmonic representation, and selection of potential versus primitive field variables. These examples are displayed with attention to these details, and with references to the literatures involved.

12) Numerical Analysis

The purpose of this chapter is to gain insight into the errors separating exact solutions and discrete FEM solutions which approximate them; and how that relates to the FEM discretization parameters. First we solve several 1D FEM systems in discrete form, exactly and compare to analytic solutions. Then we introduce Fourier (von Neumann) analysis as a tool and apply it to some common FEM approximants. There are useful tables summarizing this, in 1D and in 2D. The analysis is used to look at the 2D Laplace and Helmholtz equations.

We then turn to time-dependent problems and introduce the Propagation Factor analysis which was used also in the FD section, as a quantitative guide to stability and accuracy evaluations. The diffusion equation in 1D (x,t) is studied for stability; monotonicity; accuracy in both 2-level and 3-level -in-time contexts. In the latter, parasitic modes are introduced by the theoretically unnecessary third time level. The implications of this are carefully considered. We then make a comparable analysis of the FEM Wave Equation, in both explicit and implicit-in-time forms. The final case studied is the advection-diffusion equation including its limit of zero diffusion. Throughout the emphasis is on selection of algorithms which produce discrete systems with stable, accurate solutions; and on the possible presence of parasitic modes and their control.

Part III: Inverse Problems

13) Inverse Noise, Singular Value Decomposition, and Linear Least Squares

The Inverse Problem is introduced in the context of the precision of Matrix Inversion processes. Basic definitions are established: mean, covariance, variance. Noise models are introduced to describe variability of inputs normally thought to be known with precision. Eigenvalue theory is briefly reviewed, followed by the Singular Value Decomposition. The SVD is used to describe mean and variance of solutions to poorly posed algebraic problems - either the matrix is poorly conditioned, or the RHS is imperfectly known, or both. Linear Least Squares is then introduced and the normal equations developed. Ordinary and Weighted forms are described in a "General Least Squares" context, with enough detail to expose the practical equations and to enlighten later discussions of noise, regularization, etc.

14) Fitting Models to Data

Basic definitions are introduced for Model-Data misfit, noise, error, etc. "Fitting" is defined as a field estimation problem with imperfect model, imperfect estimates of IC's and BC's, and sparse,

noisy data. The first-order conditions for a Least Squares fit are re-described for the general (regularized) system. Direct solutions are immediately available by application of the Normal Equations. Retaining the Lagrange multipliers offers more algorithmic possibilities. The first-order conditions are rearranged to expose the "forward" and "adjoint" systems for iterative solution. Gradient descent methods are described – both steepest descent (the simple way) and conjugate gradient descent. The optimal step size is developed. Variance estimation by Monte Carlo methods is suggested in concert with this approach. Throughout the well-conditioned nature of the "forward problem" (*i.e.* the FEM model) is assumed.

Direct methods based on the method of Representers are described. This is effective for data-sparse situations. The related approach on the forward side, the unit response method, is then treated.

The same Generalized Least Squares approach is developed for the Parameter Estimation problem. This is fundamentally nonlinear; iterative methods for its solution are suggested. The problem of evaluating the Jacobi matrix for FEM forward models is considered, in concert with an iterative Adjoint method. A summary of terminology is given.

Chapters 14 and 15 are bridges to the works by Wunsch [115] and Bennett [9]; the latter describes PDE's in the continuum; the former, algebraic representations of same. The current work nests these topics in the context of interdisciplinary PDE discretization methods.

15) Dynamic Inversion

In this chapter we illustrate the ideas of Chapter 10, extended to time-dependent "forward models" where a single matrix representation of all equations is not normally practical. We look at the first-order conditions as two dynamic systems running "forward" and in "reverse", the latter being commonly called the "adjoint model" whose dependent variables are the Lagrange Multipliers. An advective-diffusive system is first studied. The Adjoint is developed, and a gradient descent algorithm given. This system admits a higher-order condensation to an Elliptic Problem in time; that is presented. The same general treatment is given to a FEM description of the Wave or Telegraph Equation. Regularization methods in these contexts are described: methods of reduction of degrees of freedom; the weight matrix; and its heuristic construction using FEM.

16) Time Conventions for Real-Time Assimilation

This chapter begins with a general discussion of the relativity of data, model output, and simulated and real time. The context is that of the forecaster. The conceptual framework supports useful "time" diagrams depicting model and data use in hindcast, nowcast, and forecast, assuming a data-assimilative operation. Various types of communication delays are discussed along with a real example. Important time conventions are described in an Appendix for geophysical dynamics, including those applicable to harmonic descriptions of oceanic tides.

17) Skill Assessment for Data-Assimilative Models

This chapter is a discussion of nomenclature applicable to Data Assimilation. (The field is characterized by wide diversity here; that hampers progress.) The discussion covers Forward and Inverse Models; Truth, Estimate, Prediction, and Skill; and Accuracy, Bias, Precision, and Noise. An example Observational System Simulation Experiment is given from the author's experience.

18) Statistical Interpolation

This chapter is a bridge to the wide (and largely self-referencing) literature on the mapping

of geostatistical fields. It is variously referred to as Objective Analysis, Optimal Interpolation, Gauss-Markov Estimation, and Kriging. It is the proper end point of this book. Much of the inversion literature reproduces the findings in these fields – but it proceeds along the lines of "stochastically-forced differential equations" (SDE) which are described here in discrete (FD or FEM) form. We describe the basic problem of field estimation from this viewpoint; develop a simple demonstration of the Gauss-Markov theorem; discuss sampling and estimating finite fields, specifically FEM fields; and the use of analytic covariance functions as prior estimates. The use of SDE to obtain the field covariance is illustrated in two simple examples. Finally, the equivalence of OA or OI as developed here, with Kriging and with Least Squares approaches, is discussed and in a limited way demonstrated.

Chapter 18 is a gateway to the Geostatistics and Kriging fields, represented by Cressie [27], Goovaerts [36] and Kitanidis [51], and the comprehensive work on Data Analysis, *e.g.* Daley [28]. We have tried to maintain interdisciplinarity in the presentation here.

Part I

The Finite Difference Method

Chapter 1

Introduction

1.1 From Algebra to Calculus and Back

Partial Differential Equations (PDE's) describe relations among mathematical *fields* – smooth functions of more than one variable – and their derivatives. Where field smoothness is interrupted, Boundary Conditions (BC's) are required which supplement the PDE with data or constraints. The classic Environmental media (hydrosphere, lithosphere, atmosphere, ionosphere, cryosphere, biosphere) are generally described by fields and PDE's. Practical PDE solution today is one of the major frontiers of applied mathematics and computation, and therefore of Environmental science and engineering.

Ordinary Differential Equations (ODE's) are generally the single-variable limit of the PDE. These are normally studied first – their genesis, analysis, and of course their solution, whether analytical or numerical. There are many similarities in approach between PDE's and ODE's. We are following convention in assuming that the reader is familiar with ODE's in general and has solved some in particular.

Historically, algebra precedes calculus. Scientific descriptions which involved dynamics were first approximated in terms of algebraic differences. These lead to calculus as the limiting case of refinement, among other things relieving the tedium of calculating approximations by hand or on small machines. In modern computation, we have *big* machines; but they are still discrete and finite. So we *reverse* this mathematical process – we go from PDE's described in the calculus, to algebraic approximations which can be implemented on these machines.[1]

While all good discretizations converge to the same limit in calculus, each has different detailed properties in its finite form. So we are intrinsically interested in the correspondence between these discrete forms and the limiting PDE. Ideally, we would like discretization *methods* with which to go backwards from calculus to algebra, confidently. In this text we will look at two common classes of discretization methods, Finite Difference and Finite Element. Understanding the calculus-algebra correspondence is one of the classical objectives of Numerical Analysis.

[1] Of course the algebra actually implemented will be remarkably precise, but still approximate.

1.2 Distributed, Lumped, Discrete Systems

These are engineering terms for common representations of dynamical systems.[2] As used here, the *Distributed System* (or equivalently, the *Continuous System*) is PDE-based (the PDE plus its BC's and data); it is our starting point. Discretization of one or more of its dimensions, to an algebraic approximation, represents a *lumping* of that continuum into a finite algebraic representation. Classical phenomena typically distinguish between space and time coordinates in the calculus. As used here, a *Lumped System* has the space dimensions converted to algebra; but the time dimension left continuous. The process is a partial discretization of PDE's into approximating ODE's. Because these are assumed familiar to the reader, this is a useful intermediate point. Discretization of the time domain into algebra leads us to the fully *Discrete System*, with all dimensions reduced to algebra.

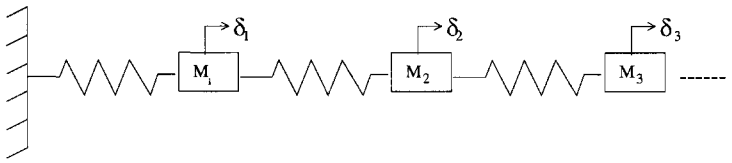


Figure 1.1: Lumped System

Many Lumped Systems originate naturally, where the *spatial* phenomena are simply discrete as originally conceived. For example, consider the problem of analyzing vibrations in a 1-D mechanical system of masses and springs depicted in Figure 1.1. Elementary application of Newton's second law gives the balance between acceleration and net force:

$$M_i \frac{d^2}{dt^2}\delta_i = K_{i+1}(\delta_{i+1} - \delta_i) - K_i(\delta_i - \delta_{i-1}) + F_i \tag{1.1}$$

(F_n accounts for the force applied at the right end; otherwise F is null.) There are N coupled 2nd order ODE's, one for each displacement δ_i. Assembling these, and accounting for their termination, we get:

$$[M]\{\frac{d^2\delta}{dt^2}\} = [K]\{\delta\} + \{F\} \tag{1.2}$$

Solution of this canonical system of ODE's is well understood analytically and numerically. General numerical integration over time would employ well-accepted methods – Runge-Kutta, Adams, etc. This would typically be achieved by first reducing the system to 2N coupled 1st order ODE's by introducing the velocities u_i as additional variables, a standard manipulation:

$$\{\frac{d\delta}{dt}\} = \{u\} \tag{1.3}$$

$$[M]\{\frac{du}{dt}\} = [K]\{\delta\} + \{F\} \tag{1.4}$$

Alternatively, we can solve in the frequency domain with the Fourier transform $\frac{d^2}{dt^2} \rightarrow -\omega^2$

$$[K + \omega^2 M]\{\delta\} = -\{F\} \tag{1.5}$$

[2]The systems language can be ambiguous. In applied work, *system* often refers to nature; while we will generally use its mathematical sense, a *system of equations* that (hopefully) constitutes an abstraction of nature.

and this solution approach is efficient for periodic motion. Both of these Lumped System approaches are well-developed.

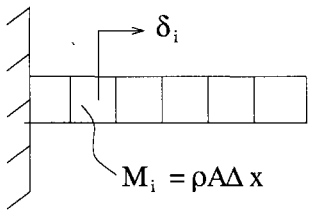


Figure 1.2: Distributed System

As an example of a Distributed System, consider an elastic rod (Figure 1.2) with elastic modulus E, density ρ, cross-section A. A typical ODE-oriented formulation strategy might be to create an approximate Lumped System

$$M_i \frac{d^2}{dt^2}\delta_i = AE\frac{(\delta_{i+1} - \delta_i)}{\Delta x} - AE\frac{(\delta_i - \delta_{i-1})}{\Delta x} + F_i \tag{1.6}$$

where M_i is the lumped mass and the effective spring constant $\mathcal{K} = \frac{AE}{\Delta x}$. Effectively, we have an approximate algebraic approach which is mathematically identical to the Lumped System above. *If one can handle only ODE's, this is the end of the story.* But the classical PDE derivation proceeds further, reaching the continuum by taking the limit of vanishingly small Δx:

$$\rho\frac{\partial^2\delta}{\partial t^2} = E\frac{\partial^2\delta}{\partial x^2} + f \tag{1.7}$$

This is the Wave Equation, in the continuous variable $\delta(x)$. It is one of the canonical PDE's of concern to us.

This approach to PDE derivation is used widely: get an approximate Lumped System and take it to its limiting refinement. One can afford to be a little casual with the lumped approximation, because the subsequent limiting process can be forgiving.

One route to practical PDE solution is to go *backward*, reversing the limiting process just taken, and deal with the approximate Lumped System. The outcome can be nonunique, but intuitive, as many lumped approximations lead to the same limit. So the tendency to eliminate the PDE limiting and unlimiting processes can be dangerous, leaving us without guidance from mathematics. But the practical and common idea is to replace the spatial derivatives with algebraic approximations which can be used with machines. Thus the Distributed System reverts ultimately to a Lumped one, ideally with an accompanying mathematical framework – the Finite Difference "method" provides the simplest of these. It leads from calculus to recognizable, intuitive lumped systems, a bonus.

Relative to the *time-domain*, the ideas retained from ODE's and/or Lumped System analysis generally have their utility here in the PDE world. For example, we can introduce the velocity field u as above into equation 1.7 and get an equivalent system with lower-order time derivatives:

$$\frac{\partial\delta}{\partial t} = u \tag{1.8}$$

$$\rho\frac{\partial u}{\partial t} = E\frac{\partial^2\delta}{\partial x^2} + f \tag{1.9}$$

And, the Fourier transform of the Wave Equation 1.7 gives us the Helmholtz Equation

$$-\rho\omega^2\delta = E\frac{\partial^2\delta}{\partial x^2} + f \tag{1.10}$$

which has its special appeal for periodic phenomena.

There is a general rule of thumb relative to Lumped and Distributed (Continuous) Systems arising in classical mathematical physics:[3]

- *Time* domain considerations are basically similar. The same ideas generally produce analogous outcomes, those normally mastered in the study of ODE's.
- In the *Space* dimensions, Lumped Systems employ algebraic relations; while Distributed Systems involve relations using differential calculus. The correspondence between Lumped and the Distributed Systems is one of the basic subjects of study in Numerical Analysis of PDE's.

Generally we will find the Lumped System to be a useful gathering point here, one which is familiar from prior study.

1.3 PDE Solutions

The common analytic solution strategies which we study include transform methods (Laplace, Fourier); separation of variables; characteristic methods; and similarity transformations (*e.g.* $x/\sqrt{t}$). Notice that each of these well-developed approaches shares the approach of eliminating a dimension, simplifying the PDE ultimately to (tractable) ODE's.

These are powerful methods, and indispensable to science and engineering. But we generally reach limits related to irregular boundaries, variable coefficients, and nonlinearities; the few existing solutions are very restrictive. Even in the absence of these complications, arbitrary forcing, while tractable, can rapidly inject cumbersome convolution or Greens Function manipulations which are likely to require numerical evaluation.

These are serious limitations from a practical viewpoint. However it is critical to emphasize the value of analytic solutions. They can be differentiated, integrated, analyzed; they facilitate sensitivity analysis; they provide clues to necessary and sufficient BC's; and, they inform intuition about generic behavior. Among these important features, none are easily obtained with numerical solutions, and several are impossible. Thus it is essential to understand and respect the body of analytic solutions. One should not embark on a numerical strategy without first mastering the best that analysis can offer.

Some general observations about Numerical solutions are useful. As a rule, these follow a classical route through the generation of ODE's (or equivalently, Lumped Systems). But the result is an *algorithm* for generating a solution. Each solution itself will be very specific – and just a set of numbers, probably with dimensions. Further, the numbers are subject to many approximations which are not apparent at the point of solution presentation.

Accordingly, it is the algorithm which we are interested in here. We will look at several generic aspects of these, looking for quantitative guidance on algorithm design and selection. We will be exposing some basic dimensions:

[3]Naturally, this space-time conceptual separation may not apply to PDE's originating in non-classical phenomena.

- *accuracy*: what is the size of the error (the unknowable discrepancy which will exist between exact and numerical solutions) and on what parameters does it depend?
- *stability*: is that error amplified by an algorithm?
- *convergence*: does the error vanish as the solution is refined, and at what rate?
- *conservation*: what are the discrete forms of energy, mass, etc, and how well can they be conserved even though any solution is necessarily approximate?

Existing analytic solutions can provide an important standard of truth, within the limitations of their range. But for real problems, the true answer is unknown *a priori*, so the meaning and evaluation of error is less straightforward. We need a systematic, mathematical approach to these issues and our field of inquiry is "Numerical Analysis".

1.4 IC's, BC's, Classification by Characteristics

Consider the ODE $\frac{d^2U}{dx^2} = f$. Apart from specifying the forcing $f(x)$, we need two conditions, U and/or $\frac{dU}{dx}$ to satisfy the undetermined constants of integration. There are two general types of problems:

- the Initial Value Problem (IVP): U, $\frac{dU}{dx}$ are specified at the same "initial" point x_0;
- the Boundary Value Problem (BVP): one condition is given at each of two different points x_1, x_2.

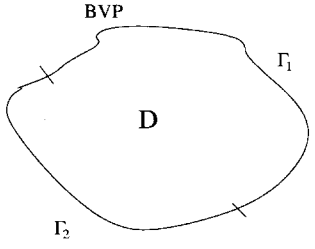
There is a rough equivalence between an IVP and its associated BVP for ODE's, and this is exploited by the "shooting" method. But already in 1-D, that method is subject to ill-conditioning. In the PDE world, there is nothing analogous to this. We must distinguish carefully between these conditions. We will need to distinguish the type of derivative specified; and the location where it is to be specified.

A uniqueness proof: Poisson Equation

We will begin by example. Consider the PDE

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f \tag{1.11}$$

with $U = a(s)$ on Γ_1, $\frac{\partial U}{\partial n} = b(s)$ on Γ_2. The two boundaries Γ_1, Γ_2 enclose the domain D, simply. Imagine two solutions U_1 and U_2 which satisfy these conditions:



$$\begin{array}{rlrl} \nabla^2 U_1 = f & \text{on D} & \nabla^2 U_2 = f \\ U_1 = a & \text{on } \Gamma_1 & U_2 = a \\ \frac{\partial U_1}{\partial n} = b & \text{on } \Gamma_2 & \frac{\partial U_2}{\partial n} = b \end{array}$$

Let $U_3 = U_2 - U_1$ be the difference between the two solutions. Then subtracting the PDE and BC's we get their homogeneous forms:

$$\begin{array}{rl} \nabla^2 U_3 = 0 & \text{on D} \\ U_3 = 0 & \text{on } \Gamma_1 \\ \frac{\partial U_3}{\partial n} = 0 & \text{on } \Gamma_2 \end{array}$$

We have the divergence theorem and the chain rule (see Appendix). From the chain rule,

$$\int\int\int \nabla \cdot (U_3 \nabla U_3)\, dv = \int\int\int \nabla U_3 \cdot \nabla U_3\, dv + \int\int\int U_3 \nabla^2 U_3\, dv \tag{1.12}$$

The divergence theorem gives us

$$\oint U_3 \frac{\partial U_3}{\partial n}\, ds = \int\int\int \nabla U_3 \cdot \nabla U_3\, dv + \int\int\int U_3 \nabla^2 U_3\, dv \tag{1.13}$$

The left side must vanish due to the BC's. The second term on the right must vanish due to the PDE. So we have

$$0 = \int\int\int \nabla U_3 \cdot \nabla U_3\, dv \tag{1.14}$$

Since $\nabla U_3 \cdot \nabla U_3$ is non-negative everywhere, we conclude that ∇U_3 must vanish everywhere, and thus U_3 = constant on D. Finally, $U_3 = 0$ if Γ_1 not null. Thus with this one proviso, we have a unique solution to the problem with the stated BC's. And if Γ_1 null, then U_3 is everywhere an undetermined constant.

Classification of BC's

Conditions on the boundary surface of a PDE domain need to be characterized as involving either the unknown function's value there, or its derivative. In the latter case we need to know a) the level of derivative and b) its direction. Intuitively, for a PDE involving second derivatives, we are talking about first derivatives at most. And the local normal direction on a boundary is special; knowledge of a function's value on a boundary surface implies knowledge of its tangential derivatives, but not of its normal ones. So intuitively, we expect derivative information to be posed in terms of the normal directional derivative, $\frac{\partial}{\partial n}$.

Accordingly, we will use this classification for BC's:

Type 1	U given	Dirichlet
Type 2	$\frac{\partial U}{\partial n}$ given	Neumann
Type 3	$aU + b\frac{\partial U}{\partial n}$ given	Mixed, Robbins, Cauchy, Radiation

Generalizing the previous result, we have that for the Poisson Equation, we need 1 and only 1 BC from this selection on **all** points of a **closed boundary**.

These BC types will need to be supplemented for higher-order PDE's; and for vector problems.

Classification of Equations

In accord with the previous BC discussion of second-order equations: consider the second order PDE:

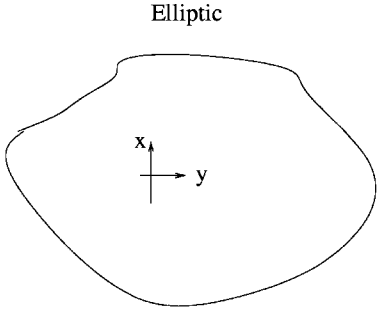
$$a\frac{\partial^2 U}{\partial x^2} + b\frac{\partial^2 U}{\partial x \partial y} + c\frac{\partial^2 U}{\partial y^2} = f \tag{1.15}$$

We restrict ourselves to the *Quasilinear* case: a, b, c, and f depend only on U, $\frac{\partial U}{\partial x}$, $\frac{\partial U}{\partial y}$, x, y. Note this is nonlinear, but there are no products among the highest derivatives. The theory of characteristics supports the following classification [43]:

$b^2 - 4ac > 0$	Hyperbolic	*e.g.*	Wave equation	$\frac{\partial^2 U}{\partial t^2} - \frac{\partial^2 U}{\partial x^2} = 0$
$b^2 - 4ac = 0$	Parabolic	*e.g.*	Diffusion equation	$\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2} = 0$
$b^2 - 4ac < 0$	Elliptic	*e.g.*	Laplace equation	$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0$

These different PDE classes have their own special requirements for necessary and sufficient conditions at their domain limits.

Elliptic equations typically describe equilibrium problems with no time dependence. All independent variables are spatial directions and are roughly equivalent in the form of the second derivatives. For these problems we need BC's on a closed surface.

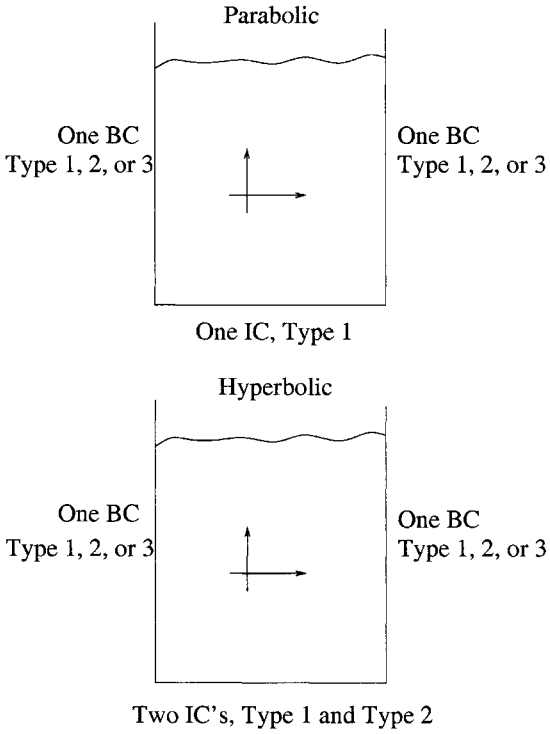


One BC (Type 1, 2, or 3) everywhere
on a closed surface

Parabolic equations typically describe propagation problems; time appears usually with first derivatives only, and the solution progresses forward in t, not backward. The spatial portion of the PDE is typically elliptic and this describes the steady-state solution. Spatial BC's are the same as in elliptic problems. The temporal domain requires a single, Type I initial condition which will propagate forward into unbounded time.

Hyperbolic equations are similar to parabolic ones, in that *propagation in time from initial conditions* is implied. But in these problems, we have a *second time derivative*; so the IC/BC blend is similar to the Parabolic case, but two IC's, Type I and Type II, are required at the same point in time. These propagate forward into unbounded time. Shocks may be supported in hyperbolic solutions.

Notice that we have recovered the distinctive sense of initial and boundary conditions here. This reflects physical intuition about the words: *boundary* connoting spatial constraints, *initial* connoting a temporal initialization. The classical PDE's of mathematical physics support this



intuitive distinction, with the domains easily separated into orthogonal steady-state (elliptic) and transient (hyperbolic/parabolic) subdomains. Naturally, this is convenient but not a mathematical necessity.

Fuller analysis leading to these classifications appears in many standard works, *e.g.* Hildebrand [43].

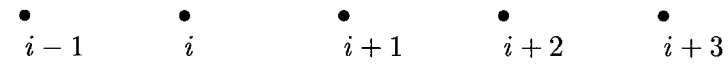
Chapter 2

Finite Difference Calculus

In this chapter we review the calculus of finite differences. The topic is classic and covered in many places. The Taylor series is fundamental to most analysis. A good reference for beginners is Hornbeck [45].

2.1 1-D Differences on a Uniform Mesh

Our objective is to develop differentiation formulas which deal only with functions U which are sampled at discrete grid points X_i: $U(X_i) \equiv U_i$. The sampling grid is assumed to lay out in the natural way, ordered with X, left to right as below.



Assuming equal mesh spacing $h \equiv X_{i+1} - X_i$ for all i, we have the Taylor series:

$$U_{i+1} = U_i + h\frac{\partial U_i}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 U_i}{\partial x^2} + O(h^3) \tag{2.1}$$

$$\begin{aligned} \frac{\partial U_i}{\partial x} &= \frac{U_{i+1} - U_i}{h} - \frac{h}{2!}\frac{\partial^2 U_i}{\partial x^2} + O(h^2) \\ &= \frac{\Delta U_i}{h} + O(h) \end{aligned} \tag{2.2}$$

where the leading error is "of order h (which can be made arbitrarily small with mesh refinement.)" For $\frac{\partial^2 U_i}{\partial x^2}$, we write another Taylor series

$$U_{i+2} = U_i + 2h\frac{\partial U_i}{\partial x} + \frac{(2h)^2}{2!}\frac{\partial^2 U_i}{\partial x^2} + \frac{(2h)^3}{3!}\frac{\partial^3 U_i}{\partial x^3} + \ldots \tag{2.3}$$

Adding these such that $\frac{\partial U_i}{\partial x}$ cancels gives:

$$\frac{\partial^2 U_i}{\partial x^2} = \frac{U_{i+2} - 2U_{i+1} + U_i}{h^2} - h\frac{\partial^3 U_i}{\partial x^3} + \ldots \equiv \frac{\Delta^2 U_i}{h^2} + O(h) \tag{2.4}$$

Generally, approximations to higher derivatives are obtained by adding one or more points; each additional point permits an $O(h)$ expression to the next derivative. The notation for the result is

$$\frac{\partial^n U_i}{\partial x^n} = \frac{\Delta^n U_i}{h^n} + O(h) \tag{2.5}$$

with $\Delta^n U_i$ indicating a difference expression among $U_i \rightarrow U_{i+n}$. These are called the *Forward Differences* and are tabulated in tTable 2.1 below. They have the recursive property

$$\Delta^n U_i = \Delta(\Delta^{n-1} U_i) \tag{2.6}$$

The core operator Δ indicates the "first forward difference".

Backward Differences are defined in the analogous way:

$$U_{i-1} = U_i - h\frac{\partial U_i}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 U_i}{\partial x^2} + \ldots \tag{2.7}$$

$$\frac{\partial^n U_i}{\partial x^n} = \frac{\nabla^n U_i}{h^n} + O(h) \tag{2.8}$$

$$\nabla^n U_i = \nabla(\nabla^{n-1} U_i) \tag{2.9}$$

These are tabulated in Table 2.2.

Both of these approximations are first-order in the mesh spacing h. Higher order approximations are generated by involving more points.

$$U_{i+1} = U_i + h\frac{\partial U_i}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 U_i}{\partial x^2} + \frac{h^3}{3!}\frac{\partial^3 U_i}{\partial x^3} + \ldots \tag{2.10}$$

$$U_{i+2} = U_i + 2h\frac{\partial U_i}{\partial x} + \frac{(2h)^2}{2!}\frac{\partial^2 U_i}{\partial x^2} + \frac{(2h)^3}{3!}\frac{\partial^3 U_i}{\partial x^3} + \ldots \tag{2.11}$$

Combining equations 2.10 and 2.11 with weights 1 and A, we get

$$\begin{aligned} U_{i+1} + AU_{i+2} = (1+A)U_i &+ (1+2A)h\frac{\partial U_i}{\partial x} \\ &+ (1+4A)\frac{h^2}{2!}\frac{\partial^2 U_i}{\partial x^2} + (1+8A)\frac{h^3}{3!}\frac{\partial^3 U_i}{\partial x^3} + \ldots \end{aligned} \tag{2.12}$$

We want the first derivative in terms of U_i, U_{i+1}, and U_{i+2}. If we choose A such that $(1+4A) = 0$, the second derivative term will vanish and the third derivative term will be the leading error term.

$$\frac{\partial U_i}{\partial x} = \frac{[AU_{i+2} + U_{i+1} - (1+A)U_i] - (1+4A)\frac{h^2}{2!}\frac{\partial^2 U_i}{\partial x^2} - (1+8A)\frac{h^3}{3!}\frac{\partial^3 U_i}{\partial x^3} + \ldots}{(1+2A)h} \tag{2.13}$$

$$1 + 4A = 0; \quad A = -1/4 \tag{2.14}$$

$$\frac{\partial U_i}{\partial x} = \frac{-U_{i+2} + 4U_{i+1} - 3U_i}{2h} - \frac{h^2}{3!}\frac{\partial^3 U_i}{\partial x^3} + O(h^3) \tag{2.15}$$

The leading error is $O(h^2)$. This is the second-order correct, forward difference approximation to the first derivative. Higher derivatives at this accuracy can be obtained by adding extra points, as in the $O(h)$ formulas. Tables 2.3 and 2.4 below record these and their backward difference counterparts.

The obvious supplement to these one-sided differences are the *Central Difference* approximations. Assuming a uniform mesh, these combine the forward and backward formulas such that the leading errors cancel. The result is an extra order of accuracy for the same number of points. For example:

$$\frac{\partial U_i}{\partial x} = \frac{\Delta U_i}{h} - \frac{h}{2!}\frac{\partial^2 U_i}{\partial x^2} + O(h^2) \tag{2.16}$$

$$\frac{\partial U_i}{\partial x} = \frac{\nabla U_i}{h} + \frac{h}{2!}\frac{\partial^2 U_i}{\partial x^2} + O(h^2) \tag{2.17}$$

Combining these,

$$\frac{\partial U_i}{\partial x} = \frac{\nabla U_i + \Delta U_i}{2h} + O(h^2) \tag{2.18}$$

$$\nabla U_i + \Delta U_i = U_{i+1} - U_{i-1} \equiv \delta U_i \tag{2.19}$$

The symbol δ in this context indicates the central difference operator; and the centered approximation to the first derivative is

$$\frac{\partial U_i}{\partial x} = \frac{\delta U_i}{2h} + O(h^2) \tag{2.20}$$

This is accurate to second order in h. Higher derivatives can be obtained by adding more points, symmetrically. The $O(h^2)$ centered differences are summarized in Tables 2.5 and 2.6 below.

Summary - Uniform Mesh

The Taylor Series provides difference formulas and error estimates for derivatives of arbitrary order and precision. The procedure is systematic and, as is shown in the next sections, easily generalized to nonuniform meshes and to multiple dimensions. The 1-D results on a uniform mesh may be summarized as:

Forward difference

$$\frac{d^n U_i}{dx^n} = \frac{\Delta^n U_i}{h^n} + O(h)$$

Backward difference

$$\frac{d^n U_i}{dx^n} = \frac{\nabla^n U_i}{h^n} + O(h)$$

Centered difference

$$\frac{d^n U_i}{dx^n} = \frac{\delta^n U_i}{(1 \text{ or } 2)h^n} + O(h^2)$$

All of these have N+1 points with nonzero weights. The centered formulas provide an extra order of accuracy for the same number of points.

To attain *higher-order accuracy*, more points need to be added. In the *uncentered* cases, we have

$$\begin{array}{lcccc} n^{th} \text{ derivative} & + & O(h^2) & \Rightarrow & n+2 \text{ pts.} \\ n^{th} \text{ derivative} & + & O(h^3) & \Rightarrow & n+3 \text{ pts.} \\ n^{th} \text{ derivative} & + & O(h^4) & \Rightarrow & n+4 \text{ pts.} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

Table 2.1: Forward difference representations, $O(h)$. [45].

	f_i	f_{i+1}	f_{i+2}	f_{i+3}	f_{i+4}
$hf'(x_i)$	-1	1			
$h^2f''(x_i)$	1	-2	1		
$h^3f'''(x_i)$	-1	3	-3	1	
$h^4f^{iv}(x_i)$	1	-4	6	-4	1

Table 2.2: Backward difference representations, $O(h)$. [45].

	f_{i-4}	f_{i-3}	f_{i-2}	f_{i-1}	f_i
$hf'(x_i)$				-1	1
$h^2f''(x_i)$			1	-2	1
$h^3f'''(x_i)$		-1	3	-3	1
$h^4f^{iv}(x_i)$	1	-4	6	-4	1

Table 2.3: Forward difference representations, $O(h^2)$. [45].

	f_i	f_{i+1}	f_{i+2}	f_{i+3}	f_{i+4}	f_{i+5}
$2hf'(x_i)$	-3	4	-1			
$h^2f''(x_i)$	2	-5	4	-1		
$2h^3f'''(x_i)$	-5	18	-24	14	-3	
$h^4f^{iv}(x_i)$	3	-14	26	-24	11	-2

Table 2.4: Backward difference representations, $O(h^2)$. [45].

	f_{i-5}	f_{i-4}	f_{i-3}	f_{i-2}	f_{i-1}	f_i
$2hf'(x_i)$				1	-4	3
$h^2f''(x_i)$			-1	4	-5	2
$2h^3f'''(x_i)$		3	-14	24	-18	5
$h^4f^{iv}(x_i)$	-2	11	-24	26	-14	3

Table 2.5: Central difference representations, $O(h^2)$. [45].

	f_{i-2}	f_{i-1}	f_i	f_{i+1}	f_{i+2}
$2hf'(x_i)$		-1	0	1	
$h^2f''(x_i)$		1	-2	1	
$2h^3f'''(x_i)$	-1	2	0	-2	1
$h^4f^4(x_i)$	1	-4	6	-4	1

Table 2.6: Central difference representations, $O(h^4)$. [45]

	f_{i-3}	f_{i-2}	f_{i-1}	f_i	f_{i+1}	f_{i+2}	f_{i+3}
$12hf'(x_i)$		1	-8	0	8	-1	
$12h^2f''(x_i)$		-1	16	-30	16	-1	
$8h^3f'''(x_i)$	1	-8	13	0	-13	8	-1
$6h^4f^4(x_i)$	-1	12	-39	56	-39	12	-1

and so on; whereas for the *centered* cases, we have extra accuracy for the same number of points:

$$\begin{array}{lcccc} n^{th} \text{ derivative} & + & O(h^4) & \Rightarrow & n+3 \text{ pts.} \\ n^{th} \text{ derivative} & + & O(h^6) & \Rightarrow & n+5 \text{ pts.} \\ n^{th} \text{ derivative} & + & O(h^8) & \Rightarrow & n+7 \text{ pts.} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

For the same number of points, the centered formulas always stay one order ahead of the uncentered formulas. Points are added alternately at the center of the formula, or in symmetric pairs.

These rules apply equally well to 1-D differences on nonuniform meshes (see below), with the exception that the special accuracy of Centered Differences is lost.

2.2 Use of the Error Term

The leading error terms are important; since they may interact, they should be kept in detail in all derivations. For example, we may construct a higher-order approximation from two lower-order approximations as follows:

$$\frac{\partial U_i}{\partial x} = \frac{\Delta U_i}{h} - \frac{h}{2}\frac{\partial^2 U_i}{\partial x^2} + O(h^2) \tag{2.21}$$

Substitute a difference formula for the leading error term $\frac{\partial^2 U_i}{\partial x^2}$

$$\frac{\partial^2 U_i}{\partial x^2} = \frac{\Delta^2 U_i}{h^2} + O(h) \tag{2.22}$$

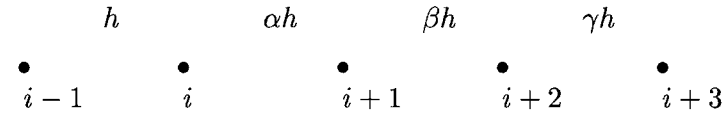
This will push the error term to $O(h) \cdot O(h)$:

$$\begin{aligned} \frac{\partial U_i}{\partial x} &= \frac{\Delta U_i}{h} - \frac{h}{2}[\frac{\Delta^2 U_i}{h^2} + O(h)] + O(h^2) \\ &= \frac{\Delta U_i}{h} - \frac{1}{2}\frac{\Delta^2 U_i}{h} + O(h^2) \\ &= [\frac{(U_{i+1} - U_i)}{h} - \frac{1}{2}\frac{(U_{i+2} - 2U_{i+1} + U_i)}{h}] + O(h^2) \qquad (2.23) \\ \frac{\partial U_i}{\partial x} &= \frac{[-3U_i + 4U_{i+1} - U_{i+2}]}{2h} + O(h^2) \qquad (2.24) \end{aligned}$$

This is the same as in the forward difference Tables derived directly from Taylor series. This procedure has obvious generality; it will produce a difference expression whose order is the product of its two parts.

2.3 1-D Differences on Nonuniform Meshes

The Taylor Series procedure outlined above is not restricted to uniform meshes. Consider the following 5-point grid:



Suppose we wish to find difference formulas for derivatives at node i. We proceed to express all the other nodal values in Taylor series about i:

$$\begin{Bmatrix} U_{i-1} \\ U_{i+1} \\ U_{i+2} \\ U_{i+3} \end{Bmatrix} = U_i \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} + \frac{h}{1!}\frac{dU_i}{dx} \begin{Bmatrix} -1 \\ \alpha \\ \alpha+\beta \\ \alpha+\beta+\gamma \end{Bmatrix} + \frac{h^2}{2!}\frac{d^2U_i}{dx^2} \begin{Bmatrix} 1 \\ \alpha^2 \\ (\alpha+\beta)^2 \\ (\alpha+\beta+\gamma)^2 \end{Bmatrix} + \frac{h^3}{3!}\frac{d^3U_i}{dx^3} \begin{Bmatrix} -1 \\ \alpha^3 \\ (\alpha+\beta)^3 \\ (\alpha+\beta+\gamma)^3 \end{Bmatrix} + \frac{h^4}{4!}\frac{d^4U_i}{dx^4} \begin{Bmatrix} 1 \\ \alpha^4 \\ (\alpha+\beta)^4 \\ (\alpha+\beta+\gamma)^4 \end{Bmatrix} + \ldots \tag{2.25}$$

Now form a weighted sum of the four equations; let the weights be (1,A,B,C) (the first one is arbitrary since we can always multiply the result by a constant). The result:

$$\begin{aligned} U_{i-1} + AU_{i+1} + BU_{i+2} + CU_{i+3} &= U_i(1+A+B+C) \\ &+ h\frac{dU_i}{dx}(-1+\alpha A+(\alpha+\beta)B+(\alpha+\beta+\gamma)C) \\ &+ \frac{h^2}{2!}\frac{d^2U_i}{dx^2}(1+\alpha^2 A+(\alpha+\beta)^2 B+(\alpha+\beta+\gamma)^2 C) \\ &+ \frac{h^3}{3!}\frac{d^3U_i}{dx^3}(-1+\alpha^3 A+(\alpha+\beta)^3 B+(\alpha+\beta+\gamma)^3 C) \\ &+ \frac{h^4}{4!}\frac{d^4U_i}{dx^4}(1+\alpha^4 A+(\alpha+\beta)^4 B+(\alpha+\beta+\gamma)^4 C) \end{aligned} \tag{2.26}$$

Now we have at our disposal the three parameters (A,B,C). Suppose we want a difference formula for $\frac{dU_i}{dx}$ which only involves u_{i-1}, u_i, u_{i+1}. Clearly then, B and C must be zero; and if we select A such that the coefficient of $\frac{d^2U_i}{dx^2}$ vanishes, then we will create an O(h^2) approximation for $\frac{dU_i}{dx}$:

$$1+\alpha^2 A = 0 \Rightarrow A = -\frac{1}{\alpha^2} \tag{2.27}$$

and thus

$$U_{i-1} + AU_{i+1} - U_i(1+A) = h\frac{dU_i}{dx}(-1+\alpha A) + \frac{h^3}{3!}\frac{d^3U_i}{dx^3}(-1+\alpha^3 A) + \ldots \tag{2.28}$$

or

$$\frac{dU_i}{dx} = \frac{U_{i-1} + AU_{i+1} - U_i(1+A)}{h(\alpha A - 1)} - \frac{h^2}{3!}\frac{d^3U_i}{dx^3}\left(\frac{\alpha^3 A - 1}{\alpha A - 1}\right) + \ldots \tag{2.29}$$

Substituting for A we get

$$\frac{dU_i}{dx} = \frac{-\alpha^2 U_{i-1} + U_i(\alpha^2 - 1) + U_{i+1}}{h(\alpha+1)\alpha} - \alpha\frac{h^2}{3!}\frac{d^3U_i}{dx^3} + ... \tag{2.30}$$

It is readily checked that when $\alpha = 1$ we obtain the familiar central difference formula.

If we wish to achieve higher accuracy, we must involve another point. Retaining $B \neq 0$, for example, we may set the coefficients of the second and third derivatives equal to zero in 2.27:

$$1 + \alpha^2 A + (\alpha+\beta)^2 B = 0 \Rightarrow A = \frac{-(1+\alpha+\beta)}{\alpha^2\beta} \tag{2.31}$$

$$-1 + \alpha^3 A + (\alpha+\beta)^3 B = 0 \Rightarrow B = \frac{(1+\alpha)}{(\alpha+\beta)^2\beta} \tag{2.32}$$

The resulting difference formula will have a leading error term

$$-\frac{h^3}{4!}\frac{d^4U_i}{dx^4}\left[\frac{1+\alpha^4 A + (\alpha+\beta)^4 B}{-1+\alpha A + (\alpha+\beta)B}\right] \tag{2.33}$$

The above procedure may be expressed in more generality as follows. Suppose we want expressions for derivatives at some grid point. Without loss of generality we take this (temporarily) to be the origin of the coordinate system. Then denoting by μ_i the difference of U_i - U_0, the Taylor series is

$$\mu_i = \sum_{n=1}^{\infty} \frac{1}{n!}\frac{d^nU_0}{dx^n}x_i^n \tag{2.34}$$

If we invent weights W_i, then

$$\sum_i W_i\mu_i = \sum_{n=1}^{\infty}\frac{1}{n!}\frac{d^nU_0}{dx^n}\sum_i W_i x_i^n = \sum_{n=1}^{\infty}\frac{1}{n!}\frac{d^nU_0}{dx^n}\omega_n \tag{2.35}$$

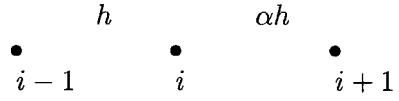
where the index i runs over all grid points, and ω_n is the n^{th} moment of the weights about the origin:

$$\omega_n = \sum_i W_i x_i^n \tag{2.36}$$

A first order derivative difference expression for $\frac{d^NU_0}{dx^N}$ can thus be obtained by setting the first $N-1$ moments of $\mathbf{W}$ equal to zero, which can be achieved with exactly N nonzero weights. Recalling that $\mu_i = U_i$ - U_0, this yields $N+1$ node points in the expression for the N^{th} derivative. Higher order expressions can be obtained by making ω_{N+1} and progressively higher moments equal to zero.

2.4 Alternative to Taylor Series: Polynomial Fit

A different procedure is to fit a polynomial or other interpolant to discrete samples; and differentiate the result. For example, consider the 3-point mesh shown below.



We will fit a second-order polynomial $\widehat{U}(x)$ to samples of U at the three mesh points:

$$\begin{aligned} \widehat{U} &= ax^2 + bx + c && (2.37)\\ U_{i-1} &= ah^2 + b(-h) + c && (2.38)\\ U_i &= c && (2.39)\\ U_{i+1} &= a(\alpha h)^2 + b(\alpha h) + c && (2.40) \end{aligned}$$

The fit is obtained by solving for the three coefficients

$$\begin{bmatrix} h^2 & -h & 1 \\ 0 & 0 & 1 \\ (\alpha h)^2 & (\alpha h) & 1 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{Bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{Bmatrix} \tag{2.41}$$

Inverting this gives

$$\begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{Bmatrix} \dfrac{(\alpha[U_{i-1} - U_i] + [U_{i+1} - U_i])}{(\alpha^2 + \alpha)h^2} \\ \dfrac{(-\alpha^2[U_{i-1} - U_i] + [U_{i+1} - U_i])}{(\alpha^2 + \alpha)h} \\ U_i \end{Bmatrix} \tag{2.42}$$

and the polynomial is differentiated to provide the difference formulas at any point x:

$$\frac{\partial \widehat{U}}{\partial x} = 2ax + b \tag{2.43}$$

$$\frac{\partial^2 \widehat{U}}{\partial x^2} = 2a = 2\left[\frac{U_{i+1} - U_i(1+\alpha) + \alpha U_{i-1}}{\alpha(\alpha+1)h^2}\right] \tag{2.44}$$

These results are identical to those obtained from Taylor Series estimates at the three grid points. (Student should verify this.) This procedure has the advantage of estimating derivatives everywhere, not just at the mesh points; but it lacks the truncation error estimate.

2.5 Difference Formulas with Cross-Derivatives

Generally, the 1-D formulas can be used in higher dimensions (although there are other options). The special case is the mixed derivative with respect to 2 or more dimensions. There are two approaches. First, we can operate with the 2-D Taylor series:

$$\begin{aligned} U(x+\Delta x, y+\Delta y) &= U|_{x,y} + (\Delta x\frac{\partial}{\partial x} + \Delta y\frac{\partial}{\partial y})U|_{x,y} \\ &+ \frac{1}{2!}(\Delta x\frac{\partial}{\partial x} + \Delta y\frac{\partial}{\partial y})^2 U|_{x,y} \\ &+ \frac{1}{3!}(\Delta x\frac{\partial}{\partial x} + \Delta y\frac{\partial}{\partial y})^3 U|_{x,y} + \ldots \end{aligned} \tag{2.45}$$

where

$$(\Delta x\frac{\partial}{\partial x} + \Delta y\frac{\partial}{\partial y})^2 = \Delta x^2\frac{\partial^2}{\partial x^2} + 2\Delta x\Delta y\frac{\partial^2}{\partial x\partial y} + \Delta y^2\frac{\partial^2}{\partial y^2} \tag{2.46}$$

and so on. From here, the procedure is generally the same as in 1-D case: write Taylor series for all points in terms of $U, \partial U, \ldots$ at point where ∂U is wanted; mix together to get desired accuracy.

The alternative approach is to operate with the 1-D formula already in hand. For example: on an (i,j) mesh with uniform mesh spacing (h,k):

$$\frac{\partial^2 U}{\partial x \partial y} = \frac{\partial}{\partial x}(\frac{\partial U}{\partial y}) \simeq \frac{\left[\frac{U_{j+1}-U_{j-1}}{2k}\right]_{i+1} - \left[\frac{U_{j+1}-U_{j-1}}{2k}\right]_{i-1}}{2h} \tag{2.47}$$

This should be intuitively correct to second order, since centered differences are being invoked. But so far we lack the leading error term. We can get this from the 1-D formula,

$$\frac{\partial U}{\partial x}|_i = \frac{U_{i+1} - U_{i-1}}{2h} - \frac{h^2}{6}\frac{\partial^3 U}{\partial x^3}|_i + \ldots \tag{2.48}$$

Differentiating this,

$$\frac{\partial}{\partial y}(\frac{\partial U}{\partial x})|_{ij} = \frac{\frac{\partial U}{\partial y}|_{i+1,j} - \frac{\partial U}{\partial y}|_{i-1,j}}{2h} - \frac{h^2}{6}\frac{\partial^4 U}{\partial x^3 \partial y}|_{i,j} + \ldots \tag{2.49}$$

By the same formula:

$$\frac{\partial U}{\partial y}|_i = \frac{U_{j+1} - U_{j-1}}{2k} - \frac{k^2}{6}\frac{\partial^3 U}{\partial y^3}|_j + \ldots \tag{2.50}$$

$$\begin{aligned} \frac{\partial^2 U}{\partial y \partial x}|_{ij} &= \frac{1}{2h}\left[(\frac{U_{i+1,j+1} - U_{i+1,j-1}}{2k}) - (\frac{U_{i-1,j+1} - U_{i-1,j-1}}{2k})\right] \\ &- \frac{1}{2h}\frac{k^2}{6}\left[\frac{\partial^3 U}{\partial y^3}|_{i+1,j} - \frac{\partial^3 U}{\partial y^3}|_{i-1,j}\right] - \frac{h^2}{6}\frac{\partial^4 U}{\partial x^3 \partial y}|_{i,j} \end{aligned} \tag{2.51}$$

There is an apparent asymmetry in the error terms. Also, the $\frac{k^2}{h}$ would in general be a fatal problem, reducing the accuracy to first-order. But

$$\frac{\left[\frac{\partial^3 U}{\partial y^3}|_{i+1,j} - \frac{\partial^3 U}{\partial y^3}|_{i-1,j}\right]}{2h} = \frac{\partial}{\partial x}(\frac{\partial^3 U}{\partial y^3}) + O(h^2) \tag{2.52}$$

So the leading error terms are

$$-\frac{k^2}{6}\frac{\partial^4 U}{\partial x \partial y^3} - \frac{h^2}{6}\frac{\partial^4 U}{\partial y \partial x^3} \tag{2.53}$$

Now we have symmetry, as expected from the form of the difference expression. And the accuracy is second-order in h and k, independently.

Chapter 3

Elliptic Equations

3.1 Introduction

Elliptic equations describe pure boundary-value problems. They require boundary conditions on a surface completely surrounding a closed domain. In classical mathematical physics, Elliptic equations generally describe *equilibrium* problems, for which the closed domain is intuitive and natural. In many cases, these problems are the steady-state limit of a more dynamical (transient) problem, the evolution of which can be represented by adding a time dimension to the problem – specifically, appending terms in $\frac{\partial}{\partial t}$ and/or $\frac{\partial^2}{\partial t^2}$ to an otherwise Elliptic operator. In that case, the boundary condition requirements do not change; but additional initial conditions are added to constrain the time-domain evolution.

A characteristic algebraic structure emerges from FD treatment of Elliptic equations and their BC's. We explore that below, by example, and then look at some considerations about algebraic solution. The discretized Elliptic system generally presents a structured set of simultaneous algebraic equations. Understanding the solution of this algebra is prerequisite to solving the dynamic (Hyperbolic or Parabolic) system; the Elliptic algorithms which work can be invoked in each time step of the dynamic problems.

3.2 A 1-D Example

Consider the 1-D equation

$$\frac{d}{dx}K(x)\frac{dU}{dx} = r(x) \tag{3.1}$$

with $K(x)$ and $r(x)$ known. Assuming that K is differentiable, we have the equivalent form

$$K\frac{d^2U}{dx^2} + \frac{dK}{dx}\frac{dU}{dx} = r \tag{3.2}$$

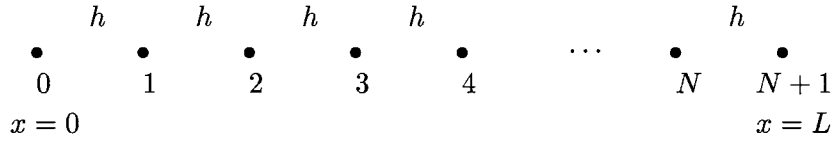
We pose this problem with Dirichlet boundary conditions

$$U(0) = U_o; \quad U(L) = U_L \tag{3.3}$$

and closure

$$K(x) = ax^2 \tag{3.4}$$

We seek a numerical solution on a simple uniform mesh:



with $h \equiv \Delta x = L/(N+1)$. A second-order FD representation of equation 3.2, centered at node i, is

$$ax_i^2\left(\frac{U_{i+1} - 2U_i + U_{i-1}}{h^2}\right) + 2ax_i\left(\frac{U_{i+1} - U_{i-1}}{2h}\right) = r_i \tag{3.5}$$

This is valid to $O(h^2)$ for all $i = [1 : N]$. Grouping terms, we have

$$U_{i-1}\left[\frac{ax_i^2}{h^2} - \frac{2ax_i}{2h}\right] + U_i\left[\frac{-2ax_i^2}{h^2}\right] + U_{i+1}\left[\frac{ax_i^2}{h^2} + \frac{2ax_i}{2h}\right] = r_i \tag{3.6}$$

or,

$$U_{i-1}[A_i] + U_i[B_i] + U_{i+1}[C_i] = r_i \qquad i = [1 : N] \tag{3.7}$$

The FD expressions at nodes 1 and N spill over onto the boundaries, where the values U_0 and U_{N+1} are known because of the BC's. These 2 relations are re-expressed with all known information on the right-hand side:

$$U_1[B_1] + U_2[C_1] = r_1 - [A_1]U_o \tag{3.8}$$

$$U_{N-1}[A_N] + U_N[B_N] = r_i - [C_N]U_L \tag{3.9}$$

The result is the tridiagonal system illustrated in Figure 3.1

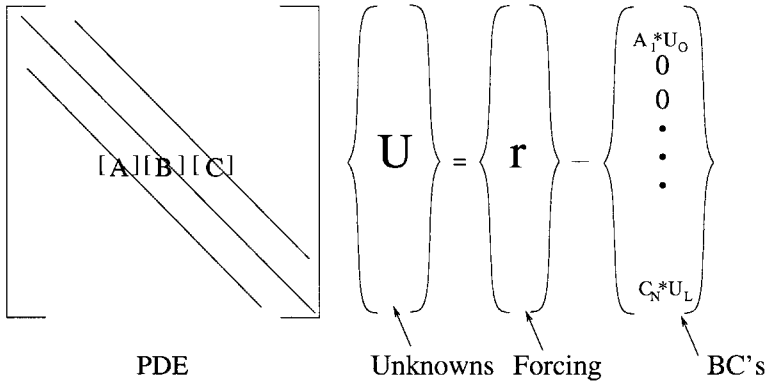


Figure 3.1: Tridiagonal structure of the 1-D Elliptic example.

This system expresses the canonical algebraic form of the PDE approximation. The differential operator, modified to suit the BC form, is realized in FD form in the LHS matrix. The vector of unknowns are the function U approximated at the nodes. The RHS vector contains two contributions, one from the inhomogeneous PDE forcing term, the other from the BC data. Each line of the matrix equation is a FD approximation; line i is centered at node i.

There are other approximations to this ODE. So far we have worked with equation 3.2, which resulted in

$$A_i = \frac{K_i}{h^2} - \frac{1}{2h}\frac{dK_i}{dx} \tag{3.10}$$

$$B_i = -2\frac{K_i}{h^2} \tag{3.11}$$

$$C_i = \frac{K_i}{h^2} + \frac{1}{2h}\frac{dK_i}{dx} \tag{3.12}$$

Alternatively, we may proceed directly from equation 3.1, using the second-order approximation

$$\frac{d}{dx}K(x)\frac{dU}{dx} = \frac{1}{h}\left[K_{i+\frac{1}{2}}\frac{U_{i+1}-U_i}{h} - K_{i-\frac{1}{2}}\frac{U_i - U_{i-1}}{h}\right] \tag{3.13}$$

This leads to the same matrix form, with

$$A_i = \frac{1}{h^2}K_{i-\frac{1}{2}} \tag{3.14}$$

$$B_i = -\frac{1}{h^2}\left(K_{i-\frac{1}{2}} + K_{i+\frac{1}{2}}\right) \tag{3.15}$$

$$C_i = \frac{1}{h^2}K_{i+\frac{1}{2}} \tag{3.16}$$

These two alternatives can be seen to be Taylor Series approximations of each other, and identical for linear variation in $K(x)$. This form does not require explicit differentiation of K; that is achieved numerically, in the differencing.

Either of these two systems is readily solved by the Thomas Algorithm, a realization of direct LU decomposition for tridiagonal systems. While general inversion requires $O(N^3)$ floating point operations and $O(N^2)$ storage, tridiagonal LU requires only $O(N)$ operations; and the original matrix storage requirement $3N$ is sufficient to contain the LU factorization. So the Thomas approach is regularly used in many tridiagonal contexts.

Neumann BC

Next consider the substitution of the Neumann condition at $x = L$:

$$\frac{dU}{dx}(L) = a \tag{3.17}$$

We will look at four options. The first,

$$\frac{U_{N+1} - U_N}{h} = a \quad \rightarrow \quad U_{N+1} = U_N + ha \tag{3.18}$$

is first-order correct. It is simply implemented in row N as:

$$U_{N-1}[A_N] + U_N[B_N + C_N] = r_N - ha[C_N] \tag{3.19}$$

The single first-order error created dominates the convergence rate of this system; the reader is invited to confirm this by solving this or a comparable problem and comparing with its analytical solution.

To reinstate $O(h^2)$ convergence, we can utilize a second-order backward difference approximation:

$$\frac{U_{N-1} - 4U_N + 3U_{N+1}}{2h} = a \quad \rightarrow \quad U_{N+1} = \frac{4}{3}U_N - \frac{1}{3}U_{N-1} + 2ha \tag{3.20}$$

Row N now becomes

$$U_{N-1}[A_N - \frac{1}{3}C_N] + U_N[B_N + \frac{4}{3}C_N] = r_N - 2ha[C_N] \tag{3.21}$$

An alternate approach to $O(h^2)$ is to relocate the nodes to the right, such that nodes N and $N+1$ are equidistant from the Neumann boundary. The FD mesh is stretched by one half-cell to the right to accomplish this and still keep a uniform mesh:

$$h = \frac{L}{N + \frac{1}{2}} \tag{3.22}$$

Now the boundary algebra is identical to the first-order expression above:

$$\frac{U_{N+1} - U_N}{h} = a \quad \rightarrow \quad U_{N+1} = U_N + ha \tag{3.23}$$

$$U_{N-1}[A_N] + U_N[B_N + C_N] = r_N - ha[C_N] \tag{3.24}$$

The system is formally the same as the first-order option; but the coefficients are adjusted to the new value of h, and the nodes at which the answers are obtained are shifted in location. As a result we have $O(h^2)$ everywhere. Node $N+1$ in this case is referred to as a *shadow node* since it lies outside the formal problem domain.

Finally, a further shift of the mesh such that node N lies at $X = L$ leaves nodes $N+1$ and $N-1$ equidistant from the boundary. In this case,

$$h = \frac{L}{N} \tag{3.25}$$

and the boundary condition is approximated to second order as

$$\frac{U_{N+1} - U_{N-1}}{2h} = a \quad \rightarrow \quad U_{N+1} = U_{N-1} + 2ha \tag{3.26}$$

Row N becomes

$$U_{N-1}[A_N + C_N] + U_N[B_N] = r_N - 2ha[C_N] \tag{3.27}$$

Mixed BC

Suppose we have a Type 3 condition at $x = L$:

$$\frac{dU}{dx}(L) = a + bU(L) \tag{3.28}$$

Working with the final case from above, we have to second order:

$$\frac{U_{N+1} - U_{N-1}}{2h} = a + bU_N \quad \rightarrow \quad U_{N+1} = U_{N-1} + 2ha + 2hbU_N \tag{3.29}$$

Row N becomes

$$U_{N-1}[A_N + C_N] + U_N[B_N + 2hbC_N] = r_N - 2ha[C_N] \tag{3.30}$$

Analogous modifications may be achieved following the other approaches to the Neumann BC.

The reader is encouraged to implement these various approximations and confirm their convergence rates relative to an analytic solution, before proceeding to higher dimensions. As a rule, the overall convergence rate is limited by the weakest equation in the system.

3.3 2-D Example: Poisson Equation on a Regular Grid

The 2-D extension of the previous example is straightforward. We consider the Poisson Equation with boundary conditions as shown. We will discretize this on the grid shown in Figure 3.2. Note

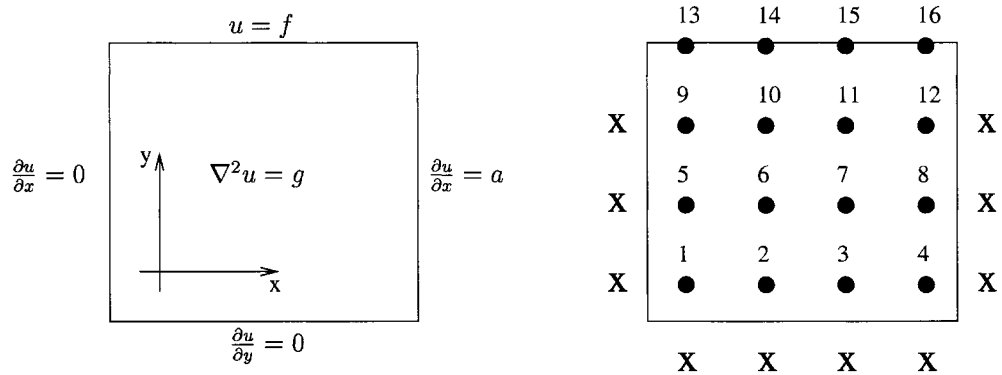


Figure 3.2: The 2-D Elliptic example. × indicates a Shadow Node.

especially the locations of the nodes (•) relative to the boundaries, and the placement of *shadow nodes* (×) across the Type II boundaries.

Molecules

Interior. The PDE is discretized on the interior as the sum of two 1-D centered FD approximations as in Figure 3.3, with $h = \Delta x$ and $k = \Delta y$.

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \simeq \frac{\delta_x^2 U_i}{h^2} + \frac{\delta_y^2 U_i}{k^2} = g \tag{3.31}$$

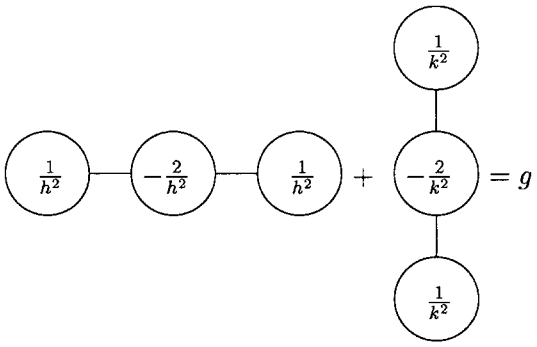


Figure 3.3: 2D Molecule as sum of 1-D molecules.

For convenience we define the mesh ratio $\beta = h^2/k^2$, and this leads to the *Computational Molecule* which is valid at *all interior nodes* (Figure 3.4.)

Boundaries. Now we have to address all the boundaries. Along the *left boundary*, we have a homogeneous Neumann Condition. Its FD approximation is

$$\frac{U_\bullet - U_\times}{h} = 0. \tag{3.32}$$

and therefore the coefficient of the shadow node is "reflected" into the computational domain:

$$U_\times = U_\bullet \tag{3.33}$$

Along the *bottom* boundary, the analogous procedure holds. Boundary molecules appear in Figure 3.5 with these FD BC's incorporated.

Along the *right* boundary we have an inhomogeneous Type II condition. The coefficient of the shadow node is reflected, and also there is an addition to the RHS:

$$\frac{U_\times - U_\bullet}{h} = a \tag{3.34}$$

$$U_\times = U_\bullet + ah \tag{3.35}$$

Along the *top* boundary, we have the Dirichlet condition at the known nodes 13-16, immediately above the active nodes. The Dirichlet data is migrated to the right-hand (known) side of the molecules for nodes 9 through 12; the Dirichlet nodes have been removed from the algebra, in favor of their data.

Finally, there are the four *Corners*. Each has 2 modifications, one for each side. For example, node 4 is the combination of the bottom and right boundary molecules.

All of these boundary molecules are summarized in Figure 3.5. Throughout this discretization, we have followed two basic rules:

- Type I boundaries: the PDE approximation is not needed. The BC is perfect, so we have used it directly. The molecules are arranged to reach out to the boundary.
- Type II, III boundaries: the PDE spills over the boundary, creating shadow nodes. Merging that with the discrete BC closes the system on these boundaries.

We will come back to the Type I boundaries later, where the neglected PDE approximation will become quite useful.

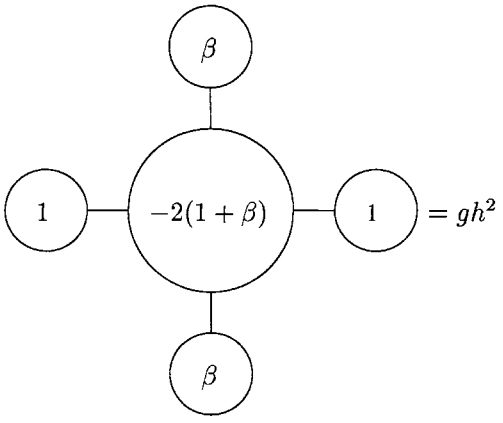


Figure 3.4: Interior Molecule

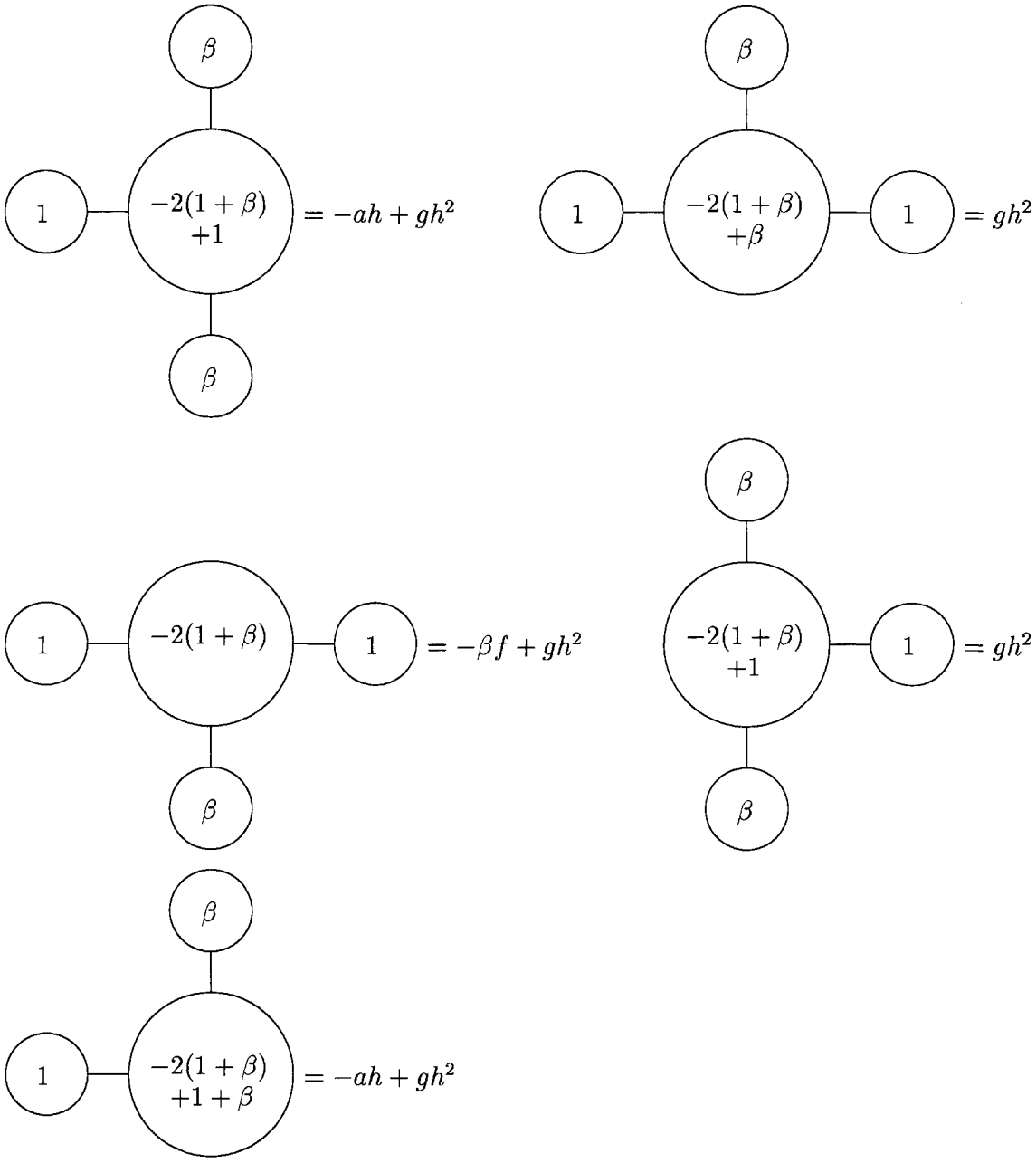


Figure 3.5: Boundary molecules for the 2-D example, plus the composite molecule at Corner node 4.

Matrix Assembly and Direct Solution

Assembling this algebra, we have the matrix structure shown in Figure 3.6.

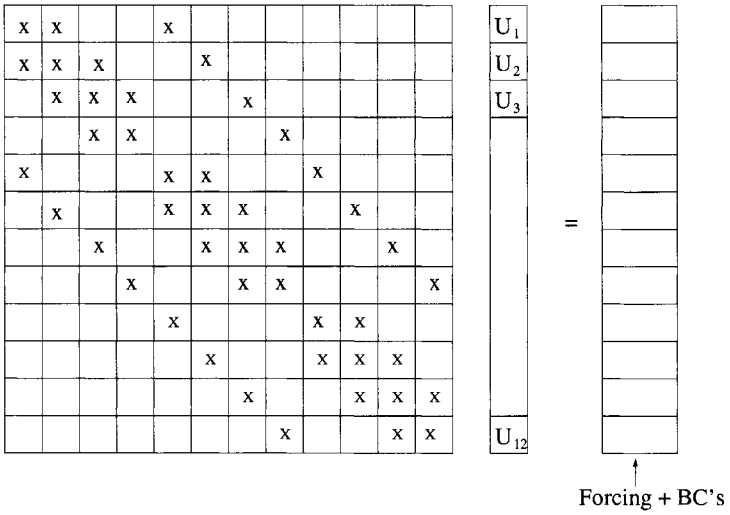


Figure 3.6: Matrix structure of the 2-D example.

The matrix is sparse, with at most 5 nonzero entries per row, regardless of the number of nodes. Further, with the natural node numbering system used, it is banded and pentadiagonal. The storage here has been made as compact as possible, with all data lying within the *Bandwidth* B. Here $B = 9$ and more generally, B is twice the number of nodes in the x-direction, plus 1. There are systematic alterations due to the boundary conditions. The central tridiagonal structure reflects the approximation to $\frac{\partial^2 U}{\partial x^2}$. The outer diagonals are needed for $\frac{\partial^2 U}{\partial y^2}$. The reader is encouraged to confirm this structure.

Solution strategies for such a system fall into two distinct categories, *Direct* and *Iterative*. Direct algorithms achieve the exact algebraic solution in a finite number of operations. They can involve relatively complex coding, and are non-repetitive in their simplest forms. These strategies are capable of exploiting sparse and banded structures; and particularly those which operate on banded matrices are very sensitive (in terms of speed) to the node numbering scheme. For example, the popular LU decomposition methods have the property that all intermediate matrices may be stored within the bandwidth of the original matrix; hence there is a significant economy with LU decomposition for 2-D FD systems; and the node numbering should be adjusted to minimize the bandwidth. These algorithms also have a 2-step structure, wherein the matrix is first factored (LU decomposed) and then the system solved. Since the factorization step is the most time-consuming, it is efficient to do that only once when multiple right-hand sides are involved. The popular Thomas Algorithm is an example of banded LU decomposition for tridiagonal systems.

To illustrate the LU idea, we have

$$[K]\{x\} = [L][U]\{x\} = \{b\} \tag{3.36}$$

with $[K] \equiv [L][U]$ constituting the FD molecules and $\{x\}$ the nodal unknowns. Introducing the intermediate vector $\{y\} = [U]\{x\}$, we have

$$[L]\{y\} = \{b\} \tag{3.37}$$

The solution for $\{y\}$ is straightforward, since the first row of this equation has only 1 nonzero coefficient, the second row, two, etc. Knowing $\{y\}$, the solution for $\{x\}$ is similarly simple:

$$[U]\{x\} = \{y\} \tag{3.38}$$

The work involved in the LU factorization is $O(B^2 \cdot N)$, with B the bandwidth; and in the solution for $\{y\}$ and $\{x\}$, $O(B \cdot N)$. As mentioned above, the memory requirements for $[K]$ and its factors are limited to $B \cdot N$.

Iterative Solution

Iterative methods are opposite in many ways. They achieve the exact algebraic solution only after an infinite number of operations; are monotonously repetitive; and can involve simple coding. Several classic iterative methods are listed below which can be implemented directly from the FD molecules. Sparseness is important to economy; but bandedness is irrelevant, so there are few constraints on node numbering. Several algorithms are sensitive to node numbering insofar as it affects the order of computations and that can affect convergence. There are both "point" iterative methods, in which individual nodes are updated separately; and "block" or "line" methods, in which groups of nodes are updated together by solving a subset of the full matrix system.

A standard trio of point iterative methods is illustrated for this example, with $\beta = 1$ (*i.e.* $h = k$). The interior molecule is the familiar 5-point formula illustrated in Figure 3.7.

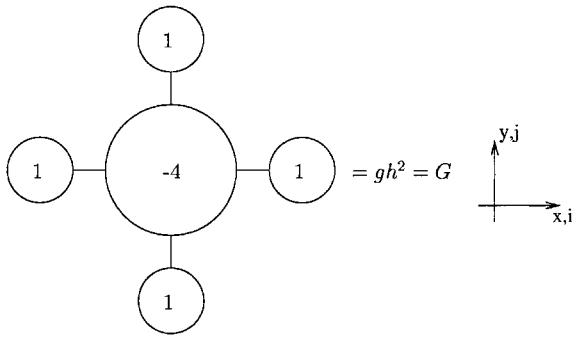


Figure 3.7: The 2-D Poisson molecule with $\beta = 1$.

Using the natural (i, j) node numbering system, we have the interior equation

$$U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{ij} = G \tag{3.39}$$

Jacobi: In this method we "solve" for U_{ij} in terms of its neighbors:

$$U_{ij}^{l+1} = \frac{1}{4}[U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}]^l + G \tag{3.40}$$

with superscript indicating iteration level. Boundary molecules would be modified appropriately. This iteration would be started from an initial guess (*e.g.* the mean of the Type I BC's) and continued until a suitable stopping rule is satisfied, typically in terms of the size of the update. Notice that we are effectively averaging the neighbors from the previous iteration. Programming of this method is easy, and may proceed directly from the FD molecules with proper decision-making at boundaries. Two arrays are needed, U^l and U^{l+1}. The iteration is independent of node numbering since all the U^{l+1} are computed before any are used in the iteration.

Gauss-Seidel: This is a perturbation of the Jacobi method. The idea is to always use the latest information, so we have a mixture of U^l and U^{l+1} on the right-side of the equation:

$$U_{ij}^{l+1} = \frac{1}{4}[U_{i+1,j}^{l} + U_{i-1,j}^{l+1} + U_{i,j+1}^{l} + U_{i,j-1}^{l+1}] + G \tag{3.41}$$

Like Jacobi, this method is simple to program directly from the FD molecules. There is no need, however, to distinguish between U^l and U^{l+1} computationally; the algorithm can be realized by overwriting U_{ij} as soon as it is computed. Because of this feature, the way in which the nodes are ordered makes a difference to the process convergence. Also, since U^{l+1} is used as soon as it is available, we expect this process to converge (or diverge!) faster than Jacobi.

SOR (Successive Over-Relaxation): The idea here is to accelerate/dampen the Gauss-Seidel process. If we identify $\tilde{U}_{ij}^{l+1}$ as the Gauss-Seidel estimate for U_{ij}^{l+1}, then SOR is a blend of this and the previous SOR iterate U_{ij}^{l}:

$$\tilde{U}_{ij}^{l+1} = \frac{1}{4}[U_{i+1,j}^{l} + U_{i-1,j}^{l+1} + U_{i,j+1}^{l} + U_{i,j-1}^{l+1}] + G \tag{3.42}$$

$$\begin{aligned} U_{ij}^{l+1} &= \omega\tilde{U}_{ij}^{l+1} + (1-\omega)U_{ij}^{l} \\ &= U_{ij}^{l} + \omega(\tilde{U}_{ij}^{l+1} - U_{ij}^{l}) \\ &= \tilde{U}_{ij}^{l+1} + (\omega - 1)(\tilde{U}_{ij}^{l+1} - U_{ij}^{l}) \end{aligned} \tag{3.43}$$

$\omega = 1$ reproduces Gauss-Seidel. $\omega < 1$ can be seen to introduce damping relative to GS; and $\omega > 1$ accelerates the GS process. Generally, we expect $0 < \omega < 2$ for stability.

3.4 Operation Counts

It is useful here to summarize the generalities we have for Elliptic matrix solution methods. We will assume a regular FD domain (unit cube) with Dirichlet boundaries. With

N the number of unknowns
B the matrix bandwidth
M the number of iterations required to converge
s the number of nonzero coefficients per equation
n the number of unknowns in each physical dimension

and a compact molecule, we have the relations summarized in Table 3.1. (The iterative count assumes each iteration requires only a handful of efficient sparse matrix multiplications.)

	Inversion	LU	Iterative
Memory	N^2	BN	sN
Operations	N^3	B^2N	sNM

Table 3.1: Scaling for generic matrix solution strategies. Inversion is the Gold Standard for well-conditioned Elliptic problems.

Fundamentally it is the 1-D node count $n = 1/h$ which tells us about *resolution*, so we need to render Table 3.1 in terms of n, as in Table 3.2.

	1-D			2-D			3-D		
N	n			n^2			n^3		
B	3			n			n^2		
s	3			5			7		
	Inv	LU	Iter	Inv	LU	Iter	Inv	LU	Iter
Memory	n^2	$3n$	$3n$	n^4	n^3	$5n^2$	n^6	n^5	$7n^3$
Operations	n^3	$9n$	$3nM$	n^6	n^4	$5n^2M$	n^9	n^7	$7n^3M$

Table 3.2: Scaling in terms of $n = 1/h$.

One can see in Table 3.2 the progression from 1 to 3 physical dimensions. There has been a stable reliance on LU methods in 1- and 2-D for a long time, reflecting contemporary machinery. But for large problems these are already fading in terms of practical memory requirements, a trend evident in 2-D and overwhelming in 3-D. So the operation counts are critical. Generally, iterations can be expected to slow down as N (or n) increases, so a key property of iterative methods will be M, the number of iterations required for convergence.

For example, we know (see Table 4.1) that for Jacobi and Gauss-Seidel, we have $M \sim n^2$ in 2-D; and that SOR can achieve $M \sim n$ for optimal relaxation. So even in 2-D, the simple iterative methods are competitive.

Many important applications today are fundamentally 3-D, and memory requirements alone are making iterative methods necessary. There is a big window of opportunity in the *3-D iteration count*, from n^7 (the operation count for 3-D LU) to Mn^3. Finding practical iterative methods which have $M \sim n^4$ is a major frontier in 3-D. Recall that $N = n^3$ for 3-D; so a practical 3-D target is achieving $M \sim N^{4/3}$.

3.5 Advective-Diffusive Equation

One of the most common PDE's is the Advective-Diffusive equation, which in steady-state is Elliptic. In typical occurrence, it results from a conservation statement governing the flux q of a physical quantity

$$\nabla \cdot q = 0 \tag{3.44}$$

plus a constitutive relationship between q and a scalar U:

$$q = -D\nabla U + VU \tag{3.45}$$

The result is the familiar Advective-Diffusive equation

$$D\nabla^2 U - V \cdot \nabla U = 0 \tag{3.46}$$

The 1-D form is

$$D\frac{d^2U}{dx^2} - V\frac{dU}{dx} = 0 \tag{3.47}$$

We will assume D and V are constant and > 0. It is useful to introduce the dimensionless form in terms of the coordinate $\chi = x/L$

$$\frac{d^2U}{d\chi^2} - P_e\frac{dU}{d\chi} = 0 \tag{3.48}$$

wherein the dimensionless Peclet number $P_e = \frac{VL}{D}$ arises naturally as the ratio of advective to diffusive effects. We will examine the FD form with centered second-order differencing for the second derivative, and three different first derivative approximations:

$$\frac{\delta^2 U_i}{h^2} - P_e(?) = 0 \tag{3.49}$$

Here h is normalized: $h = \Delta\chi = \Delta x/L$. The three approximating molecules are illustrated in Figure 3.8.

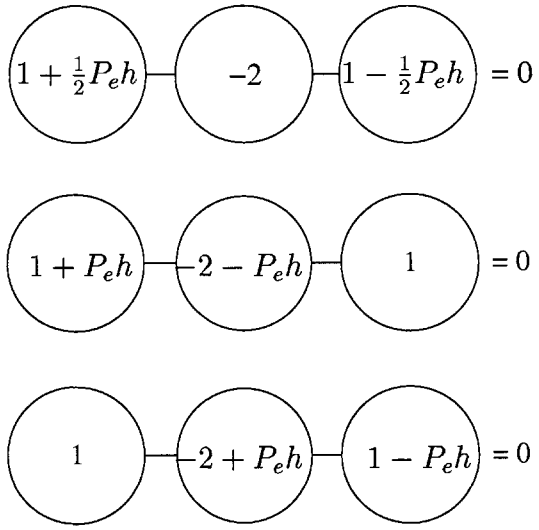


Figure 3.8: Centered, Upstream, and Downstream difference approximations to the Advective-Diffusive equation.

Centered Differencing

The centered molecule represents the difference equation

$$U_{i-1}[1 + \frac{P_e h}{2}] - 2U_i + U_{i+1}[1 - \frac{P_e h}{2}] = 0 \tag{3.50}$$

This is the baseline case; it is $O(h^2)$ in the Taylor series sense. We have weak diagonal dominance when

$$P_e h < 2 \tag{3.51}$$

$P_e h = \frac{V\Delta x}{D}$ is commonly referred to as the *cell Peclet number.* It is *the* measure of resolution here. Diagonal dominance is lost when this resolution is coarse.

The difference equations can be solved exactly as follows. We seek a solution of the form

$$U_i = \rho^i \tag{3.52}$$

Substituting into the difference equation, we have

$$U_i = \rho U_{i-1} \qquad\qquad U_{i+1} = \rho^2 U_{i-1}$$

and thus the quadratic equation

$$\left([1 + \frac{P_e h}{2}] - 2\rho + [1 - \frac{P_e h}{2}]\rho^2\right) U_{i-1} = 0 \tag{3.53}$$

Solving this for ρ gives us

$$\begin{aligned}\rho &= \frac{2 \pm \sqrt{4 - 4[1 + \frac{P_e h}{2}][1 - \frac{P_e h}{2}]}}{2[1 - \frac{P_e h}{2}]} \\ &= \frac{1 \pm \sqrt{1 - \{1 - \frac{P_e h}{2}\}^2}}{[1 - \frac{P_e h}{2}]} \end{aligned} \tag{3.54}$$

$$\rho = \frac{1 \pm \frac{P_e h}{2}}{1 - \frac{P_e h}{2}} \tag{3.55}$$

$$\rho_1 = 1 \quad ; \quad \rho_2 = \frac{1 + \frac{P_e h}{2}}{1 - \frac{P_e h}{2}} \tag{3.56}$$

The solution is thus a linear combination of these two modes

$$U_i = A + B\rho_2^i \tag{3.57}$$

with A and B free to fit the two boundary conditions.

The exact solution to the continuous PDE may be similarly obtained, as

$$U = A + B \exp(P_e x) \tag{3.58}$$

and it is apparent that a measure of accuracy is the correspondence between $\exp(P_e x)$ and ρ_2. As $h \to 0$, ρ converges with $O(P_e h)^3$, as can be verified from a Taylor series expansion:

$$\rho_2 \to \exp(P_e h) \quad as \quad P_e h \to 0 \tag{3.59}$$

For large h, however, ρ becomes negative and correspondence with the analytic solution is lost. The numerical solution in that case exhibits spurious node-to-node oscillations. Note that the onset of this oscillatory behaviour accompanies the loss of diagonal dominance.

$$\rho_2 < 0 \quad for \quad P_e h \equiv \frac{V \Delta x}{D} > 2 \tag{3.60}$$

As an example, consider the BC's $x = 0, U = 1, x = 1, U = 0$. The analytic solution is sketched in Figure 3.9, with increasing steepness near the "downstream" boundary with increasing Peclet number. The well-resolved case, $\frac{V \Delta x}{D} < 2$, exhibits the same qualitative behaviour with increasing fidelity as $h \to 0$. The poor resolution case is sketched in Figure 3.10, with the oscillatory mode having no analytic counterpart. The spurious oscillations in this case are a symptom of poor resolution.

Upstream Weighting

As an alternative, consider the first-order backward (in the sense of advection) differencing of the term $\frac{du}{dx}$ as shown in Figure 3.8.

$$\frac{\delta^2 U_i}{h^2} - P_e(\frac{U_i - U_{i-1}}{h}) = 0 \tag{3.61}$$

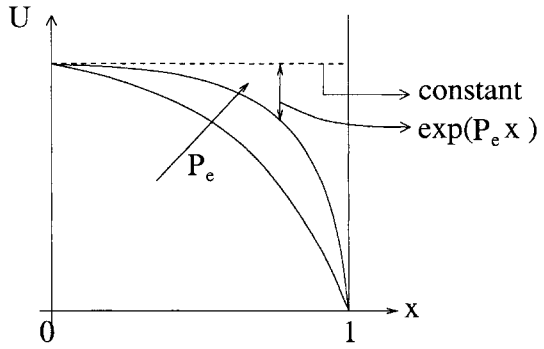


Figure 3.9: Analytic solution to the Advective-Diffusive equation, with P_e as a parameter.

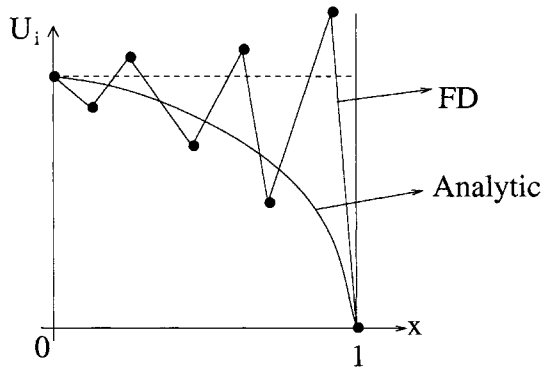


Figure 3.10: Numerical solution to the Advective-Diffusive equation, with poor resolution $P_e h$. The spurious oscillatory mode is the manifestation of $\rho < 0$.

Notice that this approximation, although only first-order correct, retains diagonal dominance for any positive value of P_e. Using the same procedure as above, the difference equations are

$$[1 + P_e h] - [2 + P_e h]\rho + \rho^2 = 0 \tag{3.62}$$

$$\begin{aligned} \rho &= \frac{2 + P_e h \pm \sqrt{[2 + P_e h]^2 - 4[1 + P_e h]}}{2} \\ &= \frac{2 + P_e h \pm P_e h}{2} \end{aligned}$$

$$\rho_1 = 1 \quad ; \quad \rho_2 = 1 + P_e h \tag{3.63}$$

Again, there are two solutions: ρ_1 (a constant); and ρ_2 which is *never negative.* Thus we arrive at the conclusion that the upstream differencing leads to monotone solutions for this simple case, with no spurious oscillations.

Accuracy depends, as above, on the correspondence between ρ_2 and $\exp(P_e h)$. Taylor series confirm that ρ_2 is an $O(h^2)$ approximation (per step), *i.e.* less accurate than its centered counterpart, but monotone over the full range of resolution.

Downstream Weighting

Finally, consider the forward differencing of $\frac{du}{dx}$,

$$\frac{\delta^2 U_i}{h^2} - P_e(\frac{U_{i+1} - U_i}{h}) = 0 \tag{3.64}$$

as shown in Figure 3.8. The analogous analysis leads us quickly to

$$1 + (-2 + P_e h)\rho + (1 - P_e h)\rho^2 = 0 \tag{3.65}$$

$$\rho_1 = 1 \quad ; \quad \rho_2 = \frac{1}{1 - P_e h} \tag{3.66}$$

This again is an $O(h^2)$ approximation to $exp(P_e h)$ as in the upstream case. But now we have re-acquired oscillatory solutions when $P_e h > 1$.

Summarizing, we have the following conditions for *monotone solutions*:

$$Upstream: \quad P_e h < \infty \tag{3.67}$$

$$Centered: \quad P_e h < 2 \tag{3.68}$$

$$Downstream: \quad P_e h < 1 \tag{3.69}$$

The Taylor series per-step truncation errors in ρ are $O(h^2)$ for the uncentered approximations, and $O(h^3)$ for centered. The accumulation of truncation error over many steps is one order lower, in each case. We have recovered the Taylor series conclusions about these molecules: second-order for centered, first-order otherwise. And we expect to see those rates of convergence for small h. The centered approximations are best at high resolution, but risk qualitative infidelity at low resolution.

The strong dependence on the cell Peclet number $P_e h \equiv \frac{V \Delta x}{D}$, which is effectively the dimensionless mesh size for this operator, is generally retained in approximations to more complicated (multidimensional, transient, nonlinear) forms of this equation.

Chapter 4

Iterative Methods for Elliptic Equations

Direct methods for linear algebra become cumbersome for large systems, even when they are sparse or banded. This is especially true in 3-D applications where the bandwidth is necessarily broad and sparse, and matrix factorization can dominate computer resources. The alternative iterative approach avoids matrix factorization, and emphasizes simple matrix multiplies and adds. These can be very fast for sparse matrices, and in addition are intuitively related directly to the FD molecules. Iterative methods require an infinite number of repetitions, so a stopping rule is necessary; direct methods terminate at a finite number of operations.

There is a vast literature concerning iterative methods for solving large algebraic systems. The material given here is introductory, preparatory to fuller treatments; see for example Ames[2], Golub and van Loan[35], or Weiss [110].

4.1 Bare Essentials of Iterative Methods

Let us express the collection of FD equations as

$$[A]\{u\} = \{v\} \tag{4.1}$$

We wish to solve this by an iterative method

$$\{u\}^l = [G]\{u\}^{l-1} + \{r\} \tag{4.2}$$

where $[G]$ is the *iteration matrix* and the superscript l indicates sequential iteration number. This is a stationary iteration, *i.e.* neither $[G]$ nor $\{r\}$ depend on the iteration count l. Since we require that $\{u\} = [G]\{u\} + \{r\}$, and substituting $\{u\} = [A^{-1}]\{v\}$, we obtain the requirement that

$$\{r\} \;=\; [I - G][A^{-1}]\{v\} \tag{4.3}$$

This constrains our iterative method: given $[G]$, $\{r\}$ must be consistent.

Errors

We define the error vector at the end of the iteration l to be

$$\{\epsilon\}^l = \{u\}^l - \{u\} = \{u\}^l - [A^{-1}]\{v\} \tag{4.4}$$

Since

$$\{u\}^l = [G]\{u\}^{l-1} + \{r\} \tag{4.5}$$

and

$$\{u\} = [G]\{u\} + \{r\} \tag{4.6}$$

we obtain by subtraction the recursion relation for the errors:

$$\{\epsilon\}^l = [G]\{\epsilon\}^{l-1} \tag{4.7}$$

Equivalently,

$$\{\epsilon\}^l = [G]^{\wedge l}\{\epsilon\}^0 \tag{4.8}$$

(Here we have a notation conflict; by $[G]^{\wedge l}$ we mean "$[G]$ taken to the power l".) This relationship is utilized in proofs of convergence, *etc.* But in computational practice it cannot be used since by hypothesis the array $\{u\}$ (the exact algebraic solution of the FD equations) is unknown. Note that $\{\epsilon\}^l$ as defined here is *not* the difference between the exact analytic (calculus) solution to a PDE and any finite solution. It is the distance between *algebraic* truth and its iterative approximation.

Increments

The increment to the solution vector which occurs during the l^{th} iteration is

$$\{\delta\}^l = \{u\}^l - \{u\}^{l-1} \tag{4.9}$$

Writing equation (4.2) for $\{u\}^l$ and also for $\{u\}^{l-1}$ and subtracting yields the recursion relation for the increments:

$$\{\delta\}^l = [G]\{\delta\}^{l-1} \tag{4.10}$$

While the errors $\{\epsilon\}$ cannot be measured, the increments $\{\delta\}$ can and thus $\{\delta\}$ provides useful computational information on the progress of an iterative method.

Residuals

Another measurable vector is the residual of the algebraic system after the l^{th} iteration, which must vanish as $l \to \infty$:

$$\{R\}^l = \{v\} - [A]\{u\}^l \tag{4.11}$$

or, taking advantage of equation (4.4),

$$\{R\}^l = [A]\left([A^{-1}]\{v\} - \{u\}^l\right) = -[A]\{\epsilon\}^l \tag{4.12}$$

Use of the recursions for $\{\epsilon\}^l$ (equations 4.7 and 4.8) yields

$$\{R\}^l = -[A][G]\{\epsilon\}^{l-1} = [A][G][A^{-1}]\{R\}^{l-1} \tag{4.13}$$

and

$$\{R\}^l = [A][G]^{\wedge l}[A^{-1}]\{R\}^0 \tag{4.14}$$

Convergence

The iteration (4.2) converges if and only if the spectral radius ρ of the iteration matrix $[G]$ is less than unity. By definition, ρ is the largest (in absolute value) eigenvalue of $[G]$. Noting that the eigenvalues of $[A][G][A^{-1}]$ are the same as those of $[G]$, we have in the limit of large l:

$$\{\epsilon\}^l \simeq \rho\{\epsilon\}^{l-1} \tag{4.15}$$

$$\{\delta\}^l \simeq \rho\{\delta\}^{l-1} \tag{4.16}$$

$$\{R\}^l \simeq \rho\{R\}^{l-1} \tag{4.17}$$

The second and third of these are useful in estimating ρ. For example,

$$\rho \simeq \frac{||\delta^l||}{||\delta^{l-1}||} \tag{4.18}$$

where the $||\delta||$ notation indicates the Euclidean norm (length) of a vector of length M:

$$||\delta|| = (\sum_{i=1}^{M} \delta_i^2)^{1/2} \tag{4.19}$$

Other norms of the form

$$||\delta|| = (\sum_{i=1}^{M} |\delta_i|^N)^{1/N} \tag{4.20}$$

are also useful, the most common being N=1 and the limiting case N=∞:

$$||\delta|| = \max_i |\delta_i| \tag{4.21}$$

4.2 Point Iterative Methods

Consider the second-order, quasilinear elliptic equation

$$\frac{\partial}{\partial x}(a\frac{\partial U}{\partial x}) + \frac{\partial}{\partial y}(c\frac{\partial U}{\partial y}) + d\frac{\partial U}{\partial x} + e\frac{\partial U}{\partial y} + fU = g \tag{4.22}$$

with a, c, d, e, f and g known functions of x, y, U and the first derivatives of U. Note the coefficient of the mixed derivative $\frac{\partial^2 U}{\partial x \partial y}$ has been set to zero. We assume the following properties:

$a > 0$ (this is an arbitrary choice)
$c > 0$ (necessary for an elliptic operator if $a > 0$)
$f \leq 0$ (stability of the underlying physical process)

We will use the following difference approximation for the second derivatives:

$$\frac{\partial U}{\partial x}(a\frac{\partial U}{\partial x}) = \frac{1}{h}\left[a_{i+1/2}(\frac{U_{i+1} - U_i}{h}) - a_{i-1/2}(\frac{U_i - U_{i-1}}{h})\right] \tag{4.23}$$

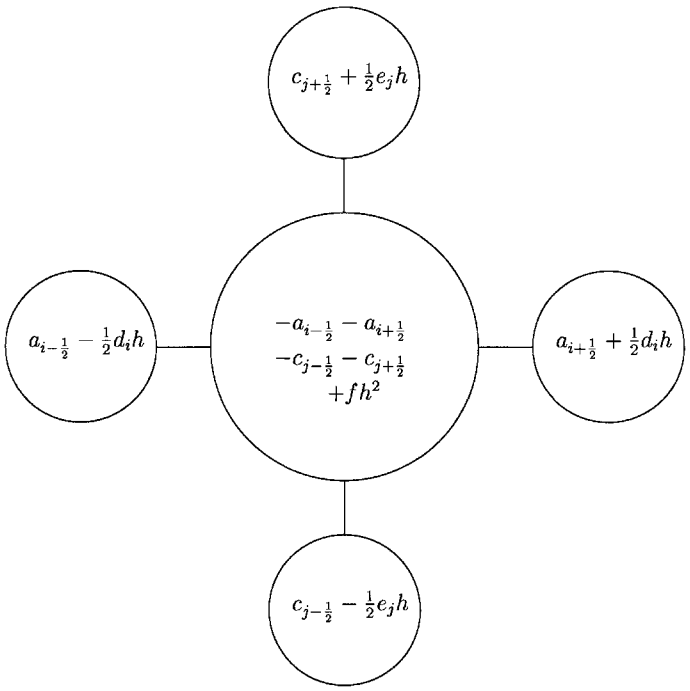


Figure 4.1: Molecule for the elliptic operator (4.22)

and centered differences for the first derivative terms. The resulting molecule is $O(h^2)$ on a uniform mesh and is illustrated in Figure 4.1.

In matrix form, we have

$$[A]\{U\} = \{V\} \tag{4.24}$$

where $V = h^2 g$ plus contributions from the inhomogeneous BC's. The following properties of $[A]$ are defined:

- *Diagonal Dominance*: $A_{ii} \geq \sum_{j \neq i} |A_{ij}|$, with strict inequality for some i. For small h, this will be guaranteed in the weak sense; and a single Dirichlet boundary condition will provide the strict inequality necessary. We will lose diagonal dominance when $\frac{dh}{2}$ exceeds a, or when $\frac{eh}{2}$ exceeds c. This is the Peclet number problem identified earlier in the 1-D analysis.
- *Symmetry*: $(A_{ij} = A_{ji})$ this property is guaranteed for the self-adjoint case $d = e = 0$, even when we have variable coefficients a, c. This is the consequence of the specific treatment of the second derivatives used here.
- *Irreducibility*: this property amounts to the requirement that all V_i impact all U_i. This is guaranteed for an elliptic problem on a simply connected domain.

We partition $[A]$ into its entries below, on, and above the main diagonal:

$$[A] = [C]\ (below) + [D]\ (Diagonal) + [E]\ (above) \tag{4.25}$$

Recalling its definition (equation 4.2), we can construct the iteration matrix $[G]$ for the standard trio of point iterative methods:

- Jacobi:[1]

$$[C]\{U\}^l + [D]\{U\}^{l+1} + [E]\{U\}^l = \{V\} \tag{4.26}$$
$$\{U\}^{l+1} = -[D^{-1}][C + E]\{U\}^l + [D^{-1}]\{V\} \tag{4.27}$$
$$[G_J] = -[D^{-1}][C + E] \tag{4.28}$$

- Gauss-Seidel:

$$[C + D]\{U\}^{l+1} = -[E]\{U\}^l + \{V\} \tag{4.29}$$
$$[G_G] = -[C + D]^{-1}[E] \tag{4.30}$$

- SOR:

$$[C]\{U\}^{l+1} + [D]\{\tilde{U}\} = -[E]\{U\}^l + \{V\} \tag{4.31}$$
$$\{U\}^{l+1} = \omega\{\tilde{U}\} + (1-\omega)\{U\}^l \tag{4.32}$$
$$[\omega C + D]\{U\}^{l+1} = [(1-\omega)D - \omega E]\{U\}^l + \omega\{V\} \tag{4.33}$$
$$[G_\omega] = [\omega C + D]^{-1}[(1-\omega)D - \omega E] \tag{4.34}$$

We know that these iterations will converge if and only if $\rho(G) < 1$. ($\rho(G)$ is the spectral radius of G). The following are known:

- [A] Diagonally Dominant: Jacobi and G-S will converge.
- [A] Symmetric, Positive Definite: Jacobi, G-S, and SOR will converge, the latter requiring $0 < \omega < 2$.
- Generally if Jacobi converges, G-S will converge faster and SOR will be optimal for some value of ω.

As an example, there is a simple demonstration of convergence for Jacobi and Gauss-Seidel when $[A]$ is diagonally dominant. First, define θ_i as a measure of diagonal dominance for row i of $[A]$:

$$\theta_i = \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} \tag{4.35}$$

and recall that the error satisfies the homogeneous form of the iteration:

$$\{\epsilon\}^{l+1} = [G]\{\epsilon\}^l \tag{4.36}$$

Writing the **Jacobi** iteration, we have:

$$\{\epsilon\}_i^{l+1} = \frac{-1}{a_{ii}}[\sum_{j \neq i} a_{ij}\epsilon_j^l] \tag{4.37}$$

[1] $[G_J]$ is the Jacobi iteration matrix, etc.

$$|\epsilon_i^{l+1}| \leq \sum_{j \neq i} \frac{|a_i j|}{|a_{ii}|} |\epsilon_j|^l \tag{4.38}$$

$$|\epsilon_i^{l+1}| \leq \theta_i ||\epsilon||^l \tag{4.39}$$

where $||\epsilon||$ is the infinity norm of $\{\epsilon\}$,

$$||\epsilon||^l = \max_j |\epsilon_j|^l \tag{4.40}$$

In the worst case,

$$||\epsilon||^{l+1} \leq \theta_{max} ||\epsilon||^l \tag{4.41}$$

and thus $\theta_{max} < 1$ is sufficient for convergence. For the elliptic equation used here, with no Dirichlet BC's, $\theta_{max} = 1$ and thus Jacobi will not diverge.

A similar demonstration for **Gauss-Seidel** can be made. Here we will assume $\theta_i < 1$ for all i.

$$\epsilon_1^{l+1} = \frac{-1}{a_{ii}} [\sum_{j=2}^{N} a_{1j} \epsilon_j^l \Rightarrow |\epsilon_1^{l+1}| \leq \theta_1 ||\epsilon||^l \leq ||\epsilon||^l \tag{4.42}$$

$$\epsilon_2^{l+1} = \frac{-1}{a_{22}} [a_{21} \epsilon_1^{l+1} + \sum_{j=3}^{N} a_{2j} \epsilon_j^l] \tag{4.43}$$

$$|\epsilon_2|^{l+1} \leq \frac{|a_{21}|}{|a_{22}|} ||\epsilon||^l + \sum_{j=3}^{N} \frac{|a_{2j}|}{|a_{22}|} |\epsilon_j|^l \leq \theta_2 ||\epsilon||^l \tag{4.44}$$

and so on. Thus, diagonal dominance is sufficient for Gauss-Seidel convergence.

There is a general set of findings for if matrix $[A]$ (equation 4.24) is Symmetric, Consistently Ordered, and has "Property A" [102]. First,

$$\rho_{GS} = \rho_J^2 \tag{4.45}$$

and thus G-S will converge or diverge faster than Jacobi. Second, there is an optimal value of ω which minimizes ρ_ω for SOR:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho_J^2}} = \frac{2}{1 + \sqrt{1 - \rho_{GS}}} \tag{4.46}$$

Thus it is possible to operate an SOR iteration with $\omega = 1$ initially, in order to estimate ρ_{GS}, and to switch to ω_{opt} once that estimate is reliable.

Finally, we summarize some results for $\nabla^2 U = 0$ on a square, length π, with Dirichlet BC's:

$$\rho_J = \cos h \simeq 1 - h^2/2 \tag{4.47}$$

$$\rho_{GS} = \cos^2 h \simeq 1 - h^2 \tag{4.48}$$

$$\rho_{\omega_{opt}} \simeq 1 - 2h \tag{4.49}$$

The approximations are the limiting ones for small h. Ideally, ρ remains as far as possible from 1, such that each iteration is effective. In each case, however, $\rho \to 1$ as $h \to 0$. This is a *serious tendency*, indicating that *these iterations slow down as resolution increases.* There is a double effect: each iteration requires more computation, and we need more iterations overall. It is clear

that the $0(h)$ convergence of SOR with optimal ω is an asset here; the retuning of ω as h decreases counteracts some of the loss of iterative effectiveness experienced by Jacobi and G-S.

In the limit of large l, we have

$$||\epsilon||^{l+M} = \rho^M ||\epsilon||^l \tag{4.50}$$

Thus for error reduction by the factor κ, we need $\rho^m = \kappa$ *i.e.* the required number of iterations is

$$M = ln(\kappa)/ln(\rho) \tag{4.51}$$

Combining the above, again for small h, we have

$$ln(\rho_J) \simeq -h^2/2 \tag{4.52}$$

$$ln(\rho_{GS}) \simeq -h^2 \tag{4.53}$$

$$ln(\rho_{\omega_{opt}}) \simeq -2h \tag{4.54}$$

and therefore for fixed κ, we have the relative iteration counts as follows:

$$M_J \simeq \frac{2}{h^2} \tag{4.55}$$

$$M_{GS} \simeq \frac{1}{h^2} = \frac{1}{2} M_J \tag{4.56}$$

$$M_{\omega_{opt}} \simeq \frac{1}{2h} = \frac{h}{2} M_{G-S} \tag{4.57}$$

The relative speedup in SOR as h becomes small is apparent. (Remember that these are 2-D results.) The impact of M on overall iteration efficiency in multi-dimensional applications is discussed herein at Table 3.2 and its attendant text.

Ames [2] provides more complete detail and an excellent link to the older literature. Westlake [112] (Appendix B therein) contains a useful collection of theorems on Eigenvalue Bounds.

4.3 Block Iterative Methods

In the previous methods, each new value U_{ij}^{l+1} is updated alone – hence their characterization as *point* iterative methods. A generalization of these methods is possible, wherein a *block* or *group* of unknowns is updated simultaneously. These go by various designations: block, line, group, or implicit iterative methods. Each iteration necessarily involves solving a matrix equation for the block of simultaneous updates. Of course this matrix needs to be much simpler than the original matrix being solved; otherwise the iteration is a bad idea. A good reference is Ames [2].

Figure 4.2 illustrates the Jacobi and Gauss-Seidel block methods in molecular form. Both molecules require a stationary tridiagonal matrix solution for each iteration, representing the implicit x-derivative portion of the molecule. As above, the iteration counts scale as $-ln(\kappa)/-ln(\rho)$, where κ is the error reduction factor and ρ is the spectral radius of the iteration. For the Laplace Equation on a square, the Line and Point versions are compared in Table 4.1. For Jacobi and G-S, the payoff for the tridiagonal matrix solution is a decrease by a factor of two in the number of iterations required. For SOR, the reduction is by the factor $\sqrt{2}$. These modest iteration count improvements need to be balanced against the compute time per iteration added by the line methods.

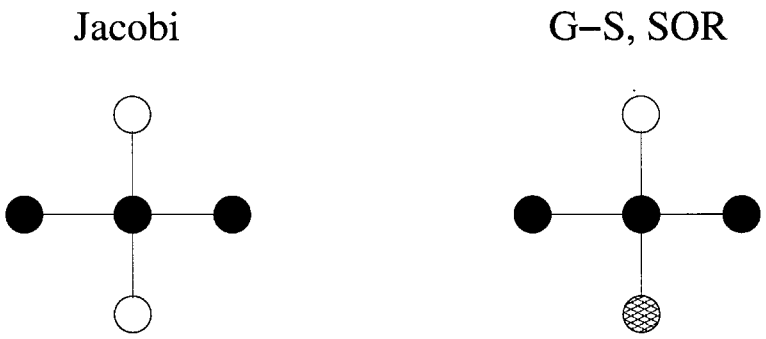


Figure 4.2: Jacobi and Gauss-Seidel/SOR (right) Line Iterative molecules. Black circles indicate the current unknown line; white, lagged in the iteration; hatched, current iteration but already computed in the previous line.

	Point	Line
Jacobi	$2/h^2$	$1/h^2$
G-S	$1/h^2$	$1/2h^2$
SOR(optimal)	$1/2h$	$1/\sqrt{2}\cdot 2h$

Table 4.1: $-1/ln(\rho)$ for point and line iterations, 2-D Laplace on a square. The number of iterations M required for a given error reduction is proportional to this.

Alternating Direction Methods

Each of the above methods is implicit in the x-directed part of the Laplacian, but explicit in the y part. This generates an obvious prejudice in the molecule. A straightforward "fix" involves alternating the direction of the implicitness: first x-implicit, then y-implicit, then x, and so forth. This preserves the tridiagonal structure in each iteration. For example, consider the elliptic equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + fu = g \tag{4.58}$$

and its FD form:

$$\partial_x^2 U_{ij} + \partial_y^2 U_{ij} + fh^2 U_{ij} = gh^2 \tag{4.59}$$

Step 1 is implicit in x:

$$(\partial_x^2 + fh^2/2)U_{ij}^{l+1} - \omega U_{ij}^{l+1} = -(\partial_y^2 + fh^2/2)U_{ij}^{l} - \omega U_{ij}^{l} + gh^2 \tag{4.60}$$

Note that an iteration parameter ω has been inserted into the calculation, to regulate convergence through the term $\omega(U_{ij}^{l+1} - U_{ij}^{l})$. The term fU has been split between the two iteration levels.

Step 2 is implicit in y:

$$(\partial_y^2 + fh^2/2)U_{ij}^{l+2} - \omega U_{ij}^{l+2} = -(\partial_x^2 + fh^2/2)U_{ij}^{l+1} - \omega U_{ij}^{l+1} + gh^2 \tag{4.61}$$

It is emphasized that this is a two-step procedure; the intermediate result (*e.g.* the odd-numbered iterations) is unreliable, containing unwanted information which is removed as the second step is

completed. The overall algorithm can be realized with only stationary, tridiagonal matrix solution technology.

The molecule for Step 1 is shown in Figure 4.3. The Step 2 molecule simply interchanges the role of x- and y- differences.

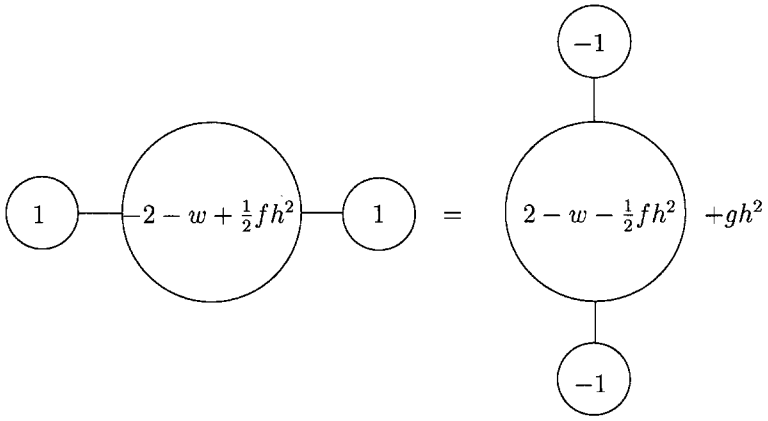


Figure 4.3: Iterative ADI molecule for Step 1, the "x-sweep". The y-derivative operates on known information from the previous iteration.

The iteration matrix for this process is obtained as follows. First, introduce the matrix form of the FD molecule:

$$([X]+[Y]+[F])\{U\}=\{R\} \tag{4.62}$$

where $[X]\{U\}$ is the matrix form of the x-derivative, etc. The right-side vector $\{R\}$ contains gh^2 plus contributions from inhomogeneous boundary conditions. In these terms, we have:

Step 1:

$$[X+\frac{1}{2}F-\omega I]\{U\}^{l+1}=[-Y-\frac{1}{2}F-\omega I]\{U\}^{l}+\{R\} \tag{4.63}$$

Step 2:

$$[Y+\frac{1}{2}F-\omega I]\{U\}^{l+2}=[-X-\frac{1}{2}F-\omega I]\{U\}^{l+1}+\{R\} \tag{4.64}$$

Eliminating the intermediate result, we have the iteration matrix

$$[G_{ADI}]=[Y+\frac{1}{2}F-\omega I]^{-1}[-X-\frac{1}{2}F-\omega I][X+\frac{1}{2}F-\omega I]^{-1}[-Y-\frac{1}{2}F-\omega I] \tag{4.65}$$

Analysis of the spectrum of $[G_{ADI}]$ and/or Fourier Analysis of the difference equations provides the following results:

- convergence for $\omega > 0$; divergence for $\omega < 0$
- neutral ($\rho = 1$) for $\omega = 0$; the iteration stalls or accumulates roundoff error monotonically
- convergence is slowest for the smoothest solution modes; faster for highly variable modes
- $\rho \rightarrow 1$ as $h \rightarrow 0$, as in all the other iterative methods studied
- $f < 0$ speeds convergence
- There is an optimal ω for the Laplace Equation on a square, Type 1 BC's:

$$\omega_{opt}=\left[\left(-\frac{h^2f}{2}+4sin^2(\frac{\pi}{2N})\right)\left(-\frac{h^2f}{2}+4cos^2(\frac{\pi}{2N})\right)\right]^{1/2} \tag{4.66}$$

where N is the maximum number of nodes in either the x- or y- direction. (Smith [102]) Thus for large N,

$$\omega_{opt} \to \left[\left(-\frac{h^2 f}{2}\right)\left(-\frac{h^2 f}{2}+4\right)\right]^{1/2} \tag{4.67}$$

- as $h \to 0$: ADI convergence $\sim$ SOR convergence (both with optimal ω).

One typically disregards the intermediate (l+1) solution as unreliable. It is possible to define a sequence of ω^l values, which can dramatically improve ADI convergence, such that $-\ln(\rho) \sim h^{1/M}$ for a sequence of length M. A more detailed review of these ideas is given in [2].

4.4 Helmholtz Equation

As an example, consider the Wave Equation

$$\frac{\partial^2 U}{\partial t^2} = \frac{\partial^2 U}{\partial x^2} - fU \tag{4.68}$$

We assume positive, real f on physical grounds. This is a Hyperbolic equation; its Fourier transform leads to the very useful (Elliptic) Helmholtz equation

$$\frac{\partial^2 U}{\partial x^2} + (\omega^2 - f)U = 0 \tag{4.69}$$

where ω is the Fourier frequency, assumed real. Analytic solutions exist of the form $e^{j\sigma x}$, with $j = \sqrt{-1}$ and the dispersion relation

$$\sigma^2 = \omega^2 - f \tag{4.70}$$

For $\omega^2 - f > 0$ there will be spatially-periodic solutions with constant amplitude. Otherwise, for large f, solutions will be spatially "trapped" *i.e.* pure exponentials decaying in either the positive or negative x-direction.

A conventional FD rendering is

$$U_{i-1} - [2 - W^2 + F]U_i + U_{i+1} = 0 \tag{4.71}$$

with $W = \omega h$, $F = fh^2$, and h the mesh spacing. We instantly have a problem with diagonal dominance. There are 2 possibilities for *achieving* diagonal dominance:

- case A: $2 - W^2 + F > 2$, or $W^2 - F < 0$. This corresponds to the trapped, spatially decaying analytic solutions, exactly reproducing the analytic threshold independent of h.
- case B: $-2 + W^2 - F > 2$, or $W^2 - F > 4$. This corresponds to very poorly resolved periodic solutions with large h; their diagonal dominance is produced by poor resolution.

Well-resolved periodic solutions will have $0 < W^2 - F < 4$. This represents a broad class of important problems; they will not be diagonally dominant, and we can expect problems with simple iterative methods.

This loss of diagonal dominance will not translate necessarily into a poor FD solution, obtained *e.g.* by noniterative methods. If we assume a solution to equation 4.71 in the form $U_i = \lambda^i$, we obtain the characteristic quadratic

$$\lambda^2 - [2 - W^2 + F]\lambda + 1 = 0 \tag{4.72}$$

for which the solution is

$$\lambda = \frac{[2 - W^2 + F]}{2} \pm \sqrt{\left(\frac{[2 - W^2 + F]}{2}\right)^2 - 1} \tag{4.73}$$

or equivalently,

$$\lambda = \frac{[2 - W^2 + F]}{2} \pm j\sqrt{1 - \left(\frac{[2 - W^2 + F]}{2}\right)^2} \tag{4.74}$$

(remember j is the imaginary unit). The first form (4.73) is natural for the trapped case $W^2 - F < 0$. In that case λ is purely real, representing geometric growth and/or decay with x, qualitatively the same as the analytic solution. (The positive option will exceed unity; the negative option will fall between 0 and 1.) The fidelity of these will depend on the resolution h.

The second form (4.74) is useful for the periodic case $W^2 - F > 0$. In this case λ is complex and

$$|\lambda|^2 = 1 \tag{4.75}$$

Solutions will be periodic in x, with uniform amplitude, qualitatively like their analytic counterparts. Again, resolution will determine quantitative skill. But the loss of diagonal dominance for these quality elliptic solutions is important to note here; and the likely failure of simple point iterative methods which work for other Elliptic problems.

4.5 Gradient Descent Methods

Returning to the basic assembly of FD relations, we have for the linear elliptic equation

$$[A]\{U\} = \{V\} \tag{4.76}$$

with $[A]$ and $\{V\}$ known. The residual of this equation was defined before,

$$\{R\} \equiv \{V\} - [A]\{U\} \tag{4.77}$$

The goal of an iterative method is the practical vanishing of $\{R\}$. A sequence of approximations $\{U\}^l$ is implied, and a sequence of $\{R\}^l$ also. *We restrict ourselves here to symmetric, positive definite* $[A]$.

Related is the metric

$$\Phi = \frac{1}{2}\{U\}^T[A]\{U\} - \{U\}^T\{V\} \tag{4.78}$$

(The superscript T indicates transposition.) This defines a surface in the N-dimensional space of possible $\{U\}$. It is unbounded, with a unique minimum. Its gradient is

$$\{\nabla\Phi\} = -\{R\} \tag{4.79}$$

The negative gradient defines the local direction of steepest descent. The point where $\{\nabla\Phi\} = 0$ is the extremum of Φ. That is the point where $[A]\{U\} = \{V\}$; a successful iteration finds that point.

A *gradient descent* method thus seeks to go downhill on the Φ surface, toward the extremum, by updating its position in U-space. Locally, the way down is given by the negative gradient - conveniently, the residual here. The linear form of this descent is

$$\{U\}^{l+1} = \{U\}^l + [H]\{R\}^l \tag{4.80}$$

Clearly this iteration will stop if and when the gradient (residual) is zero *i.e.* when the governing equations are satisfied. The answer $\{U\}$ is then in hand.[2]

In addition to a stopping rule, there are two questions in each iteration:
- what is the direction of descent?
- how far to go in that direction?

The first question essentially asks about the unit vector parallel to $[H]\{R\}^l$; the second question asks for its scalar size. The *steepest descent* method always selects the direction of the negative gradient:

$$\{U\}^{l+1} = \{U\}^l + \alpha \{R\}^l \tag{4.81}$$

Essentially, $[H] = \alpha [I]$ and there is only one free scalar parameter, the step size α.

For example, the Jacobi iteration is conveniently rearranged using the matrix partitioning of equation 4.25, $[A] = [C + D + E]$:

$$\begin{aligned} [D]\{U\}^{l+1} &= -[C+E]\{U\}^l + \{V\} && (4.82)\\ &= [D]\{U\}^l - [C+D+E]\{U\}^l + \{V\} && (4.83)\\ &= [D]\{U\}^l + R^l && (4.84) \end{aligned}$$

So we have for Jacobi,

$$[H] = [D]^{-1} \tag{4.85}$$

If the diagonals D_i are all the same (true for Laplace on a square with Dirichlet BC's), then we have in Jacobi a *steepest descent method with fixed step size* $1/D_i$. Pre-scaling all the FD equations by their own diagonals (always a good idea) adjusts the Jacobi method to be exactly this; $[D]$ becomes $[I]$ in the scaled equations.

With the direction of steepest descent selected, one can seek the *best* step size. This constitutes a 1-D optimization – minimize Φ along a given line – and it admits a closed form solution for the *optimal step size*:

$$\alpha = \frac{\{R\}^T \{R\}}{\{R\}^T [A] \{R\}} \tag{4.86}$$

Use of this constitutes the *Method of Steepest Descent with Optimal Step Size.* Since the direction is the same, this optimum cannot be inferior to the diagonally-scaled Jacobi.

Putting the Gauss-Seidel method in this context gives us

$$\begin{aligned} [C+D]\{U\}^{l+1} &= -[E]\{U\}^l + \{V\} && (4.87)\\ &= [C+D]\{U\}^l + R^l && (4.88)\\ [H] &= [C+D]^{-1} && (4.89) \end{aligned}$$

The known speedup over Jacobi is coincident here with *not* using steepest descent; there is natural geometric sense in this when Φ contours are awkward.

So we imagine a more general descent process with a sequence of directions $\{d\}^l$ which may not be parallel to the negative gradient. An iteration based on these ideas is:

$$\{U\}^l = \{U\}^{l-1} + \alpha^l \{d\}^l \tag{4.90}$$

[2] Elementary substitution reveals that $[H]$ here is related to the iteration matrix $[G]$ defined above (equation 4.2): $[G] = [I - HA]$

Once the direction of descent $\{d\}^l$ is selected somehow, the optimal step size can be found analogously:

$$\alpha^l = \frac{\{d\}^{lT} \{R\}^{l-1}}{\{d\}^{lT} [A] \{d\}^l} \tag{4.91}$$

The Conjugate Gradient Method is a classic rendering of these ideas. It selects the directions to be linearly independent of their predecessors, and specifically to be *A-conjugate* to them:

$$\{d\}^{lT} [A] \{d\}^m = 0 \tag{4.92}$$

The directions turn out to be cognizant of the current gradient, as expected; but they are neither parallel to it nor orthogonal to it. Such a direction sequence, coupled with the optimal step size, has some remarkable properties, including the surprise that it converges to the exact solution in N steps – it is a direct method! The direction sequence is easily and efficiently computed. And, early progress toward the solution is rapid. But, because round-off error is amplified with this method, it is looked upon as an iterative method which should be terminated early, certainly before N iterations are reached.

The early form of this method appeared in 1952 [42]. It and its variants ("Conjugate Direction Methods") now constitute an important class of iterative methods which deserve a fuller exposition. The reader is referred to Golub and van Loan [35] and Weiss [110].

Chapter 5

Parabolic Equations

5.1 Introduction

Next we turn to time-dynamic problems. To start with, we will consider the class of Parabolic PDE's. The canonical form is

$$\frac{\partial U}{\partial t} = L(U) \tag{5.1}$$

with $L(U)$ an *Elliptic* operator. The standard example is the diffusion equation

$$\frac{\partial U}{\partial t} = \nabla \cdot (D \nabla U) \tag{5.2}$$

with t corresponding physically to time and ∇ operating in 1- to 3-dimensional physical space. Intuitively, these are problems where the time domain is open and directed forward. IC's determine the future, not the past; BC's bound the temporal evolution within a closed spatial domain; their influence propagates forward in time but not backward. The graphical description of the necessary and sufficient conditions adopted in Section 1.4 is redrawn here for the 2-D (x, t) Parabolic PDE.

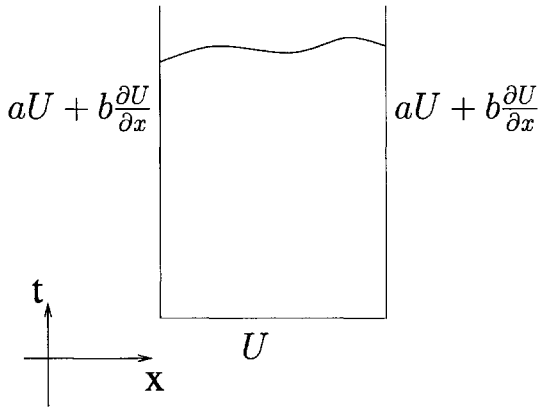


Figure 5.1: Necessary and sufficient conditions for the Parabolic equation.

Essentially, we need one IC as a function of x, $U(x)$ at $t = 0$; and two BC's throughout all time. The form of the BC's are unchanged from the Elliptic discussion – either Type 1, Type 2, or Type 3 is needed everywhere on a closed (spatial) boundary surrounding the elliptic dimensions. The distinction here is that this boundary information is needed as a function of time, for all $t \geq 0$.

Everything we have learned about discretizing elliptic operators will remain the same here. Accordingly, it will be useful to think about discretization in two stages. Starting from the PDE, we will first discretize the elliptic part, leaving time continuous. Formally, this converts the continuous or "Distributed" System of the PDE into a Lumped System of ODE's:

$$\frac{dU_i}{dt} = L_i(U_i) \qquad i = [1, N] \tag{5.3}$$

with $L_i(U_i)$ the FD operator or "molecule" for the elliptic part of the PDE. Implied in this lumping is the discretization of $U(x,t)$ into the finite, N-dimensional vector of nodal functions $U_i(t)$, the application of BC's, and the enforcement of the PDE approximation only at the discrete centers of the molecules i. This is no surprise, and generally the identical maneuver used in discretizing elliptic equations *per se*. The Lumped System adds dynamics, in this case parabolic dynamics. The Lumped System description is useful for the discussion of BC's, conservation properties, and some general observations about dynamics which are independent of the temporal discretization. Notice that we are doing something classical: practicing space-time separation of variables, and of the PDE operator itself.

Temporal discretization can be approached in two different ways:

- using standard FD formulae for $\frac{d}{dt}$ in 5.3, for example,

$$\frac{U_i^{l+1} - U_i^l}{\Delta t} = L_i(U_i^{l+\theta}) \tag{5.4}$$

  with $U_i^{l+\theta}$ defined at some intermediate time $t + \theta\Delta t$; and Δt the timestep from l to $l+1$;

- integrating 5.3 over an interval Δt. Example:

$$U_i^{l+1} - U_i^l = \int_t^{t+\Delta t} L_i(U_i) = \Delta t \overline{L_i(U_i)} \tag{5.5}$$

  with the overline indicating a temporal average.

Both of these approaches lead to the same result: a system of difference equations in a set of unknowns U_i^l, with superscript l indicating a point in time. The function U gets replaced with a discrete set of values on a space-time lattice; and we have a comparable set of space-time molecules defining the difference equations. We refer to this system as the Discrete System; it is the last algebraic maneuver, and the form which will be implemented computationally.

On this lattice, we imagine the solution propagating forward from IC's, constrained by BC's at the edge of the lattice. We will look at two different methods of propagation: "point" or explicit-in-time methods, which propagate one value at a time, independently of its soon-to-be-discovered neighbors; and "line" ("block") or implicit-in-time methods, which propagate a complete new set of values at the same time. This distinction is analogous to that made in the discussion of iterative methods for elliptic operators. In fact, there is a mathematical equivalence between certain iterative methods for elliptic equations and certain discrete systems representing Parabolic (and Hyperbolic) equations; and the equivalence can be exploited in the design of algorithms. It is worth noting that for Parabolic equations, the Continuous system propagates as a line; that is, BC's are felt instantaneously and simultaneously throughout the domain, for all time at and subsequent to the time of their imposition.

5.2 Examples: Discrete Systems

Each of the Discrete Systems examined here share the same lineage:

$$\frac{\partial U}{\partial t} = D\frac{\partial^2 U}{\partial x^2} \qquad (Distributed\ System) \tag{5.6}$$

$$\frac{dU_i}{dt} = \frac{D}{h^2}\delta_x^2 U_i \qquad (Lumped\ System) \tag{5.7}$$

with δ_x^2 the second centered difference operator and h the mesh spacing. We will examine different Discrete versions, reflecting different time-domain details.

Euler

The Discrete System for the Euler method is

$$U_i^{l+1} - U_i^l = \frac{D\Delta t}{h^2}\delta_x^2 U_i^l \tag{5.8}$$

Here we encounter the first basic dimensionless number, the Richardson number r:

$$r \equiv \frac{D\Delta t}{h^2} \tag{5.9}$$

r is the dimensionless time step. h^2/D sets the time scale for internal adjustments within the discrete solution space. The discrete system is restated as

$$U_i^{l+1} - U_i^l = rU_{i-1}^l - 2rU_i^l + rU_{i+1}^l \tag{5.10}$$

and the FD "molecule" for this discrete system is shown in Figure 5.2. The lumped system is centered-in-space. The discrete system is

- an explicit-in-time or Euler integration of the lumped system; or equivalently,
- a forward-difference-in-time approximation to the lumped system

As such its order of approximation is $O(h^2 + \Delta t)$. It represents pointwise propagation from IC's, as illustrated in Figure 5.3. Stemming from this, we can imagine that the numerical space-time lattice of values U_i^l is a "house of cards", and anticipate that there is a maximum value of timestep r beyond which the calculation will fall apart. Essentially, *expect conditional stability for this scheme.* In fact, essentially all pointwise propagation schemes have this conditional stability requirement. The solution must not get too far ahead of itself before the elliptic part of the molecule is exercised.

This discrete system has the peculiarity of a zone in which the solution is uniquely dependent on IC's, *i.e.* totally aloof from the BC's. This is illustrated in Figure 5.3. In other words, BC information propagates diagonally on the space-time lattice, with delay accumulating with distance from the boundary. This property is not shared with either the distributed system or the lumped system; it is solely an artifact of the time discretization. (Notice that such a zone is appropriate, at least qualitatively, for a Hyperbolic system wherein wavelike behaviour and finite delay time is intrinsic.)

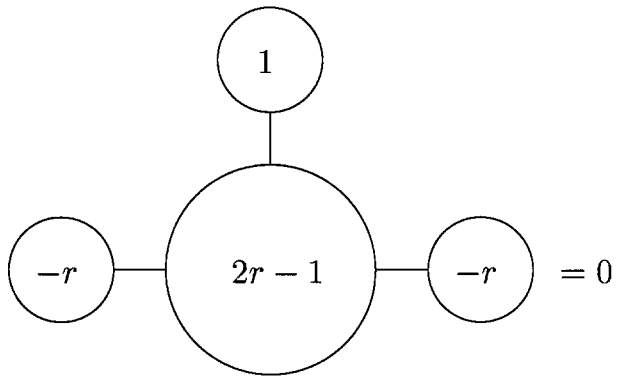


Figure 5.2: Euler discrete system for the diffusion equation. This is conditionally stable.

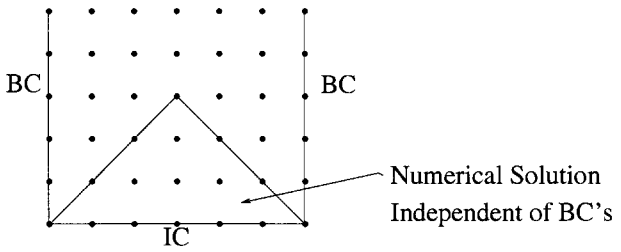


Figure 5.3: Zone of unique IC influence for Euler and Leapfrog systems.

Leapfrog

An alternative explicit system is obtained by invoking three adjacent time levels in each molecule. The discrete system is

$$U_i^{l+1} - U_i^{l-1} = 2r\delta_x^2 U_i^l \tag{5.11}$$

(Figure 5.4.) In time, we have either the midpoint integration rule, or centered differencing, over the interval of length $2\Delta t$. Both lead to second-order truncation errors in t; hence this system is second-order correct overall, $O(h^2 + \Delta t^2)$. Notice that this system is not self-starting – we need IC's at two levels in time in order to generate anything new. This suggests a basic flaw in conception of this discrete system, whereby we are effectively asking for $\partial U/\partial t$ at the start. If that can be generated, then we are in business; but the opportunity to generate something unrelated to the PDE, out of roundoff or other imprecision, is clearly inserted at the outset.

This method is unconditionally unstable, in part due to this fact. We will have more to say about this elsewhere, but for now we note the fact. Because this molecule is a close facsimile of that which is so successful for elliptic problems, we take home the lesson that algorithms which work well for one type of problem are not easily transported to other types of PDE's. Further, the fact that this system is $O(\Delta t^2)$ does not confer any grace; it cannot be used for any practical calculation despite its apparent accuracy. Here is a perfect example of intuition gone astray.

Like the Euler system, the Leapfrog system represents pointwise propagation from IC's, Figure

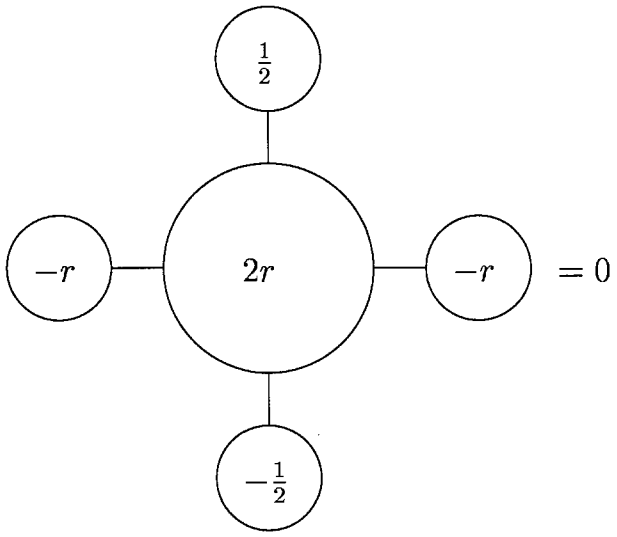


Figure 5.4: Leapfrog molecule for the diffusion equation. This is unconditionally unstable.

5.3. However in the Leapfrog case, all values of r are unstable so this is largely an academic observation.

Backward Euler

A third discrete system is the reverse of the Euler system above. This is gotten by either backward Euler integration in time, or backward differencing of the time derivative. The discrete system is

$$U_i^{l+1} - U_i^l = r\delta_x^2 U_i^{l+1} \tag{5.12}$$

and its molecule is given in Figure 5.5. Like the forward or explicit Euler system above, this system is $O(h^2 + \Delta t)$ in truncation error. But the reversed treatment of the time derivative reverses the instability, such that going forward in time, *it is unconditionally stable!* No value of r, no matter how bad the accuracy, is large enough to cause instability. Intuitively, the elliptic operator is always applied at the latest time level, as it is being computed. There is no hiding place for an instability to set in even when r is huge; in that case we are effectively solving for the steady-state, which we know is well-conditioned from elliptic studies of the Laplace equation.

2-Level Implicit

A blend of the two Euler systems is obvious:

$$U_i^{l+1} - U_i^l = r\theta\delta_x^2 U_i^{l+1} + r(1-\theta)\delta_x^2 U_i^l \tag{5.13}$$

This reproduces both previous schemes for $\theta = 0$ and 1, respectively. The special value $\theta = 1/2$ corresponds to central differencing in t, or the trapezoidal rule integration in t. Either way, we obtain enhanced temporal accuracy, with truncation error $O(h^2 + \Delta t^2)$. This is universally referred

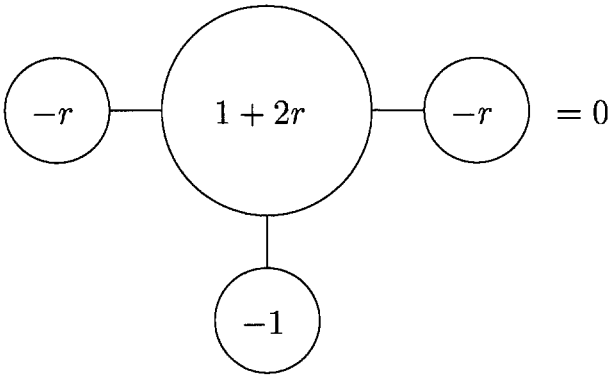


Figure 5.5: Backward Euler molecule. This is unconditionally stable.

to as the *Crank-Nicolson* system. It has the following stability properties:

$$\theta \geq 0.5 \qquad \textit{Unconditional Stability} \tag{5.14}$$

$$\theta < 0.5 \qquad \textit{Conditional Stability} \tag{5.15}$$

and the stability condition is dependent on r and θ:

$$r \leq r_{max}(\theta) \tag{5.16}$$

This condition will be derived below. This molecule appears in Figure 5.6; the system generalizes the previous two Euler systems.

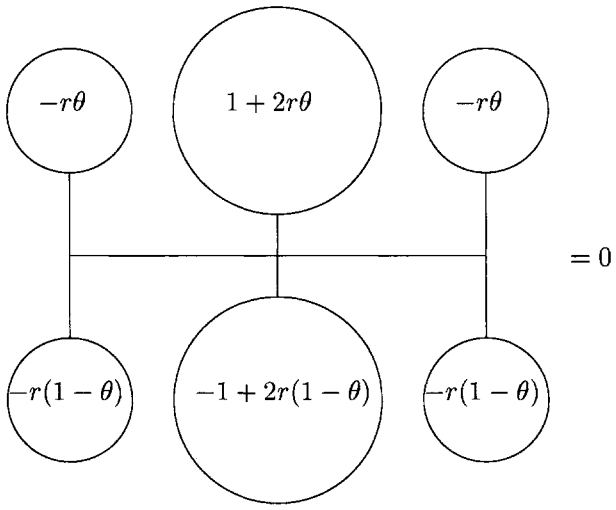


Figure 5.6: 2-Level Implicit molecule. $\theta = 0.5$ is the Crank-Nicolson system.

5.3 Boundary Conditions

Boundary conditions are exerted on these systems as in the elliptic problems studied above. For example, suppose we have a Type II BC at the leftmost boundary, $i = 0$:

$$-D\frac{\partial U}{\partial x} = q_0 \tag{5.17}$$

In the Lumped system, irrespective of temporal discretization, we have the FD approximation at node 0:

$$\frac{U_{-1} - U_1}{2h} = \frac{q_0}{D} \tag{5.18}$$

Now the Lumped system at node 0 will invoke the fictitious value of U_{-1}, which is beyond the boundary:

$$\frac{dU_0}{dt} = \frac{D}{h^2}(U_{-1} - 2U_0 + U_1) \tag{5.19}$$

and the Lumped BC may be invoked to eliminate this "shadow" variable. Combining these and eliminating U_{-1} we get:

$$\frac{dU_0}{dt} = \frac{D}{h^2}(2U_1 - 2U_0) + \frac{2}{h}q_0 \tag{5.20}$$

or equivalently,

$$\frac{h}{2}\frac{dU_0}{dt} = \frac{D(U_1 - U_0)}{h} + q_0 \tag{5.21}$$

(We will have more use for the second form later.) From here we can proceed to any of the discrete systems. For example, the Euler molecule at node 0 would become as illustrated in Figure 5.7, wherein the scaled flux Q_0 has been introduced:

$$Q_0 \equiv \frac{\Delta t}{h}q_0 \tag{5.22}$$

Notice that the boundary flux q_0 is evaluated at level l here (it is lagged), consistent with the sense of forward Euler integration of the Lumped System in time.

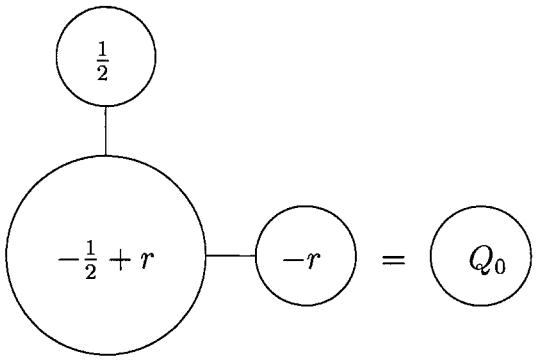


Figure 5.7: Euler system incorporating Type II BC.

It is easy to extend this example to the Type III BC:

$$-D\frac{\partial U}{\partial x} = \alpha(U_0^* - U_0) = q_0 - \alpha U_0 \tag{5.23}$$

wherein U_0^* is an equilibrium value (given as data), α is an inverse time scale of adjustment, and $q_0 \equiv \alpha U_0^*$. Either way, this BC has two independent pieces of data: (α, U_0^*) or (α, q_0). The Lumped BC is

$$\frac{U_{-1} - U_1}{2h} = \frac{1}{D}(q_0 - \alpha U_0) \tag{5.24}$$

and combining this as before with the Lumped PDE at the boundary, we get

$$\frac{dU_0}{dt} = \frac{D}{h^2}(2U_1 - 2U_0) + \frac{2}{h}(q_0 - \alpha U_0) \tag{5.25}$$

or equivalently,

$$\frac{h}{2}\frac{dU_0}{dt} = \frac{D(U_1 - U_0)}{h} + q_0 - \alpha U_0 \tag{5.26}$$

Proceeding as above, the molecule for the Euler discrete system with Type III BC would be as given in Figure 5.8. Here we have defined the dimensionless rate constant A, with scaling analogous to Q_0 above:

$$A \equiv \frac{\Delta t}{h}\alpha \tag{5.27}$$

The Type III BC has introduced the intrinsic time constant α/h into the system. We have set the Type III term involving AU explicitly, at time level l, consistent with the Euler idea. This may be expected to introduce internal dynamics on that time scale, and thus we may anticipate instabilities when A is large. There are obvious alternatives; we offer this as an example only.

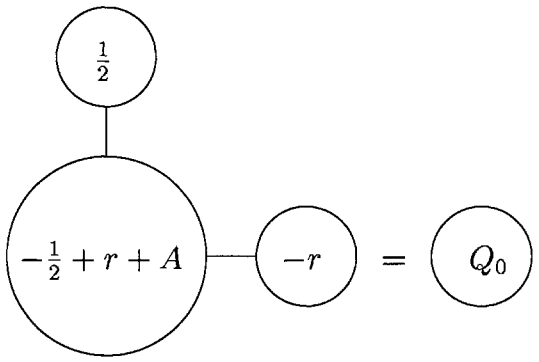


Figure 5.8: Euler system incorporating Type III BC.

Other combinations of BC's and Discrete Systems could be handled similarly.

5.4 Stability, Consistency, Convergence

In the PDE literature there are three classic terms pertaining to the quality of a numerical solution. These are:

- *Convergence* refers to the agreement between the PDE solution (unknown) $U(x,t)$ and the numerical array U_i^l, which is only available numerically. Essentially, the convergence question is, "does $U_i^l \to U(x_i, t^l)$ as h, $\Delta t \to 0$ independently?"

- *Consistency* addresses the correspondence between the *PDE operator* and the *discrete operator* or molecule. "Does $L_i \to L$ as $h, \Delta t \to 0$ independently?" Essentially, does the FD molecule $\to$ PDE? This is a weaker question than convergence; it is easier to demonstrate. The Taylor series truncation errors examined earlier, expressed for the complete FD molecule, are one gateway to this question.
- *Stability* concerns the boundedness of the numerical solution U_i^l, assuming bounded BC's, IC's, and forcing. For linear problems, the homogeneous response to IC's may be studied, as $t \to$ large.

A basic rule of thumb which emerges is: stability + consistency $\Rightarrow$ convergence. Since convergence is the hard one, effort in the other two categories is more abundant. Below we illustrate these ideas in the context of the diffusion equation in one space variable. The interested reader is invited to look at more complete texts *e.g.* Ames [2].

Convergence - Lumped System

We are interested in the solutions to the PDE and its lumped representation:

$$\frac{\partial \mu}{\partial t} = D\frac{\partial^2 \mu}{\partial x^2} \quad \Leftrightarrow \quad \frac{dU_i}{dt} = \frac{D}{h^2}\partial_x^2 U_i \tag{5.28}$$

$$\mu(x_i, t) \simeq U_i(t) \tag{5.29}$$

and we ask about the difference ϵ between the solutions,

$$\epsilon_i(t) = U_i(t) - \mu_i(t) \tag{5.30}$$

given that equations 5.28 and 5.29 govern. The exact solution μ satisfies the Lumped system with the truncation error appended:

$$\frac{d\mu_i}{dt} = \frac{D}{h^2}\delta_x^2 \mu_i - \frac{h^2}{12}\frac{\partial^4 \mu}{\partial x^4} + \ldots \tag{5.31}$$

and therefore the error satisfies

$$\frac{d\epsilon_i}{dt} - \frac{D}{h^2}\delta_x^2 \epsilon_i = \frac{h^2}{12}\frac{\partial^2 \mu}{\partial x^4} - \ldots \tag{5.32}$$

subject to the IC $\epsilon_i = 0$. The complete solution for ϵ is therefore the convolution of the forcing by the truncation error, which exists uniquely but is unknown:

$$\epsilon_i(t) = \int_{\tau=0}^{t} g(t-\tau)\left(\frac{h^2}{12}\frac{\partial^4 \mu_i(\tau)}{\partial x^4} - \ldots\right) d\tau \tag{5.33}$$

Here $g(t-\tau)$ is the impulse response of $(\frac{d}{dt} - \frac{D}{h^2}\partial_x^2)$. If the discrete system is stable, then $|g| < G$, ie, g is bounded and thus

$$|\epsilon_i(t)| < \frac{Gh^2}{12}\int_{\tau=0}^{t} |\frac{\partial^4 \mu_i}{\partial x^4}(\tau)| d\tau \tag{5.34}$$

and at any point in time t, we have second-order convergence:

$$\epsilon_i(t) = O(h^2) \tag{5.35}$$

Alternatively: if μ is bounded and continuous, then the truncation error is bounded

$$|\frac{\partial^4 \mu_i}{\partial x^4} - \ldots| < M \tag{5.36}$$

and we get

$$|\epsilon_i(t)| < \frac{Mh^2}{12} \int_{\tau=0}^{t} |g(t-\tau)| d\tau \tag{5.37}$$

Further, if $\int_{\tau=0}^{t} |g(t-\tau)| d\tau < N$ for arbitrarily large t, (essentially, the impulse response is finite and decays to 0 over long time):

$$|\epsilon_i(t)| < \frac{MNh^2}{12} \tag{5.38}$$

The error ϵ_i does not threaten to grow unbounded as $t \to \infty$, in addition to vanishing as $h^2 \to 0$. This is a demonstration of convergence. Many additional features could be introduced.

Convergence – Discrete System

Consider the analogous development for the Euler discrete system:

$$U_i^{l+1} = rU_{i-1}^l + (1-2r)U_i^l + rU_{i+1}^l \tag{5.39}$$

and

$$\mu_i^{l+1} = r\mu_{i-1}^l + (1-2r)\mu_i^l + r\mu_{i+1}^l + O(h^2+k)k \tag{5.40}$$

(Here $k \equiv \Delta t$.) Subtracting the two, with discrete error $\epsilon_i^l = U_i^l - \mu_i^l$, we get

$$\epsilon_i^{l+1} = r\epsilon_{i-1}^l + (1-2r)\epsilon_i^l + r\epsilon_{i+1}^l + O(h^2+k)k \tag{5.41}$$

IC's are $\epsilon_i^0 = 0$. Notice that for $r < 1/2$, $(1-2r) > 0$ and $|1-2r| = 1-2r$. So taking the absolute value we get

$$|\epsilon_i^{l+1}| \leq |r||\epsilon_{i-1}|^l + |1-2r||\epsilon|_i^l + |r||\epsilon|_{i+1}^l + |O(h^2+k)k| \tag{5.42}$$

Now define the largest error at any time,

$$||\epsilon||^l \equiv \max_i |\epsilon_i^l| \tag{5.43}$$

and the upper limit on the truncation error A, and we obtain

$$||\epsilon||^{l+1} \leq ||\epsilon||^l + A(h^2+k)k \tag{5.44}$$

From the IC's, $||\epsilon||^0 = 0$. So $||\epsilon||^l \leq A(h^2+k)k$, $||\epsilon||^2 \leq A(h^2+k)2k$, and so on:

$$||\epsilon||^l \leq A(h^2+k)lk \tag{5.45}$$

and at any given point in time, the error bound is linear in $t = lk$ and proportional to (h^2+k) and to the truncation error magnitude, which generally scales with the solution size:

$$||\epsilon||^l \leq A(h^2+k)t \tag{5.46}$$

As expected intuitively, the convergence is $O(h^2+k)$, the same as that obtained from the FD molecule truncation terms. Additionally, we find that the solution has potential for linear buildup in error over time – not an instability *per se*, but still a liability. this proof requires $r = \frac{\triangle t D}{\triangle x^2} < 1/2$, so we may anticipate problems with $r > 1/2$ *i.e.* large Δt. Finally, we need to be careful that 5.46 is an upper bound only. It is not an equality; and there are certainly more precise (lower) upper bounds for this error.

Consistency

There is a classic example of inconsistency: the DuFort-Frankel Scheme. This is a modification of the Leapfrog method; recall that Leapfrog is unconditionally unstable for the diffusion equation. In an effort to save it, we might rewrite it as

$$\frac{U_i^{l+1} - U_i^{l-1}}{2k} = D\frac{U_{i-1}^l - (U_i^{l+1} + U_i^{l-1}) + U_{i+1}^l}{h^2} - \frac{k^2}{6}\frac{\partial^3 U}{\partial t^3} + D\frac{h^2}{12}\frac{\partial^4 U}{\partial t^4} \tag{5.47}$$

where the approximation $(U_i^{l+1} + U_i^{l-1})$ has been substituted for $2U_i^l$ which normally appears in the Laplacian in this system. Ostensibly this is a benign change, a temporal average, of order Δt^2. By beefing up the diagonal of the system at the unknown level, perhaps it will add stability. In fact, it does – unconditional stability! But there is a price paid – it is inconsistent.

To see this, rewrite the temporal average as

$$U_i^{l+1} + U_i^{l-1} = (U_i^{l+1} - 2U_i^l + U_i^{l-1}) + 2U_i^l \tag{5.48}$$

Then, the DuFort-Frankel scheme is

$$\frac{U^{l+1} - U^{l-1}}{2k} = D\frac{U_{i-1}^l - 2U_i^l + U_{i+1}^l}{h^2} - D\frac{k^2}{h^2}\left(\frac{U_i^{l+1} - 2U_i^l + U_i^{l-1}}{k^2}\right) \tag{5.49}$$

But the last term is a problem:

$$\frac{U_i^{l+1} - 2U_i^l + U_i^{l-1}}{k^2} \to \frac{\partial^2 U_i^l}{\partial t^2} + O(k^2) \tag{5.50}$$

So if $k/h = \beta =$ a constant, we have

$$D\beta^2\frac{\partial^2 U}{\partial t^2} + \frac{\partial U}{\partial t} = D\frac{\partial^2 U}{\partial x^2} \tag{5.51}$$

This is consistent with the hyperbolic telegraph equation! But if $k \to 0$ faster than h: $\beta \to 0$ and this is a parabolic system. So we have different end points dependent on the path or process of resolution. But the requirement of consistency is that the discrete system operator approach the PDE *independently of the path, i.e.* independent of β in this case. So we find that this system is not consistent with the diffusion equation.

Stability

Stability as defined here pertains to the discrete solution. That is the solution which will be implemented on machines. If the discrete solution is bounded, *i.e.* it will not become infinitely large, then it is stable. A strict sense of this is that the system decays to zero at long time in the absence of forcing. In other words, its homogeneous response to IC's decays to zero at long time. The marginal case, where the homogeneous response neither decays nor grows, represents "neutral stability". Now all practical implementations of discrete systems are subject to continuous inputs of noise through their imperfect algebra. So we can think of the homogeneous response as the temporal convolution of every conceivable mode of noise, in addition to the response to the IC's *per se*. An unstable system accumulates and amplifies noise; a stable one causes it to decay away. Notice here that stability is unrelated to any metric of accuracy. It is a practical consideration, a necessary precondition for simply getting a reproducible solution.

Our approach is heuristic: imagine a spatial mode of the solution, study the homogeneous response to it. For a linear system, all modes must be stable since they will certainly occur randomly through roundoff on any finite machine. So a single unstable mode is sufficient to qualify a system as unstable.

We will concentrate here on the worst-resolved spatial mode: $\{+1, -1, +1, -1, +1, -1, +1, \ldots\}$. This is the so-called "washboard" or "$2\Delta x$" mode. In terms of Fourier transforms, it exists at the lowest resolvable point in the discrete spatial spectrum, the Nyquist point. This mode has the greatest Taylor Series truncation error ($\sim \frac{\partial^4 U}{\partial x^4}$ for the second derivative). Although not apparent until the next section, it is the most prone to instability for the systems introduced here, and therefore controls the stability. *For this washboard mode*, we have the approximate Laplacian

$$U_{i+1} - 2U_i + U_{i-1} = -4U_i \tag{5.52}$$

Assuming an infinitely long spatial domain, there are no boundaries and all molecules are identical.

The **Euler** discrete system is

$$U_i^{l+1} - U_i^l = -4rU_i^l \tag{5.53}$$

for all i. Solving this, we get

$$U_i^{l+1} = (1 - 4r)U_i^l \tag{5.54}$$

Immediately we have a 3-way result for the series U_i^{l+1}:

- for small r: the solution *decays monotonically.* This occurs when $0 < 1 - 4r < 1$. Since $r > 0$, we have

$$r < \frac{1}{4} \tag{5.55}$$

  Since the analytic solution generally behaves this way, this range of r provides at least qualitative fidelity.

- for intermediate r: the solution *decays in an alternating series.* This occurs when $-1 < 1 - 4r < 0$:

$$\frac{1}{4} < r < \frac{1}{2} \tag{5.56}$$

  In this range, the solution is stable but exhibits solutions which alternate in sign, which is *not* qualitatively faithful to the PDE.

- for large r: the solution *grows without bound,* in an alternating series. This occurs when $1 - 4r < -1$, or

$$\frac{1}{2} < r \tag{5.57}$$

  For this range of r, the system is simply *unstable.*

Now r is the dimensionless timestep. As r grows, so does the temporal truncation error. At $r = \frac{1}{4}$ we lose all resemblance to the exact solution, and this gets progressively worse until at $r = \frac{1}{2}$ and beyond the lack of fidelity becomes fatal. The *Euler stability condition* is therefore

$$r \leq \frac{1}{2} \tag{5.58}$$

with the equality characterizing neutral stability. Recall that these are necessary conditions; this characterizes only one mode. (In this case, this washboard mode turns out to be the critical, controlling mode.)

In the same light we can examine the **Leapfrog** system for this mode:

$$U_i^{l+1} - U_i^{l-1} = 2r\delta_x^2 U_i^l = -8rU_i^l \tag{5.59}$$

This equation has solutions of the form

$$U_i^{l+1} = \gamma U_i^l \tag{5.60}$$

and on substitution we get the quadratic equation

$$\gamma^2 + 8r\gamma - 1 = 0 \tag{5.61}$$

and its solution is

$$\gamma = \frac{-8r \pm \sqrt{(8r)^2 + 4}}{2} = -4r \pm \sqrt{(4r)^2 + 1} \tag{5.62}$$

The negative option clearly exceeds unity in absolute value. Therefore we have

$$|U_i^{l+1}| = |\gamma||U_i^l| > |U_i^l| \tag{5.63}$$

and the *Leapfrog system is unstable for all values of* r.

While we are here, look at the (inconsistent) **DuFort-Frankel** system. Earlier we asserted that this scheme was unconditionally stable. The present analysis for the washboard mode gives

$$\gamma^2 - 1 = 2r(-\gamma - \gamma^2 - 1 - \gamma) \tag{5.64}$$

$$(1 + 2r)\gamma^2 + (4r)\gamma + (-1 + 2r) = 0 \tag{5.65}$$

This has roots

$$\gamma = \frac{-2r \pm 1}{1 + 2r} \tag{5.66}$$

The negative option has magnitude unity irrespective of r; the positive option always has magnitude < 1. For this mode, the system is *neutrally stable for all* r.

Finally, let's look at the **general 2-level system** 5.13:

$$U_i^{l+1} - U_i^l = r\delta_x^2\Big(\theta U_i^{l+1} + (1-\theta)U_i^l\Big) \tag{5.67}$$

For the washboard mode we have

$$U_i^{l+1} - U_i^l = -4r\Big(\theta U_i^{l+1} + (1-\theta)U_i^l\Big) \tag{5.68}$$

And rearrangement gives us

$$\frac{U_i^{l+1}}{U_i^l} = \frac{1 - 4r(1-\theta)}{1 + 4r\theta} \tag{5.69}$$

From this we see that we will have *monotone* solutions in time, as long as we have

$$r < \frac{1}{4(1-\theta)} \tag{5.70}$$

For larger r, $\frac{U_i^{l+1}}{U_i^l}$ becomes negative. This oscillatory behaviour will nevertheless remain stable when

$$r < \frac{1}{2(1-2\theta)} \tag{5.71}$$

For larger r, we have instability. Immediately we see that increasing θ improves the stability; and that this system is unconditionally stable when $\theta \geq 1/2$ (*i.e.* no value of r produces instability).

Summarizing the various limits:

- Euler, $\theta = 0$:
  - monotone when $r < \frac{1}{4}$
  - stable when $r < \frac{1}{2}$
- Crank-Nicolson, $\theta = 1/2$:
  - monotone when $r < \frac{1}{2}$
  - stable unconditionally
- Backward Euler, $\theta = 1$:
  - monotone unconditionally
  - stable unconditionally

Here we have looked only at one mode of the system, the most poorly-resolved one. The next section generalizes the analysis to include all possible modes. The stability results here will survive as rules of thumb governing the most critical modes.

5.5 Accuracy: Fourier (von Neumann) Analysis

Now we will generalize the stability analysis of the previous section, in a way which allows us to consider all possible modes of the system. To do this we will decompose the solution into its spatial Fourier spectrum; and look at the evolution of that. In doing so, we will be able to make a metric of fidelity to the PDE solution, in terms of properties of its Fourier transform.

The reader is assumed to be familiar with the Discrete Fourier Transform; Press *et al.* [99] contains a useful exposition.

Continuous System

First, review the classic use of the Fourier transform *vis a vis* the PDE

$$\frac{\partial U}{\partial t} = D\frac{\partial^2 U}{\partial x^2} \tag{5.72}$$

Assume a space-time separation of the solution of the particular form

$$U(x,t) = Ae^{\alpha t}e^{j\sigma x} \tag{5.73}$$

$j = \sqrt{-1}$ is the imaginary unit; σ is a (real) spatial wavenumber; α is the corresponding growth rate in time; A is the amplitude for this spatial (Fourier) mode at $t = 0$. In general, we will need a complete, continuous spectrum of $0 \leq \sigma \leq \infty$ to be able to represent an arbitrary set of IC's. But for now assume a single value of σ. Each wavenumber σ has a spatial wavelength λ, and $\lambda\sigma = 2\pi$. Inserting 5.73 into the PDE, we obtain the *Dispersion Relation*

$$\alpha = -D\sigma^2 \tag{5.74}$$

which characterizes the PDE response at wavenumber σ. The superposition of all spatial waveforms (all σ) present gives the full response. The following characteristics are well-known and can be inferred here:

- The solution is stable: all σ decay monotonically since all $\alpha < 0$)
- The longest waves decay slowest ($\alpha \to 0$ as $\lambda \to \infty$)
- The solution gets smoother over time, as the sharper features are represented by the bigger σ (smaller λ)

Lumped System

The lumped system $\frac{dU_i}{dt} = \frac{D}{h^2}\delta^2 U_i$ can be approached in the same manner. Here we have to be a little careful to respect the limits of the Fourier spectrum. For a uniform, infinitely long mesh, the complete spectrum is continuous but there is a *Nyquist cutoff point at* $\lambda = 2h$. Use of wavenumbers representing shorter wavelengths is redundant, as each such mode has an indistinguishable counterpart at $\lambda > 2h$. So for the infinite (unbounded) lumped system, the complete Fourier spectrum is given in terms of either σ or equivalently, λ:

$$0 \leq \sigma \leq \pi/h \tag{5.75}$$

$$\infty \geq \lambda \geq 2h \tag{5.76}$$

As above, we need to deal with the discrete difference operator δ^2 as it affects a given Fourier mode. Direct expansion, and some trigonometric identities, give

$$\begin{aligned} \delta^2 U_i &= U_{i-1} - 2U_i + U_{i+1} \\ &= (e^{-j\sigma h} - 2 + e^{j\sigma h})U_i = (2\cos\sigma h - 2)U_i \\ &= -4U_i \left(\sin\frac{\sigma h}{2}\right)^2 \end{aligned} \tag{5.77}$$

A little further processing reduces this to

$$\frac{\delta^2 U_i}{h^2} = -\sigma^2 U_i \left(\frac{\sin\frac{\sigma h}{2}}{\frac{\sigma h}{2}}\right)^2 \tag{5.78}$$

Notice that we have created a new dimensionless number, $S \equiv \sigma h \equiv \frac{2\pi h}{\lambda}$. This is a measure of resolution; small S means small h relative to the wavelength at hand. Large S is coarse resolution of a given wavelength. So one further step gives the useful shorthand

$$\frac{\delta^2 U_i}{h^2} = -\sigma^2 C^2 U_i \tag{5.79}$$

$$C(S) = \frac{\sin(S/2)}{S/2} \tag{5.80}$$

$$S = \sigma h \tag{5.81}$$

This is a remarkably simple result. It compares directly with the analytic form of Fourier differentiation, with the added factor C^2 containing all of the discretization effects. Notice that C approaches unity at small S; that is, for very long wavelenghts relative to the mesh, the FD difference formula becomes increasingly perfect. From 5.75 the limits of S and L are

$$0 \leq S \leq \pi \tag{5.82}$$

$$\infty \geq L \geq 2 \tag{5.83}$$

($L \equiv \lambda/h$ is a convenient measure of the wavelength resolution. $LS = 2\pi$.) The right-side of these limits is the Nyquist point. If we now make the substitutions into the lumped system, we get

$$\alpha U_i = -D\sigma^2 C^2 U_i \tag{5.84}$$

and the lumped system dispersion relation is

$$\alpha = -D\sigma^2 C^2 \tag{5.85}$$

The discretization factor C carries all of the discrepancy from the exact solution. If it were unity (which it is at perfect resolution, $S = 0$), then the discrete system would have a perfect dispersion relation.

We can go further by inserting a Taylor series for $C(S)$, and get-

$$\alpha = -D\sigma^2 \left[1 - \frac{(\sigma h)^2}{12} + \ldots\right] = -D\sigma^2 \left[1 - \frac{(\frac{2\pi h}{\lambda}^2)}{12} + \ldots\right] \tag{5.86}$$

Notice these facts:

- α is correct to $O(h^2)$ for any σ
- α is undefined for $\lambda < 2h$
- accuracy depends on $\sigma h = \frac{2\pi h}{\lambda}$ *i.e.* the mesh spacing h is meaningful only relative to λ
- the lumped system is stable: $\alpha < 0$ for all σ

A plot of $\frac{\alpha}{-D\sigma^2}$ vs λ/h is sketched in Figure 5.9. The analytic solution has $\frac{\alpha}{-D\sigma^2} = 1$. The plot for the lumped system reveals that

- the lumped is underdamped relative to the analytic (distributed) system;
- the error is greater for small λ/h; and
- the error is monotone, between the limits 0 at $S = 0$, and $1 - 4/\pi^2 \approx 0.6$ at the Nyquist point $S = \pi$.

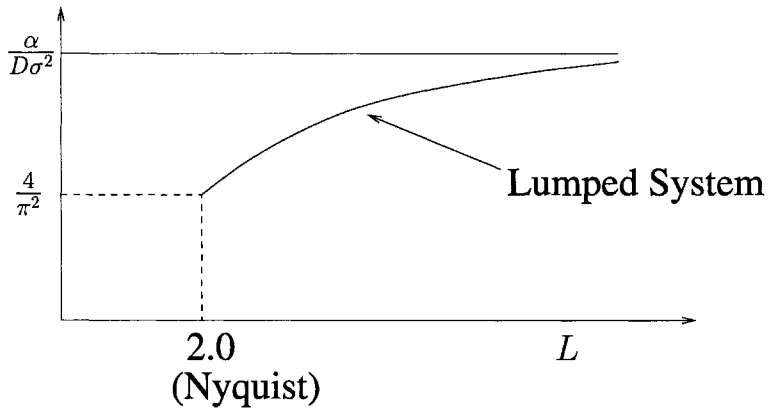


Figure 5.9: Dispersion relation for lumped system representation of 1-D diffusion equation. $L = \lambda/h$.

Discrete System

To handle the discrete system, we need to return to a device introduced earlier in the discussion of stability:

$$\gamma \equiv \frac{U_i^{l+1}}{U_i^l} \tag{5.87}$$

This is the ratio of solutions separated by one timestep. In the discrete system, γ plays the role of the separable time-solution $e^{\alpha t}$:

$$U_i^l = \gamma^l e^{j\sigma x_i} \tag{5.88}$$

and γ is related to α:

$$\gamma = e^{\alpha \Delta t} \tag{5.89}$$

The ratio of the solution after l timesteps, to that now, is γ^l; and the elapsed time is $\tau = l\Delta t$. So we have

$$\gamma^l = e^{\alpha l \Delta t} = e^{\alpha \tau} \tag{5.90}$$

We may speak of α or of γ; they convey the same information. But strictly speaking, only γ is defined in the mathematics of the discrete system. Notice here that this represents a subtle but appropriate shift of emphasis from α, the solution growth *rate*, to γ, the solution itself after finite time Δt.

(Aside: For the lumped system, γ may be gotten from α via 5.86:

$$e^{\alpha \Delta t} = e^{-D\sigma^2\left[1 - \frac{(\sigma h)^2}{12} + \ldots\right]\Delta t} \tag{5.91}$$

The first term is the analytic (continuous) version. So the lumped system has

$$\gamma_L = \gamma_C \cdot e^{D\sigma^2\left[\frac{(\sigma h)^2}{12} + \ldots\right]\Delta t} \tag{5.92}$$

and one more Taylor Series for the exponential finishes this comparison:

$$\gamma_L = \gamma_C \cdot \left[1 + D\sigma^2 \frac{(\sigma h)^2}{12}\Delta t + \ \ldots\right] \tag{5.93}$$

The temporal accuracy is linear in Δt. The integration over time causes defects in α to accumulate, linearly with the temporal interval, for perfect time-integration.)

Euler System

Now, back to the Discrete System. We start with the explicit Euler system:

$$U_i^{l+1} = U_i^l + r\delta^2 U_i^l \tag{5.94}$$

We have for the Fourier expansion,

$$\delta^2 U_i = -S^2 C^2 U_i = -4\sin^2\left(\frac{S}{2}\right) U_i \tag{5.95}$$

and for the temporal expansion

$$U_i^{l+1} = \gamma U_i^l \tag{5.96}$$

Substitution gives us

$$\gamma = 1 - 4r\sin^2\frac{S}{2} \tag{5.97}$$

Now γ, the ratio of new to old solution, should represent a pure decay, *i.e.* it should lie between 0 and 1. But we see that we will lose that property when r is too big, $4r\sin^2\frac{S}{2} > 1$. So the criterion for a *monotone* mode is

$$r < \frac{1}{4\sin^2\frac{S}{2}} \tag{5.98}$$

Stability requires $|\gamma| < 1$. We lose that when r is even bigger, $4r\sin^2\frac{S}{2} > 2$. So the *stability* requirement is

$$r < \frac{1}{2\sin^2\frac{S}{2}} \tag{5.99}$$

Now S lies between 0 and π. So in both cases, the worst mode is $S = \pi$, *i.e.* the poorly-resolved mode at the Nyquist point. So the criterion for purely *monotone* solutions is

$$r < \frac{1}{4} \tag{5.100}$$

Operating with r above this limit will provoke non-monotone behaviour. The short waves are the first to go; then with increasing r, more and more of the short end of the spectrum becomes nonmonotone. Ultimately, as r goes beyond the threshold $\frac{1}{2}$, the stability of the worst-case Nyquist modes is lost. Operation beyond this point is impossible; it will lead to unstable growth of either roundoff error or noisy IC's. So the *stability limit* is

$$r < \frac{1}{2} \tag{5.101}$$

These findings corroborate what we did in the previous section relative to stability and monotonicity; there we examined only the Nyquist mode. (That is the limiting mode in these cases.)

Assuming we have a stable discrete system, we can analyze its fidelity relative to the continuous system. For the continuous system (the PDE), we have

$$\gamma_C = e^{-rS^2} \tag{5.102}$$

$$= 1 - rS^2 + \frac{r^2S^4}{2!} - \frac{r^3S^6}{3!} + \ \ldots \tag{5.103}$$

where again we have used a Taylor Series expansion. For the Euler discrete system, we have γ_E and its Taylor Series expansion

$$\gamma_E = 1 - 4r\sin^2\frac{S}{2} \tag{5.104}$$

$$= 1 - \frac{2rS^2}{2!} + \frac{2rS^4}{4!} - \frac{2rS^6}{6!} + \ldots \tag{5.105}$$

The leading error is the S^4 term. The discrepancy is

$$\frac{r^2S^4}{2} - \frac{rS^4}{12} = \left(\frac{r}{2} - \frac{1}{12}\right)rS^4 \tag{5.106}$$

and this vanishes for $r = 1/6$. So we get an extra boost in accuracy for this special value![1]

In Figure 5.10 we sketch the dependence of γ_E on spatial wavelength. The zones of monotone, oscillatory, and unstable temporal evolution are set forth, as a function of dimensionless timestep r. This provides a measure of qualitative fidelity to the analytic solution, which exhibits monotone

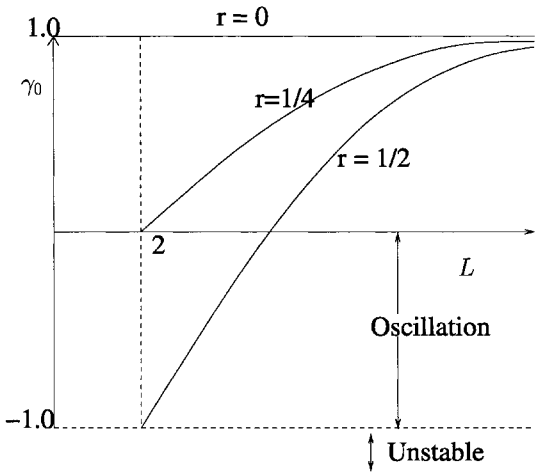


Figure 5.10: Illustration of Euler discrete system behaviour for a single timestep: γ versus L; $L = \lambda/h$.

decay ($0 \leq \gamma \leq 1$) for all σ.

2-Level Implicit System

Next, lets look at the general 2-level discrete system as displayed in Figure 5.6

$$U_i^{l+1} - U_i^l = r\delta_x^2\Big(\theta U_i^{l+1} + (1-\theta)U_i^l\Big) \tag{5.107}$$

[1]Recall from above that the Lumped system ($r = 0$) was *under*damped across the whole Fourier spectrum. It can be verified that the Euler system is *over*damped when $r = 1/4$, which is the onset of nonmonotone behaviour for the Nyquist mode. Evidently there is a crossover point in between, where extra accuracy is obtained – the leading Δt error just cancels the leading Δx error! The expression here suggests that this is at $r = 1/6$. The student is invited to confirm this by plotting these relations; and by examining the Taylor series truncation analysis of the original FD molecule for this discrete system.

Proceeding as above, we obtain

$$\gamma_\theta - 1 = -4r \sin^2 \frac{S}{2} \Big(\theta \gamma_\theta + (1 - \theta) \Big) \tag{5.108}$$

$$\gamma_\theta = \frac{1 - 4r(1-\theta) \sin^2 \frac{S}{2}}{1 + 4r\theta \sin^2 \frac{S}{2}} \tag{5.109}$$

Of course this reproduces the Euler system when $\theta = 0$. Like that case, $\gamma_\theta \leq 1$ for all $(r,\ \theta)$; but there is the potential for nonmonotone modes when r is too big. That occurs when $\gamma_\theta \leq 0$; so the requirement for *monotonicity* is

$$r \leq \frac{1}{4(1-\theta) \sin^2 \frac{S}{2}} \tag{5.110}$$

As usual, the shortest-wavelength modes lose monotonicity first; it controls the *monotonicity requirement*

$$r \leq \frac{1}{4(1-\theta)} \tag{5.111}$$

Continued growth in r expands the nonmonotone part of the spectrum. Stability is lost when $\gamma_\theta \leq -1$. Assuming $\theta < \frac{1}{2}$, we have the *stability requirement*

$$r \leq \frac{1}{2(1-2\theta) \sin^2 \frac{S}{2}} \tag{5.112}$$

and again, the limiting mode is at the Nyquist point. The practical *stability criterion* is then governed by this mode:

$$r \leq \frac{1}{2(1-2\theta)} \tag{5.113}$$

When $\theta \geq \frac{1}{2}$, we have unconditional stability (*i.e.* stability independent of r). But this does not guarantee monotonicity – that requirement remains as stated above.

As we found in the Euler system, there is a special increment of accuracy when

$$r = \frac{1}{6(1-2\theta)} \tag{5.114}$$

where the leading spatial and temporal truncation errors cancel.

Figure 5.11 is a sketch of γ versus λ/h for the Crank-Nicolson system, $\theta = 1/2$. It is qualitatively similar to Euler, except for the unconditional stability and the expanded range of monotone performance. The student is encouraged to develop and plot these relationships to fix ideas.

Propagation Factor

All of this analysis of γ addresses qualitative fidelity to the continuous system – in terms of stability and monotonicity. As a measure of quantitative accuracy, γ alone falls short, on 2 counts. First, we need to normalize γ by its analytic (continuous) counterpart since the latter system decays in a specific σ-dependent fashion. This is readily accomplished. Second, $\gamma \to 1$ as $r \to 0$ for any consistent approximation, as does the analytic (continuous) system. So the gap between these vanishes at small r, leaving little information.

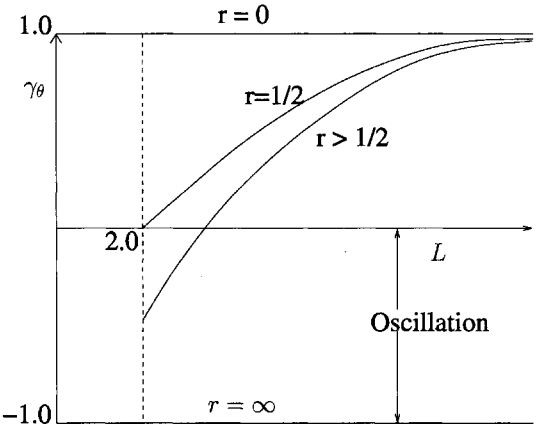


Figure 5.11: As in Figure 5.10. Illustration of Crank-Nicolson ($\theta = 1/2$) system behaviour for a single timestep: γ versus λ/h.

To accommodate these issues, we normalize γ and introduce a characteristic time $\tau = N\Delta t$ for the comparison:

$$T \equiv \left(\frac{\gamma_D}{\gamma_C}\right)^N \tag{5.115}$$

T is the *Propagation Factor.* It quantifies the ratio of the Discrete solution to the Continuous one, in the future, after elapsed time τ, starting from the same IC's. τ is held fixed, so N increases as Δt decreases. We take as the characteristic time the e-folding or relaxation timescale for the analytic solution,

$$\tau = \frac{1}{D\sigma^2} \tag{5.116}$$

Notice that τ depends on σ. With this, the Propagation Factor for the Discrete Diffusion Equation is

$$T = \frac{(\gamma_D)^N}{e^{-1}} \tag{5.117}$$

$$N = \frac{1}{D\Delta t\sigma^2} = \frac{1}{rS^2} \tag{5.118}$$

Now we have in T a valid measure of accuracy. The student is encouraged to prepare plots or T versus λ/h for the discrete systems developed here (*i.e.* various combinations of r, θ), before proceeding further. An example plot appears in Figure 5.12. $T = 1$ is perfect accuracy. $T > 1$ indicates that the discrete solution exceeds the continuous one (underdamped case) – *not* that the system is necessarily unstable. $T < 1$ indicates overdamping, and necessarily, stability. One finds in general, $T \to 1$ as $\frac{\lambda}{h} \to \infty$ for all combinations of $(r,\ \theta)$, or else we do not have convergence. In the nonmonotonic region, T is undefined.

Example: Implicit Leapfrog System

Here we revisit the idea of a 3-level-in-time system. Before we saw this in the form of the (explicit) leapfrog system, which was unconditionally unstable for diffusion problems; and the Dufort-Frankel

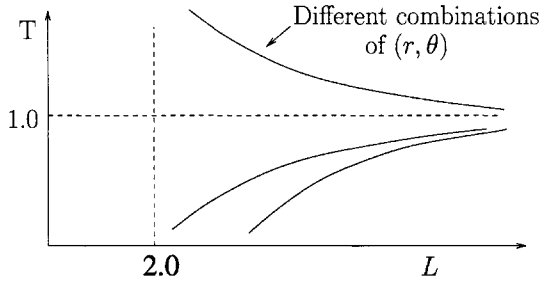


Figure 5.12: Sketch of Propagation Factor plot, T versus $L = \lambda/h$. $T = 1$ represents perfect accuracy. T is undefined for non-monotone modes.

system, which was stable but inconsistent. Both of these were centered-in-time, hence second-order accurate. Here we try to stabilize the system by adding implicitness, avoiding the inconsistency.

The system we will examine is

$$\frac{U_i^{l+1} - U_i^{l+1}}{2\Delta t} = \frac{D}{h^2}\delta^2 \left[\frac{\theta}{2}(U_i^{l+1} + U_i^{l-1}) + (1-\theta)U_i^l\right] \tag{5.119}$$

This is illustrated in molecular form in Figure 5.13.

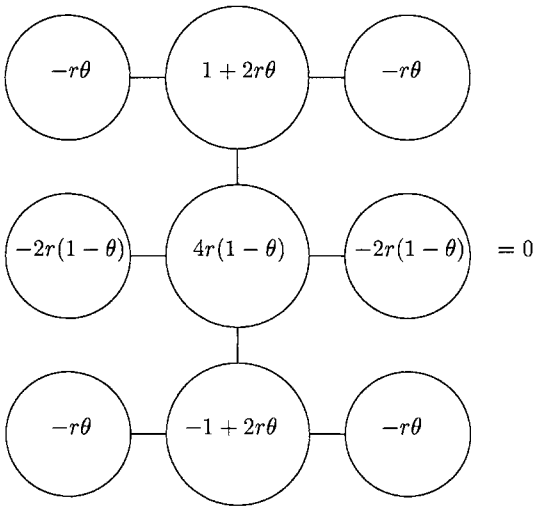


Figure 5.13: Implicit leapfrog system.

This system is centered in space and time, and hence second-order correct. For $\theta = 0$ it reverts to the Leapfrog system, which is unstable and useless. $\theta = 1$ reproduces the Crank-Nicolson system with the caveat that there are two independent solutions, an "odd" one and an "even" one, which are completely uncoupled. This is illustrated in Figure 5.14. Both odd and even solutions would have the same, favorable C-N dynamics on a time interval of $2\Delta t$. So we can think of this system as a blend of the two: one which is unconditionally stable, the other which is unconditionally unstable. For mildly nonlinear diffusion, it can be an attractive way to time-center the nonlinearity without requiring iteration in each timestep, and still retain second-order temporal accuracy for the linear dynamics. Because of the three levels in time, the system is not self-starting; it needs additional information in terms of IC's at $t = 0$ as well as $t = -\Delta t$. In this lies the seeds of its potential failure.

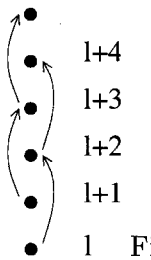


Figure 5.14: The odd-even decoupling of the implicit leapfrog system.

The Fourier transform of this system gives us

$$\gamma - \frac{1}{\gamma} = -\left[8r\sin^2\left(\frac{S}{2}\right)\right]\left[\frac{\theta}{2}(\gamma + \frac{1}{\gamma}) + (1-\theta)\right] \tag{5.120}$$

If we temporarily identify $R(S) \equiv 4r\sin^2\left(\frac{S}{2}\right)$, $(0 \leq R \leq 4r)$, then this quadratic equation is

$$\gamma^2\left[1 + R\theta\right] + \gamma\left[2R(1-\theta)\right] + \left[-1 + R\theta\right] \tag{5.121}$$

The roots are

$$\gamma = \frac{-R(1-\theta) \pm \sqrt{R^2(1-\theta)^2 - [1+R\theta][-1+R\theta]}}{[1+R\theta]} \tag{5.122}$$

$$= \frac{-R(1-\theta) \pm \sqrt{R^2(1-2\theta)+1}}{[1+R\theta]} \tag{5.123}$$

There are 2 roots – at any value of σh, there are two different temporal dynamics, γ^+ and γ^-. Both must be stable if a solution is to be possible. The requirement $|\gamma| \leq 1$ for this system leads to[2]

$$2R|1-\theta| \leq 2R\theta \tag{5.124}$$

$$|1-\theta| \leq \tag{5.125}$$

$$\theta \geq 0.5 \tag{5.126}$$

So the overall assessment of this 3-level system depends unequivocally on θ:

- $\theta \geq 0.5$ gives unconditional stability
- $\theta \leq 0.5$ gives unconditional instability

("Unconditional" here indicates that there is no dependence on timestep r.) If we conceive of the system as a blend of explicit Leapfrog and implicit Crank-Nicolson systems, then the instability in the former is quenched at the halfway point.

Case A: $\theta = 1$ (Crank-Nicolson)

From 5.123 above, we have

$$\gamma = \pm\frac{\sqrt{1-R^2}}{1+R} = \pm\sqrt{\frac{(1+R)(1-R)}{(1+R)(1+R)}} = \pm\sqrt{\frac{1-R}{1+R}} \tag{5.127}$$

[2]The student is encouraged to verify this.

We know that odd and even systems are uncoupled; so γ^2 represents the progress of the solution over two time steps:

$$\gamma^2 = \frac{1-R}{1+R} = \frac{1-4r\sin^2\left(\frac{S}{2}\right)}{1+4r\sin^2\left(\frac{S}{2}\right)} \tag{5.128}$$

This is identical to the original Crank-Nicolson (2-level with $\theta = 1/2$) expression, equation 5.109, with the adjustment for the effective timestep $2\Delta t$. So we have recovered what we knew by inspection: that this system must perform identically to the C-N system operating at twice the timestep.

But we can learn more here. For $R > 1$, γ^2 is negative (and stable). So γ is *imaginary*. This characterizes the shortest wavelength modes when $r > 1/4$; increasing r increases the portion of the spectrum so affected. This corresponds to temporal oscillations of period $4\Delta t$; they are positive at $t = 0$, imaginary at $t = \Delta t$, negative at $t = 2\Delta t$, imaginary at $t = 3\Delta t$, positive at $t = 4\Delta t$, and so on; with overall size decreasing as t progresses. If we ignore the odd-numbered timesteps, the behaviour is identical to C-N behaviour when r is too big – stable but non-monotone dynamics. This characterizes both roots, γ^+ and γ^-. Neither has any merit vis-a-vis the continuous system.

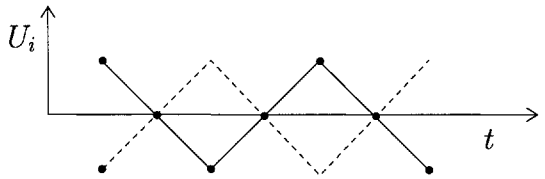


Figure 5.15: Parasitic modes of period $4\Delta t$ arising at the short-wave (poorly resolved) end of the spectrum, when $\gamma^2 < 0$. Dot and Dash lines represent time histories of two adjacent nodes.

For $R \leq 1$, γ^2 is positive (and stable). But now we need to distinguish between the two values of γ. One will be positive and will converge to the continuous version – the "physical" mode. The other will be negative, a mode which changes sign every timestep. This mode has no counterpart in the continuous system; it is "nonphysical" or "parasitic". As such, it breeds on trash in the system. If IC's and BC's are physical, *i.e.* are compatible with the continuous system, then the parasitic modes should not be introduced. But, imperfect algebra (roundoff error, imprecise data, etc.) will inject these modes into the system. And perfectly physical data will not perfectly match the imperfect discrete system, either. The parasitic modes will therefore be present in any practical simulation of this 3-level system. It is significant that the well-resolved Fourier modes have $R \to 0$, and therefore are prone to this type of parasite.

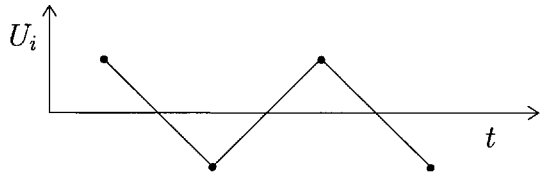


Figure 5.16: Parasitic mode of period $2\Delta t$ arising at the long-wave (well-resolved) end of the spectrum, when $\gamma^2 > 0$. Neighboring nodes have the same time history.

Even perfectly benign-sounding conditions can initiate the parasites. Imagine starting this system "at rest", $U(x) = 0$ at $t = 0$ *and* at $t = -\Delta t$; and imposing a step input in the left-hand BC for all $t > 0$. The computed sequences at Δt, $3\Delta t$, $5\Delta t$, ... would be the same as that at $2\Delta t$, $4\Delta t$, $6\Delta t$, ... The result would be timeseries with this shape shown in figure 5.18. The superposition of the two modes represents a smooth "physical" trend with an oscillation of period $2\Delta t$ superimposed.

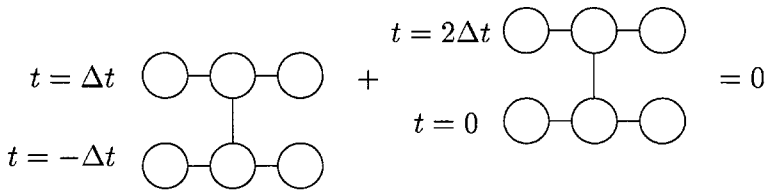


Figure 5.17: Implicit Leapfrog molecule with $\theta = 1$. This molecule is the summation of two Crank-Nicolson molecules. Odd and even solutions are *uncoupled.*

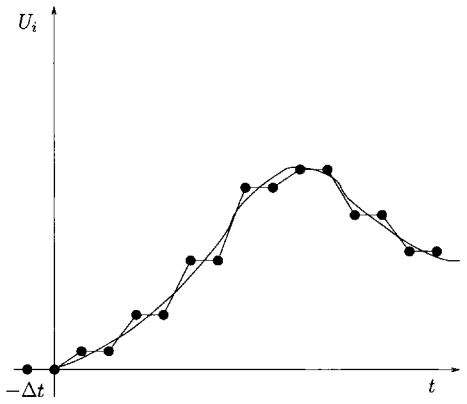


Figure 5.18: Nodal time series in which odd and even solutions are perfectly decoupled and identical.

Case B: $\theta = 1/2$ (Trapezoidal Rule)

This case is at the boundary of unconditional instability. From 5.123 above, we have

$$\gamma = \frac{-R/2 \pm 1}{1 + R/2} \tag{5.129}$$

The positive option is the "physical" one; in fact it is identical to C-N in every way. This molecule is easily visualized as the sum of two sequential C-N molecules (Figure 5.19); so the C-N solution satisfies this compound molecule.

But that is not the whole story. The negative option in 5.129 is

$$\gamma = -1 \tag{5.130}$$

This is a pure parasite. It affects every spatial mode, independent of resolution. It oscillates undamped with a period of $2\Delta t$. It is most easily inserted into the system via IC's. Suppose that the IC's at $t = 0$ and $t = -\Delta t$ exactly satisfied a C-N molecule. Then the very first 3-level step

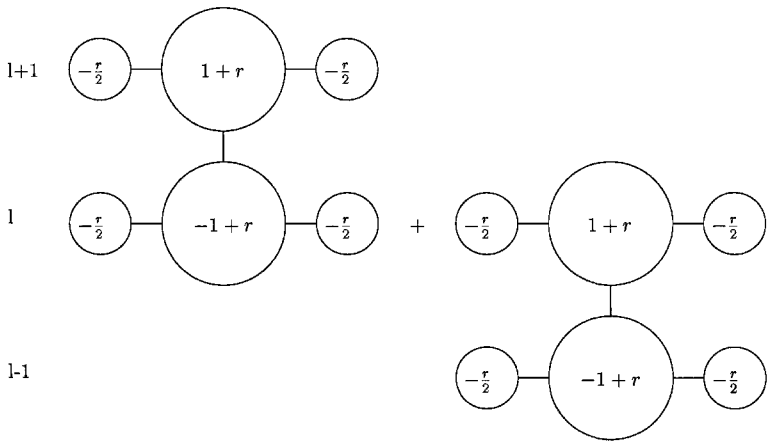


Figure 5.19: Implicit Leapfrog molecule with $\theta = 1/2$.

would have to satisfy the C-N molecule from time t to Δt. And so on this would go, each step generating the C-N solution over the last half of its 3-level molecule. However, suppose the IC's did not have this property. Then the first time step would negate the initial C-N inequality by producing its negative in the interval t to Δt. And this pattern would repeat indefinitely. The key to parasite avoidance would seem to be in the IC's. But any level of machine imprecision would certainly upset the pattern, thus continuously injecting parasitic modes even if they were not there initially. Here it is worth it to remember that the extra IC demanded by this system, at $t = -\Delta t$, is beyond that necessary/sufficient for the continuous system. So we have a liability here.

How to get rid of parasitic solutions for this system? Here are some approaches:

- Carefully manage the necessary extra IC's, and the precision.
- Generate the extra IC's by taking a C-N or other 2-level step first.
- Every so often: take another two level step, *e.g.* backward Euler, which amounts to injecting smoothing or filtering the solution.
- Avoid 3-level systems for parabolic problems where possible.

In closing this section, recall the attraction of the 3-level system to begin with: for mildly nonlinear problem we might linearize over $2\Delta t$ without causing an uncentered temporal truncation error. This would appear to be the only justification for this system, given its propensity for parasites.

5.6 Conservation Laws

Most of the PDE's of classical origin amount to conservation statements of some kind. Where there is a physical (or empirical) principal of conservation, its expression on an infinitesimal control volume leads to a PDE which expresses the principle at a point in a continuum. That statement alone is only half the process. To it we commonly add a *constitutive relation* which relates the vector (tensor) flux of the conserved quantity, to a suitable scalar (vector) potential and/or its gradient. This process toward the Poisson equation, for example, is illustrated here:

$$\begin{array}{lll} \textit{Conservation Law} & + \quad \textit{Constitutive Relation} & \Rightarrow \textit{PDE} \\ \nabla \cdot q = \sigma & \qquad q = -\kappa \nabla U & \qquad \nabla \cdot \kappa \nabla U = -\sigma \end{array} \tag{5.131}$$

q	Conserved	U	σ
Heat Flux	Thermal Energy	T	Heating Rate
Diffusion Flux	Molecules of Species	C	Chem. Reaction Rate
Mass Flux(Porous Medium)	Fluid Mass	P	Evaporation Rate
Electrostatic Field	"Force"	ϕ	Electric Charge
Gravity Field	"Force"	ϕ	Mass
Stress	Momentum	δ (displacement)	Force

Table 5.1: Conservation Analogies

q is the vector flux of the conserved quantity; σ is its source; and U is the scalar surrogate for the flux. This conceptual process leads to most of the interesting PDE's (or at least to their elliptical part). We list some of them in Table 5.1.

In terms of units, $[q]$ = (conserved quantity)/cm^2/sec ; $[\sigma]$ = (conserved quantity)/cm^3/sec in the *cgs* system. The source term σ as used here need not be a constant; nor need it be exogenous. For example, radioactive decay: $\sigma = -kC$. And for transient problems, σ represents change in local storage: $\sigma = -\frac{\partial C}{\partial t}$. So for example in a chemical transport problem,

$$\nabla \cdot D\nabla C = \frac{\partial C}{\partial t} + kC \tag{5.132}$$

we would have

$$-\sigma = \frac{\partial C}{\partial t} + kC \tag{5.133}$$

So if we stick with the lumped system, all these terms may be treated under the rubric "source" terms – in particular the storage term which is a key feature of parabolic systems.

Now the PDE is a local conservation statement, valid anywhere on its domain. It supports a global conservation statement on any subdomain by simply integrating it,

$$\int\int\int \nabla \cdot q dv = \int\int\int \sigma dv \tag{5.134}$$

The Divergence theorem is critical here:

$$\int\int\int \nabla \cdot q dv = \oint q \cdot \hat{n} ds \tag{5.135}$$

and the familiar result states that the Rate of Escape equals the sum of Internal Sources:

$$\oint q \cdot \hat{n} ds = \int\int\int \sigma dv \tag{5.136}$$

or equivalently, in terms of U:

$$-\oint \kappa \nabla U \cdot \hat{n} ds = \int\int\int \sigma dv \tag{5.137}$$

$(\nabla U \cdot \hat{n} \equiv \frac{\partial U}{\partial n})$.

Numerically, for the lumped system we cannot hope for perfection for any finite h. However, if we define our control volumes carefully, we can recover a statement of numerical conservation

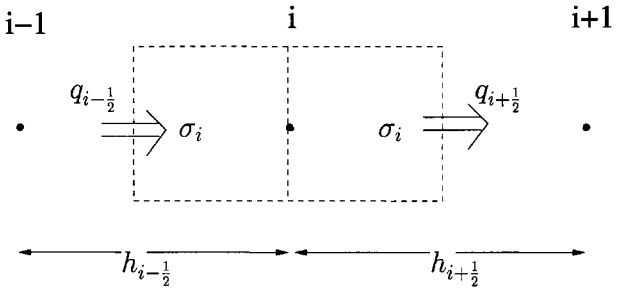


Figure 5.20: Pictorial representation of Lumped system given in equation 5.139

embedded in the lumped system. At the core we will replace the volume and surface integrals applicable to the continuous system with their numerical approximants *e.g.* $\int\int\int(\ )dv \rightarrow \sum(\ )_i \Delta v_i$.

Consider the continuous system and its lumped version

$$\frac{\partial}{\partial x}(K\frac{\partial U}{\partial x}) = -\sigma \tag{5.138}$$

$$\left[\frac{K_{i+1/2}}{h_{i+1/2}}(U_{i+1}-U_i) - \frac{K_{i-1/2}}{h_{i-1/2}}(U_i - U_{i-1})\right] = -\sigma_i \frac{(h_{i+1/2}+h_{i-1/2})}{2} \tag{5.139}$$

This reasonable approximant is $O(h)$ if h is variable; otherwise it is $O(h^2)$. It is most easily imagined as the integral of the PDE from $x_{i-1/2}$ to $x_{i+1/2}$, depicted in Figure 5.20. Notice here that we are *not* differentiating the coefficient $K(x)$ analytically; we are letting the numerical differentiation handle it. The practical advantage of this is enormous in real simulations where a smooth, differentiable function $K(x)$ is not available. There are three important things to notice about this lumped system:

- The "molecule" corresponds to a conservation statement over the two half-boxes from $x_{i-1/2}$ to $x_{i+1/2}$, with influx $q_{i-1/2} = -\frac{K_{i-1/2}}{h_{i-1/2}}(U_i - U_{i-1})$ and efflux $q_{i+1/2} = -\frac{K_{i+1/2}}{h_{i+1/2}}(U_{i+1} - U_i)$
- The lumped source term σ_i represents the average of all sources in the two half-boxes :

$$\sigma_i = \frac{\int_{x_{i-1/2}}^{x_{i+1/2}} \sigma dx}{x_{i+1/2} - x_{i-1/2}} \tag{5.140}$$

  This is especially important when σ is to be represented as Dirac delta function; or when the mesh lengths h are variable. In the context of parabolic systems, σ_i represents the average accumulation rate over the two half-boxes.

- $K_{i+1/2}$ embodies all $K(x)$ variation between x_i and x_{i+1}. Instinctively, this is some kind of average. But the average we want is the *harmonic mean* of K. To generate this, consider steady flow rate with $\sigma = 0$. In this case, $-q = K\frac{\partial u}{\partial x}$ is spatially constant. Its integration produces

$$-q\int_i^{i+1} \frac{dx}{K} = U_{i+1} - U_i \Rightarrow -q_{i+1/2} = \frac{U_{i+1}-U_i}{\int \frac{dX}{K}} \tag{5.141}$$

  So the proper lumped-system constitutive relation is based on the parameter $K_{i+1/2}$

$$\frac{K_{i+1/2}}{h_{i+1/2}} = \frac{1}{\int_i^{i+1} \frac{dX}{K}} \tag{5.142}$$

With these features in mind, the lumped system amounts to a set of finite conservation statements, on subdomains or "boxes" defined by the system equations themselves. Summing them causes the interior fluxes $q_{i+1/2}$ to cancel, giving us a global conservation statement

$$\begin{aligned} -q_{N+1/2} + q_{1/2} &= -\sum_{1}^{N} \sigma_i \frac{(h_{i+1/2} + h_{i-1/2})}{2} \\ &= -\sigma_1 h_{1/2}/2 - \sum_{i=1}^{N-1} \frac{(\sigma_i + \sigma_{i+1})}{2} h_{i+1/2} - \sigma_N h_{N+1/2}/2 \end{aligned} \tag{5.143}$$

This is obviously a conservation statement. The influx and efflux from the system of equations are balanced against a trapezoidal-rule integration of the interior sources as illustrated in Figure 5.21.

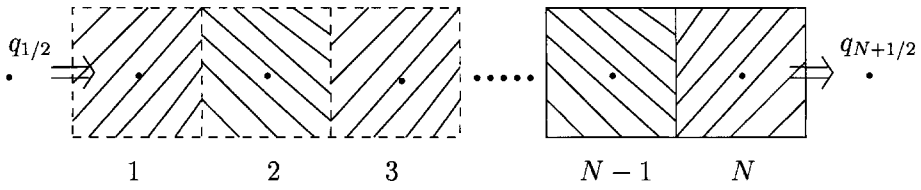


Figure 5.21: Summing the lumped system relations 5.139 gives the global conservation statement for the Dirichlet problem.

If U_0, U_{N+1} are known (Dirichlet data) then there is nothing more to the system – we have used all the relations for the conventional Dirichlet problem. But notice that the boundaries of the conservation statement are awkwardly arranged relative to the BC location. And there seems to be no role for sources near the boundaries, σ_0, σ_{N+1}. Even in the simple case of type I BC's, we would still like to compute q_0, q_{N+1} from an approximation to $\frac{\partial U}{\partial x}$. But that is not represented here, and in many applications it is crucial to compute the flux occurring on a boundary. So we would like to do better. Specifically, we would like to expand the conservation boundaries over the two half-boxes on the ends, and involve the natural quantities q_0 and q_{N+1} which are presently ignored. Since we have to do this anyway in the case of Neumann (Type II) or Type III BC's, we will consider that next.

Boundary Conditions

For the general Type II BC's our approach is

- Write the PDE approximation at the boundary. The molecule will spill over into a "shadow" or image region;
- Write the BC approximation at the boundary. It too will spill over into the shadow region. Use it to eliminate the image node;
- Merge the two to eliminate the image nodes. The result is the FD molecule or lumped system representation at the boundary, representing both PDE and BC.

Consider the left-hand boundary at $x = 0$, node 0. We will invent a shadow region to the left, node -1, as shown in Figure 5.22. The shadow region will be symmetric relative to the mesh: $h_{-1/2} = h_{1/2}$. Furthermore, we will let $K(x)$ be symmetric about the boundary, so $K_{-1/2} = K_{1/2}$. With these provisos, the PDE approximation 5.139 reduces to

$$K_{1/2} \frac{(U_{-1} - 2U_0 + U_1)}{h} = -\sigma_0 h \tag{5.144}$$

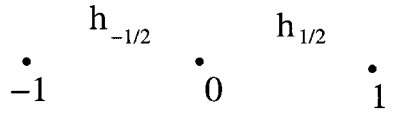


Figure 5.22: Left boundary of the conservation example. Node -1 is a shadow node; Node 0 is on the boundary.

Now the BC is in terms of the known Neumann data q_0:

$$-K\frac{\partial U}{\partial x} = q_0 \tag{5.145}$$

and its lumped representation is

$$-K_0\frac{(U_1 - U_{-1})}{2h} = q_0 \tag{5.146}$$

and since we have symmetry, K_0 is the same as $K_{1/2}$:

$$K_0 = \frac{\int_{-h}^{+h} dx}{\int_{-h}^{+h} \frac{dx}{K}} = \frac{\int_{0}^{+h} dx}{\int_{0}^{+h} \frac{dx}{K}} = K_{1/2} \tag{5.147}$$

Now, we put the two relations together. Solving 5.146 for U_{-1} we get

$$U_{-1} = U_1 + \frac{2h}{K_0}q_0 \tag{5.148}$$

and its elimination from 5.144 gives us the result

$$K_{1/2}\frac{(U_1 - U_0)}{h_{1/2}} = -\frac{\sigma_0 h_{1/2}}{2} - q_0 \tag{5.149}$$

or equivalently,

$$-q_{1/2} + q_0 = \frac{-\sigma_0 h_{1/2}}{2} \tag{5.150}$$

This is the missing link in the conservation relation; it is a conservation statement in the half-box on the left side of the system. It is illustrated in Figure 5.23. The Neumann flux q_0 is indistinguishable from the sources in the half-box adjacent to the boundary, in accord with intuition.

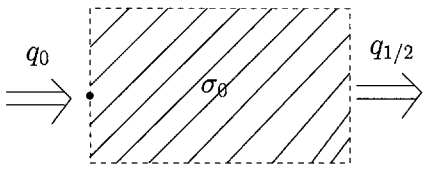


Figure 5.23: Conservation statement at Neumann Boundary.

The analogous procedure at the right side of the system gives

$$-q_{N+1} + q_{N+1/2} = \frac{-\sigma_{N+1} h_{N+1/2}}{2} \tag{5.151}$$

Adding these two boundary relations to the interior sum gives the whole-system conservation statement:

$$-q_{N+1} + q_0 = -\sum_{i=0}^{N}(\frac{\sigma_i + \sigma_{i+1}}{2})h_{i+1/2} \tag{5.152}$$

This is the final conservation statement for the Neumann system. It is exact, even though the lumped system is not. It requires special care in defining the BC's, the spatial variations in the parameters, and the source terms; the sense of all these is implicit in the lumped-system statement or FD molecule. The FD molecule also defines the sense of spatial integration (trapezoidal rule in this case).

For the type III BC we have

$$-K\frac{\partial U}{\partial x} = q_0 = a_0 U_0 + b_0 \tag{5.153}$$

The boundary balance 5.149 becomes

$$K_{1/2}\frac{(U_1 - U_0)}{h_{1/2}} + a_0 U_0 = \frac{-\sigma_0 h_{1/2}}{2} - b_0 \tag{5.154}$$

and a similar result holds for the right-hand side boundary. This BC defines an additional endogenous source in the overall balance, $a_0 U_0$; b_0 plays the role of q_0 in the Neumann case.

Now back to the Type I boundary. With U_0 given, there is no need for the boundary molecule 5.149, *unless* we want to obtain the flux q_0. In that case, 5.149 is *the equation for the boundary flux*. Its rearrangement gives us

$$q_0 = -K_{1/2}(\frac{U_1 - U_0}{h_{1/2}}) - \frac{\sigma_0 h_{1/2}}{2} \tag{5.155}$$

Now this is a remarkable result. It looks like a conventional, one-sided approximation to the constitutive relation $q = -K\frac{\partial U}{\partial x}$, with an extra term in σ added on. The added term is first-order in the mesh spacing h, so one is tempted to ignore it. However we know that a) this molecule was arrived at by blending second-order approximations for the PDE and for the BC; and b) that it conserves *exactly*. So the correct interpretation is, the first-order correction for σ near the boundary restores this otherwise one-sided, first order approximation for q to be second-order correct. A similar interpretation applies to the placement of $K(x)$ in this molecule. Use of this approach gives us the identical set of lumped-system relations among $U_i, i = [0, N+1]$ and the boundary fluxes (q_0, q_{N+1}), irrespective of the type of BC implemented.

The same general approach can guarantee an exact conservation statement in higher dimensions. For example, in 2-D, for the Poisson equation

$$\nabla \cdot (K \nabla U) = -\sigma \tag{5.156}$$

we have the lumped system on a uniform grid:

$$\begin{aligned} &\frac{1}{h}\left[K_{i+1/2}(\frac{U_{i+1} - U_i}{h}) - K_{i-1/2}(\frac{U_i - U_{i-1}}{h})\right] \\ &+\frac{1}{k}\left[K_{j+1/2}(\frac{U_{j+1} - U_j}{k}) - K_{j-1/2}(\frac{U_j - U_{j-1}}{k})\right] \quad = \quad \sigma_{ij} \end{aligned} \tag{5.157}$$

with $k = \Delta y$. Multiplying by hk gives us

$$k[-q^x_{i+1/2} + q^x_{i-1/2}] + h[-q^y_{j+1/2} + q^y_{j-1/2}] = -hk\sigma_{ij} \tag{5.158}$$

and we have a local, exact balance among influx, efflux, and sources as depicted in Figure 5.24. The issues pertaining to the definition of averages for K and σ, the cancellation of interior fluxes, and the imposition of BC's are unchanged from those just exposed in the 1-D case.

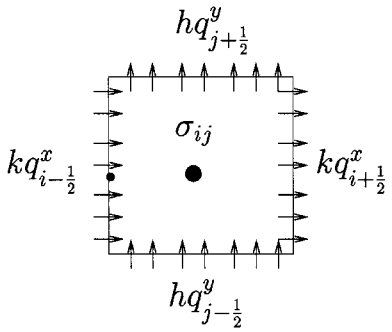


Figure 5.24: Conservation box for 2-D Poisson equation.

5.7 Two-Dimensional Problems

Most approaches to parabolic problems in 2 elliptic dimensions (x, y) or more are generalizations or the 1-D approaches described above. The generalization of the Crank-Nicolson system in multi-D remains the gold standard for parabolic equations. As an example, consider the 2-D Diffusion equation

$$\frac{\partial U}{\partial t} = D\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) \tag{5.159}$$

For a regular, square grid, the general 2-level-in-time discrete system has the molecule

$$\frac{U^{l+1}_{ij} - U^l_{ij}}{\Delta t} = \frac{D}{h^2}\delta^2\left(U^{l+1}_{ij}\theta - (1-\theta)U^l_{ij}\right) \tag{5.160}$$

with the FD version of the Laplacian

$$\delta^2 = \delta^2_x + \delta^2_y \tag{5.161}$$

This system is *pentadiagonal* in structure, like the comparable system of elliptic equations ($\frac{\partial U}{\partial t} = 0$). Grouping the known quantities at time level l we get

$$[U_{ij} - r\theta\delta^2 U_{ij}]^{l+1} = [U_{ij} + (1-\theta)r\delta^2 U_{ij}]^l \equiv g^l_{ij} \tag{5.162}$$

with dimensionless timestep $r = \frac{D\Delta t}{h^2}$ as usual. The problem amounts to an elliptic solution in each time step, with a new right-hand side built from the latest known solution. In the linear case as shown, the system is stationary. Expanding the Laplacian at the unknown time level $l+1$, we have

$$-r\theta(U_{i,j-1} + U_{i,j+1} + U_{i+1,j} + U_{i-1,j})^{l+1} + (1+4r\theta)U^{l+1}_{ij} = g^l_{ij} \tag{5.163}$$

This algebraic system is *diagonally dominant*. Hence we know that a) standard direct methods, *e.g.* LU decomposition, should be successful and indeed contemporary practice confirms this. Additionally, simple iterative methods for the comparable elliptic equation can be applied here with success. Jacobi and Gauss-Seidel methods will work for all r and θ; and there will be an optimal parameterization of SOR. So immediately we have a feasible, quality solution with Crank-Nicolson dynamics. For example, the Jacobi scheme in a single timestep can be written,

$$(1+4r\theta)U^{k+1}_{ij} = g_{ij} + r\theta(U_{i,j-1} + U_{i,j+1} + U_{i+1,j} + U_{i-1,j})^k \tag{5.164}$$

where we have dropped the superscript indicating time; and added an iteration counter k. The right-hand side g_{ij} is fixed during the iteration; it depends on the existing solution U_{ij} at time level l.

A particularly important (historically) class of iterative methods are the so-called Alternating Direction Implicit (ADI) methods. Both methods capitalize on the efficiency of tridiagonal direct solution, by making only part of the Laplacian implicit at any point. Hence these are "line", "block", or "implicit" iterative methods. ADI methods are conceived in either of 2 ways:

- as an iterative solution to elliptic problems, and hence as an iterative solution to parabolic problems in a single timestep
- as a time-stepping algorithm for parabolic problems.

In the former, ADI iteration is employed to convergence in each timestep. In the latter, we employ only one ADI iteration per timestep, and move on. Effectively we bury the iterative residual in the temporal error.

Iterative ADI

To expose this in context of an elliptic solver, we rewrite 5.162 in the form

$$[\delta^2 U_{ij} - \frac{1}{r\theta} U_{ij}]^{l+1} = \frac{-1}{r\theta} g_{ij}^l \tag{5.165}$$

The time superscripts l will be dropped but are still implied. The idea is to proceed in 2 steps, first implicit in the δ_x terms, explicit in the δ_y terms; then do the opposite:

- Step 1:

$$-\omega U_{ij}^{k+1} + (\delta_x^2 - \frac{1}{2r\theta}) U_{ij}^{k+1} = -(\delta_y^2 - \frac{1}{2r\theta}) U_{ij}^k - \frac{g_{ij}}{r\theta} - \omega U_{ij}^k \tag{5.166}$$

- Step 2 :

$$-\omega U_{ij}^{k+2} + (\delta_y^2 - \frac{1}{2r\theta}) U_{ij}^{k+2} = -(\delta_x^2 - \frac{1}{2r\theta}) U_{ij}^{k+1} - \frac{g_{ij}}{r\theta} - \omega U_{ij}^{k+1} \tag{5.167}$$

Both steps are tridiagonal. Notice two things in addition to the alternate splitting of the Laplacian. First, the diagonal term $\frac{1}{r\theta} U_{ij}$ has been shared equally at old and new iteration level; and second, an acceleration term has been added to both sides, ωU_{ij}. This iteration is unconditionally stable for $\omega > 0$. This is *iterative stability, i.e.* it pertains to taking a single step, iteratively. The stability over time is dictated by the stability of the basic discrete system, which depends on r and θ. The rules of thumb developed earlier in one elliptic dimension generally pertain here, typically modified by a factor of 2 in 2-D.

A single ADI iteration requires execution of *both* steps. Otherwise the iteration is unstable.

ADI as Time-Stepping

Here we divide the *dynamics* into two parts, first implicit in x and then in y. But we do not iterate within a timestep. Like the iterative solution, a single timestep requires both steps.

- Step 1:

$$\frac{U_{ij}^{l+1} - U_{ij}^l}{\Delta t} = \frac{D}{h^2} (\delta_x^2 U_{ij}^{l+1} + \delta_y^2 U_{ij}^l) \tag{5.168}$$

- Step 2:

$$\frac{U_{ij}^{l+2} - U_{ij}^{l+1}}{\Delta t} = \frac{D}{h^2}(\delta_x^2 U_{ij}^{l+1} + \delta_y^2 U_{ij}^{l+2}) \tag{5.169}$$

Rewriting to resemble the iterative method we have

- Step 1:

$$(\delta_x^2 - \frac{1}{r})U_{ij}^{l+1} = (-\delta_y^2 - \frac{1}{r})U_{ij}^{l} \tag{5.170}$$

- Step 2:

$$(\delta_y^2 - \frac{1}{r})U_{ij}^{l+2} = (-\delta_x^2 - \frac{1}{r})U_{ij}^{l+1} \tag{5.171}$$

Here $\frac{1}{r}$ plays the role of ω above. The time-stepping algorithm looks like ADI iteration for $\delta^2 U = 0$, with $\omega = \frac{1}{r}$. Hence this system is unconditionally stable!

Fourier analysis of ADI time-stepping reveals it to be a close cousin of the basic Crank-Nicolson system; its dynamics are an $O(\Delta t^2)$ approximation of the C-N dynamics. This result is based on comparing a full ADI timestep of length $2\Delta t$. The intermediate point, following only one-half of the timestep, is always unreliable.

Ames [2] offers the following demonstration of ADI consistency. It is based on the consistency of the Crank-Nicolson scheme. Begin by operating on the ADI system thus:

$$(\delta_x^2 + \frac{1}{r})[(\delta_x^2 - \frac{1}{r})U^{l+1} = (-\delta_y^2 - \frac{1}{r})U^{l}] \tag{5.172}$$

$$(\delta_x^2 - \frac{1}{r})[(\delta_x^2 + \frac{1}{r})U^{l+1} = (-\delta_y^2 + \frac{1}{r})U^{l+2}] \tag{5.173}$$

Now subtracting 5.173 from 5.172 we get

$$\begin{aligned} 0 &= (\delta_x^2 + \frac{1}{r})(-\delta_y^2 - \frac{1}{r})U^l - (\delta_x^2 - \frac{1}{r})(-\delta_y^2 + \frac{1}{r})U^{l+2} \\ &= \delta_x^2\delta_y^2(-U^l + U^{l+2}) + \frac{1}{r^2}(-U^l + U^{l+2}) - \frac{1}{r}(\delta_x^2 + \delta_y^2)(U^l + U^{l+2}) \end{aligned} \tag{5.174}$$

Multiplying by r^2,

$$r(\delta_x^2 + \delta_y^2)(U^l + U^{l+2}) = (U^{l+2} - U^l) + r^2\delta_x^2\delta_y^2(U^{l+2} - U^l) \tag{5.175}$$

and dividing by $2\Delta t$:

$$\frac{D}{h^2}(\delta_x^2 + \delta_y^2)(\frac{U^l + U^{l+2}}{2}) = \frac{(U^{l+2} - U^l)}{2\Delta t} + D^2\Delta t^2 \frac{\delta_x^2}{h^2}\frac{\delta_y^2}{h^2}(\frac{(U^{l+2} - U^l)}{2\Delta t}) \tag{5.176}$$

Ignoring the truncation error terms, these difference equations are exactly the C-N difference equations for a timestep of $2\Delta t$. Operating on the truncation terms, we obtain the final result:

$$D\nabla^2 U + O(h^2) = \frac{\partial U}{\partial t} + O(\Delta t^2) + D^2\Delta t^2\left[\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}\frac{\partial U}{\partial t} + O(h^2 + \Delta t^2)\right] \tag{5.177}$$

So for a full 2-part timestep, ADI is a consistent, second-order approximant to Crank-Nicolson.

5.8 Nonlinear Problems

Here we will sketch a few ideas about the nonlinear diffusion equation,

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}(D\frac{\partial U}{\partial x}); \quad D = D(U) \tag{5.178}$$

A standard two-level scheme is

$$\begin{aligned} \frac{U_i^{l+1} - U_i^l}{\Delta t} &= \left[D_{i+1/2}(\frac{U_{i+1} - U_i}{h}) - D_{i-1/2}(\frac{U_i - U_{i-1}}{h})\right]^{l+1} \frac{\theta}{h} \\ &+ \left[D_{i+1/2}(\frac{U_{i+1} - U_i}{h}) - D_{i-1/2}(\frac{U_i - U_{i-1}}{h})\right]^{l} \frac{1-\theta}{h} \end{aligned} \tag{5.179}$$

To get $O(\Delta t^2)$, iteration is necessary in each timestep. A typical (simple) approach is to iterate as if D were not dependent on U, and form a diffusion equation along conventional lines. This is illustrated in Figure 5.25. Following solution for U, $D(U^{l+1})$ would be recomputed using the most recent iterate, and the same timestep reformed and recalculated.

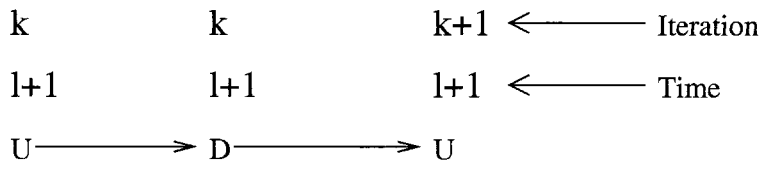


Figure 5.25: Simple iteration on nonlinear $D(U)$ within a timestep. Time is fixed while the iteration advances. Each computation of U is a discrete Elliptic PDE solution.

This is easy to program and debug; values of D are lagged and each try at the time step is just like a linear step. The iteration is nonstationary – the matrices for a timestep need to be recomputed in each iteration. So speed of iterative convergence is important. The solution algorithm will be tridiagonal for 1-D (pentadiagonal for 2-D, etc.) and if we are lucky, this will work. If it does, the $D(x)$ sensitivity is probably small.

For large $\frac{dD}{dU}$ we may need to do something more sophisticated. In this category we have the Newton-Raphson approach to nonlinear solution. It is less common, and more work, since the system to be solved in each iteration does *not* resemble a simple diffusion system. This method converges fast – with each subsequent convergence error the square of its predecessor – provided we have a good initial guess. In a time-stepping context, a good guess is always available – the value at the end of the last step.

Here is the Newton-Raphson method for coupled nonlinear equations. Denote the finite difference "molecule" centered at node i as f_i:

$$f(U_1, U_2, \ldots U_N) = 0 \tag{5.180}$$

A Taylor series expansion gives

$$0 = f_i(\bar{U}^k) + \frac{df_i}{dU_1}|_k(U_1^{k+1} - U_1^k) + \frac{df_i}{dU_2}|_k(U_2^{k+1} - U_2^k) + \ldots \tag{5.181}$$

and keeping only the first-order terms we get

$$0 \simeq f_i(\bar{U}^k) + (\frac{df_i}{dU_1}, \frac{df_i}{dU_2}, \ldots \frac{df_i}{dU_N}).(\Delta U^{k+1}) \tag{5.182}$$

This is for a single equation i; assembling all the equations gives

$$\begin{bmatrix} \frac{\partial f_1}{\partial U_1} & \frac{\partial f_1}{\partial U_2} & \cdots & \frac{\partial f_1}{\partial U_N} \\ \frac{\partial f_2}{\partial U_1} & \frac{\partial f_2}{\partial U_2} & \cdots & \frac{\partial f_2}{\partial U_N} \\ \vdots & & & \\ \frac{\partial f_N}{\partial U_1} & \frac{\partial f_N}{\partial U_2} & \cdots & \frac{\partial f_N}{\partial U_N} \end{bmatrix}^k \begin{Bmatrix} \Delta U_1 \\ \Delta U_2 \\ \vdots \\ \Delta U_N \end{Bmatrix}^{k+1} = \begin{Bmatrix} -f_1 \\ -f_2 \\ \vdots \\ -f_N \end{Bmatrix}^k \tag{5.183}$$

The coefficient matrix is the Jacobian $[J]$; $\Delta U_i^{k+1} = U_i^{k+1} - U_i^k$ is the update in iteration k. In compact form,

$$[J]^k \{\Delta U\}^{k+1} = \{-f\}^k \tag{5.184}$$

This iteration will generally be nonstationery, as above. And for a tridiagonal system it will be tightly banded, (possibly tridiagonal), etc. But it is harder to program and debug, since the Jacobian of all the FD molecules needs to be obtained. That is the penalty for faster convergence.

For example: let f_i be the FD molecule 5.179:

$$f_i = \left\{ D_{i+1/2} \frac{U_{i+1} - U_i}{h} - D_{i-1/2} \frac{U_i - U_{i-1}}{h} \right\}^{l+1} \frac{\theta}{h} - \frac{U_i^{l+1}}{\Delta t} + g_i(U^l) \tag{5.185}$$

with g_i known and constant through the iteration. Then we have

$$\frac{\partial f_i}{\partial U_i^{l+1}} = \left[-\frac{D_{i+1/2}}{h} - \frac{dD_{i+1/2}}{dU_i} \frac{U_i}{h} - \frac{-D_{i-1/2}}{h} - \frac{dD_{i-1/2}}{dU_i} \frac{U_i}{h} \right] \frac{\theta}{h} - \frac{1}{\Delta t} \tag{5.186}$$

$$\frac{\partial f}{\partial U_{i+1}^{l+1}} = \left[\frac{D_{i+1/2}}{h} + \frac{dD_{i+1/2}}{dU_{i+1}} \frac{U_{i+1}}{h} \right] \frac{\theta}{h} \tag{5.187}$$

$$\frac{\partial f}{\partial U_{i-1}^{l+1}} = \left[\frac{D_{i-1/2}}{h} + \frac{dD_{i-1/2}}{dU_{i-1}} \frac{U_{i-1}}{h} \right] \frac{\theta}{h} \tag{5.188}$$

Alternatively, we could extrapolate over Δt from the start of the time step (one iteration of the strategy mentioned first; an $O(\Delta t)$ approach); or use a centered, 3-level scheme to achieve second-order accuracy. The latter can be very attractive for stronger nonlinearities, provided the parasitic mode can be dealt with (not obvious).

Alternatively we can get an $O(\Delta t^2)$ explicit scheme by pursuing a second-order explicit time stepping algorithm. For the diffusion equation

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \tag{5.189}$$

we have the second derivative

$$\frac{\partial^2 U}{\partial t^2} = \frac{\partial^2 (\frac{\partial U}{\partial t})}{\partial x^2} = \frac{\partial^4 U}{\partial x^4} \tag{5.190}$$

and the Taylor Series extrapolation gives

$$U^{l+1} = U^l + \frac{\partial U^l}{\partial t} \Delta t + \frac{1}{2!} \frac{\partial^2 U}{\partial t^2} \Delta t^2 + \ldots \tag{5.191}$$

Hence, a time-stepping system can be built on the extrapolation

$$U^{l+1} = U^l + \frac{\partial^2 U^l}{\partial x^2} \Delta t + \frac{1}{2!} \frac{\partial^4 U^l}{\partial x^4} \Delta t^2 \tag{5.192}$$

The first two terms on the right are the usual Euler system. The higher derivatives have spread the FD footprint beyond its usual range for the Laplacian. Probably the most important practical implication is at the boundaries, where comparable sophistication is needed for any of the conventional BC's. Otherwise, one loses the $O(\Delta t^2)$ advantage of this idea.

Chapter 6

Hyperbolic Equations

6.1 Introduction

Here we concern ourselves with the general hyperbolic form

$$\frac{\partial^2 U}{\partial t^2} - \nabla \cdot C^2 \nabla U = f(U, \frac{\partial U}{\partial t}, \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}) \tag{6.1}$$

We have isolated the highest derivatives on the left-side, and separated the dimension t from the rest of the system, which alone is *elliptic*. The canonical example is the Telegraph equation

$$\frac{\partial^2 U}{\partial t^2} + \tau \frac{\partial U}{\partial t} - C^2 \frac{\partial^2 U}{\partial x^2} = 0 \tag{6.2}$$

which we will study in detail. The sign separating the elliptical operator and $\frac{\partial^2}{\partial t^2}$ is crucial. As stated, the t-dimension is completely different from the others. Reverse this sign and we have a wholly elliptic problem, with t no different from the other coordinates. In practice hyperbolic equations commonly represent the time-evolution of systems which are elliptic in steady-state but which support wave-like transients. So it is natural to use t for this dimension here. The first derivative term $\tau \frac{\partial U}{\partial t}$ normally represents a loss or damping effect; as stated, $\tau > 0$ corresponds to physically stable processes.

The necessary and sufficient conditions for a unique solution are sketched in Figure 6.1. The elliptic dimensions require the customary choice of Type I, II, or III BC, everywhere. The time domain requires *two* IC's. Intuitively we expect to require two pieces of information to nail down the $\frac{\partial^2}{\partial t^2}$ term; but the idea of setting them at different points in time, like in an elliptic dimension, is not allowed. Notice that there is a domain of independence from the BC's, where IC's alone determine the solution. This is a basic feature of Hyperbolic problems – there is a finite time delay required for information to reach any interior point.

We can always decompose a hyperbolic PDE into a pair of coupled first-order (in time) PDE's. For example, for the Telegraph equation, we introduce the additional dependent variable $V(x,t)$ as follows:

$$\frac{\partial U}{\partial t} + \tau U = C_1 \frac{\partial V}{\partial x} \tag{6.3}$$

$$\frac{\partial V}{\partial t} = C_2 \frac{\partial U}{\partial x} \tag{6.4}$$

This pair of equations is equivalent to 6.2 provided that

$$C_1 C_2 \equiv C^2 \tag{6.5}$$

We refer to this pair of equations as the *Primitive Pair*. Differentiating 6.3 by t, and 6.4 by x, allows elimination of V and we recover the telegraph equation, as promised. The analogous operation allows elimination of U and we obtain the same telegraph equation in V also:

$$\frac{\partial^2 U}{\partial t^2} + \tau \frac{\partial U}{\partial t} - C^2 \frac{\partial^2 U}{\partial x^2} = 0 \tag{6.6}$$

$$\frac{\partial^2 V}{\partial t^2} + \tau \frac{\partial V}{\partial t} - C^2 \frac{\partial^2 V}{\partial x^2} = 0 \tag{6.7}$$

Necessary and sufficient conditions in terms of U and V are readily obtained from those on U in Figure 6.1; they are shown in Figure 6.2. A Type I BC on U is equivalent to a Type II BC on V, and vice-versa. And if we know IC's U and $\frac{\partial U}{\partial t}$ then we also know V and $\frac{\partial V}{\partial t}$.

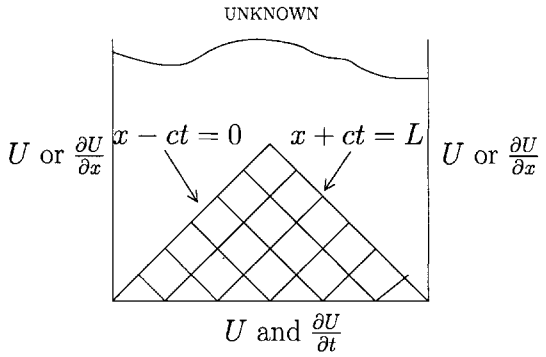


Figure 6.1: Initial and Boundary Conditions required for Hyperbolic problems. In the hatched area, the solution is independent of the BC's.

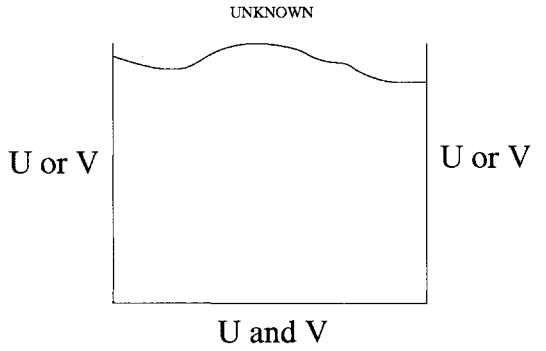


Figure 6.2: Initial and Boundary Conditions required for Hyperbolic problems – in terms of the primitive pair (U, V).

Examples follow. In each case, the primitive pair is obtained from basic principles; the higher-order telegraph equation is obtained thereafter.

1. **Acoustic Waves.** Let a fluid medium have density ρ, velocity V, and pressure P. The classical formulation involves conservation of mass and momentum, plus a constitutive relation among pressure and density.

Mass conservation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho V}{\partial x} = 0 \tag{6.8}$$

Momentum conservation:

$$\frac{\partial \rho V}{\partial t} + \frac{\partial P}{\partial x} = 0 \tag{6.9}$$

Constitutive relation:

$$P = P(\rho) \Rightarrow \frac{\partial P}{\partial \rho} \equiv \beta(\rho) \tag{6.10}$$

The primitive pair is in terms of ρ and ρV:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho V}{\partial x} = 0 \qquad \frac{\partial \rho V}{\partial t} + \beta \frac{\partial \rho}{\partial x} = 0 \tag{6.11}$$

and combination yields the classical wave equation in ρ:

$$\frac{\partial^2 \rho}{\partial t^2} - \beta \frac{\partial^2 \rho}{\partial x^2} = 0 \tag{6.12}$$

governing, for example, sound propagation in air. This formulation is lossless.

2. **Shallow Water Waves**. Let a free-surface waterbody be characterized by constant density $\rho \simeq \rho_0$, depth H, free surface elevation ζ, and horizontal velocity V. As in the acoustics example, we have conservation of mass and momentum plus a constitutive relation expressing the hydrostatic assumption.

Mass conservation:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial HV}{\partial x} = 0 \tag{6.13}$$

Momentum conservation:

$$\frac{\partial HV}{\partial t} + \frac{H}{\rho_0} \frac{\partial P}{\partial x} + \tau HV = 0 \tag{6.14}$$

Hydrostatic pressure:

$$P = \int_z^\zeta \rho g dz' \Rightarrow \frac{\partial P}{\partial x} = \rho_0 g \frac{\partial \zeta}{\partial x} \tag{6.15}$$

The primitive pair is in terms of ζ and HV:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (HV)}{\partial x} = 0 \qquad \frac{\partial HV}{\partial t} + gH \frac{\partial \zeta}{\partial x} + \tau HV = 0 \tag{6.16}$$

and the telegraph equation in ζ results:

$$\frac{\partial^2 \zeta}{\partial t^2} + \tau \frac{\partial \zeta}{\partial t} - \frac{\partial}{\partial x} \left(gH \frac{\partial \zeta}{\partial x} \right) = 0 \tag{6.17}$$

The loss term τ here represents friction at the bottom of the water column.

3. **Elastic Waves**. For an elastic medium of constant density $\rho \simeq \rho_0$, undergoing deformation with displacement field U, velocity V, strain ϵ, and stress F: we have kinematic relationships among the motion variables, plus conservation of mass and momentum (force balance); plus a constitutive relation between stress and strain.

Kinematics:

$$\epsilon = \frac{\partial U}{\partial x} \qquad V = \frac{\partial U}{\partial t} \tag{6.18}$$

Volume conservation:

$$\frac{\partial \epsilon}{\partial t} = \frac{\partial V}{\partial x} \tag{6.19}$$

Momentum conservation

$$\rho_0 \frac{\partial V}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{6.20}$$

Constitutive equations:

$$F = -K\epsilon \tag{6.21}$$

The primitive pair is in terms of ϵ, V:

$$\rho_0 \frac{\partial V}{\partial t} - \frac{\partial K\epsilon}{\partial x} = 0 \qquad\qquad \frac{\partial \epsilon}{\partial t} - \frac{\partial V}{\partial x} = 0 \tag{6.22}$$

and the wave equations in V and in U are

$$\rho_0 \frac{\partial^2 V}{\partial t^2} - \frac{\partial}{\partial x} K \frac{\partial V}{\partial x} = 0 \tag{6.23}$$

$$\rho_0 \frac{\partial^2 U}{\partial t^2} - \frac{\partial}{\partial x} K \frac{\partial U}{\partial x} = 0 \tag{6.24}$$

4. **Electric Transmission Line**. This is the origin of the term "telegraph equation". We have electric charge per unit length Q, current I, electric potential e, and three properties of the line, per unit length: Inductance I, capacitance C, and resistance R. Physical principles include conservation of electric charge, a force balance on charged particles, and the constitutive relation governing capacitance:

   Charge conservation

$$\frac{\partial Q}{\partial t} + \frac{\partial I}{\partial x} = 0 \tag{6.25}$$

   Force balance

$$L\frac{\partial I}{\partial t} + IR = -\frac{\partial e}{\partial x} \tag{6.26}$$

   Constitutive relation

$$Ce = Q \tag{6.27}$$

   These give us the primitive pair in the variables e, I:

$$C\frac{\partial e}{\partial t} + \frac{\partial I}{\partial x} = 0 \qquad\qquad L\frac{\partial I}{\partial t} + IR + \frac{\partial e}{\partial x} = 0 \tag{6.28}$$

   and the telegraph equation:

$$\frac{\partial^2 e}{\partial t^2} + \frac{R}{L}\frac{\partial e}{\partial t} - \frac{1}{LC}\frac{\partial^2 e}{\partial x^2} = 0 \tag{6.29}$$

These applications obviously share a unified underlying structure. In multiple (elliptic) dimensions, the primitive pair often involves a scalar and a vector, or in some applications, two vectors.

6.2 Lumped Systems

We have two alternate descriptions of the same phenomena: the telegraph equation and the primitive pair. With proper IC's and BC's, these descriptions are equivalent. But their Lumped System representations are different in apparently small ways. So we need to be careful to distinguish them.

For the telegraph equation, consider a conventional second-order centered approximation to the elliptic operator on a uniform mesh. This is **Lumped System # 1**:

$$\frac{d^2U_i}{dt^2} + \tau\frac{dU_i}{dt} - C^2\frac{\delta_x^2 U_i}{h^2} = 0 \tag{6.30}$$

The same discrete system could be expressed for V. The two sets of variables U_i and V_i would be completely independent, coupled only in the specification of their boundary and initial conditions. This system is very robust as we shall see.

Now the primitive pair, with second-order centered differencing of the first derivatives, **Lumped System # 2**:

$$\frac{dU_i}{dt} + \tau U_i = C_1\frac{(V_{i+1} - V_{i-1})}{2h} \tag{6.31}$$

$$\frac{dV_i}{dt} = C_2\frac{(U_{i+1} - U_{i-1})}{2h} \tag{6.32}$$

These two equation sets are completely interwoven and must be solved simultaneously.

Lumped System # 2 (6.31, 6.32) reduces to something *close* to Lumped System # 1 (6.30). But it is not identical, unlike the perfect correspondence of the two continuous systems. To expose this, we will eliminate the variables V_i from 6.31 as follows. First, take the time derivative of 6.31:

$$\frac{d^2U_i}{dt^2} + \tau\frac{dU_i}{dt} = \frac{C_1}{2h}\left(\frac{dV_{i+1}}{dt} - \frac{dV_{i-1}}{dt}\right) \tag{6.33}$$

From 6.32 we have

$$\frac{dV_{i+1}}{dt} = C_2\frac{(U_{i+2} - U_i)}{2h} \qquad \frac{dV_{i-1}}{dt} = C_2\frac{(U_i - U_{i-2})}{2h} \tag{6.34}$$

and so by substitution, we achieve the desired elimination:

$$\frac{d^2U_i}{dt^2} + \tau\frac{dU_i}{dt} = \frac{C_1C_2}{4h^2}(U_{i+2} - 2U_i + U_{i-2}) \tag{6.35}$$

Now in spirit this is the same as 6.30; but the Laplacian term is spread over a $4h$ footprint, instead of its more compact form δ^2 which spans only $2h$. For well-resolved spatial modes, these will be close to each other. But for poorly-resolved modes, there is a significant difference. Since the latter tend to dominate the stability of systems, and are responsible for parasitic modes when they are present, we can expect consideration of this difference to become important. Also, as we shall see, the $4h$ footprint significantly complicates the enforcement of BC's.

For completeness, a similar manipulation removes the U_i from 6.32, starting with its time differentiation:

$$\frac{d^2V_i}{dt^2} = \frac{C_2}{2h}\left(\frac{dU_{i+1}}{dt} - \frac{dU_{i-1}}{dt}\right) \tag{6.36}$$

$$= \frac{C_2}{2h}\left(C_1\frac{V_{i+2} - V_i}{2h} - C_1\frac{V_i - V_{i-2}}{2h}\right) - \frac{C_2\tau}{2h}(U_{i+1} - U_{i-1}) \tag{6.37}$$

The result is identical:

$$\frac{d^2V_i}{dt^2} + \tau\frac{dV_i}{dt} = \frac{C_1C_2}{4h^2}(V_{i+2} - 2V_i + V_{i-2}) \tag{6.38}$$

6.3 Harmonic Approach

Before developing the discrete systems there is an alternate, and very powerful, way forward. Since many hyperbolic problems involve finding *periodic* solutions, then we assume from the start that the solutions are time-harmonic:

$$U_i(t) = \mathcal{U}_i e^{j\omega t} \tag{6.39}$$

$$V_i(t) = \mathcal{V}_i e^{j\omega t} \tag{6.40}$$

$j = \sqrt{-1}$ is the imaginary unit. $\mathcal{U}_i$ is the complex amplitude of the solution U_i at node i; it includes amplitude and phase. ω is the radian frequency of the motion, assumed known from the forcing. Insertion of this expression into **Lumped System 1** produces

$$\left[(j\omega)^2 + \tau_j\omega\right]\mathcal{U}_i e^{j\omega t} - \frac{C^2}{h^2}\delta_x^2\,\mathcal{U}_i e^{j\omega t} = 0 \tag{6.41}$$

Dividing by ω^2 we encounter the Courant Number K:

$$K \equiv \frac{C^2}{\omega^2 h^2} \tag{6.42}$$

and the Harmonic form of Lumped System 1 is:

$$\left[(-1 + j\frac{\tau}{\omega}) - K\delta_x^2\right]\mathcal{U}_i = 0 \tag{6.43}$$

Effectively we have reduced the system to *elliptic* form, by transforming away the time dimension.

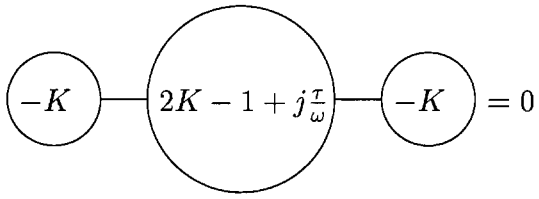


Figure 6.3: FD molecule for Harmonic System # 1, equation 6.43

Figure 6.3 illustrates this Harmonic System. The system is Tridiagonal, with complex coefficients. Generally the solutions $\mathcal{U}$ will be complex also. When $\frac{\tau}{\omega} << 1$ we have the lossless case and the matrix is real. In that case, the real and imaginary parts of $\mathcal{U}_i$ are independent of each other, and can be solved for independently. Boundary conditions are conventional elliptic types, and pose no special problems. Implementation of this system is straightforward in a computer language which supports complex data types; and it *works well* in terms of solution skill. Note however, that it is *not* diagonally dominant; so elliptic solvers which require this property are not effective here.

The same Harmonic System # 1 pertains to $\mathcal{V}$, the Fourier transform of V, with the dual boundary conditions.

Now let's introduce the system boundaries and the BC's. In Figure 6.4 we have a uniform 1-D mesh, with both $\mathcal{U}$ and $\mathcal{V}$ defined at the nodes. Suppose the BC's are Type I relative to $\mathcal{U}$. Then HS1 comprises N equations 6.43 for $i = [1 : N]$, plus the BC's $\mathcal{U}_0$ and $\mathcal{U}_{N+1}$ from the Dirichlet data. The solution for $\mathcal{U}$ is complete, in conventional elliptic form. This works.

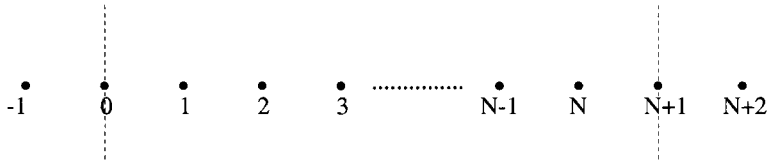


Figure 6.4: One-dimensional mesh with boundaries at nodes 0 and $N + 1$.

Now for the $\mathcal{V}$ solution. The same HS1 pertains on the interior, $i = [1 : N]$. But the BC's are the dual of those for $\mathcal{U}$. From equation 6.3, the Dirichlet data are transformed into Neumann data:

$$\frac{\partial \mathcal{V}}{\partial x} = \frac{j\omega + \tau}{C_1}\mathcal{U} \tag{6.44}$$

Now the path is clear; it is the path of the elliptic Neumann problem. Write the PDE approximation at the boundary, invoking a shadow node $i = -1$:

$$-K\mathcal{V}_{-1} + \left(2K - 1 + j\frac{\tau}{\omega}\right)\mathcal{V}_0 - K\mathcal{V}_1 = 0 \tag{6.45}$$

and the FD version of the BC:

$$\frac{1}{2h}\left(\mathcal{V}_1 - \mathcal{V}_{-1}\right) = \frac{j\omega + \tau}{C_1}\mathcal{U}_0 \tag{6.46}$$

and combine them to eliminate $\mathcal{V}_{-1}$:

$$\left(2K - 1 + j\frac{\tau}{\omega}\right)\mathcal{V}_0 - 2K\mathcal{V}_1 = -2j\frac{K}{K_1}\left(1 - j\frac{\tau}{\omega}\right)\mathcal{U}_0 \tag{6.47}$$

($K_1 \equiv C_1/\omega h$.) This is the boundary molecule. It is $O(h^2)$. The analogous procedure embeds the Dirichlet data $\mathcal{U}_{N+1}$ in the right-side boundary relation. There are now $N + 2$ equations in the $N + 2$ unknown $\mathcal{V}_i$.

The above procedure can be worked in reverse; suppose we have a Dirichlet problem in $\mathcal{V}$. Solve the system of interior molecules subject to the Dirichlet data. Then, with $\mathcal{V}_i, i = [0, N + 1]$ known, use 6.47 to determine the proper value of $\mathcal{U}_0$ and the analogous relation at the right-side for $\mathcal{U}_0$. Now solve the Dirichlet problem for $\mathcal{U}$.

The bottom line: HS1 works for both Dirichlet and Neumann problems. Except on the boundaries, $\mathcal{U}$ and $\mathcal{V}$ are completely uncoupled. At a boundary, $\mathcal{U}$ and $\mathcal{V}$ are coupled in the two dual implementations of the single piece of data required there. It is therefore possible to avoid calculation of one of these, say $\mathcal{V}$, if the only interest is in $\mathcal{U}$.

Now let's look at **Lumped System # 2** and its Harmonic representation. Introduction of $\mathcal{U}$ and $\mathcal{V}$ into LS2, equations 6.31, 6.32, we have

$$(j\omega + \tau)\mathcal{U}_i = \frac{C_1}{2h}(\mathcal{V}_{i+1} - \mathcal{V}_{i-1}) \tag{6.48}$$

$$j\omega \mathcal{V}_i = \frac{C_2}{2h}(\mathcal{U}_{i+1} - \mathcal{U}_{i-1}) \tag{6.49}$$

Introducing the dimensionless numbers[1] $K_1 = C_1/\omega h$ and $K_2 = C_2/\omega h$ we get

$$\left(1 - j\frac{\tau}{\omega}\right)\mathcal{U}_i = -j\frac{K_1}{2}(\mathcal{V}_{i+1} - \mathcal{V}_{i-1}) \tag{6.50}$$

$$\mathcal{V}_i = -j\frac{K_2}{2}(\mathcal{U}_{i+1} - \mathcal{U}_{i-1}) \tag{6.51}$$

These systems are completely coupled and must be solved simultaneously for $\mathcal{U}$ and $\mathcal{V}$. Figure 6.5 illustrates the situation.

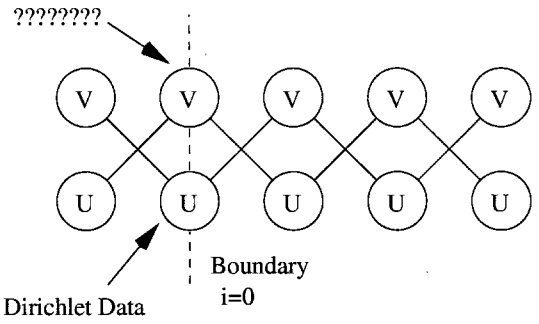


Figure 6.5: One-dimensional mesh showing relation of Lumped (Harmonic) System 1 to boundary on left. There is a Dirichlet BC on $\mathcal{U}$.

There are two potentially separate systems of equations: a) those with involving odd-numbered $\mathcal{U}_i$ and even-numbered $\mathcal{V}_i$; and b) even $\mathcal{U}_i$, odd $\mathcal{V}_i$. With Dirichlet data for $\mathcal{U}_0$, system b) is constrained. But system a) lacks a condition on $\mathcal{V}_0$.

The proper condition is a Neumann BC for $\mathcal{V}_0$, equation 6.44. But the usual approach – first derivative centered on node 0 – leaves $\mathcal{V}_0$ unconstrained. And the use of equation 6.49 does not help, since it does not involve $\mathcal{V}_0$ or $\mathcal{V}_{-1}$. One way out is to use a second-order forward difference for $\frac{\partial \mathcal{V}}{\partial x}$, involving $\mathcal{V}_0$, $\mathcal{V}_1$ and $\mathcal{V}_2$, to enforce the Neumann Condition. That is illustrated in Figure 6.6. In fact, we need not think of this explicitly as the enforcement of a Neumann BC at all – it

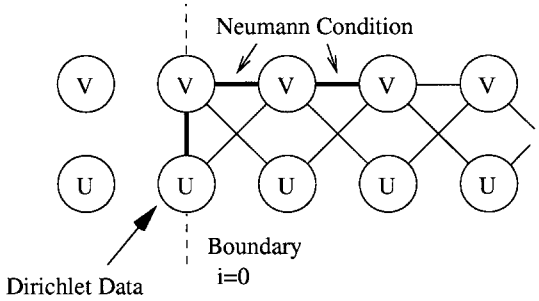


Figure 6.6: Use of second-order forward difference for $\frac{\partial \mathcal{V}}{\partial x}$ at node 0.

amounts to using a one-sided replacement for 6.48 at the boundary of the Harmonic System. This is perhaps the most natural approach.

[1]The product $K_1 K_2 = K$, the Courant number.

An alternative is to use the telegraph-type boundary condition as above. We introduce the molecule from Harmonic System 1 at node 0, equation 6.45. This links the three variables $\mathcal{V}_{-1}$, $\mathcal{V}_0$, and $\mathcal{V}_1$ together. Then the Neumann condition 6.46 involving $\mathcal{V}_{-1}$, $\mathcal{V}_1$ and the Dirichlet data for $\mathcal{U}_0$ is invoked. The result is depicted in Figure 6.7, and the merger of the boundary equations is given in equation 6.47, eliminating $\mathcal{V}_{-1}$.

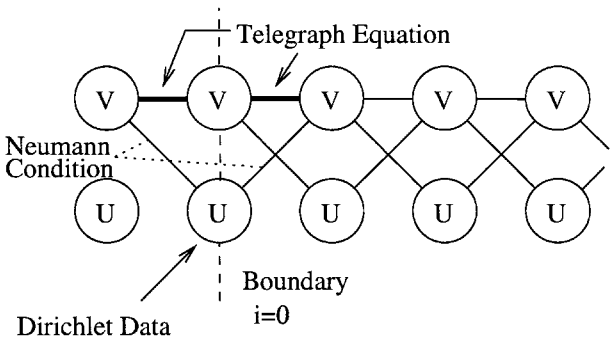


Figure 6.7: Use of telegraph equation for $\mathcal{V}_0$, combined with central differencing for $\frac{\partial \mathcal{V}}{\partial x}$ at node 0.

6.4 More Lumped Systems

Now these BC arrangements with HS2 are awkward. And the near-independence of odd and even solutions, with their potential to become uncoupled, is a threat. The complications are not a consequence of the Harmonic treatment of the time domain; they are embedded in the spatial discretization. Therefore none of these concerns get any better if we go back to the time domain – their presence can only be masked by further complexity in notation, etc. So it is natural to introduce a third Lumped System which does not have these complications. It is obtained by simply removing one of the solutions as in Figure 6.8. This is Harmonic System # 3; its time domain counterpart is Lumped System # 3. Its relations are the same as HS2/LS2, with the proviso that only odd-numbered $\mathcal{V}_i$ and even-numbered $\mathcal{U}_i$ exist. Instantly we gain a factor of 2 in the number of degrees of freedom, and get the same dynamics; or, we can cut the mesh length in half, and get better resolution for the same investment of computer resources. And, we remove the threat of odd-even uncoupling. Imposition of BC's is easy in this system; one arranges *a priori* to have the mesh terminate on either a "U" node or a "V" node, whichever variable is known at that boundary. There is no need to invent a BC which is not part of the Primitive statement – either U or V is necessary and sufficient. This general arrangement is typically called a "Staggered Mesh". As we shall see there are many varieties of staggering in multiple (elliptic) dimensions. For completeness we record here the relations for **Lumped System # 3**:

$$\frac{dU_i}{dt} + \tau U_i = C_1 \frac{(V_{i+1} - V_{i-1})}{2h} \qquad i = 2, 4, 6, \ldots \tag{6.52}$$

$$\frac{dV_i}{dt} = C_2 \frac{(U_{i+1} - U_{i-1})}{2h} \qquad i = 1, 3, 5, \ldots \tag{6.53}$$

Harmonic System # 3 is the Fourier transform of LS3; replace $\frac{\partial}{\partial t}$ with $j\omega$. To the extent that LS2 is awkward, LS3 is refreshingly straightforward. It is very popular and effective.

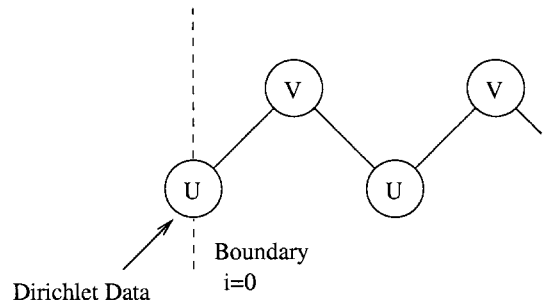


Figure 6.8: Staggered mesh. Implementation of the Primitive Pair on this mesh constitutes Lumped System # 3; its Fourier Transform is Harmonic System # 3.

Earlier we pointed out that LS2 had an effective footprint of $4h$ (equation 6.35). Now the *effective* mesh spacing in LS3 is the distance between adjacent $\mathcal{U}_i$ (or $\mathcal{V}_i$): $= 2h$. This is illustrated in Figure 6.9.

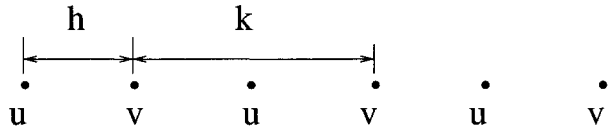


Figure 6.9: Lumped System 3. The mesh is the same as for LS1 and LS2, except that the variables are defined only at alternating nodes, with effective mesh length $k = 2h$.

Defining $k = 2h$, and retracing our steps leading to equation 6.35, we obtain

$$\frac{d^2U_i}{dt^2} + \tau\frac{dU_i}{dt} = \frac{C_1C_2}{4h^2}(U_{i+2} - 2U_i + U_{i-2}) \tag{6.54}$$

$$= \frac{C_1C_2}{k^2}(U_{i+2} - 2U_i + U_{i-2}) \tag{6.55}$$

The LS3 is identical to LS1, if the effective mesh lenghts of the two systems (distance between adjacent U_i) are made equal. Eliminating every other solution variable, and staggering the mesh, has greatly simplified our lives! There are fewer variables, no odd/even conundrum and its invitation to parasitic modes, and good correspondence with the continuous system properties. Provided complications are not introduced at the boundaries, LS1 and LS3 are achieving the same thing.

There is another kind of staggered grid, illustrated in Figure 6.10. Here we go back to defining both U and V at the same points. But instead of enforcing the PDE approximations at the nodes, we enforce them midway between the nodes. This system is used with the Primitive Pair; it is **Lumped System # 4**:

$$\frac{1}{2}\left(\frac{dU_i}{dt} + \frac{dU_{i+1}}{dt}\right) + \tfrac{\tau}{2}(U_i + U_{i+1}) = C_1\frac{(V_{i+1} - V_i)}{h_{i+1/2}} \tag{6.56}$$

$$\frac{1}{2}\left(\frac{dV_i}{dt} + \frac{dV_{i+1}}{dt}\right) = C_2\frac{(U_{i+1} - U_i)}{h_{i+1/2}} \tag{6.57}$$

Here $h_{i+1/2}$ is the distance between node i and $i+1$. This is an $O(h^2)$ system even when h is variable.

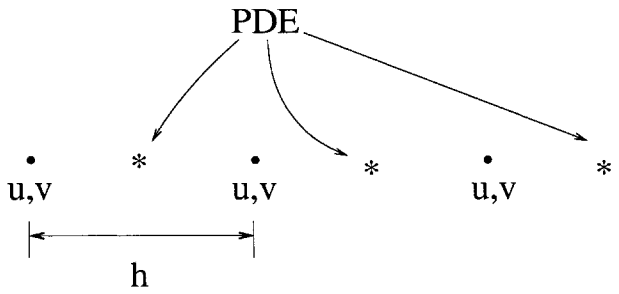


Figure 6.10: Lumped System # 4. Both U and V are defined at every node; the primitive pair of PDE's are both enforced midway between the nodes.

6.5 Dispersion Relationship

In this section we develop the dispersion relation for the continuous (analytic) and lumped systems considered so far. We are looking for correspondence between the two. Quantitative correspondence is a measure of accuracy; qualitative non-correspondence is usually an indication of the presence of parasitic modes of the lumped system which have no counterpart in the continuum.

The method of analysis – Fourier analysis of the PDE and its approximants – was introduced in Chapter 5 and the reader is advised to master that material first.

Continuous System

The governing Telegraph equation is

$$\frac{\partial^2 U}{\partial t^2} + \tau \frac{\partial U}{\partial t} - C^2 \frac{\partial^2 U}{\partial x^2} = 0 \tag{6.58}$$

We will seek solutions of the form

$$U = A e^{\alpha t} e^{j\sigma x} \tag{6.59}$$

with $j = \sqrt{-1}$. σ is the spatial wavenumber for a given Fourier mode of wavelength λ: $\sigma\lambda = 2\pi$. We will assume σ is given; we want the dispersion relationship α for a given σ. We will assume σ is real; α will generally be complex.

Inserting 6.59 into the telegraph equation we get:

$$\alpha^2 + \tau\alpha + C^2\sigma^2 = 0 \tag{6.60}$$

The solution to this quadratic equation is

$$\alpha = -\frac{\tau}{2} \pm j\sigma\sqrt{C^2 - \left(\frac{\tau}{2\sigma}\right)^2} \tag{6.61}$$

and reassembling the solution,

$$U = A e^{-\frac{\tau}{2}t} e^{j\sigma[x \pm \sqrt{C^2 - (\frac{\tau}{2\sigma})^2}t]} \tag{6.62}$$

The first part, $e^{-\frac{\tau}{2}t}$, is real and represents pure decay. It is unity in the lossless case $\tau = 0$. The second part $e^{j\sigma[x \pm \sqrt{c^2 - (\frac{\tau}{2\sigma})^2}t]}$ represents, for small τ, a propagating wave with constant amplitude

and speed $\pm\sqrt{C^2 - (\frac{\tau}{2\sigma})^2}$. The lossless speed is $\pm C$; increasing τ slows the propagation down. For large τ, the system is completely damped and waves do not propagate at all. This occurs when $\tau/2 > C\sigma = \frac{2\pi}{\lambda}C$, in which case α is real. This condition also happens at moderate τ, when the the spatial wavelength λ is very long. Otherwise, with $\tau/2 < C\sigma = \frac{2\pi}{\lambda}C$, a wave propagates and damps, and its magnitude after an elapsed time $N\Delta t$ is given by

$$\frac{|U|^{t+N\Delta t}}{|U|^t} = e^{-\tau N\Delta t/2} = [e^{-\tau\Delta t/2}]^N \tag{6.63}$$

We may also seek the dispersion relationship for the Primitive Pair:

$$\left(\frac{\partial}{\partial t} + \tau\right) U - C_1 \frac{\partial V}{\partial x} = 0 \tag{6.64}$$

$$\frac{\partial V}{\partial t} - C_2 \frac{\partial U}{\partial x} = 0 \tag{6.65}$$

In matrix form, this is

$$\begin{bmatrix} \frac{\partial}{\partial t} + \tau & -C_1 \frac{\partial}{\partial x} \\ -C_2 \frac{\partial}{\partial x} & \frac{\partial}{\partial t} \end{bmatrix} \begin{Bmatrix} U \\ V \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6.66}$$

Assuming vector solutions of the form

$$\begin{Bmatrix} U \\ V \end{Bmatrix} = \begin{Bmatrix} U_0 \\ V_0 \end{Bmatrix} e^{\alpha t} e^{j\sigma x} \tag{6.67}$$

we obtain the matrix equation

$$\begin{bmatrix} \alpha + \tau & -C_1 j\sigma \\ -C_2 j\sigma & \alpha \end{bmatrix} \begin{Bmatrix} U_0 \\ V_0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6.68}$$

For nontrivial solutions, we must have the determinant vanish. That condition is

$$\alpha^2 + \tau\alpha - C_1 C_2 (j\sigma)^2 = 0 \tag{6.69}$$

and since $C^2 \equiv C_1 C_2$, we have

$$\alpha^2 + \tau\alpha + C^2\sigma^2 = 0 \tag{6.70}$$

This quadratic equation in α is the same as that obtained from the telegraph equation. These two descriptions have identical dispersion relations.

Lumped System # 1

Lumped System #1 is the discretized telegraph equation:

$$\frac{\partial^2 U_i}{\partial t^2} + \tau \frac{\partial U_i}{\partial t} - \frac{C^2}{h^2}\delta^2 U_i = 0 \tag{6.71}$$

As before we will assume solutions of the form

$$U = e^{\alpha t} e^{j\sigma x} \tag{6.72}$$

and as before, for a given Fourier mode σ we have

$$U_{i+1} = e^{j\sigma h} U_i \qquad U_{i-1} = e^{-j\sigma h} U_i \tag{6.73}$$

and the difference operator δ^2 produces

$$\delta^2 U_i = \left[e^{j\sigma h} - 2 + e^{-j\sigma h} \right] U_i \tag{6.74}$$

$$= -\left[4 \sin^2 \left(\frac{\sigma h}{2} \right) \right] U_i \tag{6.75}$$

With these substitutions, 6.71 gives us

$$\alpha^2 + \tau\alpha + \frac{C^2}{h^2} \left[4 \sin^2 \left(\frac{\sigma h}{2} \right) \right] = 0 \tag{6.76}$$

and the solution to this quadratic equation gives the dispersion relation:

$$\alpha = -\frac{\tau}{2} \pm j\sigma \sqrt{C^2 \left[\frac{\sin \sigma h/2}{\sigma h/2} \right]^2 - \left(\frac{\tau}{2\sigma} \right)^2} \tag{6.77}$$

There is a familiar discretization factor here, $\left[\frac{\sin \sigma h/2}{\sigma h/2} \right]^2$, which ranges from roughly 0.4 at the Nyquist point, to unity at high resolution. As $\frac{\sigma h}{2} \to 0$, α becomes perfect. For finite σh, we incur a discretization error but still retain qualitative fidelity. The threshold for critical damping (no propagation) is shifted from its analytic counterpart:

$$\frac{\tau}{2} > C\sigma \left[\frac{\sin \sigma h/2}{\sigma h/2} \right] \tag{6.78}$$

Higher τ results in pure decay; lower τ results in wave propagation. Critical damping will set in at lower values of τ if the discretization is coarse. For propagating modes, we have *perfect* amplitude decay for this system; the discretization error occurs in the phase of the solution relative to the analytic. In the worst case of the Nyquist modes, $\sigma h = \pi$, we have both propagation and damping by Lumped System # 1. Significant errors will occur in the phasing of these modes; but they will propagate with finite speed; there is no opportunity for the least-resolved modes to become trapped at their source, unless due to critical damping, in which case their pure decay is qualitatively right.

Lumped System # 2

This System is based directly on the Primitive Pair:

$$\begin{bmatrix} \frac{\partial}{\partial t} + \tau & -C_1 \frac{\delta}{2h} \\ -C_2 \frac{\delta}{2h} & \frac{\partial}{\partial t} \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6.79}$$

As above, we seek solutions of the form

$$U = e^{\alpha t} e^{j\sigma x} \tag{6.80}$$

The first difference operator becomes

$$\frac{\delta U_i}{2h} = \frac{U_{i+1} - U_{i-1}}{2h} = \frac{U_i e^{j\sigma h} - U_i e^{-j\sigma h}}{2h} = j \frac{\sin \sigma h}{h} U_i \tag{6.81}$$

and the lumped Primitive Pair becomes

$$\begin{bmatrix} \alpha+\tau & -jC_1\frac{\sin\sigma h}{h} \\ -jC_2\frac{\sin\sigma h}{h} & \alpha \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6.82}$$

For nontrivial solutions, we need the determinant to vanish:

$$\alpha^2 + \tau\alpha + C^2\left[\frac{\sin\sigma h}{h}\right]^2 = 0 \tag{6.83}$$

and the roots are

$$\alpha = -\frac{\tau}{2} \pm j\sigma\sqrt{C^2\left[\frac{\sin\sigma h}{\sigma h}\right]^2 - \left(\frac{\tau}{2\sigma}\right)^2} \tag{6.84}$$

This is remarkably similar to 6.77 for LS1; the only difference is the discretization factor, $\left[\frac{\sin\sigma h}{\sigma h}\right]^2$ here for LS3, $\left[\frac{\sin\sigma h/2}{\sigma h/2}\right]^2$ for LS1. Both of these factors approximate unity, and are perfect for high resolution. But the discrepancy grows as we approach the Nyquist point, $\sigma h \to \pi$. While LS1 maintained at least qualitative fidelity there, LS2 loses it; $\left[\frac{\sin\sigma h}{\sigma h}\right]^2 \to 0$ and we have two real roots

$$\alpha = -\frac{\tau}{2} \pm \frac{\tau}{2} \tag{6.85}$$

Both Nyquist solutions fail to propagate. One decays at the rate τ; the other does not decay at all! What is happening is, these solutions are indistinguishable numerically from ones with infinite wavelength, the exact opposite of the truth! The centered first derivative detects no slope in this solution. If we go back to the matrix equation 6.82, we can see this clearly for the Nyquist mode:

$$\begin{bmatrix} \alpha+\tau & 0 \\ 0 & \alpha \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6.86}$$

The δ operator produces nothing! And the two solutions are uncoupled:

- One solution is $(\alpha+\tau)U_i = 0$. U decays with $\alpha = -\tau$, independent of V.
- The other is $\alpha V_i = 0$. V persists forever with $\alpha = 0$, independent of U.

This is a serious qualitative flaw in LS2 which has no resemblance at all to the continuum. It arises in the first centered difference being used twice, and the related $4h$ footprint of this system. LS1 uses the second centered difference, with footprint $2h$, and avoids this complication. This qualitative flaw in LS2 affects the complete spectrum between $2h \le \lambda \le 4h \quad (\pi \ge \sigma h \ge \pi/2)$.

For well-resolved spatial modes, LS2 has good fidelity to the continuum both qualitatively and quantitatively. For solutions which propagate, ($j\sigma\sqrt{positive}$), the numerical damping is perfect, while the wave speed suffers some discretization error – as in LS1 but different in detail.

Lumped System # 3

This qualitative infidelity in LS2 is absent in LS3. Recall that LS3 is the same as LS2 except that half of the equations and unknowns are removed; and the surviving variables are staggered with effective mesh spacing $k = 2h$ (Figure 6.9). Along with the elimination of odd (even) variables,

we eliminate the portion of the Fourier spectrum $\lambda < 4h$, which was the problem area above. Equivalently, we have $2k < \lambda < \infty$ for this system.

Since the equations are unchanged, we have the same dispersion relation as for LS2, equation 6.84. Re-expressing that in terms of the effective mesh spacing k, we have for LS3:

$$\alpha = -\frac{\tau}{2} \pm j\sigma\sqrt{C^2\left[\frac{\sin\sigma k/2}{\sigma k/2}\right]^2 - \left(\frac{\tau}{2\sigma}\right)^2} \tag{6.87}$$

This is the *identical* dispersion relation as for LS1 (telegraph equation), if we use the appropriate mesh spacing - the distance between solution points - in each case. (Compare with equation 6.77). So the Primitive Pair discretized on a staggered grid (LS3) has identical dispersion properties as the Telegraph Equation (LS1) at equivalent resolution. Both are free of parasitic, poorly-resolved modes which infect LS2.

Lumped System # 4

This is the LS where we center the molecules between the nodes of an unstaggered mesh as shown in Figure 6.10. On a mesh of N interior nodes, there are $2(N+2)$ nodal variables, $2(N+1)$ PDE's and 2 BC's. Proceeding as above we have

$$\begin{bmatrix} (\alpha+\tau)\dfrac{(e^{j\sigma h}+1)}{2} & -\dfrac{C_1(e^{j\sigma h}-1)}{h} \\ -\dfrac{C_2(e^{j\sigma h}-1)}{h} & \alpha\dfrac{(e^{j\sigma h}+1)}{2} \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6.88}$$

Setting the determinant of this to zero, and some trigonometric manipulation, gives us

$$\alpha(\alpha+\tau)\left[\cos^2\left(\frac{\sigma h}{2}\right)\right] + \frac{C^2}{h^2}\left[4\sin^2\left(\frac{\sigma h}{2}\right)\right] = 0 \tag{6.89}$$

The solution is

$$\alpha = -\frac{\tau}{2} \pm j\sigma\sqrt{C^2\left[\frac{\tan\sigma h/2}{\sigma h/2}\right]^2 - \left(\frac{\tau}{2\sigma}\right)^2} \tag{6.90}$$

Here we have introduced another discretization factor, $\left[\frac{\tan\sigma h/2}{\sigma h/2}\right]$. This approaches unity at high resolution, such that the Lumped System dispersion relation approaches perfection at that limit. As we approach the Nyquist point, $\tan\pi/2 \to \infty$ and the wavespeed increases without bound, while still maintaining the finite decay rate $\tau/2$. Poorly-resolved spatial modes propagate essentially instantaneously throughout the system once introduced.

Figure 6.11 (top) is a plot of the dispersion relation for the four Lumped Systems considered here, for the lossless case ($\tau = 0$). The quantities plotted are the dimensionless forms: $A = \frac{\alpha h}{jC}$ versus $S = \sigma h$. The full range of S is shown, $0 \le S \le \pi$ *i.e.* $\infty \ge \lambda \ge 2h$. For large S the spatial discretization is coarse; LS1 and LS3 do well qualitatively but their quantitative fidelity falls off. LS2 is "folded" back to zero at the Nyquist point $S = \pi$; waves at that limit neither propagate nor decay. LS4 is singular at $S = \pi$, indicating an unbounded propagation speed and an ambiguous fate for these waves, critically dependent on the boundary conditions.

All these Lumped Systems show good fidelity for well-resolved modes. A reasonable threshold for resolution is about 10 nodes per wavelength; $\frac{\lambda}{h} = 10$ corresponds to $S = .2\pi$ and Figure 6.11 (bottom) shows good quantitative and qualitative fidelity for all systems for $S \le .2\pi$.

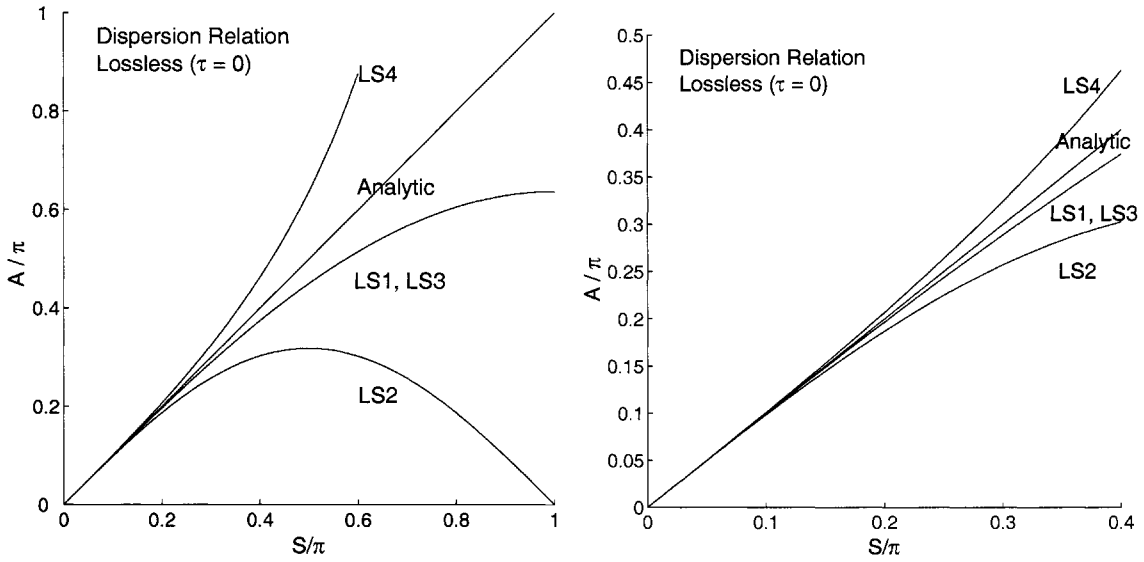


Figure 6.11: Dimensionless dispersion relation for continuous (analytic) and lumped systems. $A \equiv \frac{\alpha h}{jC}$; $S \equiv \sigma h$. Left: The full range of S is shown, from perfect resolution ($S = 0, \lambda = \infty$) to the Nyquist point ($S = \pi, \lambda = 2h$). Right: The well-resolved range of small S is highlighted. $S \leq .2\pi$ corresponds to 10 or more nodes per wavelength.

6.6 Time-Domain Simulation: Discrete Systems

Introduction

Finally we will discuss some common time-stepping simulation approaches. We will concentrate on Lumped System # 1 (telegraph equation) and #3 (primitive pair on a staggered mesh) in one Elliptic (spatial) dimension. LS2 is basically disqualified due to its support of parasitic modes. We will also look at LS4 – although it has no obvious counterpart beyond 1-D, it is useful in 1-D wave propagation studies.

We need to introduce the solution notation

$$U(x,t) \sim U_i^l \tag{6.91}$$

with superscript l indicating time levels separated by uniform increments Δt, and subscript i indicating position on a uniform x-grid with spacing h.

For analysis purposes, we extend that introduced for the dispersion studies above, in discrete form:

$$U_i^l = \gamma^l e^{j\sigma x} \tag{6.92}$$

The Fourier wavenumber σ is as defined above; γ is a discrete-time analog of α:

$$\gamma = e^{\alpha \Delta t} \tag{6.93}$$

We treated this decomposition before, in connection with Parabolic Systems. The analysis is the same here. For stability, we require

$$|\gamma| < 1 \tag{6.94}$$

for all feasible values of σ. Here however we will find that γ is, or at least should be, complex, so we will have to be a little more careful. For accuracy purposes, we define the Propagation Factor T:

$$T = \left(\frac{\gamma_d}{\gamma_c}\right)^N \tag{6.95}$$

where the subscripts refer to discrete and continuous systems. Essentially, we want γ_d to be as close as possible to its continuous-system counterpart γ_c, so a value $T = 1$ is perfection. However there is a complication – γ converges to unity anyway as Δt gets small. So we need to make the comparison at a fixed time in the future, based on a natural timescale t_o of the continuous system: $N\Delta t = t_o$. So N needs to grow as Δt shrinks.

From the continuous system,

$$\gamma_c^N = e^{\alpha t_o} = e^{\frac{-\tau t_o}{2}} e^{\pm j\sigma t_o \sqrt{C^2 - \left(\frac{\tau}{2\sigma}\right)^2}} \tag{6.96}$$

It is convenient to define t_o as the time it takes for the continuous system to propagate one wavelength:

$$\sigma t_o \sqrt{C^2 - \left(\frac{\tau}{2\sigma}\right)^2} = 2\pi \tag{6.97}$$

Substituting $N\Delta t = t_o$, and a little rearrangement, leads to

$$N = \frac{2\pi}{\sqrt{S^2\left(\frac{C\Delta t}{h}\right)^2 - \left(\frac{\tau\Delta t}{2}\right)^2}} \tag{6.98}$$

and we have three dimensionless numbers $\mathcal{K}$, S, and $\mathcal{T}$, in addition to N:

$$\mathcal{K} = C\frac{\Delta t}{h} \tag{6.99}$$

$$\mathcal{T} = \frac{\tau\Delta t}{2} \tag{6.100}$$

$$S = \sigma h \tag{6.101}$$

$$N = \frac{2\pi}{\sqrt{S^2\mathcal{K}^2 - \mathcal{T}^2}} \tag{6.102}$$

$\mathcal{K}$ is the *Courant Number*, the most important dimensionless number in hyperbolic systems.

Under these conditions the continuous system gives

$$\gamma_c^N = \left(e^{-\mathcal{T}}\right)^N \tag{6.103}$$

This is a real number; the analytic solution has decayed and propagated exactly one wavelength. The Propagation Factor is

$$T = \left(\frac{\gamma_d}{e^{-\mathcal{T}}}\right)^N \tag{6.104}$$

This is a complex number since the discrete system will have imperfect wave speed. $|T|$ gives the magnitude of the wave in the discrete system relative to that in the continuous system. $|T| = 1$ is

perfection. The argument of T (its angle in the complex plane), is the same as the argument of γ_c^N (Figure 6.12). Perfection for this metric is 2π. It is useful to observe that

$$\arg(T) = N \cdot \arg(\gamma_d) \tag{6.105}$$

$$|T| = \frac{|\gamma_d|^N}{e^{-NT}} \tag{6.106}$$

We summarize the central points:

- γ is the ratio of U_i after one time step, to its starting value
- $\gamma \leq 1$ is necessary for stability
- T is the ratio of discrete to continuous solution, starting from the same IC's, at a point in the future when a wave in the continuous system has propagated one wavelength.
- $|T|$ is the measure of amplitude fidelity. $|T| = 1$ is perfect.
- $\arg(T)$ is the measure of phase fidelity. $\arg(T) = 2\pi$ is perfect.

Below we will develop expressions for γ_d for some common discrete systems.

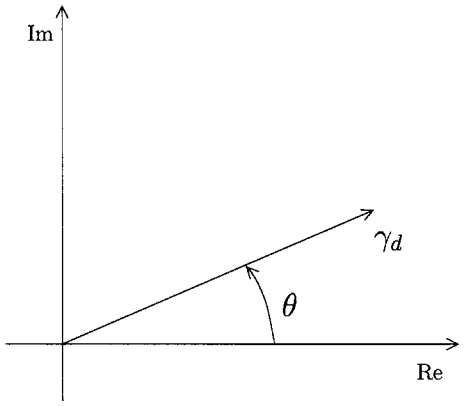


Figure 6.12: γ_d in the complex plane. $\phi = \arg \gamma_d$. Perfect phase occurs when $N\phi = 2\pi$.

Discrete System 1 (Telegraph Equation)

This is the most natural approach to the Telegraph Equation: building on LS1 which used the centered, second-order approximation δ_x^2 for the second derivative, do the same in the time domain. The Lumped System is

$$\frac{d^2U_i}{dt^2} + \tau\frac{dU_i}{dt} - \frac{C^2}{h^2}\delta_x^2 U_i = 0 \tag{6.107}$$

and the Discrete System is

$$(U^{l+1} - 2U^l + U^{l-1})_i + \frac{\tau\Delta t}{2}(U^{l+1} - U^{l-1})_i - \mathcal{K}^2\delta_x^2 U_i^l = 0 \tag{6.108}$$

The molecule is shown in Figure 6.13. Recall that $\mathcal{K} = C\frac{\Delta t}{h}$ is the Courant Number.

This system has several interesting properties

- it is centered in x and t, therefore second-order correct;

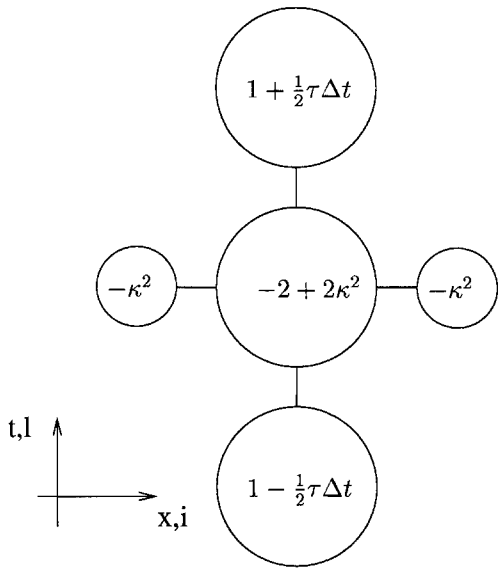


Figure 6.13: Discrete System # 1, explicit version.

- it is explicit-in-time, therefore no matrix factorization is needed;
- it is stable if $\mathcal{K}^2 < 1$, as a result of its explicitness;
- there are no parasites: LS1 was immune, and this adds none.

Particularly interesting here is the stability condition. This molecule "looks like" the standard Laplacian in 2-D, the workhorse of Elliptic systems. We found above that the equivalent molecule for Parabolic Systems was unconditionally unstable! In the Hyperbolic Case, we have conditional stability; essentially the grid $(h, \Delta t)$ must be such that the numerical solution does not build from IC's faster that the continuum would have (see Figure 6.1). That is one interpretation of the Courant Number.

We will explore some of these properties below. But first there is an interesting implicit generalization of this system:

$$(U^{l+1} - 2U^l + U^{l-1})_i + \frac{\tau\Delta t}{2}(U^{l+1} - U^{l-1})_i - \mathcal{K}^2\delta_x^2\left[\frac{\theta}{2}(U^{l+1} + U^{l-1}) + (1-\theta)U^l\right]_i = 0 \quad (6.109)$$

Here we have simply moved a fraction of the Laplacian (δ_x^2) term to the new level, balanced with an equal fraction at the old level. The implicitness is controlled by the new parameter θ; the system is centered-in time regardless of the value of θ. The molecule is illustrated in Figure 6.14. It reduces to the explicit version just described when $\theta = 0$.

This implicit system has the following properties:

- it is centered in x and t, therefore second-order correct;
- $\theta = 0$ reduces to the explicit system presented above;
- $\theta > 0$ is implicit; a matrix factorization involving the unknowns at time $l+1$ is required to get ahead
- no parasites are supported;
- stability is *unconditional* if $\theta > 1/2$, *i.e.* no value of Δt can destabilize the calculations. Relative to the explicit version, the δ_x^2 operator at level $l+1$ is responsible for the stability.
- for $\theta < 1/2$, we have conditional stability when $\mathcal{K}^2 < \frac{1}{1-2\theta}$.

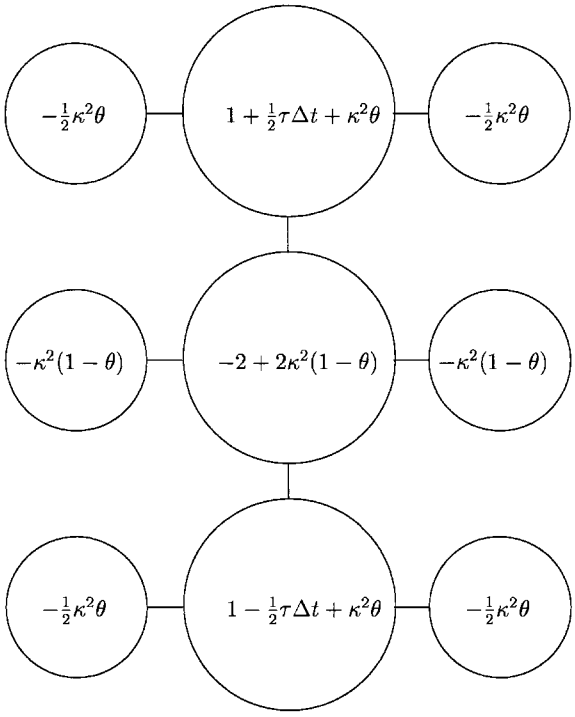


Figure 6.14: Discrete System # 1, implicit version.

We can derive some of these properties by looking at the Fourier analysis. For the general implicit scheme, we have the substitutions

$$U_i^{l+1} = \gamma U_i^l \tag{6.110}$$

$$\delta_x^2 U_i^l = -4\left[\sin\left(\frac{S}{2}\right)\right]^2 U_i^l \tag{6.111}$$

with $0 \leq S \leq \pi$. Direct application to equation 6.109 gives us the quadratic equation in γ:

$$\left(\gamma^2 - 2\gamma + 1\right) + \mathcal{T}\left(\gamma^2 - 1\right) + 4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2\left[\frac{\theta}{2}(\gamma^2+1) + (1-\theta)\gamma\right] = 0 \tag{6.112}$$

$$\begin{aligned}\gamma^2\left[1+\mathcal{T}+4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2\frac{\theta}{2}\right] &+ \gamma\left[-2 \quad +4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2(1-\theta)\right] \\ &+ \left[1-\mathcal{T}+4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2\frac{\theta}{2}\right] = 0\end{aligned} \tag{6.113}$$

Now elsewhere (see Appendix) we show that the roots of the quadratic equation

$$ax^2 + bx + c = 0 \tag{6.114}$$

are stable ($|x| \leq 1$ for each root) provided

$$\frac{c}{a} \leq 1 \tag{6.115}$$

$$|b| \leq a + c \tag{6.116}$$

(These rules require a, b, c real and $a > 0$.) Applying these, we find that this scheme is *stable* when

$$\mathcal{K}^2(1-2\theta)\left[\sin\left(\frac{S}{2}\right)\right]^2 \leq 1 \tag{6.117}$$

This is always true for $\theta \geq 1/2$; so that guarantees unconditional stability. For $\theta < 1/2$, the worst case is at the Nyquist point, $S = \pi$. In that case, we have *Courant Number-dependent stability*:

$$\mathcal{K}^2 \leq \frac{1}{(1-2\theta)} \tag{6.118}$$

For *accuracy* studies, the propagation factor is based on γ, which is obtained by solution of 6.113. We leave that to the student as an exercise. For the explicit case, $\theta = 0$, we have:

$$\gamma = \frac{(1-Q/2) \pm j\sqrt{(1-\mathcal{T})(1+\mathcal{T}) - (1-Q/2)^2}}{(1+\mathcal{T})} \tag{6.119}$$

with Q defined for convenience:

$$Q = 4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2 \tag{6.120}$$

Propagating waves are characterized as having the quantity under the radical positive; otherwise, we have critical damping. It is easy to confirm that, for propagating waves and $\theta = 0$,

$$|\gamma|^2 = \frac{1-\mathcal{T}}{1+\mathcal{T}} \tag{6.121}$$

independently of spatial resolution. The wavenumber independence is a property of the continuous system. This is a second-order correct approximation to the continuum version $e^{2\alpha\Delta t}$. For the lossless case, $|\gamma| = 1$ is perfect. All error is vested in the phasing in that case.

More generally, for $\theta \neq 0$, we have

$$|\gamma|^2 = \frac{\left[1-\mathcal{T}+4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2\frac{\theta}{2}\right]}{\left[1+\mathcal{T}+4\mathcal{K}^2\left[\sin\left(\frac{S}{2}\right)\right]^2\frac{\theta}{2}\right]} \tag{6.122}$$

There are many more interesting properties of γ and T to be explored here, including the phase errors and the onset of critical damping. We leave these to the reader as exercises.

Discrete Systems 3: Coupled 1^{st} Order Equations

This System is based on the primitive pair, a staggered spatial mesh:

$$\frac{dU_i}{dt} + \tau U_i - C_1\frac{\delta V_i}{2h} = 0 \tag{6.123}$$

$$\frac{dV_i}{dt} - C_2\frac{\delta U_i}{2h} = 0 \tag{6.124}$$

We invoke some new dimensionless number definitions:

$$\mathcal{K}_1 = \frac{C_1\Delta t}{h} \tag{6.125}$$

$$\mathcal{K}_2 = \frac{C_2\Delta t}{h} \tag{6.126}$$

in addition to the previous ones

$$\mathcal{K}^2 = \mathcal{K}_1\mathcal{K}_2 \tag{6.127}$$

$$\mathcal{K} = \frac{C\Delta t}{h} \tag{6.128}$$

$$\mathcal{T} = \frac{\tau\Delta t}{2} \tag{6.129}$$

and the Fourier substitutions

$$\frac{\delta U_i}{2h} = \frac{(U_{i+1} - U_{i+1})}{2h} = \frac{j}{h}\sin\sigma h \tag{6.130}$$

$$S = \sigma h \tag{6.131}$$

$$U_i^{l+1} = \gamma U_i^l \tag{6.132}$$

In all these *staggered-mesh systems*, the Nyquist point is at $S = \pi/2$, since the *effective mesh spacing is* $2h$.

a) Euler Explicit System. Simple Euler time-stepping gives us

$$U_i^{l+1} - U_i^l + \tau\Delta t U_i^l - \frac{C_1\Delta t}{2h}(V_{i+1} - V_{i-1})^l = 0 \tag{6.133}$$

$$V_i^{l+1} - V_i^l - \frac{C_2\Delta t}{2h}(U_{i+1} - U_{i-1})^l = 0 \tag{6.134}$$

The molecules for this system are illustrated in Figure 6.15. It is understood that the mesh spacing $2h$ separates adjacent U solutions, etc. Assuming Fourier spatial modes, we have the matrix equation

$$\begin{bmatrix} \gamma - 1 + \tau\Delta t & -j\mathcal{K}_1 \sin S \\ -j\mathcal{K}_2 \sin S & \gamma - 1 \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = 0 \tag{6.135}$$

Setting the determinant to zero we obtain the quadratic equation

$$(\gamma - 1 + \tau\Delta t)(\gamma - 1) + \mathcal{K}^2 \sin^2 S = 0 \tag{6.136}$$

$$\gamma^2 + \gamma[2\mathcal{T} - 2] + \left[1 - 2\mathcal{T} + \mathcal{K}^2 \sin^2 S\right] = 0 \tag{6.137}$$

The stability requirements ($|\gamma| \leq 1$) are

$$\frac{\mathcal{K}^2}{2} \leq \mathcal{T} \tag{6.138}$$

and

$$\mathcal{T} \leq 1 \tag{6.139}$$

These are severe. In the lossless case, we have *unconditional instability*. For many practical settings, this is a useless system. If we make the loss term τU implicit – *i.e.* move it forward to time level $(l+1)$ – the stability constraint $\mathcal{T} \leq 1$ is removed, leaving only $\frac{\mathcal{K}^2}{2} \leq \mathcal{T}$. This does not change the basic assessment here. The reader is encouraged to confirm this.

Figure 6.16 illustrates the *system of assembled molecules* for DS3a, emphasizing the grid staggering and its relation to the molecule assembly.

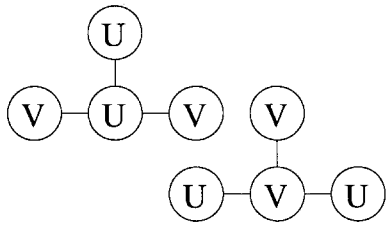


Figure 6.15: Two separate molecules for Euler discretization of the primitive pair on a staggered grid. This is Discrete System 3a. The x-axis is horizontal; the t-axis is vertical.

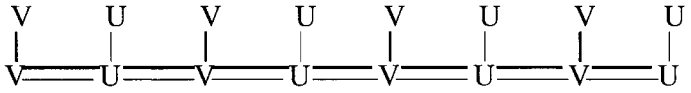


Figure 6.16: DS3a: Explicit time-stepping for LS3 as in Figure 6.15. This system is unconditionally unstable for the lossless case. Here and in the following Figures the boldface lines indicate the molecules for dV/dt; and the t-axis is vertical.

b) Explicit/Implicit Euler System. In this System, we solve explicitly for U as above. But with U^{l+1} in hand, we solve for the new V implicitly, by backward differencing of the time derivative:

$$U_i^{l+1} - U_i^l \quad +\tau\Delta t U_i^l \quad -\frac{C_1\Delta t}{2h}(V_{i+1} - V_{i-1})^l = 0 \tag{6.140}$$

$$V_i^{l+1} - V_i^l \qquad\qquad -\frac{C_2\Delta t}{2h}(U_{i+1} - U_{i-1})^{l+1} = 0 \tag{6.141}$$

This is illustrated in Figures 6.17 and 6.18. Despite the implicit molecule, the two parts are solved sequentially; no matrices need factorization in this algorithm.

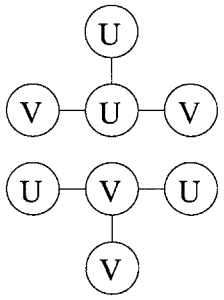


Figure 6.17: Two separate molecules for Euler Explicit/Implicit System on a staggered grid. The molecule for dU/dt is unchanged from DS3a. The dV/dt term is Backward "Euler". This is Discrete System 3b. Sequential solution requires no matrix factorization.

The matrix representation for Fourier modes is

$$\begin{bmatrix} \gamma - 1 + \tau\Delta t & -j\mathcal{K}_1 \sin S \\ -j\gamma\mathcal{K}_2 \sin S & \gamma - 1 \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = 0 \tag{6.142}$$

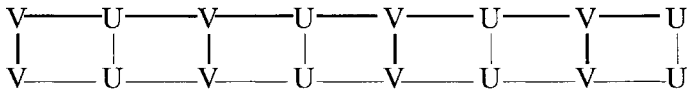


Figure 6.18: DS3b: Explicit-Implicit time-stepping for LS3 as in Figure 6.17. These molecules are explicit in U, implicit in V. Sequential solution for U first, then V, enables a complete timestep with no matrix factorization .

Only a single term has changed in this matrix – the insertion of γ in the lower left entry, reflecting its shift forward in time. The quadratic equation expressing $Det[\ ] = 0$ is:

$$\gamma^2 + \gamma\left[2\mathcal{T} - 2 + \mathcal{K}^2 \sin^2 S\right] + [1 - 2\mathcal{T}] = 0 \tag{6.143}$$

and this will be stable provided that

$$\mathcal{K}^2 \leq 4(1 - \mathcal{T}) \tag{6.144}$$

Now this is a workable system; for the lossless case, the Courant number limitation resembles a purely explicit system, which this in effect is.

Without compromising the explicitness of this system, we can move the loss term τU_i forward in time. If this term is advanced to time level $(l+1)$, the stability is improved a little: $\mathcal{K}^2 \leq 4(1+\mathcal{T})$. If we "center" the loss term:

$$U_i^{l+1} - U_i^l + \frac{\tau\Delta t}{2}(U_i^{l+1} + U_i^l) - \frac{C_1\Delta t}{2h}(V_{i+1} - V_{i-1})^l = 0 \tag{6.145}$$

and leave the dV/dt equation unchanged, the matrix becomes

$$\begin{bmatrix} \gamma - 1 + \frac{\tau\Delta t}{2}(\gamma + 1) & -j\mathcal{K}_1 \sin S \\ -j\gamma\mathcal{K}_2 \sin S & \gamma - 1 \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = 0 \tag{6.146}$$

The quadratic equation for this determinant's vanishing is

$$\gamma^2 [1 + \mathcal{T}] + \gamma\left[-2 + \mathcal{K}^2 \sin^2 S\right] + [1 - \mathcal{T}] = 0 \tag{6.147}$$

Now this is a remarkable result. Refer to equation 6.113, the comparable quadratic governing the explicit form of Discrete System 1 (Telegraph Equation with $\theta = 0$). *The present result for DS3b is identical* – if one adjusts for the fact that in DS3b we have the effective mesh spacing $2h$. Evidently the propagation properties of both systems are the same! Since DS1 is second-order in time, so must be DS3b. But DS3b used Euler time-derivatives; so evidently the forward/backward differencing causes the $O(\Delta t)$ truncation errors to cancel. In addition to this accuracy boost, we also get a workable stability condition requiring Courant Number of order unity.

c) Split-Time System

This system descends from DS3b. By staggering U and V in time as well as space, we obtain the arrangement as shown in Figure 6.19. We define one set of variables, say V, at the half-levels $l - \frac{1}{2}, l + \frac{1}{2}, l + \frac{3}{2}, \ldots$, then we have the equations

$$U_i^{l+1} - U_i^l + \tfrac{\tau\Delta t}{2}(U_i^{l+1} + U_i^l) - \frac{C_1\Delta t}{2h}(V_{i+1} - V_{i-1})^{l+\frac{1}{2}} = 0 \tag{6.148}$$

$$V_i^{l+\frac{3}{2}} - V_i^{l+\frac{1}{2}} - \frac{C_2\Delta t}{2h}(U_{i+1} - U_{i-1})^{l+1} = 0 \tag{6.149}$$

It is evident that this is just a shift in nomenclature from DS3b. Adopting the Fourier convention

$$V^{l+\frac{1}{2}} = \gamma V^{l-\frac{1}{2}} \tag{6.150}$$

we have the matrix representation

$$\begin{bmatrix} \gamma - 1 + \frac{\tau\Delta t}{2}(\gamma+1) & -j\mathcal{K}_1 \sin S \\ -j\gamma\mathcal{K}_2 \sin S & \gamma - 1 \end{bmatrix} \begin{Bmatrix} U_i^l \\ V_i^{l+\frac{1}{2}} \end{Bmatrix} = 0 \tag{6.151}$$

This is identical to DS3b, with a simple notational shift. The split-step system is clearly centered-in-time, with all first derivatives explicit – that illumines the quandary above about second-order correctness for Euler timestepping. By reference the propagation behaviour of DS3c is identical to the explicit form of DS1 (telegraph equation), also. This system can alternately be viewed as a centered, explicit treatment of both parts of the primitive pair, on a time- and space- staggered grid.

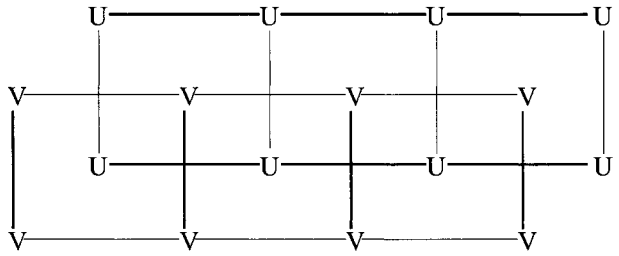


Figure 6.19: DS3c: Time-splitting discretization of LS3. U and V are defined at alternate levels in time. This system is obtained a) from the Explicit/Implicit system by shifting the dV/dt molecules and the V solution by $\Delta t/2$; or b) as a centered system of first derivatives, starting with the time-staggered mesh. The separation between U solutions is Δt. The system is completely centered and explicit.

There is one difficulty with either of these views of DS3b – they are not self-starting. The time-split grid requires U and V at separate points in time, separated by $\frac{\Delta t}{2}$; the proper IC's are U and V at the same point in time. This is an invitation for a parasite to arise in the IC implementation. (There is a $O(\Delta t)$ initialization of an $O(\Delta t^2)$ system.) The dilemma does not arise when the system is viewed as DS3b.

d) Leapfrog

In the same line of thought, we can define a centered "leapfrog" time-stepping for the primitive pair *without* staggering in the time-domain. The discrete system is

$$U_i^{l+1} - U_i^{l-1} + \tau\Delta t(U_i^{l+1} + U_i^{l-1}) - \frac{2C_1\Delta t}{2h}(V_{i+1} - V_{i-1})^l = 0 \tag{6.152}$$

$$V_i^{l+1} - V_i^{l-1} - \frac{2C_2\Delta t}{2h}(U_{i+1} - U_{i-1})^l = 0 \tag{6.153}$$

This is totally centered in time and space. The molecules are illustrated in Figure 6.20. Notice that the dU/dt molecules "leap over" the U variables defined at their centers; and likewise for the dV/dt molecules. Time-splitting simply eliminates these unused variables.

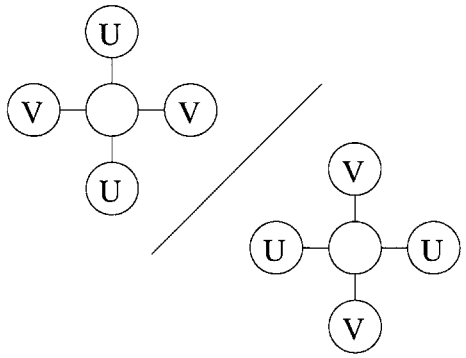


Figure 6.20: Leapfrog molecules. The mesh is staggered-in-space with effective spacing $2h$; but not in time. The molecules ignore the variables at their centers. As a result there is a parasitic mode representing odd-even uncoupling.

The Fourier matrix is

$$\begin{bmatrix} \gamma^2 - 1 + \tau\Delta t(\gamma^2+1) & -2j\gamma\mathcal{K}_1 \sin S \\ -2j\gamma\mathcal{K}_2 \sin S & \gamma^2 - 1 \end{bmatrix} \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = 0 \tag{6.154}$$

and the characteristic equation is

$$\gamma^4 [1 + 2\mathcal{T}] + \gamma^2 \left[-2 + 4\mathcal{K}^2 \sin^2 S\right] + [1 - 2\mathcal{T}] = 0 \tag{6.155}$$

If we concentrate on γ^2, then we find that this is the identical quadratic equation characterizing the split-step system, allowing for the change in effective timestep $2\Delta t$ here. Even-numbered solutions will have the same dynamics as DS3c, and by extension to DS3b and DS1 (explicit). We keep getting this result.

However, here we have an additional problem: even-odd *uncoupling*. There are 2 sets of solutions which are completely unrelated. While the root γ^2 is at least qualitatively faithful to the continuum, the roots $\gamma = \pm\gamma^2$ are problematic – one is faithful, the other is a parasite describing the odd/even relationship. The related solution mode is initiated in the IC's (this molecule requires 2 time levels of both U and V); and fed by imprecision along the way. DS3c dealt with this problem by elimination: the time-splitting eliminated either the odd or the even solutions. Given that this system doubles the number of unknowns, does not resolve the need for extra IC's, and adds nothing in terms of accuracy or qualitative fidelity, there is little to recommend it relative to DS3c.

It is interesting to consider moving the loss term τU around in time. In particular, centering it entirely leads to unconditional instability! (The reader is encouraged to develop this result.)

From Lumped System 3, we arrive at two useful time-stepping algorithms which are second-order correct: DS3b and DS3c. Although they initially look different, they are in fact the same. They both involve staggering the mesh in space and time. Their relation to the leapfrog system is illustrated in Figure 6.21.

The space-staggering eliminates problematic modes by effectively making the Nyquist cutoff at $\lambda = 4h$. The time-splitting eliminates parasitic odd-even decoupling. (These are manifestations of the same feature in x and t.) DS3c clarifies the second-order accuracy, but invites IC problems which do not arise with DS3b. Stability is conditional, with a reasonable explicit limit on $\mathcal{K}$. Extensions to 2 and 3 elliptic dimensions are reasonable and generally preserve these qualities. Accuracy for DS3b/c is the same as the explicit form of DS1.

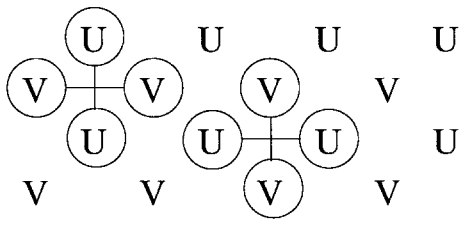


Figure 6.21: By removing the degrees of freedom at the centers of the Leapfrog molecules, we obtain a time-split system, identical to DS3c.

Discrete System 4: Implicit Four-Point Primitive

Lumped System 4 is based on a non-staggered spatial mesh, with the PDE approximants centered *between* the nodes:

$$\frac{1}{2}\left(\frac{dU_i}{dt} + \frac{dU_{i+1}}{dt}\right) + \frac{\tau}{2}(U_i + U_{i+1}) - \frac{C_1}{h}(V_{i+1} - V_i) = 0 \tag{6.156}$$

$$\frac{1}{2}\left(\frac{dV_i}{dt} + \frac{dV_{i+1}}{dt}\right) - \frac{C_2}{h}(U_{i+1} - U_i) = 0 \tag{6.157}$$

This admits a very compact, implicit time discretization involving 2 time levels and centered somewhere between them:

$$\frac{dU_i}{dt} \to \frac{U_i^{l+1} - U_i^l}{\Delta t} \qquad V_i \to \theta V_i^{l+1} + (1-\theta)V_i^l \tag{6.158}$$

$$\frac{dV_i}{dt} \to \frac{V_i^{l+1} - V_i^l}{\Delta t} \qquad U_i \to \theta U_i^{l+1} + (1-\theta)U_i^l \tag{6.159}$$

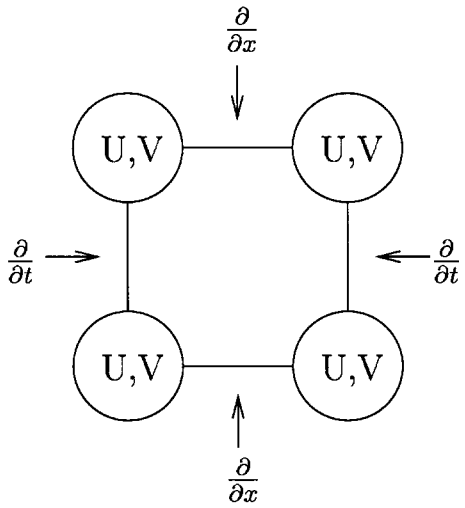


Figure 6.22: DS4. A compact implicit 4-point molecule with no timesplitting and no spatial staggering. The centered ($\theta = .5$) version is shown.

The molecule for this system is illustrated in Figure 6.22. When $\theta = .5$, it is second-order correct. The reader is encouraged to develop the analysis of this molecule. Interestingly, this value walks

the line with respect to stability: unconditional stability when $\theta > 1/2$; otherwise, unconditional instability. As an instance of LS4, it also does a reasonable qualitative job with poorly resolved spatial modes as described in the previous section. Because of its compact nature, there is no problem with parasitic temporal modes.

6.7 Lumped Systems in Higher Dimensions

The extension of the hyperbolic systems to two or more elliptic dimensions invites considerable complexity, originating in the various physical applications themselves. In particular, some of the dependent variables become vectors or tensors, while some remain scalars. This complexity is masked in 1-D where the unifying structure is most obvious, and where many of the essential ideas about numerical implementation are exposed.

We will look here at the 2-D system comprising the scalar U and the vector $\mathbf{V} = (V_x, V_y)$:

$$\frac{\partial U}{\partial t} - C_2 \left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} \right) \tag{6.160}$$

$$\frac{\partial V_x}{\partial t} + \tau V_x - C_1 \frac{\partial U}{\partial x} = 0 \tag{6.161}$$

$$\frac{\partial V_y}{\partial t} + \tau V_y - C_1 \frac{\partial U}{\partial y} = 0 \tag{6.162}$$

or equivalently, in more compact form,

$$\frac{\partial U}{\partial t} - C_2 \nabla \cdot \mathbf{V} = 0 \tag{6.163}$$

$$\frac{\partial \mathbf{V}}{\partial t} + \tau \mathbf{V} - C_1 \nabla U = 0 \tag{6.164}$$

This is the primitive pair. The 2-D telegraph equations for U and for $\mathbf{V}$ are obtained by the usual operations. For the case where C_1 and C_2 are constants, we have

$$\frac{\partial^2 U}{\partial t^2} + \tau \frac{\partial U}{\partial t} - C^2 \nabla^2 U = 0 \tag{6.165}$$

$$\frac{\partial^2 \mathbf{V}}{\partial t^2} + \tau \frac{\partial \mathbf{V}}{\partial t} - C^2 \nabla^2 \mathbf{V} = 0 \tag{6.166}$$

The Hyperbolic System as posed here favors the Fluid Mechanics instance, by associating the vector $\mathbf{V}$ with fluid velocity and the scalar U with pressure. There is a dual statement of the problem in Electricity and Magnetism. For the plane case, we have the scalar H_z representing the normal (z-directed) magnetic field, and the vector $\mathbf{E}$ representing the in-plane electric field (E_x, E_y). The primitive Maxwell System is

$$\frac{\partial H_z}{\partial t} - \frac{1}{\mu} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right) = 0 \tag{6.167}$$

$$\frac{\partial E_x}{\partial t} + \frac{\sigma}{\epsilon} E_x - \frac{1}{\epsilon} \frac{\partial H_z}{\partial y} = 0 \tag{6.168}$$

$$\frac{\partial E_y}{\partial t} + \frac{\sigma}{\epsilon} E_y + \frac{1}{\epsilon} \frac{\partial H_z}{\partial x} = 0 \tag{6.169}$$

So if we associate H_z with U, and $(-E_y, E_x)$ with (V_x, V_y) the 2-D Maxwell system is identical to 6.160-6.166 above. We will use equations 6.160-6.166 in the present exposition.

(In 3-D the Maxwell System requires vector fields for both **H** and **E**; the primitive pair is

$$\frac{\partial \mathbf{H}}{\partial t} + \frac{1}{\mu} \nabla \times \mathbf{E} = 0 \tag{6.170}$$

$$\frac{\partial \mathbf{E}}{\partial t} + \frac{\sigma}{\epsilon} \mathbf{E} - \frac{1}{\epsilon} \nabla \times \mathbf{H} = 0 \tag{6.171}$$

and there is a general vector telegraph equation for both fields. The vector identity $\nabla \times (\nabla \times \mathbf{A}) \equiv \nabla (\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}$ is useful in processing these equations.)

Lumped System # 1

This system is based on straightforward implementation of the telegraph equation in 2-D:

$$\frac{d^2 U_i}{dt^2} + \tau \frac{dU_i}{dt} - \frac{C^2}{h^2} (\delta_x^2 + \delta_y^2) U_i = 0 \tag{6.172}$$

This relies on the conventional strength of the 5-point numerical Laplacian, illustrated in Figure 6.23. It collapses to the 1-D Lumped System studied above, for 1-D situations. There are no special considerations in 2-D which are not exposed in 1-D. This system works well, either in Harmonic form (*not* diagonally dominant), or in time-stepping form, in which case there are stability considerations (discussed below).

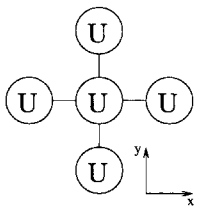


Figure 6.23: Lumped System # 1 in 2-D.

Lumped System # 2

This system is based on the primitive pair, with centered first differences replacing the first derivatives in x and y:

$$\frac{dU_i}{dt} - \frac{C_2}{2h} (\delta_x V_{xi} + \delta_y V_{yi}) = 0 \tag{6.173}$$

$$\frac{dV_{xi}}{dt} + \tau V_{xi} - \frac{C_1}{2h} (\delta_x U_i) = 0 \tag{6.174}$$

$$\frac{dV_{yi}}{dt} + \tau V_{yi} - \frac{C_1}{2h} (\delta_y U_i) = 0 \tag{6.175}$$

This is the 1-D Lumped System # 2 with straightforward extension to include y-dependence. Its implementation on a non-staggered grid is illustrated in Figure 6.24. This system suffers from the short-wave problems identified in 1-D; they are present in both spatial dimensions here.

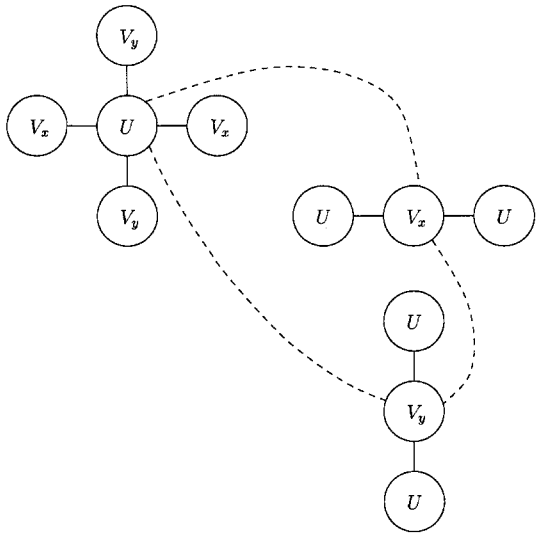


Figure 6.24: Lumped System # 2. The three separate molecules are centered-in-space; all three variables are defined at all points. The central entry in each molecule occupies the identical position in the grid, as indicated by the dotted lines.

Lumped System # 3C

Simply by shifting the molecules of LS2 in Figure 6.24, we can obtain LS3. The equations are identical, with the understanding that the variables are only defined at alternating points as in 1-D. This is depicted in Figure 6.25. This molecule is parasite-free and always favoved over LS2, for reasons which were already exposed in the 1-D analysis. It was (re)invented indendently Platzman [97], Arakawa [4, 5] and Leendertse [53] in Geophysical Fluid Dynamics (Arakawa's "C" grid), and by Yee [116] in Electromagnetism. It remains in popular use in both fields.

Lumped System # 4

This System was useful in 1-D; however there is no direct analogy in 2-D. Its generalizations produce an unequal number of equations and unknowns, making it unworkable.

Arakawa Systems

Starting from LS2, there are many ways to introduce grid staggering in 2-D. These were studied and classified by Arakawa [4, 5] and are illustrated in Figure 6.26. The Arakawa "A" scheme is non-staggered, and equivalent to LS2 herein. The "C" scheme was introduced above as LS3; the "B" and "E" schemes are also presented in Figure 6.26; they have the property that both vector components (V_x, V_y) are defined at the same points, but staggered relative to the scalar U. There are applications where this becomes important in meteorology and oceanography.

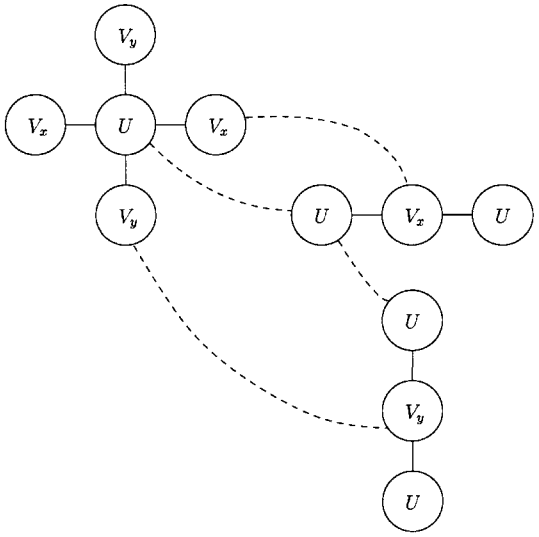


Figure 6.25: Lumped System # 3C. Staggered-grid version of Lumped System # 2. The three separate molecules are centered-in-space in an overlapping pattern as indicated by the dashed lines. Only one variable is defined at any one point of the grid. This is the Arakawa "C" pattern in Figure 6.26.

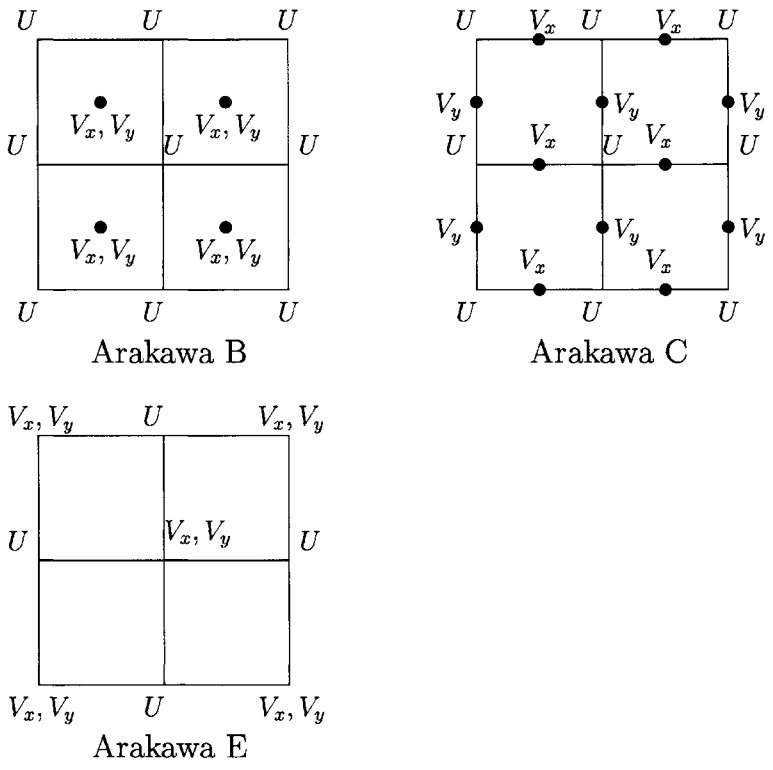


Figure 6.26: Lumped System # 3: various staggered-mesh arrangements in 2-D. The classification is from Arakawa [4, 5]. The "C" system is identical to LS3 presented in Figure 6.25.

Part II

The Finite Element Method

Chapter 7

General Principles

Fundamental to the FEM is the notion of determining an approximate solution $\hat{u}$ which comes close to the unknown true solution u for a given problem. From the outset we require that the function $\hat{u}$ will be defined everywhere in terms of a finite set of mathematical *basis functions* $\phi_j(\mathbf{x})$ whose properties are *a priori* well-known:

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{N} u_j \phi_j(\mathbf{x}) \tag{7.1}$$

The coefficients u_j are the primary unknowns of any problem once the basis has been selected. Examples of sets of basis functions $\phi_j(\mathbf{x})$ abound in mathematical physics and numerical analysis: Fourier, Chebyshev, Bessel, and the other special functions; and various polynomial bases including Taylor and Lagrange polynomials, splines, etc. In any practical problem, the basis in use will necessarily be *finite* and *incomplete* – *i.e.* incapable except in lucky cases of representing the exact solution perfectly. Any numerical solution may be viewed, then, as a 2-step process. First, select a basis which is likely to fit the unknown solution (*i.e.* contain its major features) for the particular problem. And second, determine the coefficients u_j in a reliable way.

We will discuss the latter aspect first, in the context of well-known bases. Then we look at the bases commonly used in the FEM.

7.1 The Method of Weighted Residuals

Use of a finite or incomplete basis guarantees that in general a given differential equation cannot be satisfied everywhere, leaving an imbalance or *residual* everywhere. For example, consider the Helmholtz equation

$$\nabla^2 u + fu = g \tag{7.2}$$

with f and g known. By definition, the unknown function u satisfies this equation everywhere. However approximating u in a finite basis as $\hat{u}$, we define the residual R

$$R(\mathbf{x}) \equiv \nabla^2 \hat{u} + f\hat{u} - g \tag{7.3}$$

which will generally be nonzero. Clearly, R depends on the selection of both the basis ϕ_i and the coefficients u_j. Enlightened, problem-dependent basis selection is the first step towards a small

residual. Then, given a finite basis, one must concentrate on making R small in some average way by choosing the coefficients u_j.

One way to formalize this idea of determining u_j is the Method of Weighted Residuals (MWR) – in which R is required to vanish in a weighted integral sense. In Cartesian space we have

$$\int\int\int R W_i dxdydz = 0 \tag{7.4}$$

for a set of distinct weighting functions $W_i(\mathbf{x})$, $i = 1, N$, and the integration performed over the full domain in which the differential equation governs. Throughout we will use the inner product notation $< , >$ to indicate domain (volume) integration:

$$\langle \mathbf{a}, \mathbf{b} \rangle \equiv \int\int\int \mathbf{a} \cdot \mathbf{b}\, dv \tag{7.5}$$

and the MWR is stated compactly

$$\langle R,\ W_i \rangle = 0 \qquad i\ =\ 1, \quad N \tag{7.6}$$

Equivalently, "R is orthogonal to W_i." With N basis functions ϕ_i selected *a priori*, a choice of N independent weighting functions W_i will determine the N unknown u_j. This is the essence of the Method of Weighted Residuals.

Note that for the exact solution, $R = 0$ everywhere and (7.6) is satisfied for any and all finite weighting functions W_i. Satisfaction of (7.6) is therefore a necessary but not sufficient condition for finding the true solution. If the set of weighting functions is *complete* – in the sense that it contains all possible residuals of the differential equation – then the condition (7.6) that R be orthogonal to a complete set of weighting functions leaves only one possibility: $R = 0$ everywhere *i.e.* we obtain the exact solution $\hat{u} = u$. Of course, in practice the weighting functions W_i will be finite and incomplete, just as the basis ϕ_i will not be complete. If we conceive of W_i as the first N members of a complete set, and likewise for ϕ_j, we may conceptualize the process of convergence as one in which the sets of weighting and basis functions are made progressively more complete. In the process, the possibilities for nonzero residual are diminished and R is ultimately annihilated, with $\hat{u}$ converging to u.

Several common methods may be categorized as MWR's:

- **Galerkin Method**: in which the weighting functions are identical to the basis functions:

$$W_i = \phi_i \tag{7.7}$$

  This is used extensively with finite elements and with spectral methods, the latter using orthogonal basis functions.

- **Least-Squares Method**: in which the objective is to minimize $< R^2 >$ with respect to each of the u_i independently. The result is the MWR requirement

$$\left\langle R, \frac{\partial R}{\partial u_i} \right\rangle = 0 \tag{7.8}$$

  and we recognize the weighting function

$$W_i = \frac{\partial R}{\partial u_i} \tag{7.9}$$

In the case of the Helmholtz equation (7.2), this becomes

$$W_i = \nabla^2 \phi_i + f\phi_i \tag{7.10}$$

- **Subdomain Method**: in which the weighting functions are uniform over a finite subdomain, and vanish elsewhere:

$$W_i = S_i \tag{7.11}$$

(see Figure 7.1). Use of compact, local subdomains (*e.g.* contiguous square boxes filling the plane) leads to recognizable finite difference approximations (see example below at section 7.2).

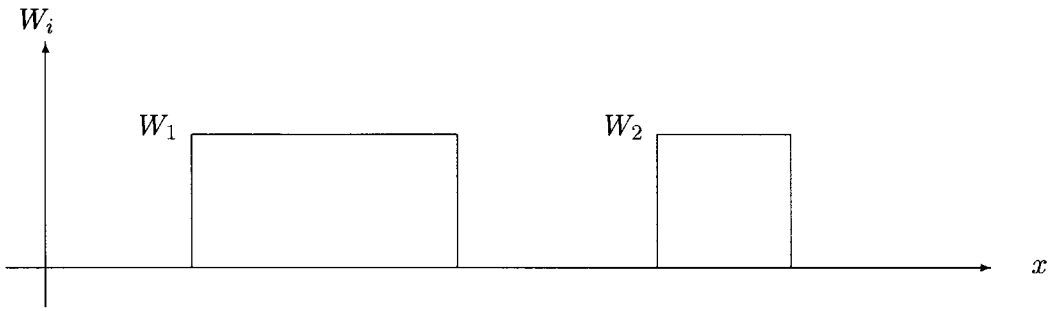


Figure 7.1: Subdomain method in 1-D: example of two weighting functions.

- **Collocation Method**: in which the residual is set to zero at discrete, predefined points $\mathbf{x}_i$. This may be cast as a MWR with Dirac delta functions as the weighting functions:

$$W_i = \delta(\mathbf{x}_i) \tag{7.12}$$

Careful selection of the collocation points $\mathbf{x}_i$ can provide powerful advantages; with certain bases, collocation and Galerkin methods can be shown to be identical.

7.2 MWR Examples

It is useful to examine some familiar 1-D examples in MWR form. For example, consider the simple approximation problem

$$\hat{u} = g(\mathbf{x}) \qquad 0 \leq x \leq l \tag{7.13}$$

where $g(\mathbf{x})$ is to be given with $g(0) = g(l) = 0$, and a Fourier sine series is desired:

$$\hat{u} = \sum_{j=1}^{N} u_j \sin(\frac{j\pi x}{l}) \tag{7.14}$$

We may consider this a "zero-order" differential equation in u; note that $\hat{u}$ exactly satisfies the boundary conditions because of the properties of the basis. The residual is

$$R = \hat{u} - g \tag{7.15}$$

and the MWR statement is

$$\left\langle \left(\sum_j u_j \sin(\frac{j\pi x}{l}) - g \right), W_i \right\rangle = 0 \quad i = 1, N \tag{7.16}$$

or, interchanging the order of summation and integration,

$$\sum_j u_j \left\langle \sin(\frac{j\pi x}{l}) \ , \ W_i \right\rangle = \langle g \ , \ W_i \rangle \qquad i = 1, N \tag{7.17}$$

In matrix form we have:

$$[A]\{u\} = \{b\} \tag{7.18}$$

with the coefficients given in terms of integrals of basis and weighting functions:

$$A_{ij} = \left\langle \sin(\frac{j\pi x}{l}), W_i \right\rangle \tag{7.19}$$

$$b_i = \langle g, W_i \rangle \tag{7.20}$$

The choice of W_i will now determine the u_i. The classic choice

$$W_i = \sin(\frac{i\pi x}{l}) \tag{7.21}$$

exploits the orthogonality of the Fourier basis, ($\left\langle \sin(\frac{i\pi x}{l}), \sin(\frac{j\pi x}{l}) \right\rangle = 0$ for $i \neq j$), rendering the $[A]$ matrix diagonal. It may be arrived at via Least Squares or Galerkin method in this case. The inversion of (7.18) is now trivial:

$$u_i = \frac{\left\langle g, \sin(\frac{i\pi x}{l}) \right\rangle}{\left\langle \sin^2(\frac{i\pi x}{l}) \right\rangle} \tag{7.22}$$

which is the classical result for the Fourier sine series.

As a second example, consider the 1-D version of the Helmholtz equation

$$\frac{d^2u}{dx^2} + fu = g \tag{7.23}$$

with $g(x)$ known, f constant, and boundary conditions as above:

$$u(0) = u(l) = 0 \tag{7.24}$$

We will use the Fourier sine basis (7.14) which fits the BC's as above. For the second derivative, we have

$$\frac{d^2\hat{u}}{dx^2} = -\sum_j u_j \left(\frac{j\pi}{l}\right)^2 \sin(\frac{j\pi x}{l}) \tag{7.25}$$

and the residual follows:

$$R(x) = \sum_j u_j \left[f - \left(\frac{j\pi}{l}\right)^2 \right] \sin(\frac{j\pi x}{l}) - g(x) \tag{7.26}$$

The weighted residual statement then takes the matrix form similar to (7.18):

$$[A']\{u\} = \{b\} \tag{7.27}$$

$$A'_{ij} = \left\langle \left[f - \left(\frac{j\pi}{l}\right)^2 \right] \sin\left(\frac{j\pi x}{l}\right), W_i \right\rangle \tag{7.28}$$

$$b_i = < g, W_i > \tag{7.29}$$

as above. The matrix $[A']$ will again be diagonalized if we exploit the orthogonality of the Fourier basis functions by choosing either Galerkin:

$$W_i = \sin\left(\frac{i\pi x}{l}\right) \tag{7.30}$$

or Least-Squares:

$$W_i = \left[f - \left(\frac{i\pi}{l}\right)^2 \right] \sin\left(\frac{i\pi x}{l}\right) \tag{7.31}$$

In either case we arrive at the same classical result by inversion of $[A']$:

$$u_i = \frac{\left\langle g, \sin\left(\frac{i\pi x}{l}\right) \right\rangle}{\left[f - \left(\frac{i\pi x}{l}\right)^2 \right] \left\langle \sin^2\left(\frac{i\pi x}{l}\right) \right\rangle} \tag{7.32}$$

i.e. the Fourier sine transform of the forcing function g divided by the transform of the Helmholtz operator.

The diagonalization of $[A']$ in (7.27) and the resultant simplicity of (7.32) is a direct consequence of the selection of basis and weighting functions which are a) orthogonal and b) eigenfunctions of the differential operator. The result (7.32) is readily generalized across the broad set of Sturm-Liouville problems [44]. For simple geometry, the orthogonal eigenfunctions of several model differential operators have been extensively studied and constitute the Special Functions of mathematical physics. Their use in such simple problems is natural and elegant. For complex geometry, however, the problem of determining the natural basis in terms of orthogonal eigenfunctions can be computationally overwhelming. In such realistic cases, then, one seeks a more humble (nonorthogonal) basis and should anticipate the need to solve a non-diagonal version of (7.27).

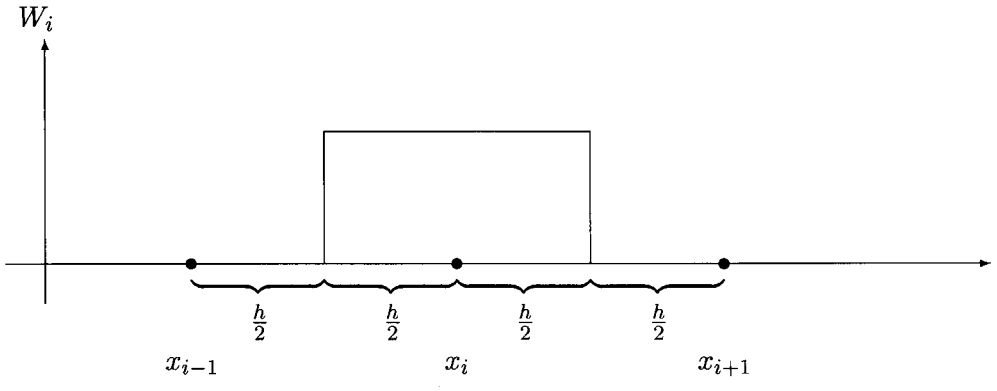


Figure 7.2: Subdomain approach on a regular grid.

As a third MWR example, consider the subdomain approach to the 1-D Helmholtz equation (7.23). We take as subdomains the contiguous finite difference cells as illustrated in Figure 7.2.

The MWR using W_i as shown gives

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} R\, dx = \frac{d\hat{u}}{dx}\Big|_{-\frac{h}{2}}^{\frac{h}{2}} + hf\overline{\hat{u}}_i - h\overline{g}_i = 0 \tag{7.33}$$

where the overbar indicates the average value on subdomain i. If for a basis we choose quadratic variation among the 3 nodes

$$\hat{u}(x) = \left[\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right]\frac{x^2}{2} + \left[\frac{u_{i+1} - u_{i-1}}{2h}\right]x + u_i \tag{7.34}$$

then it is readily verified that (7.33) reduces to

$$\left[\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right] + f\left[\frac{u_{i+1} + 22u_i + u_{i-1}}{24}\right] = \overline{g}_i \tag{7.35}$$

The first term is the conventional 2nd-order finite difference approximation to $\frac{d^2u}{dx^2}$; the second term is a local average reflecting this choice of basis and weighting functions.

Alternatively, we could choose a linear variation in $\hat{u}(x)$ between nodes; from (7.33) this would produce

$$\left[\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right] + f\left[\frac{u_{i+1} + 6u_i + u_{i-1}}{8}\right] = \overline{g}_i \tag{7.36}$$

i.e. the same expression for $\frac{d^2u}{dx^2}$ but a different average term $\overline{\hat{u}}_i$.

Finally, collocation at node i with the quadratic basis would produce the conventional finite difference form

$$\left[\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right] + fu_i = g_i \tag{7.37}$$

7.3 Weak Forms

Above it was noted that the exact solution will satisfy any set of weighted residual requirements. Clearly, the orthogonality condition $< R,\ W_i >= 0$ is *weaker* than the original, exact condition $R = 0$ everywhere, except for the limiting case where W_i is a complete set. Expanding R term-by-term gives the *weak form of the governing differential equation.* For the Helmholtz equation (7.2) we have the weak form

$$\left\langle \nabla^2 u, W_i \right\rangle + \langle fu, W_i \rangle = \langle g, W_i \rangle \tag{7.38}$$

Other weak forms are possible and desirable. For example, we may integrate the first term in (7.38) by parts, using Green's theorem:

$$\left\langle \nabla^2 u, W_i \right\rangle + \langle \nabla u, \nabla W_i \rangle = \oint \mathbf{n} \cdot W_i \nabla u\, ds \tag{7.39}$$

wherein the boundary integral encloses the domain and $\mathbf{n}$ is the unit vector normal to the boundary, directed outward. Use of (7.39) converts (7.38) to the alternate weak form of the Helmholtz equation:

$$-\langle \nabla u, \nabla W_i \rangle + \langle fu, W_i \rangle = \langle g, W_i \rangle - \oint \mathbf{n} \cdot W_i \nabla u\, ds \tag{7.40}$$

The exact solution satisfies both weak forms (7.38) and (7.40). Additionally, for linear problems, the solution error $\epsilon \equiv u - \hat{u}$ will satisfy the homogeneous version of the weak-form used to generate $\hat{u}$. For example, if (7.38) is used to generate $\hat{u}$, we have

$$\left\langle \nabla^2 \hat{u}, W_i \right\rangle + \langle f\hat{u}, W_i \rangle = \langle g, W_i \rangle \tag{7.41}$$

Subtracting (7.41) from (7.38), we have

$$\left\langle \nabla^2 \epsilon, W_i \right\rangle + \langle f\epsilon, W_i \rangle = 0 \tag{7.42}$$

7.4 Discrete Form

Finally, we state the weak forms (7.38) and (7.40) in discrete, matrix form by expressing $\hat{u}$ in the basis (7.1). As in (7.18) and (7.27), we have

$$[A]\{u\} = \{b\} \tag{7.43}$$

with A_{ij} and b_i comprising integrals of ϕ, W, and their derivatives. In the case of (7.38) we have

$$A_{ij} = \left\langle \nabla^2 \phi_j,\ W_i \right\rangle + \langle f\phi_j, W_i \rangle \tag{7.44}$$

$$b_i = \langle g, W_i \rangle \tag{7.45}$$

Similarly from (7.40) we have

$$A_{ij} = -\langle \nabla\phi_j, \nabla W_i \rangle + \langle f\phi_j, W_i \rangle \tag{7.46}$$

$$b_i = \langle g, W_i \rangle - \oint \frac{\partial u}{\partial n} W_i \, ds \tag{7.47}$$

It is the discrete form which is solved numerically. There are two basic computational steps: the evaluation of the domain and boundary integrals in $[A]$ and $\{b\}$ (the *assembly* step), and the solution of the matrix equation itself.

7.5 Boundary Conditions

For the Helmholtz equation used in the previous examples, we distinguish three types of boundary conditions:

- Type 1 (Dirichlet): u is specified;
- Type 2 (Neumann): $\nabla u \cdot \hat{\mathbf{n}}$ is specified;
- Type 3 (Mixed): a blend of u and $\nabla u \cdot \hat{\mathbf{n}}$ is specified: $\nabla u \cdot \hat{\mathbf{n}} + \alpha u = \beta$.

At all points on the closed boundary Γ, exactly one of these conditions must be specified. We refer to the boundary segment Γ_1 as the Type I portion of Γ, and similarly for Γ_2 and Γ_3. In WR terms, the Type 1 conditions are referred to as *essential* conditions, while the Type 2 and 3 conditions are *natural* conditions. This classification holds for the general 2nd-order elliptic partial differential equation.

In general it is necessary that the approximate solution $\hat{u}$ satisfy the essential boundary conditions. We may thus separate the numerical solution into homogeneous and particular parts:

$$\hat{u} = \hat{u}_p + \hat{u}_h \tag{7.48}$$

with respect to the essential boundary conditions. The particular part $\hat{u}_p$ satisfies the essential conditions and is known *a priori*; the basis for the unknown, homogeneous part of the solution is therefore required to vanish on Γ_1.

Natural conditions, on the other hand, are satisfied only approximately or "weakly" in most WR methods. The boundary integral is typically the vehicle for their enforcement. For example, in equation (7.40), in terms of $\hat{u}$ we have

$$-\langle \nabla \hat{u}, \nabla W_i \rangle + \langle f\hat{u}, W_i \rangle = \langle g, W_i \rangle - \oint W_i \mathbf{n} \cdot \nabla u \, ds \tag{7.49}$$

On the right side, the boundary integral is expressed not in terms of $\hat{u}$ but in terms of u; the natural boundary conditions enter the calculations here. This is a "weak" constraint since the slope of $\hat{u}$ is not being directly constrained; rather, the weak form is driven by the boundary data.

An obvious operational requirement for this procedure is that the integrand $W_i \mathbf{n} \cdot \nabla u$ vanish on Γ_1; otherwise the required information is not available. Therefore, we require that the weighting functions W_i vanish on Γ_1. There is obvious symmetry with the requirement for ϕ_i.

For the general Type III condition, the substitution $\nabla u \cdot \hat{\mathbf{n}} = -\alpha\hat{u} + \beta$ leads to the weak form

$$-\langle \nabla \hat{u}, \nabla W_i \rangle + \langle f\hat{u}, W_i \rangle - \oint W_i \alpha \hat{u} \, ds = \langle g, W_i \rangle - \oint W_i \beta \, ds \tag{7.50}$$

and the $\alpha\hat{u}$ term is internalized in the system.

These are operational rules. The burden of proof is to show that a particular WR method converges as the basis and weighting functions approach complete sets. The reader is referred to more fundamental texts (*e.g.* Strang and Fix, [103]) for this theory.

7.6 Variational Principles

In some cases a variational principle accompanies a differential equation and satisfaction of either is equivalent. For example, consider the functional Q, defined as

$$Q = \frac{1}{2}\langle K\nabla\hat{u}, \nabla\hat{u} \rangle - \frac{1}{2}\langle f\hat{u}, \hat{u} \rangle + \langle g, \hat{u} \rangle - \oint K \frac{\partial u}{\partial n} \hat{u} \, ds \tag{7.51}$$

and representing the sum of potential and kinetic "energy" plus "work" done on the volume and boundary, respectively. In physical systems where this functional is meaningful, Q is minimized by the function u which also satisfies the Helmholtz equation

$$\nabla \cdot K\nabla u + fu = g \tag{7.52}$$

at every point in the domain. This differential equation is obtained by minimizing Q over all admissible functions via the Calculus of Variations.

The functional Q, already in domain integral form, provides an alternate starting point for generating a weak form differential equation. With a finite basis for $\hat{u}$:

$$\hat{u} = \hat{u}_p + \sum_j u_j \phi_j \tag{7.53}$$

the first-order conditions for a minimum are easy to obtain as

$$\frac{\partial Q}{\partial u_i} = 0 \qquad i = 1, \quad N \tag{7.54}$$

and we may operate within the integrals term-by-term with

$$\frac{\partial}{\partial u_i} = \frac{\partial \hat{u}}{\partial u_i}\frac{\partial}{\partial \hat{u}} = \phi_i \frac{\partial}{\partial \hat{u}} \tag{7.55}$$

The result is the weak form

$$\langle K\nabla\hat{u}, \nabla\phi_i\rangle - \langle f\hat{u}, \phi_i\rangle + \langle g, \phi_i\rangle = \oint K\frac{\partial u}{\partial n}\phi_i \, ds \tag{7.56}$$

Admissible variations must satisfy the essential boundary conditions; therefore ϕ_i vanishes on Γ_1; and $K\frac{\partial \hat{u}}{\partial n}$ is available as boundary condition data on the rest of the boundary. It is evident that (7.56) matches term-by-term with the Galerkin MWR version of the Helmholtz equation, weak form (7.40).

This correspondence between Variational and Galerkin approaches is quite general, for linear problems, and is characterized by the dual pathways to the same weak and discrete forms illustrated in Figure 7.3.

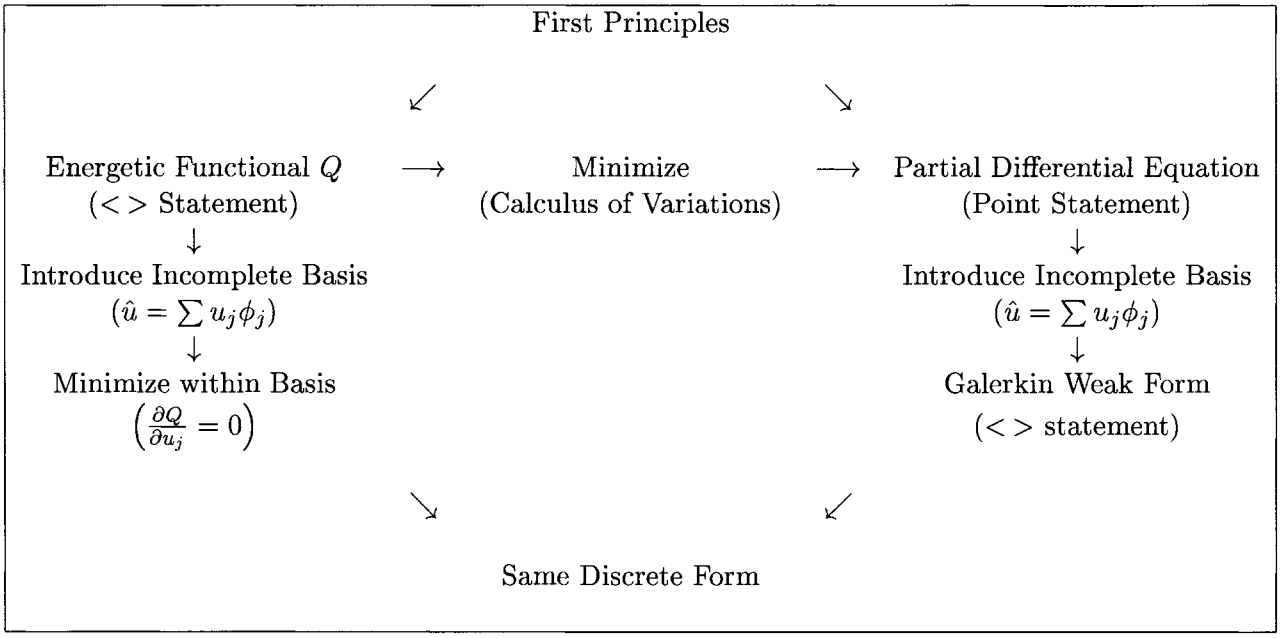


Figure 7.3: Illustrating the two pathways for obtaining identical MWR Discrete Forms for linear problems with a variational principle.

7.7 Weak Forms and Conservation Properties

A great many of the classical PDE's share an underlying formulation unity in Conservation Laws. These generally can be satisfied only approximately by finite fields. However the FEM has some intrinsic *integral* conservation properties, which originate in the choice of Weak Form. We display some of the generalities here. Much more remains to be developed in specific application contexts.

This section may be omitted on a first reading.

Divergence Forms

The simplest and very common PDE form involves a *flux divergence* formulation to achieve conservation of a physical quantity. This was discussed in connection with FD discretization (equation 5.131):

$$\begin{array}{lll} \textit{Conservation Law} & + \quad \textit{Constitutive Relation} & \Rightarrow \textit{PDE} \\ \nabla \cdot q = \sigma & \qquad q = -\kappa \nabla U & \qquad \nabla \cdot \kappa \nabla U = -\sigma \end{array} \tag{7.57}$$

Here q is the vector flux of the conserved quantity; σ is its source; and U is the scalar surrogate for the flux. Table 5.1 displays some of the specific instances of this common form. The PDE as listed is an Elliptic operator; all temporal dynamics are bundled here in the source term σ. If σ contains at most $\frac{\partial U}{\partial t}$, we have a Parabolic equation overall; $\frac{\partial^2 U}{\partial t^2}$, Hyperbolic.

The integral form of this conservation law is

$$\oint q \cdot \hat{n} \, ds = < \sigma > \tag{7.58}$$

where the Divergence Theorem (see Appendix) has been invoked, along with the usual FEM shorthand <> for volume integration. If we insert the constitutive relation (7.57), we have the equivalent statement:

$$-\oint \kappa \frac{\partial U}{\partial n} ds = < \sigma > \tag{7.59}$$

In either case, we have *a priori* the integrated balance between all the source in the domain of integration, and its efflux across the boundary. Included in σ is the internal accumulation rate for dynamic problems.

The Weak Form is

$$- < q \cdot \nabla W_i > + \oint q \cdot \hat{n} \, W_i \, ds = < \sigma \, W_i > \tag{7.60}$$

This is a discretized, local approximation to the conservative point balance expressed in the PDE. It is a statement that the *weighted* residual of this balance must vanish. The residual function itself will generally be nonzero, so the discretized conservation statement is *weak*; its residual will vanish everywhere only when the elements are infinitesimal. For *finite* elements, the local norm of this residual is a useful metric of numerical fidelity.

Despite this "weakness" for finite meshes, there is an *exact* global conservation statement. This can be arrived at by summing the WR equations. In so doing we require and exploit this important property of FEM weights

$$\sum_i^N W_i = 1 \tag{7.61}$$

everywhere, irrespective of N; and therefore

$$\nabla \sum_i W_i = \sum_i \nabla W_i = 0 \tag{7.62}$$

Therefore by summing the equations (7.60), the internal fluxes $< q \cdot \nabla W_i >$ cancel out; we get

$$\oint q \cdot \hat{n}\, ds =< \sigma > \tag{7.63}$$

The sum of the full WR system provides *perfect* conservation, despite the weakness of its parts. Numerically, the efflux perfectly balances the internal sources. This perfect balance is not dependent on refined resolution or on the details of the basis ϕ. Nor is it dependent on the manner in which the constitutive relation is represented, assuming reasonably consistent implementation. Instead, it is a constraint implied in the WR method at the outset, assuming that W has the property (7.61). In the Galerkin method with $W = \phi$, this is equivalent to requiring that the basis be able to interpolate a constant. For non-Galerkin, the constraint (7.61) is reasonable – the residual to be sampled everywhere, democratically – but not necessary. It is reasonable to require property (7.61) of any WR method, unless there is a contraindication.

This development applies immediately to Type II and III problems. But there is a problem at Dirichlet boundaries. Above, we required that "W vanish on Γ_1". The equivalent common wisdom is, "ignore the Weak Form near Dirichlet boundaries", where the solution is already known. So, the requirement (7.61) would seem to be violated.

The way forward is to view the Dirichlet problem as equivalent to a Neumann problem; and to use the Weak Formulation to *derive* the equivalent Neumann data along Type I boundaries. In practice, part of W is "removed" from the weight space W to solve the Dirichlet problem; that complementary part of W is then restored in order to derive the equivalent Neumann data. Property (7.61) is thereby preserved and conservation guaranteed; dual Neumann and Dirichlet boundary data become known; and either supports the same interior solution because the identical set of MWR equations governs.[1] The importance of this is discussed in several different physical contexts (groundwater, oceans, heat transfer, phase change, fluid mechanics) [22, 39, 64, 65, 69, 80, 81, 84]. This approach has various names attached; in [39] it is referred to as the *consistent flux method.* It produces superior solutions in addition to preserving the conservation properties described here.

Circulation Forms

A second generic form which is commonly used in vorticity and electromagnetism involves the curl operator at the heart of an elliptic PDE:

$$\nabla \times A = S \tag{7.64}$$

with A an unknown vector field of interest and S a forcing vector. It is common to express A in terms of various vector and/or scalar potentials, *e.g.*

$$A = \nabla \times F + \nabla \Phi \tag{7.65}$$

[1] Remember that numerically, q is only weakly related to $\partial U/\partial n$, so simple differentiation of the numerical Dirichlet solution will not produce exact conservation.

with the surrogate potential vector F and scalar Φ becoming the unknowns. Irrespective of the details of this closure, we have the analytical property from the circulation theorem

$$\oint (\hat{n} \times A)\ ds =< S > \tag{7.66}$$

which expresses the balance between surface circulation and the sources enclosed.

The WR discretization utilizes a related circulation theorem (Appendix):

$$< A \times \nabla W_i > + \oint (\hat{n} \times A) W_i\ ds =< S\ W_i > \tag{7.67}$$

This relation is the weak form, and the same comments apply to it as above for the divergence-based operators. Summation of the WR equations, and again assuming property (7.61), we have

$$\oint (\hat{n} \times A)\ ds =< S > \tag{7.68}$$

As above, we recover an exact replica of a global analytic property, independent of discretization or basis, and with implications for dealing with boundary conditions. An elaboration of developments of this in computational Electromagnetism is available in [78].

Summary

- Given the reasonable constraint $\sum W_i = 1$, the MWR Divergence and Curl operators both share the property that intramural transports cancel. As a result, the global Divergence and Circulation theorems are reproduced exactly, for the finite fields generated. This is a simple consequence of using Weak Forms with these operators "integrated by parts".

- There are important implications for boundary condition implementation. The Weak Form exposes the natural BC in the boundary integrals. It must be used in a dual way for either Dirichlet or Neumann problems. Effectively there is a dual pair of BC's, one and only one of which must be specified, the other being derivable.

- Local conservation balances are always enforced in the "weak" sense – they approach perfection as the function spaces (W, ϕ) approach completeness, *i.e.* as the element size becomes infinitesimal. For *finite* elements, suitable local norms of the residual function are good guides for mesh refinement and/or basis enrichment [1, 6].

- Extensions of these ideas to interior flux estimation and conservation statements on subdomains are natural [21].

- If integrals are integrated numerically, for example by popular Gauss Quadrature, this finding remains sound; the requirement $\sum W = 1$ is restricted to the Gauss Points [64].

Important criteria for basic MWR formulation are implied for more complex problems. This is illustrated in the example below.

Example: Advective-Diffusive-Reactive Equation

Consider for example the PDE

$$\frac{\partial C}{\partial t} + rC + \nabla \cdot q = 0 \tag{7.69}$$

and its integral statement

$$< (\frac{\partial}{\partial t} + r)C > + \oint q \cdot \hat{n}\, ds = 0 \tag{7.70}$$

Either of these expresses the integrated balance between Domain Accumulation and Decay, and Boundary Efflux.

There are many possible Weak statements of this. Exploiting Integration by Parts from the outset, we have

$$< \frac{\partial C}{\partial t} W_i > + < rC\ W_i > - < q \cdot \nabla W_i > + \oint q \cdot \hat{n} W_i\, ds = 0 \tag{7.71}$$

On summation we recover a perfect replica of equation 7.70 above. So this particular WR discretization shares an exact conservation property with the PDE.

The typical closure is in terms of advective and nonadvective fluxes

$$q = vC - D\nabla C \tag{7.72}$$

with advective field v and diffusivity D. This closure may be inserted into the Weak-Form statements above where needed; assuming reasonable consistency, the perfect conservation balance is preserved. The internal fluxes cancel without further constraining how they are discretized or how the integrations are approximated. The only constraint is that $\sum W_i = 1$ everywhere.

In a second Weak Form, it is common to separate advective and nonadvective flux components, and to differentiate the advective part *before* discretization:

$$\frac{\partial C}{\partial t} + rC + C\nabla \cdot v + v \cdot \nabla C - \nabla \cdot D\nabla C = 0 \tag{7.73}$$

and to invoke the relation between v and its sources σ:

$$\nabla \cdot v = \sigma \tag{7.74}$$

(The case $\sigma = 0$ is common.) Putting these two together, we obtain the PDE

$$\frac{\partial C}{\partial t} + rC + \sigma C + v \cdot \nabla C - \nabla \cdot (D\nabla C) = 0 \tag{7.75}$$

A common Weak Form of this is:

$$< (\frac{\partial C}{\partial t} + rC + \sigma C)W_i > + < v \cdot \nabla C\ W_i > + < D\nabla C \cdot \nabla W_i > - \oint D\frac{\partial C}{\partial n}\ W_i\, ds = 0 \tag{7.76}$$

where the diffusive part of q has been integrated by parts, but not the advective part. Summing these, we get

$$< (\frac{\partial C}{\partial t} + rC + \sigma C) > + < v \cdot \nabla C > - \oint D\frac{\partial C}{\partial n}\, ds = 0 \tag{7.77}$$

and the balance among the two terms σC and $v \cdot \nabla C$ needs further attention. Typically, v is itself a discrete field, and must be represented approximately in a finite basis. We then need a Weak Form of equation (7.74). *Assuming integration by parts*, we have:

$$- < v \cdot \nabla W_i > + \oint v \cdot \hat{n} \, W_i \, ds =< \sigma W_i > \tag{7.78}$$

(Note there are other options here.) Multiplying each of these by the nodal value C_i and summing, we acquire in the integrands $\sum C_i W_i$. If $W_i = \phi_i$, *i.e.* the continuity weights are equal to the basis for C, then $\sum C_i W_i$ is the numerical field C, everywhere; and therefore we have

$$- < v \cdot \nabla C > + \oint C v \cdot \hat{n} \, ds =< \sigma C > \tag{7.79}$$

This saves the conservative balance by restoring the advective flux across the boundary, and getting rid of the internal terms:

$$< (\frac{\partial C}{\partial t} + rC) > + \oint C v \cdot \hat{n} \, ds - \oint D \frac{\partial C}{\partial n} \, ds = 0 \tag{7.80}$$

Invoking the constitutive relation simplifies the boundary integrals:

$$< (\frac{\partial C}{\partial t} + rC) > + \oint q \cdot \hat{n} \, ds = 0 \tag{7.81}$$

In addition to the required property $\sum W_i = 1$, this formulation requires a Galerkin relation between the continuity equations and the C basis; and the use of integration by parts of both divergence terms. These features together guarantee exact global conservation.

Notice that we are, essentially, adding nothing to the basic formulation here. All we are doing is adding the MWR equations actually used, and cancelling the internal identities. But very specific constraints are implied on choice of Weak Form, basis, and representation of boundary fluxes, if this is to work.

Extra care is needed in these developments to distinguish the numerical field C and its derivatives. In particular, the numerical q contains a diffusive (gradient) contribution which is normally only *weakly* related to $\frac{\partial}{\partial n} \sum C_i \phi_i$. Failure to recognize this can lead to formulation error. The reader is referred to [64] for more detailed discussion of this, and also of the implications of approximate numerical integration.

There are some general observations illustrated here. As a rule, preserving exact conservative properties is critically dependent on the path: the order of approximation, integration, and differentiation, and of course the PDE itself. Obtaining the proper weak form *before* the numerical basis is introduced allows operating with the PDE in the continuum, where integration and differentiation are perfect. Weak formulations that preserve the natural conservation statements in the PDE, *i.e.* the Divergence and/or Curl operators in original form, are likely to produce recognizable discrete balances involving natural boundary conditions.

We cannot count on perfect numerical integration. Integration by parts of the divergence or curl operators, before introducing closure or bases, injects the natural BC's into the integrated conservation statements, using the properties of the continuum.

Chapter 8

A 1-D Tutorial

8.1 Polynomial Bases – the Lagrange Family

The overwhelming majority of FE methods use polynomials as the basis functions. Polynomials have been extensively studied as approximants to functions with various degrees of smoothness, and their manipulation both analytically and computationally is straightforward. In particular, all quantities needed by Weighted Residual Methods are integrals of the bases and their derivatives; these are especially simple for polynomials.

Consider a one-dimensional domain x with pre-defined points or "nodes" x_j. For our purposes, the simplest polynomial form is the Lagrange family:

$$\phi_i(x) = \prod_{i \neq j} \left(\frac{x - x_j}{x_i - x_j} \right) \tag{8.1}$$

For $N+1$ nodes, $\phi_i(x)$ provides a continuous polynomial of order N with the property $\phi_i(x) = \delta_{ij}$ *i.e.* ϕ_i is unity at its "home" node, and vanishes at all other nodes; and it is the unique N^{th} order polynomial with this property. The most general N^{th} order polynomial variation on this domain is obtained with an arbitrary combination of ϕ_1, ϕ_2,..., ϕ_{N+1}:

$$\hat{u}(x) = \sum_{i=1}^{N+1} u_i \, \phi_i(x) \tag{8.2}$$

and it is immediately clear that the coefficients u_i are the values of the polynomial evaluated at node i:

$$u_i = \hat{u}(x_i) \tag{8.3}$$

and that $\hat{u}(x)$ is the unique N^{th} order interpolant among the given u_i. Thus, the Lagrange family readily facilitates polynomial interpolation among data or functions sampled at nonuniform intervals.

For example, the three members of the quadratic Lagrange family are illustrated in Figure 8.1, wherein

$$\phi_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \tag{8.4}$$

and similarly for ϕ_2, ϕ_3.

In passing, we note that the Lagrange family of any order N has the property $\sum_i \phi_i(x) = 1$ for any x; this property is necessary if the basis is to exactly represent a constant.

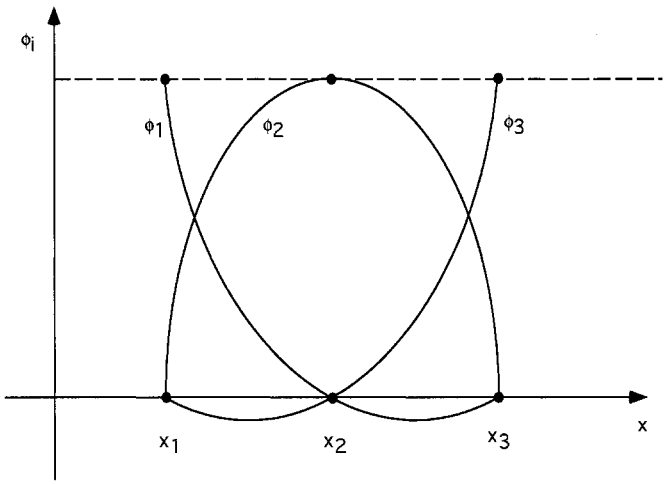


Figure 8.1: Quadratic Lagrange element in 1-D.

8.2 Global and Local Interpolation

Consider the data of Table 8.1

Table 8.1: Interpolation data.

x_i	0	1	2	3	4	5
u_i	5	7	2	1	9	8

and its 5^{th} order interpolating polynomial. Suppose we are interested in interpolating the value $\hat{u}(x = 4.5)$. Straightforward evaluation yields $\hat{u}(x = 4.5) = 12.11$ for this "global" interpolation! Alternately, we may use the most "local" interpolation, *i.e.* linear variation between x_4 and x_5, which yields $\hat{u}(x = 4.5) = 8.5$. Results for intermediate values of N are shown in Table 8.2. It is clear that the more global the interpolant, the more likely is the occurrence of extreme points lying between the nodes (reflecting the possibility of N zeros of an N^{th} order polynomial). The 5^{th} order Lagrange polynomials have "global support" in this example, *i.e.* they are generally nonzero everywhere on the domain, and all nodal data, no matter how far removed, affects the interpolation everywhere.

Table 8.2: Interpolated result at $x = 4.5$

N	1	2	3	5
$\hat{u}(x = 4.5)$	8.5	9.6	10.75	12.11

As a second example, consider the function $u = \frac{1}{x}$ sampled as shown in Table 8.3. Clearly, $\frac{1}{x}$ is not a polynomial, but we hope for a good approximation in the polynomial basis. Interpolation of $\hat{u}(x = 4.5)$ and extrapolation of $\hat{u}(x = 5.5)$, with various degrees of polynomial, is shown below in Table 8.4. In this case, it is clear that $N = 1$ or 2 is superior in both interpolation and extrapolation, and that increasing sophistication in the polynomial basis is counterproductive. These examples

illustrate some of the hazards associated with global interpolation; in general, globally-supported bases are not employed in FE analysis at least partly because of these undesirable features.

Table 8.3: Sampled values of $u = \frac{1}{x}$.

x_i	1	2	3	4	5
u_i	1	.5	.333	.250	.200

Table 8.4: Interpolation and extrapolation results.

N	Exact	1	2	3	4
$\hat{u}(4.5)$	.222	.225	.2206	.224	.216
$\hat{u}(5.5)$	.167	.15	.183	.133	.333

The alternative, local interpolation, is a universal and fundamental principle of the FEM. The domain of interest is divided into contiguous pieces or *elements* as shown in Figure 8.2. Each element then supports its own, self-contained interpolating basis $\phi_i^e(x)$ and interpolant $\hat{u}^e(x)$. As suggested in Figure 8.2, neighboring elements may use different order interpolation if desired. This eliminates some of the pitfalls associated with global interpolation, and also provides the opportunity to have higher-order local interpolation only on selected elements where it is necessary.

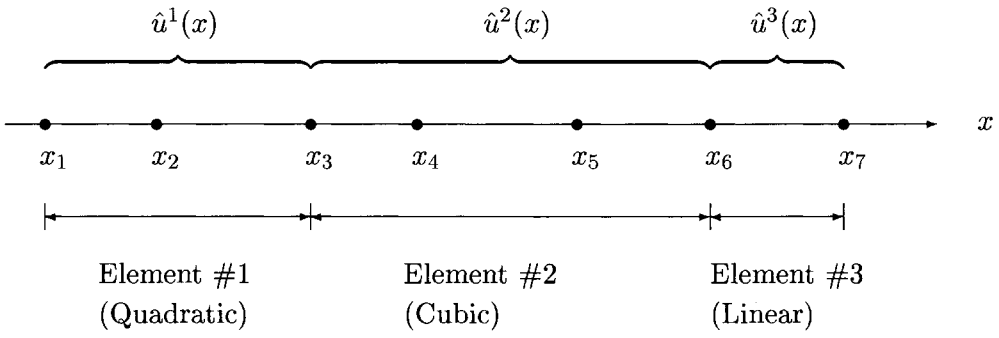


Figure 8.2: Discretization of x domain into three elements.

Elements, then, are the basic *building blocks* of the FEM, from two perspectives:

- the basis itself, *i.e.* the local interpolation; and
- the integration of the WRM.

The second point will be detailed later, but observe here that all WRM quantities involve integrals which may be evaluated on an element-by-element basis and summed, provided there are no singularities lurking at the junctions between elements.

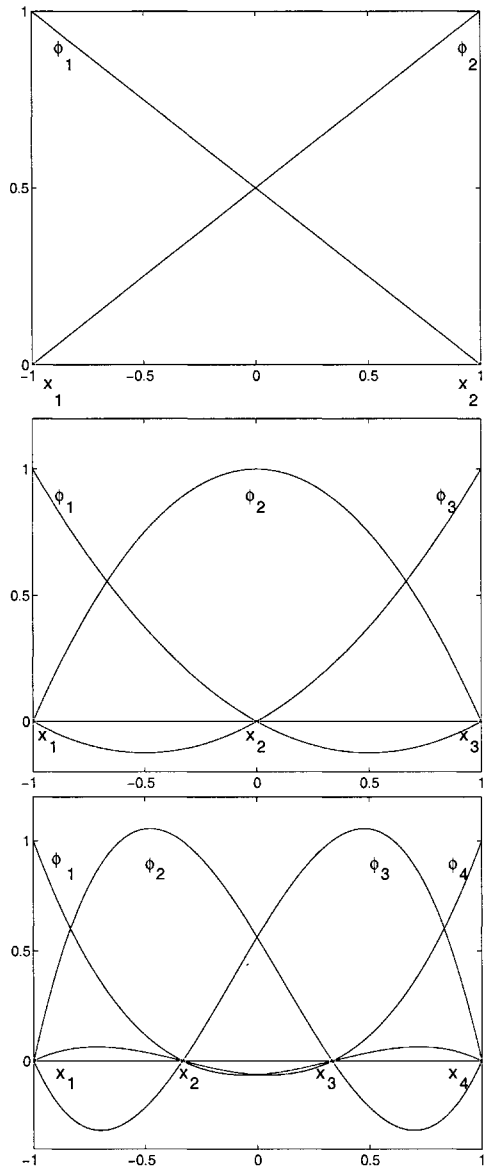


Figure 8.3: Lagrange Bases in 1 dimension.

8.3 Local Interpolation on Elements

An individual element may be viewed in isolation, as a self-contained interpolating unit. All elements of a given *type* are mathematically identical from this local perspective. For example, equations 8.1 and 8.2 suffice for the whole Lagrange family; all that is needed is to specify the (x_i, u_i) pairs for a particular element. Figure 8.3 shows generic Lagrangian elements of different orders. A few local conventions (left-to-right node ordering in 1-D) make the element-level description even more universal.

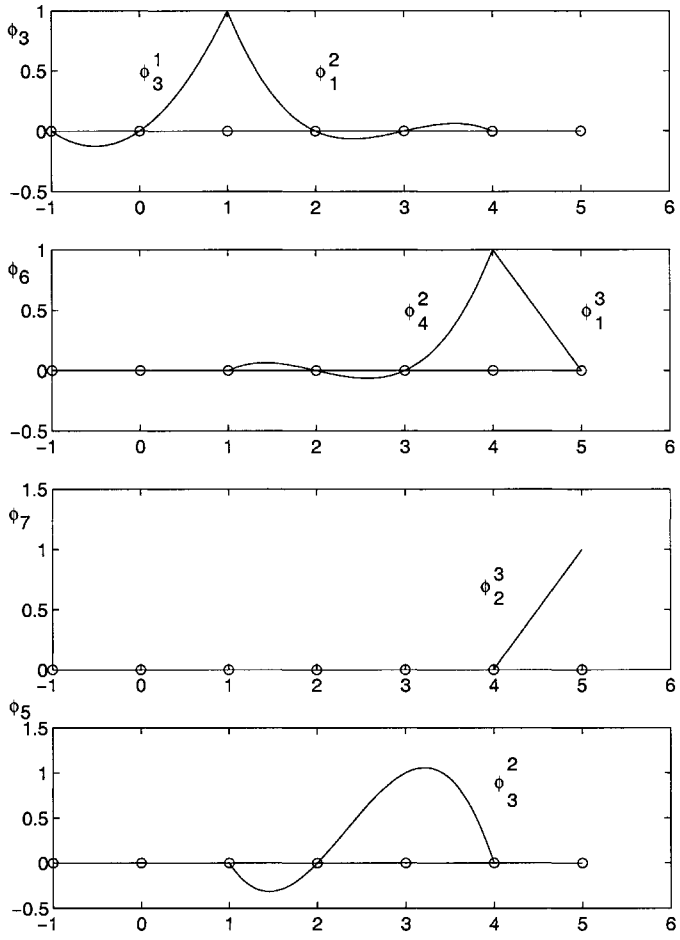


Figure 8.4: Global variation of selected bases on the mesh of Figure 8.2

Here it is useful to introduce the element-based notation ϕ_l^e:

$$\phi_l^e(x) = \prod_{l \neq j} \left(\frac{x - x_j}{x_l - x_j} \right) \quad for\ x\ in\ element\ e \tag{8.5}$$

$$= 0 \qquad otherwise. \tag{8.6}$$

In these terms, the global variation of, say $\phi_3(x)$ in Figure 8.2 would be expressed as the sum of all element variations:

$$\phi_3(x) = \phi_3^1(x) + \phi_1^2(x) \tag{8.7}$$

as illustrated in Figure 8.4. Figure 8.4 also illustrates the global variation of selected other bases on this mesh.

8.4 Basis Function Continuity – Hermite Polynomials

It is apparent that polynomial bases will be smooth with smooth derivatives within any element; but that their continuity will be interrupted at element boundaries. The Lagrange family described

above is constructed so that the bases are continuous at element boundaries, but their first and higher derivatives will be discontinuous there (see Figure 8.4). It follows that all functions expressed in these bases will have the same continuity properties. We refer to these as C^0 elements, *i.e.* they support functions whose global continuity is limited to the zeroth derivative.

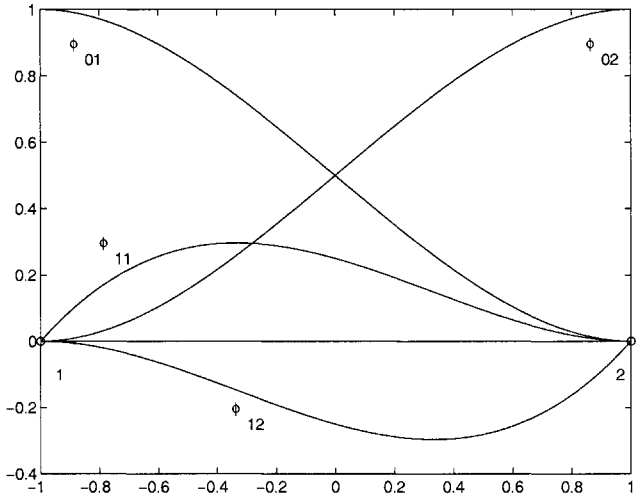


Figure 8.5: Hermitian cubic bases.

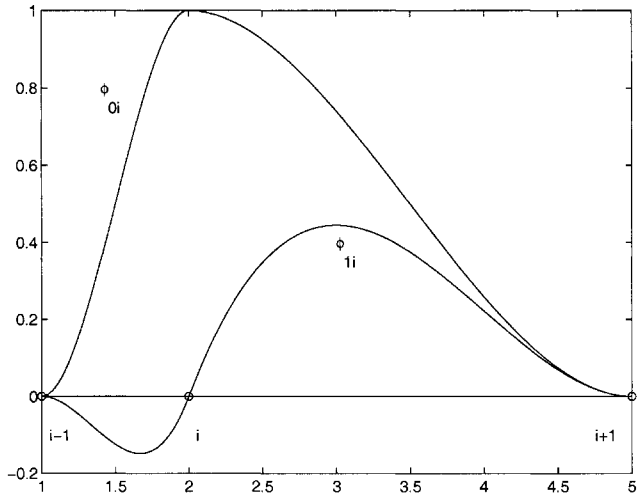


Figure 8.6: Global variation of Hermitian cubic bases associated with node i.

Higher-order continuity can be built into the basis. Consider as a starting point the simplest C^0 linear element, wherein local variation of the form $\phi = cx + d$ is defined by nodal values at the two endpoints of the element. We may increase the order of interpolation to quadratic by adding a third constraint sufficient to determine the extra constant in $\phi = bx^2 + cx + d$. In the Lagrangian family, this is achieved by adding a third node on the interior of the element, leaving the continuity at element boundaries unchanged. An alternative is to constrain the slope of the function at one of the existing nodes, thereby enabling continuity of the first derivative at that node. Cubic variation is obtained by treating both nodes in this way – *i.e.* the value and slope of the function are specified at both nodes.

The result is the standard Hermite cubic element, which provides C^1 interpolation on 2-node elements. The bases are illustrated in Figure 8.5. We need to distinguish two types of bases on this element: (ϕ_{01}, ϕ_{02}) associated with the value of the function (its "zeroth derivative"), and (ϕ_{11}, ϕ_{12}) associated with its first derivative. A function expressed in the Hermite basis requires nodal values of u and its derivative, u_{0i} and u_{1i}:

$$\hat{u}(x) = \sum_i (u_{0i}\phi_{0i} + u_{1i}\phi_{1i}) \tag{8.8}$$

Constraints on the four bases are listed in Table 8.5. There are four constraints for each of the

Table 8.5: Nodal values of Hermite cubic bases.

	ϕ_{01}	ϕ_{02}	ϕ_{11}	ϕ_{12}
$\phi(x_1)$	1	0	0	0
$\phi(x_2)$	0	1	0	0
$\frac{\partial\phi}{\partial x}(x_1)$	0	0	1	0
$\frac{\partial\phi}{\partial x}(x_2)$	0	0	0	1

cubic functions: 2 constraints on the function value, and 2 on its slope. The cubics are readily obtained:

$$\phi_{01} = 2\xi^3 - 3\xi^2 + 1 \tag{8.9}$$

$$\phi_{02} = -2\xi^3 + 3\xi^2 \tag{8.10}$$

$$\phi_{11} = (x_2 - x_1)\left(\xi^3 - 2\xi^2 + \xi\right) \tag{8.11}$$

$$\phi_{12} = (x_2 - x_1)\left(\xi^3 - \xi^2\right) \tag{8.12}$$

where ξ is the natural local coordinate:

$$\xi \equiv \frac{(x - x_1)}{(x_2 - x_1)} \tag{8.13}$$

Figure 8.6 depicts the global view of ϕ_{0i} and ϕ_{1i} for a single node i. As in the Lagrangian case, this basis is locally supported *i.e.* it is nonzero only in elements containing node i. The continuity of $\frac{\partial\phi}{\partial x}$ is apparent. (Not as apparent is the necessary discontinuity of the higher derivatives.)

An alternate local coordinate can be defined as

$$\chi \equiv 2\frac{(x - x_1)}{(x_2 - x_1)} - 1 \tag{8.14}$$

which is zero at the center of the element and ranges from -1 at the left node to +1 at the right. In this coordinate, the basis functions are

$$\phi_{01} = \frac{\chi^3 - 3\chi + 2}{4} \tag{8.15}$$

$$\phi_{02} = \frac{-\chi^3 + 3\chi + 2}{4} \tag{8.16}$$

$$\phi_{11} = (x_2 - x_1)\left(\frac{\chi + 1}{2}\right)\left(\frac{\chi - 1}{2}\right)^2 \tag{8.17}$$

$$\phi_{12} = (x_2 - x_1)\left(\frac{\chi + 1}{2}\right)^2\left(\frac{\chi - 1}{2}\right) \tag{8.18}$$

This will be a more natural coordinate system for numerical integration on an element (see later section).

Higher-order continuity can be obtained by extending this procedure – for example, 5th order polynomials on a 2-node element can provide C^2 continuity, etc.

8.5 Example

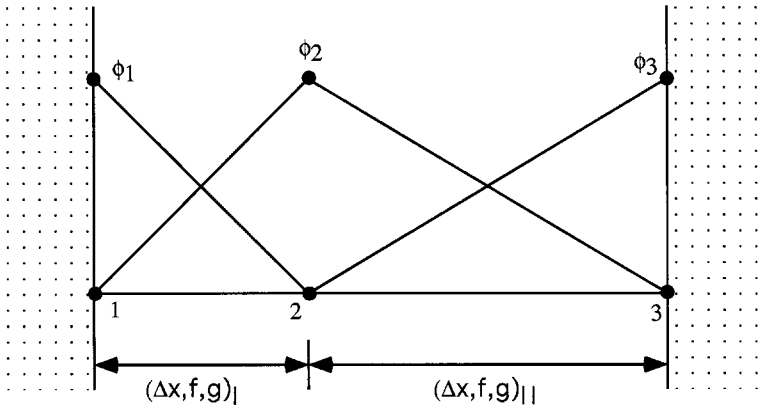


Figure 8.7: Mesh of two linear elements for the example in section 8.5.

Let's work a 1-D example before going further. Consider the Helmholtz equation introduced above (equation 7.2):

$$\frac{d^2u}{dx^2} + fu = g \tag{8.19}$$

with boundary conditions at $x = (0,\ L)$. As in section 7.3, we have the weak form

$$-\int_0^L \frac{du}{dx}\frac{dW_i}{dx}\,dx + \int_0^L f u W_i\,dx = \int_0^L g W_i\,dx - \left(\frac{du}{dx}W_i\right)\Bigg|_{x=0}^{x=L} \tag{8.20}$$

We will use the Galerkin method ($W_i = \phi_i$) with the simplest linear bases on the two-element mesh shown in Figure 8.7. Following the development in section 7.4, we have

$$[A]\{u\} = \{b\} \tag{8.21}$$

$$A_{ij} = -\int_0^L \frac{d\phi_j}{dx}\frac{d\phi_i}{dx}\,dx + \int_0^L f\phi_j\phi_i\,dx \tag{8.22}$$

$$b_i = \int_0^L g\phi_i\,dx - \left(\frac{du}{dx}\phi_i\right)\Bigg|_{x=0}^{x=L} \tag{8.23}$$

Recall that each row of the matrix equation is the residual weighted with ϕ_i, which we will denote here as WR_i. We will consider each row of the matrix equation by itself.

Row 1 is WR_1, the residual weighted with ϕ_1. It is easy to see that $\int(\,)\,dx$ in this row need only be evaluated over the first element, because ϕ_1 is supported only on element I. Similarly, we will have nonzero values for A_{11} and A_{12}, but A_{13} will be zero, because ϕ_3 is not supported on

ξ	$2\left(\frac{x-x_1}{x_2-x_1}\right)-1$
$\phi_1(\xi)$	$\left(\frac{1-\xi}{2}\right)$
$\phi_2(\xi)$	$\left(\frac{1+\xi}{2}\right)$
$\frac{d\phi_1}{dx}$	$-\frac{1}{\Delta x}$
$\frac{d\phi_2}{dx}$	$\frac{1}{\Delta x}$
$\langle 1 \rangle$	Δx
$\langle \phi_i \rangle$	$\frac{\Delta x}{2}$
$\langle \phi_i \phi_j \rangle$ $i \neq j$	$\frac{\Delta x}{6}$
$\langle \phi_i \phi_i \rangle$	$\frac{\Delta x}{3}$
$\langle \phi_i^2 \phi_j \rangle$ $i \neq j$	$\frac{\Delta x}{12}$
$\langle \phi_i^3 \rangle$	$\frac{\Delta x}{4}$
$\langle a(x)\frac{d\phi_1}{dx} \rangle$	$-\frac{1}{\Delta x}\langle a(x) \rangle$
$\langle a(x)\frac{d\phi_2}{dx} \rangle$	$\frac{1}{\Delta x}\langle a(x) \rangle$

Table 8.6: Integrals of ϕ and its derivatives for linear elements.

element I. Table 8.6 gives exact integrals of the various quantities needed, over a single element. Direct evaluation of our case gives

$$A_{11} = -\int_I \frac{d\phi_1}{dx}\frac{d\phi_1}{dx}\,dx + \int_I f\phi_1\phi_1\,dx = \Delta x_I\left[-\left(\frac{1}{\Delta x_I}\right)^2 + \frac{f_I}{3}\right] \tag{8.24}$$

$$A_{12} = -\int_I \frac{d\phi_1}{dx}\frac{d\phi_2}{dx}\,dx + \int_I f\phi_1\phi_2\,dx = \Delta x_I\left[\left(\frac{1}{\Delta x_I}\right)^2 + \frac{f_I}{6}\right] \tag{8.25}$$

and the right-hand side is

$$b_1 = \int_I g\phi_1\,dx - \left(\frac{du}{dx}\phi_1\right)\Bigg|_{x=0}^{x=L} = \frac{\Delta x_I}{2}g_I + \frac{du}{dx}(x=0) \tag{8.26}$$

(The boundary contribution at $x = L$ has vanished because $\phi_1 = 0$ there; and ϕ_1 is unity at $x = 0$, its home node.) We may now assemble the first row of the matrix equation, WR_1:

$$\left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_I u_1 + \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_I u_2 = \left(\frac{g\Delta x}{2}\right)_I + \frac{du}{dx}(x=0) \tag{8.27}$$

Row 2 is $\int r, \phi_2\,dx$. Since ϕ_2 is supported in both elements, we need to integrate over both. In element I we have contributions to A_{21} and A_{22}:

$$A_{21}^I = -\int_I \frac{d\phi_2}{dx}\frac{d\phi_1}{dx}\,dx + \int_I f\phi_2\phi_1\,dx = \Delta x_I\left[\left(\frac{1}{\Delta x_I}\right)^2 + \frac{f_I}{6}\right] \tag{8.28}$$

$$A_{22}^I = -\int_I \frac{d\phi_2}{dx}\frac{d\phi_2}{dx}\,dx + \int_I f\phi_2\phi_1\,dx = \Delta x_I\left[-\left(\frac{1}{\Delta x_I}\right)^2 + \frac{f_I}{3}\right] \tag{8.29}$$

(The superscript indicates a contribution to the integration from one element.) The right-hand side contribution to this element is similarly obtained:

$$b_2^I = \int_I g\phi_2\,dx - \left(\frac{du}{dx}\phi_2\right)\Bigg|_{x=0}^{x=L} = \frac{\Delta x_I}{2}g_I \tag{8.30}$$

Note that both boundary contributions in WR_2 vanish because $\phi_2 = 0$ at both boundaries.

WR_2 is also supported on element II; by similar reasoning, we have contributions to A_{22}, A_{23}, and b_2:

$$A_{22}^{II} = -\int_{II} \frac{d\phi_2}{dx}\frac{d\phi_2}{dx}\,dx + \int_{II} f\phi_2\phi_2\,dx = \Delta x_{II}\left[-\left(\frac{1}{\Delta x_{II}}\right)^2 + \frac{f_{II}}{3}\right] \tag{8.31}$$

$$A_{23}^{II} = -\int_{II} \frac{d\phi_2}{dx}\frac{d\phi_3}{dx}\,dx + \int_{II} f\phi_2\phi_3\,dx = \Delta x_{II}\left[\left(\frac{1}{\Delta x_{II}}\right)^2 + \frac{f_{II}}{6}\right] \tag{8.32}$$

$$b_2^{II} = \int_{II} g\phi_2\,dx - \left(\frac{du}{dx}\phi_2\right)\Bigg|_{x=0}^{x=L} = \frac{\Delta x_{II}}{2}g_{II} \tag{8.33}$$

Accumulating all terms in WR_2 gives us

$$\left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_I u_1+\left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_I u_2+\left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_{II} u_2+\left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_{II} u_3 = \left[\frac{g\Delta x}{2}\right]_I+\left[\frac{g\Delta x}{2}\right]_{II} \tag{8.34}$$

Row 3 is analogous to row 1; its assembly as above gives us

$$\left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_{II} u_2+\left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_{II} u_3 = \left[\frac{g\Delta x_{II}}{2}\right]_{II}-\frac{du}{dx}(x=L) \tag{8.35}$$

The *matrix equation* (8.21) for this 2-element system may now be expressed in detail as

$$\begin{bmatrix} \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_I & \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_I & 0 \\ \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_I & \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_I+\left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_{II} & \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_{II} \\ 0 & \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right]_{II} & \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right]_{II} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$= \begin{Bmatrix} \left[\frac{g\Delta x}{2}\right]_I \\ \left[\frac{g\Delta x}{2}\right]_I+\left[\frac{g\Delta x}{2}\right]_{II} \\ \left[\frac{g\Delta x}{2}\right]_{II} \end{Bmatrix} + \begin{Bmatrix} \frac{du}{dx}(x=0) \\ 0 \\ -\frac{du}{dx}(x=L) \end{Bmatrix} \tag{8.36}$$

Before going further, it is useful to note that this discrete system contains recognizable (and therefore credible) finite difference expressions. Consider the interior equation, WR_2. If we introduce the average local element size, $\Delta x_i \equiv (\Delta x_I + \Delta x_{II})/2$, upon rearrangement we find

$$\frac{1}{\Delta x_i}\left[\frac{u_3-u_2}{\Delta x_{II}}-\frac{u_2-u_1}{\Delta x_I}\right] + \frac{1}{\Delta x_i}\left[\frac{\Delta x_{II} f_{II}\,(u_3+2u_2)}{6}+\frac{\Delta x_I f_I\,(2u_2+u_1)}{6}\right] = \frac{1}{\Delta x_i}\left[\frac{\Delta x_{II} g_{II}+\Delta x_I g_I}{2}\right] \tag{8.37}$$

When the mesh is uniform, $\Delta x_I = \Delta x_{II} = \Delta x$, the form is instantly recognized as a finite difference expression with averaging for the undifferentiated terms:

$$\frac{u_3-2u_2+u_1}{\Delta x^2}+\frac{u_3 f_{II}+u_2(2f_{II}+2f_I)+u_1 f_I}{6}=\frac{g_{II}+g_I}{2} \tag{8.38}$$

which is reminiscent of the comparable subdomain method example earlier in Chapter 7. In the case of constant f, the averaging of the term fu reduces to Simpsons Rule:

$$\frac{u_3-2u_2+u_1}{\Delta x^2}+\frac{f\,(u_3+4u_2+u_1)}{6}=\frac{g_{II}+g_I}{2} \tag{8.39}$$

Under these same restrictions, WR_1 becomes

$$\frac{u_2 - u_1}{\Delta x} = \frac{du}{dx}(x=0) + \frac{\Delta x}{2}\left[g_I - f\frac{(2u_1 + u_2)}{3}\right] \tag{8.40}$$

We recognize here a one-sided difference approximation to the Neumann boundary condition at $x = 0$, with additional terms proportional to Δx. In fact, these terms convert an otherwise $\bigcirc(\Delta x)$ approximation to one which is $\bigcirc(\Delta x^2)$. By the standard Taylor series expansion, we have

$$\frac{u_2 - u_1}{\Delta x} = \frac{du}{dx}(x=0) + \frac{\Delta x}{2}\frac{d^2u}{dx^2}(x=0) + \bigcirc(\Delta x^2) \tag{8.41}$$

For the governing Helmholtz equation, we have $\frac{d^2u}{dx^2} = g - fu$, and the Δx terms in (8.40) are clearly approximate this to first-order at $x = 0$. Hence (8.40) is an $O(\Delta x^2)$ approximation as stated. Analogous structure is evident in WR_3 for the boundary at $x = L$.

- It is evident that the general form (8.36) produces *difference equations on irregular meshes* with variable coefficients, systematically structured by the Method of Weighted Residuals.

8.6 Boundary Conditions

The discrete form (8.36) contains 3 equations in 5 unknowns u_1, u_2, u_3, and the two boundary slopes $\frac{du}{dx}_1$ and $\frac{du}{dx}_3$. Necessary and sufficient conditions for a unique solution require one condition at each of the boundaries, thereby closing the system. We classify these as follows:

- Type 1 (Dirichlet): u is specified;
- Type 2 (Neumann): $\frac{du}{dx}$ is specified: $\frac{du}{dx} = \beta$;
- Type 3 (Mixed): a blend of u and $\frac{du}{dx}$ is specified: $\frac{du}{dx} + \alpha u = \beta$.

Implementation of the Neumann condition is straightforward – the data β is inserted directly and naturally into the right-side vector of equation 8.36. The term "natural" boundary condition suits this situation perfectly.

For the Type 1 boundary condition, say at $x = 0$, we have the situation where u_1 is known *a priori*, but $\frac{du}{dx}$ is not. The practical approach is to solve WR_2 and WR_3 first, with u_1 given:

$$\begin{bmatrix} 1 & 0 & 0 \\ \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_I & \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_I + \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_{II} & \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_{II} \\ 0 & \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_{II} & \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_{II} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$= \begin{Bmatrix} 0 \\ \left[\frac{g\Delta x}{2}\right]_I + \left[\frac{g\Delta x}{2}\right]_{II} \\ \left[\frac{g\Delta x}{2}\right]_{II} \end{Bmatrix} + \begin{Bmatrix} u_1 \\ 0 \\ -\beta_3 \end{Bmatrix} \tag{8.42}$$

where we have assumed a Type 2 boundary condition at $x = L$. After this is solved, all values of u_i are known, and WR_1 can be invoked to recover $\frac{du}{dx}$ at $x = 0$:

$$\frac{du}{dx}(x=0) = \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_I u_1 + \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_I u_2 - \left[\frac{g\Delta x_I}{2}\right]_I \tag{8.43}$$

For a Type 3 boundary, say at $x = L$, we proceed initially as with a Type 2 condition by inserting the expression $\frac{du}{dx} = \beta - \alpha u_3$ in the right-hand side of WR_3:

$$\left[\frac{1}{\Delta x} + \frac{f\Delta x_{II}}{6}\right]_{II} u_2 + \left[-\frac{1}{\Delta x} + \frac{f\Delta x_{II}}{3}\right]_{II} u_3 = \left[\frac{g\Delta x_{II}}{2}\right]_{II} - \beta_3 + \alpha u_3 \tag{8.44}$$

The term αu_3 is unknown and must be moved to the left side of the equation:

$$\left[\frac{1}{\Delta x} + \frac{f\Delta x_{II}}{6}\right]_{II} u_2 + \left[-\frac{1}{\Delta x} + \frac{f\Delta x_{II}}{3} - \alpha\right]_{II} u_3 = \left[\frac{g\Delta x_{II}}{2}\right]_{II} - \beta_3 u_3 \tag{8.45}$$

The resulting matrix equation, for a Type 1 boundary at $x = 0$ and Type 3 at $x = L$, is:

$$\begin{bmatrix} 1 & 0 & 0 \\ \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_I & \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_I + \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_{II} & \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_{II} \\ 0 & \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_{II} & \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_{II} - \alpha \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$= \begin{Bmatrix} u_1 \\ \left(\frac{g\Delta x}{2}\right)_I + \left(\frac{g\Delta x}{2}\right)_{II} \\ \left(\frac{g\Delta x}{2}\right)_{II} \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ -\beta_3 \end{Bmatrix} \tag{8.46}$$

The case $\alpha = 0$ is the Type 2 case.

The general structure emerges in this simple system:

- The coefficient matrix A comprises domain integrals of basis functions and weighting functions, the particulars of which involve the particulars of the WR method, the discretization of the domain into elements, and the governing equation. This matrix is modified in cases of Type 1 or Type 3 boundaries.
- The vector of unknowns contains the unknown nodal values of the FEM system.
- A right-hand side vector containing inhomogeneous terms in the governing equation.
- An additional right-hand side vector containing boundary condition information.

Finally, consider a larger system with N nodes and $N - 1$ elements. The structure of WR_1 and WR_3 are unchanged, and on the interior we need only replicate the structure of WR_2. The result is easily generalized from (8.34):

$$\begin{aligned} &\left[\frac{1}{\Delta x} + \frac{\Delta x}{6}f\right]_{i-1} u_{i-1} + \left[-\frac{1}{\Delta x} + \frac{\Delta x}{3}f\right]_{i-1} u_i \\ &+ \left[-\frac{1}{\Delta x} + \frac{\Delta x}{3}f\right]_i u_i + \left[\frac{1}{\Delta x} + \frac{\Delta x}{6}f\right]_i u_{i+1} = \left[\frac{g\Delta x}{2}\right]_{i-1} + \left[\frac{g\Delta x}{2}\right]_i \end{aligned} \tag{8.47}$$

and the matrix will have a general tridiagonal structure:

$$\begin{bmatrix} A_{11} & A_{12} & 0 & \dots & \dots & \dots & 0 \\ A_{21} & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & A_{i,i-1} & A_{i,i} & A_{i,i+1} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & A_{N-1,N} \\ 0 & \dots & \dots & \dots & 0 & A_{N,N-1} & A_{N,N} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_{i-1} \\ u_i \\ u_{i+1} \\ \vdots \\ u_N \end{Bmatrix}$$

$$= \begin{Bmatrix} \left(\frac{g\Delta x}{2}\right)_I \\ \left(\frac{g\Delta x}{2}\right)_I + \left(\frac{g\Delta x}{2}\right)_{II} \\ \vdots \\ \left(\frac{g\Delta x}{2}\right)_{i-1} + \left(\frac{g\Delta x}{2}\right)_i \\ \vdots \\ \left(\frac{g\Delta x}{2}\right)_{N-2} + \left(\frac{g\Delta x}{2}\right)_{N-1} \\ \left(\frac{g\Delta x}{2}\right)_{N-1} \end{Bmatrix} + \begin{Bmatrix} \frac{du}{dx}(x=0) \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ -\frac{du}{dx}(x=L) \end{Bmatrix} \tag{8.48}$$

Implementation of boundary conditions for this larger system is unchanged from the 3 node case.

8.7 Element-Level Representation: the Element Matrix

It is clear that the above procedure for assembling the discrete system (8.36) can become arduous rapidly, and that a structured approach is necessary to succeed. But it is also clear that there is a natural structure in the matrix $[A]$ – one that we have sought to preserve in the notation. For example, the quantities $\frac{1}{\Delta x}$ and $\frac{f\Delta x}{6}$ occur repeatedly in exactly 2 different combinations and these are the only quantities needed to build $[A]$ in this case. We take advantage of this by introducing the notion of an element-level matrix whose contributions to the global system are easily assembled.

Imagine for the moment that the FE mesh included only the first element. Then the discrete system would be limited to only nodes 1 and 2, and to those integrals evaluated on element I:

$$\begin{bmatrix} \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right] & \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right] \\ \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right] & \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right] \end{bmatrix}_I \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$= \begin{Bmatrix} \left[\frac{g\Delta x}{2}\right] \\ \left[\frac{g\Delta x}{2}\right] \end{Bmatrix}_I + \begin{Bmatrix} \frac{du_1}{dx} \\ -\frac{du_2}{dx} \end{Bmatrix}_I \tag{8.49}$$

As before, the two rows of the equation are parts of WR_1 and WR_2, respectively. WR_1 is complete and identical to that described above; WR_2 here is only that part contributed from integration on element I.

Similarly, if only element II were present, then the discrete system would be limited to those contributions from II:

$$\begin{bmatrix} \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right] & \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right] \\ \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right] & \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right] \end{bmatrix}_{II} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix}$$

$$= \begin{Bmatrix} \left[\frac{q\Delta x}{2}\right] \\ \left[\frac{q\Delta x}{2}\right] \end{Bmatrix}_{II} + \begin{Bmatrix} \frac{du_2}{dx} \\ -\frac{du_3}{dx} \end{Bmatrix}_{II} \tag{8.50}$$

Either of these is the valid discrete system for its element in isolation. When both elements are present, the discrete system is obtained by adding the individual element representations. (Recall that each is an integral of the weighted residual; the addition is simply extending the domain of integration.) The addition of these two matrix equations gives (8.36) above, with the sole exception of the right-hand side vector involving the natural boundary conditions. Here we obtain the vector

$$\begin{Bmatrix} \frac{du_1}{dx} \\ -\frac{du_2}{dx} \\ 0 \end{Bmatrix}_{I} + \begin{Bmatrix} 0 \\ \frac{du_2}{dx} \\ -\frac{du_3}{dx} \end{Bmatrix}_{II} \tag{8.51}$$

Assuming continuity of the natural boundary condition $\frac{du}{dx}$ at the boundary between I and II, the term $\frac{du_2}{dx}_{II} - \frac{du_2}{dx}_{I}$ vanishes, and we have recovered the original 2-element discrete system by superposition of the two element-level WR representations. This superposition is valid for the general case of N elements.

Clearly, we are superposing identical representations of a single, generic element. If we introduce the local node numbering system $i = 1, 2$ with node 1 on the left of the element as in Figure 8.3, we can generalize the element contributions to the system in terms of an element matrix:

$$[A]_E = \begin{bmatrix} \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right] & \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right] \\ \left[\frac{1}{\Delta x}+\frac{f\Delta x}{6}\right] & \left[-\frac{1}{\Delta x}+\frac{f\Delta x}{3}\right] \end{bmatrix}_{E} \tag{8.52}$$

an element list of variables

$$\{u\}_E = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}_E \tag{8.53}$$

and a right-hand side contribution from E:

$$\{b\}_E = \begin{Bmatrix} \left[\frac{q\Delta x}{2}\right] \\ \left[\frac{q\Delta x}{2}\right] \end{Bmatrix}_E + \begin{Bmatrix} \frac{du_1}{dx} \\ -\frac{du_2}{dx} \end{Bmatrix}_E \tag{8.54}$$

The discrete system for a single isolated element is

$$[A]_E \{u\}_E = \{b\}_E \tag{8.55}$$

This element-level representation would be the complete discrete system if there were only one element. It embodies the differential equation, the particular weighted residual method, and the choice of weighting and basis functions. *The complete discrete system is just the summation of the element systems:*

$$\sum_E [A]_E \{u\}_E = \sum_E \{b\}_E \tag{8.56}$$

In our example case, $\frac{d^2u}{dx^2} + fu = g$ using Galerkin, the general form is exactly equation (8.21), evaluated on a single element:

$$[A]_E = \left[-\int_E \left(\frac{d\phi_j}{dx}\frac{d\phi_i}{dx} + f\phi_j\phi_i \right) dx \right]$$

$$\{b\}_E = \left\{ \int_E g\phi_i \, dx \right\} - \left\{ \frac{du}{dx}\phi_i \right\} \Bigg|_{x_l}^{x_r} \tag{8.57}$$

(The subscripts r, l indicate the right and left endpoints of an element). Recall the sense of the matrix equation: the indices i, j range over those ϕ which are supported on a given element; and each row is WR_i, integrated over the element.

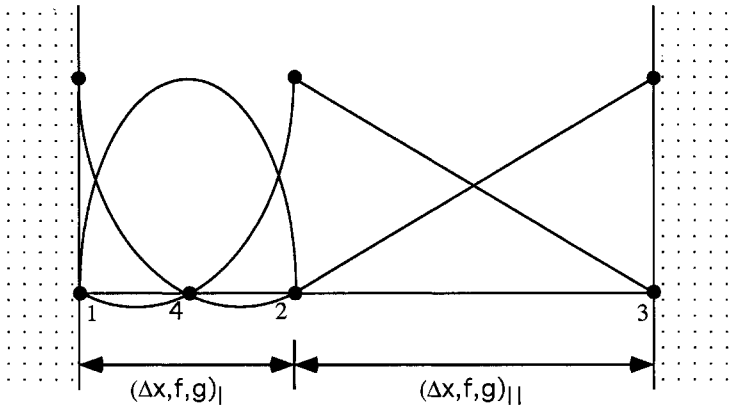


Figure 8.8: Modification of 2-element mesh (Figure 8.7 for quadratic variation in element I. Node 4 is at the element center.

Suppose we were to enhance the 2-element discretization of the previous example by enriching the basis in element I from linear to quadratic, as shown in Figure 8.8. The element matrix is formally the same as (8.57) above; but with three bases supported in element I, the system will be 3×3. Table 8.7 gives all necessary integrals for direct evaluation; the result is:

$$\begin{bmatrix} \left[-\frac{7}{3\Delta x} + \frac{2f\Delta x}{15}\right] & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right] & \left[-\frac{1}{3\Delta x} - \frac{f\Delta x}{30}\right] \\ \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right] & \left[-\frac{16}{3\Delta x} + \frac{8f\Delta x}{15}\right] & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right] \\ \left[-\frac{1}{3\Delta x} - \frac{f\Delta x}{30}\right] & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right] & \left[-\frac{7}{3\Delta x} + \frac{2f\Delta x}{15}\right] \end{bmatrix}_I \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}_I$$

$$= \left\{ \begin{array}{c} \left[\frac{g\Delta x}{6}\right] \\ \left[\frac{4g\Delta x}{6}\right] \\ \left[\frac{g\Delta x}{6}\right] \end{array} \right\}_I + \left\{ \begin{array}{c} \frac{du_1}{dx} \\ 0 \\ -\frac{du_3}{dx} \end{array} \right\}_I \tag{8.58}$$

where the subscripting indicates local node numbers as in Table 8.7.

Element II is unchanged by this mesh upgrade, and therefore its contribution to the discrete system is unchanged from (8.50). All that is left is to add the two discrete systems together.

Now the insertion of the information from element I requires a little care. The local-to-global node correspondence is

$$\{1,\ 2,\ 3\}_I \Leftrightarrow \{1,\ 4,\ 2\} \tag{8.59}$$

Therefore we expect all rows and columns of $[A]_I$ and $\{b\}_I$ to map accordingly into the 4×4 global versions. The contributions of element I alone to the complete system are therefore

$$\left[\begin{array}{cccc} \left[-\frac{7}{3\Delta x} + \frac{2f\Delta x}{15}\right]_I & \left[-\frac{1}{3\Delta x} - \frac{f\Delta x}{30}\right]_I & 0 & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I \\ \left[-\frac{1}{3\Delta x} - \frac{f\Delta x}{30}\right]_I & \left[-\frac{7}{3\Delta x} + \frac{2f\Delta x}{15}\right]_I & 0 & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I \\ 0 & 0 & 0 & 0 \\ \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I & 0 & \left[-\frac{16}{3\Delta x} + \frac{8f\Delta x}{15}\right]_I \end{array} \right] \left\{ \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \right\}$$

$$= \left\{ \begin{array}{c} \left[\frac{g\Delta x}{6}\right]_I \\ \left[\frac{g\Delta x}{6}\right]_I \\ 0 \\ \left[\frac{4g\Delta x}{6}\right]_I \end{array} \right\} + \left\{ \begin{array}{c} \left[\frac{du_1}{dx}\right]_I \\ -\left[\frac{du_2}{dx}\right]_I \\ 0 \\ 0 \end{array} \right\} \tag{8.60}$$

where all node indices are global. Addition of element II's contributions is achieved with the same local-to-global conversion as in the earlier example:

$$\{1,\ 2\}_{II} \Leftrightarrow \{2,\ 3\} \tag{8.61}$$

and the complete discrete system for the two elements is

$$\left[\begin{array}{cccc} \left[-\frac{7}{3\Delta x} + \frac{2f\Delta x}{15}\right]_I & \left[-\frac{1}{3\Delta x} - \frac{f\Delta x}{30}\right]_I & 0 & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I \\ \left[-\frac{1}{3\Delta x} - \frac{f\Delta x}{30}\right]_I & \left[-\frac{7}{3\Delta x} + \frac{2f\Delta x}{15}\right]_I + \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_{II} & \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_{II} & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I \\ 0 & + \left[\frac{1}{\Delta x} + \frac{f\Delta x}{6}\right]_{II} & \left[-\frac{1}{\Delta x} + \frac{f\Delta x}{3}\right]_{II} & 0 \\ \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I & \left[\frac{8}{3\Delta x} + \frac{f\Delta x}{15}\right]_I & 0 & \left[-\frac{16}{3\Delta x} + \frac{8f\Delta x}{15}\right]_I \end{array} \right] \left\{ \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \right\}$$

	$j=1$	$j=2$	$j=3$
$\langle\phi_j\rangle$	$\frac{h}{3}$	$\frac{4h}{3}$	$\frac{h}{3}$
$\langle\phi_1\phi_j\rangle$	$\frac{4h}{15}$	$\frac{2h}{15}$	$-\frac{h}{15}$
$\langle\phi_2\phi_j\rangle$	$\frac{2h}{15}$	$\frac{16h}{15}$	$\frac{2h}{15}$
$\langle\phi_3\phi_j\rangle$	$-\frac{h}{15}$	$\frac{2h}{15}$	$\frac{4h}{15}$
$\langle\frac{d\phi_j}{dx}\rangle$	-1	0	1
$\langle\phi_1\frac{d\phi_j}{dx}\rangle$	$-\frac{3}{6}$	$\frac{4}{6}$	$-\frac{1}{6}$
$\langle\phi_2\frac{d\phi_j}{dx}\rangle$	$-\frac{4}{6}$	0	$\frac{4}{6}$
$\langle\phi_3\frac{d\phi_j}{dx}\rangle$	$\frac{1}{6}$	$-\frac{4}{6}$	$\frac{3}{6}$
$\langle\frac{d\phi_1}{dx}\frac{d\phi_j}{dx}\rangle$	$\frac{7}{6h}$	$-\frac{8}{6h}$	$\frac{1}{6h}$
$\langle\frac{d\phi_2}{dx}\frac{d\phi_j}{dx}\rangle$	$-\frac{8}{6h}$	$\frac{16}{6h}$	$-\frac{8}{6h}$
$\langle\frac{d\phi_3}{dx}\frac{d\phi_j}{dx}\rangle$	$\frac{1}{6h}$	$-\frac{8}{6h}$	$\frac{7}{6h}$

Table 8.7: Integrals of ϕ and its derivatives for 1-D quadratic elements (mid-element node centered). $h = \frac{\Delta x}{2}$ is the spacing between nodes; Δx is the element length; $\langle 1\rangle = 2h$.

$$= \begin{Bmatrix} \left[\frac{g\Delta x}{6}\right]_I \\ \left[\frac{g\Delta x}{6}\right]_I + \left[\frac{g\Delta x}{2}\right]_{II} \\ \left[\frac{g\Delta x}{2}\right]_{II} \\ \left[\frac{4g\Delta x}{6}\right]_I \end{Bmatrix} + \begin{Bmatrix} \left[\frac{du_1}{dx}\right]_I \\ -\left[\frac{du_2}{dx}\right]_I + \left[\frac{du_2}{dx}\right]_{II} \\ -\left[\frac{du_3}{dx}\right]_{II} \\ 0 \end{Bmatrix} \tag{8.62}$$

and as before, continuity of $\frac{du_2}{dx}$ will cause the *a priori* cancellation of those terms.

8.8 Assembly of Element Systems: the FE Incidence List

The above procedure for adding together the Discrete Subsystems from each element can be formalized. The essential idea is a generalization of the local-to-global mapping used above at equations (8.59) and (8.61). We define the finite element Incidence List, IN, as the map linking local and global node numbers:

$$i \Leftrightarrow IN(E, i) \tag{8.63}$$

where

- E is the element number,
- i is the local node number in element E, and
- IN is the corresponding global node number.

Note that the natural view here is from the local or element perspective, looking outward towards the full system representation.

Using this notation, we may readily construct the global contributions to, say, matrix $[A]$ in the form

$$A_E(i,j) \Rightarrow A(IN(E,i),\, IN(E,j)) \tag{8.64}$$

$$b_E(i) \Rightarrow b(IN(E,i)) \tag{8.65}$$

Here the symbol $\Rightarrow$ indicates "contributes to" additively. $x \Rightarrow y$ indicates "x contributes to y".

Table 8.8: Incidence List for the 1-D mesh in Figure 8.9.

E	$IN(E, 1)$	$IN(E, 2)$	$IN(E, 3)$
1	4	7	2
2	2	5	6
3	1	3	4

As an example, consider the 1-D quadratic mesh shown in Figure 8.9, where the elements and nodes are purposefully given in unnatural order. Using the local node numbering convention given in Table 8.7, we have the Incidence List shown in Table 8.8. Each element will contribute a 3×3 subsystem of the form (8.58). The contribution of element 1 to the global discrete system will be

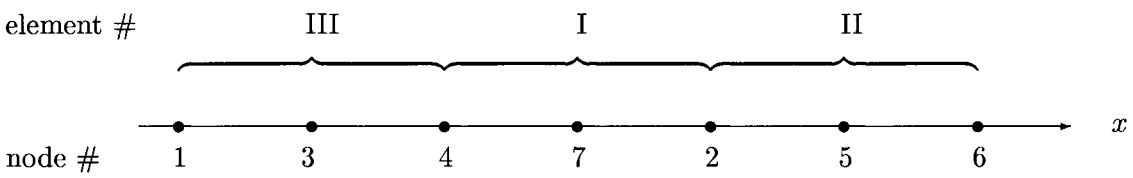


Figure 8.9: Three-element mesh of quadratics.

as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_I(3,3) & 0 & A_I(3,1) & 0 & 0 & A_I(3,2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_I(1,3) & 0 & A_I(1,1) & 0 & 0 & A_I(1,2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_I(2,3) & 0 & A_I(2,1) & 0 & 0 & A_I(2,2) \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_7 \end{Bmatrix} = \begin{Bmatrix} 0 \\ b_I(3) \\ 0 \\ b_I(1) \\ 0 \\ 0 \\ b_I(2) \end{Bmatrix} \tag{8.66}$$

and this structure will be valid for any differential equation. From a computational or procedural perspective, if we start with empty arrays, then we may simply visit each element in turn, evaluate the local subsystem of equations, and add it via the incidence list to the global system as above. The above would be the result after evaluating element I; adding element II to the list, we would have

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_I(3,3)+A_{II}(1,1) & 0 & A_I(3,1) & A_{II}(1,2) & A_{II}(1,3) & A_I(3,2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_I(1,3) & 0 & A_I(1,1) & 0 & 0 & A_I(1,2) \\ 0 & A_{II}(2,1) & 0 & 0 & A_{II}(2,2) & A_{II}(2,2) & 0 \\ 0 & A_{II}(3,1) & A_{II}(3,2) & A_{II}(3,3) & 0 & 0 & 0 \\ 0 & A_I(2,3) & 0 & A_I(2,1) & 0 & 0 & A_I(2,2) \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_7 \end{Bmatrix} = \begin{bmatrix} 0 \\ b_I(3)+b_{II}(1) \\ 0 \\ b_I(1) \\ b_{II}(2) \\ b_{II}(3) \\ b_I(2) \end{bmatrix} \tag{8.67}$$

and so on. This generic process is referred to as the *Assembly* process. It achieves the assembly of the complete discrete system by systematically evaluating and linking (summing) its individual subsystems for each element.

8.9 Matrix Structure

The assembly process highlights a basic structure of any FE matrix system: it is *sparse*, *i.e.* the matrix comprises mostly zeros, with only a handful of nonzero entries in any row or column. Clearly, nonzero coefficients occur only when created in at least one element matrix; and therefore a coefficient A_{ij} will remain zero unless nodes i and j co-occur in at least one element. We define a "neighboring node" relationship: j and i are neighbors if nodes j and i co-occur in at least one element.

Sparsity results directly for the use of locally-supported bases. Since A_{ij} involves the product of two locally-supported bases (and/or their derivatives), then it vanishes for most i, j combinations. We define a "neighbor" relationship among the bases: j and i are neighbors if ϕ_j and ϕ_i are co-supported on at least one element. For C^0 elements, this is equivalent to the statement about neighboring nodes.

The sparse structure is predictable *a priori* from the incidence list. We need only examine this list element-by-element, noting all i, j combinations which occur any each element – these will be the locations of nonzero coefficients, *for any differential equation.* For example, consider the 3-element mesh of Table 8.8. Element I provides us with all combinations of 4, 7, and 2. For element II, combinations of 2, 5, and 6; and element III, 1, 3,and 4. The union of all these nonzero locations is the logical structure:

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{8.68}$$

with 1 indicating a nonzero coefficient. This structure is clearly emergent in the assembly example above, equation (8.67).

Let's define some sparse matrix measures:

- N is the number of nodes
- N_s is the average number of nonzero entries per row
- $s = N/N_s$ is the matrix sparsity
- $d = N_s/N$ is the matrix density

The number of nonzero coefficients is $N_s N = dN^2 = N^2/s$.

Clearly, N_s is a consequence of the average number of neighbors in the mesh. This number will be generally unaffected by the algebraic size of the overall system. Therefore, increasing the number of elements (either by enlarging the domain or refining its discretization) increases the sparsity (decreases the density). So, if we are clever, we can achieve FE algorithms whose operations counts are reduced from general, full matrix requirements (N^2 or N^3) by factors of $1/s$ or $1/s^2$ (d or d^2). Equivalently, N^2 or N^3 becomes N_s or $N_s^2 N$. With N_s constant, such an algorithm would scale linearly with N.

For example, consider a mesh of 1-D quadratic elements, The mid-element bases ϕ_i have support in only 1 element, and therefore have only 2 neighbors. Including the diagonal, then, there will be exactly 3 nonzero entries in row i. Row i for these functions will have exactly 3 nonzero entries. An endpoint basis ϕ_i is supported in 2 elements, with a total of 4 neighbors. There will be exactly 5 nonzero entries in row i. (This may be truncated to 3 nonzero entries if node i is a boundary node.) This structure is independent of the number of elements; and so we have for this mesh, $N_s = 4$.

In Figure 8.10 we introduce another sparse matrix measure, the bandwidth. We define the half-bandwidth N_h as the maximum "distance" between the diagonal and the last nonzero entry in

any row of A. The bandwidth $N_b = 2N_h + 1$ is the maximum width of the nonzero entries. (These measures include the intervening zero entries). There exist very effective direct (noniterative) solution techniques for the matrix equation $Ax = b$ when A is banded. These methods are based on LU decomposition, wherein all intermediate calculations fit within (and fill) the bandwidth of the matrix. So while direct inversion of a full matrix requires $O(N^3)$ operations, the banded solvers can achieve this in $O(N_b^2 N)$ operations; and the storage required reduces from $O(N^2)$ entries to $O(N_b N)$.

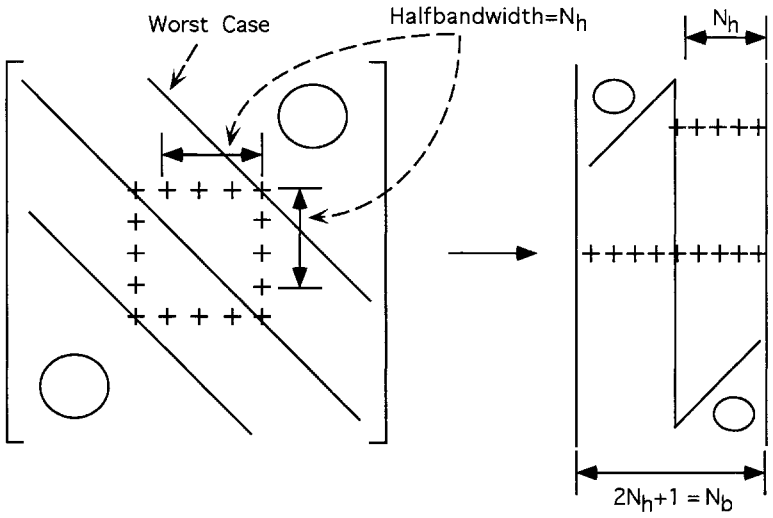


Figure 8.10: Half-bandwidth, bandwidth, and general storage modes. The row index is preserved; the column index is shifted.

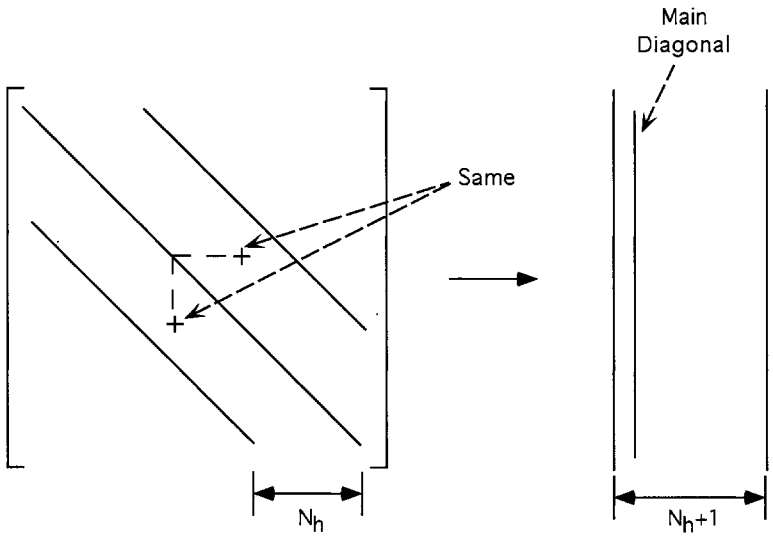


Figure 8.11: Symmetric banded storage mode.

The half-bandwidth is easily computed from the incidence list – it is the greatest difference in node numbers occurring in any element. For the mesh of Figure 8.9, $N_h = 5$. N_h is clearly sensitive to the detailed node numbering. A simple rearrangement of the node numbers to the natural order (increasing monotonically from the left) reduces N_h to 3. This simple rearrangement would result

in significant savings in both memory and run time if direct matrix solution via LU decomposition were used. Two practical features are noteworthy:

- The use of LU decomposition for 2-D FE problems is widespread and powerfully efficient.
- Simple mesh editing - for example insertion of a new node into an established mesh – can have a disastrous impact on bandwidth, as the latest added node $(N+1)$ becomes a neighbor, potentially, to node 1.

As a result, bandwidth reduction algorithms are of immense practical importance in practical 2-D FE work. These renumber the nodes of an existing mesh with bandwidth minimization as the objective.

These observations about bandwidth and LU decomposition fade in importance for 3-D FE work, because the minimum possible bandwidth on realistic meshes becomes unacceptably large for direct solution methods. For example, a uniform discretization of a cube with M nodes on a side has a minimum half-bandwidth of approximately M^2. So the banded storage required would be M^5 and the runtime would be of order M^7. For reasonable discretization levels – say, $M = 50$ – the storage requirement alone is prohibitive ($50^5 \approx 3 \times 10^8 words$)[1] in many practical environments. The comparable effect in 2-D is however quite practical: $N_h = M$, storage M^3, runtime M^4.

8.10 Variable Coefficients

In the above example we have simplified things by assuming the coefficients for the Helmholtz equation f, g were adequately represented by constants on each element. In that case, integrals of the form $\int_0^L f\phi_j\phi_i\,dx$ are especially simple at the element level:

$$\int_E f\phi_j\phi_i\,dx = f_E \int_E \phi_j\phi_i\,dx \tag{8.69}$$

Effectively, the variation of the coefficients is portrayed as piecewise constant function with discontinuities at element boundaries, as in Figure 8.12.

More generally, we may expand the coefficients in terms of any known basis:

$$f(x) = \sum_k f_k\psi_k(x) \tag{8.70}$$

The most common case is to use the same basis for the coefficients as is used for the solution itself *i.e.* $\psi = \phi$. In many FE descriptions, this is tacitly assumed unless stated otherwise. In Figure 8.12 we display this case. Data support for this case consists of nodal values of the coefficient, as opposed to element values as above. Once the basis for the coefficients is established, the WR method provides an unambiguous statement of how this coefficient variation is embedded in the discrete system – *e.g.* the integral (8.69) above.

These two types of coefficient variation both have intrinsic strengths and weaknesses. In cases where an essential discontinuity in a coefficient occurs, then a discontinuous, element-based representation is natural, and the mesh needs to be designed with node placement at the discontinuity. In cases where smooth coefficient variation is desirable, then the C^0 or higher continuity is appropriate, supported by nodal data. Hybrid strategies for specifying coefficient variation are of course possible.

[1] 2.4 Gigabytes for 64-bit words

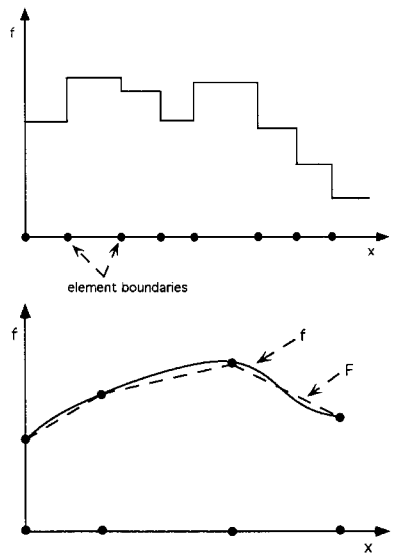


Figure 8.12: Smooth function $f(x)$ represented as (top) a piecewise constant function, supported by element values; and (bottom) represented as a C^o function $F(x)$, supported by nodal values on linear elements.

8.11 Numerical Integration

In the above examples we have integrated the various terms exactly in order to expose the structure of the difference equations which result. This procedure becomes tedious rapidly as complexity is added to either the basis or the governing differential equation, and it is therefore rarely used in practice. The more general and easily automated approach is to perform the integrations numerically – *i.e.* to sample the integrand at discrete points and construct a weighted sum which approximates the integral:

$$\int_e f(x)\,dx \approx \sum_{k=1}^{n} W_k f(x_k) \tag{8.71}$$

The x_k are called quadrature points, the W_k weights. If the (x_k, W_k) pairs are carefully selected, the approximation can be exact for certain types of integrands.

For example, n quadrature points can always exactly integrate a polynomial of order $n-1$, provided the weights are chosen to match the points. It is easy to verify that, for an arbitrary selection of x_k, the weights W_k are given by

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \dots & x_n^{n-1} \end{bmatrix} \begin{Bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \\ W_n \end{Bmatrix} = \begin{Bmatrix} \int_e dx \\ \int_e x\,dx \\ \int_e x^2\,dx \\ \vdots \\ \int_e x^{n-1}\,dx \end{Bmatrix} \tag{8.72}$$

We can do better if the x_k are also carefully chosen to fit the integrand. The Gauss family of quadrature formulas provides the highest level of precision available for polynomial integrands: n quadrature points can exactly integrate a polynomial of order $(2n-1)$. This is achieved by locating

Table 8.9: Gauss-Legendre quadrature. The order of polynomial interpolation is $2n-1$. ξ is the normalized independent variable on the interval $[-1,1]$.

n	ξ_k	W_k
2	± 0.57735027	1
3	0 ±0.77459667	0.88888889 0.55555556
4	±0.33998104 ±0.86113631	0.65214515 0.34785485
5	0 ±0.53846931 ±0.90617985	0.56888889 0.47862867 0.23692689
6	±0.23861919 ±0.66120939 ±0.93246951	0.46791393 0.36076157 0.17132449
7	0 ±0.40584515 ±0.74153119 ±0.94910791	0.41795918 0.38183005 0.27970539 0.1294849662
8	±0.18343464 ±0.52553241 ±0.79666648 ±0.9602898565	0.3626837834 0.31370665 0.22238103 0.10122854

the x_k at the zeros of the n^{th} member of a family of orthogonal functions. Table 8.9 displays the (x_k, W_k) pairs for the Legendre polynomials, which provides integration on the interval $(-1, 1)$ with all quadrature points on the interior of the interval. In computational practice, these can be generated to any desired level of significant Figures using standard algorithms.

This family is widely used in FEM analysis, and is commonly referred to as "Gauss-Legendre Quadrature." To make it work in our context, we need one additional adjustment – the conversion from the global integration over dx to the generic integration over $d\xi$ requires the mapping transformation $J \equiv dx/d\xi$ to relate the x and ξ domains:

$$\int_e f(x)\,dx = \int_e f(x)\frac{dx}{d\xi}dx \approx \sum_{k=1}^{n} W_k f\left(x_k\right) J\left(x_k\right) \tag{8.73}$$

In the simple cases discussed above, the constant $J = (\Delta x)_e/2$ maps x onto the interval $(-1, 1)$ as required by the Gauss-Legendre formulas. For more complex elements, we will find that J will be a function of ξ. (This will be discussed in a later chapter.)

Note the general weakness in the quadrature approach: the integrand is assumed to be a polynomial; and for smooth integrands, increasing the order of numerical integration (increasing the number of quadrature points) will generally improve the accuracy of the integration. However, all bets are off if the integrand has a discontinuity or a singularity; these integrals will require specialized care.

The literature on numerical integration is vast and its findings are well-articulated in most introductory numerical analysis texts (*e.g.* [20]). The general family of Gauss-type formulas extends beyond polynomial integration, generalizing to all sets of orthogonal bases. The Discrete Fourier Transform is a common example of a quadrature approximation which is exact for Fourier bases up to a limiting wavenumber.

8.12 Assembly with Numerical Quadrature

Systematic use of numerical quadrature adds an additional layer of generic structure and simplicity to the assembly process. Above in Section 8.8 we noted that for the general discrete system $[A]\{u\} = \{B\}$, with A and B comprising integrals:

$$A_{ij} = \int a_{ij}\,dx \tag{8.74}$$

$$B_i = \int b_i\,dx \tag{8.75}$$

then the integrals can be assembled as a sum of element contributions:

$$[A] = \sum_E [A]_E \tag{8.76}$$

$$\{B\} = \sum_E \{B\}_E \tag{8.77}$$

We now find an additional level of structure: $[A]_E$ is itself a sum of contributions from quadrature points within that element:

$$[A] = \sum_E [A]_E = \sum_E \sum_k [A]_k = \sum_E \sum_k W_k\,[a]_k\,J_k \tag{8.78}$$

$$\{B\} = \sum_E \{B\}_E = \sum_E \sum_k \{B\}_k = \sum_E \sum_k W_k \{b\}_k J_k \tag{8.79}$$

We identify $[a]$, the "Gauss Point Matrix", and its right-hand side equivalent $\{b\}$, as the essential contributors to the discrete system, originating at the lowest (most local) level: the individual quadrature point.

The basic FEM requirement is to numerically evaluate *integrands* generated by the WR method *at a single quadrature point*, in a generic element with simple local basis functions. The balance of the work in obtaining the discrete system is generic, structured assembly.

The "view from the inside" begins at the Gauss point k and follows its contribution to the global system, as in (8.65):

$$a(i,\, j)_k J_k W_k \Rightarrow \quad A_E(i,j) \quad \Rightarrow A\,(IN(E,i),\, IN(E,j)) \tag{8.80}$$

$$b(i)_k J_k W_k \Rightarrow \quad b_E(i) \quad \Rightarrow B\,(IN(E,i)) \tag{8.81}$$

The intermediate concept of the Element matrix is not actually needed numerically, and we may write the assembly in its most direct form:

$$a(i,\, j)_k J_k W_k \quad \Rightarrow \quad A\,(IN(E,i),\, IN(E,j)) \tag{8.82}$$

$$b(i)_k J_k W_k \quad \Rightarrow \quad B\,(IN(E,i)) \tag{8.83}$$

where all indices are at the local (element) level, and as above, $\Rightarrow$ indicates "contributes to" additively. Note that a and b are *integrands*, not integrals, as generated by the WR method. The assembly indicated takes care of everything from that point on.

Chapter 9

Multi-Dimensional Elements

9.1 Linear Triangular Elements

Linear triangles are the simplest 2-D elements, providing entry-level linear interpolation of the form

$$f(x) = a + bx + cy \tag{9.1}$$

They are a universal point of departure for 2-D applications.

Local Interpolation

In Figure 9.1 we show the basic triangular element in the (x, y) plane. Its vertices are numbered locally, $l = (1, 2, 3)$ in *counterclockwise order* by convention and have global coordinates (x_i, y_i). We assume no restrictions on the shape of the triangle.

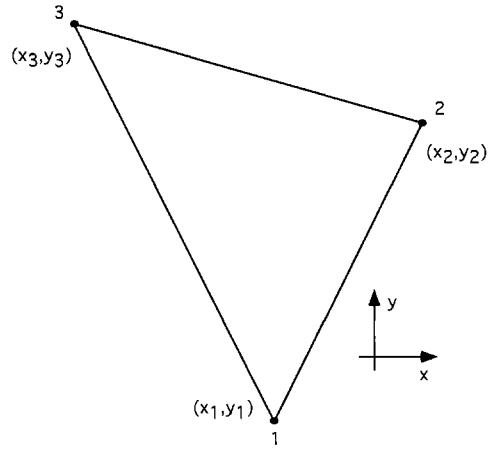


Figure 9.1: Linear triangular element with counterclockwise node numbering convention.

The triangle area is the basic measure of its size:

$$A = \frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$$= \frac{1}{2}[x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)]$$

$$= -\frac{1}{2}[y_1(x_2 - x_3) + y_2(x_3 - x_1) + y_3(x_1 - x_2)] \tag{9.2}$$

It is natural to define the element Cartesian lengths

$$\Delta x_1 = x_2 - x_3 \qquad \Delta y_1 = y_2 - y_3 \tag{9.3}$$

$$\Delta x_2 = x_3 - x_1 \qquad \Delta y_2 = y_3 - y_1 \tag{9.4}$$

$$\Delta x_3 = x_1 - x_2 \qquad \Delta y_3 = y_1 - y_2 \tag{9.5}$$

and the area is more simply expressed as

$$A = \frac{1}{2}\sum_1^3 x_l \, \Delta y_l = -\frac{1}{2}\sum_1^3 y_l \, \Delta x_l \tag{9.6}$$

The relevant "grid size" measures here are not aligned with the Cartesian grid, but with the triangle itself. The internodal spacing

$$\Delta s_l = \sqrt{\Delta x_l^2 + \Delta y_l^2} \tag{9.7}$$

provide one measure of local discretization; each has an orthogonal length equal to the altitude H_l defined in Figure 9.2

$$H_l = \frac{2A}{\Delta s_l} \tag{9.8}$$

Other measures of grid size are useful: for example $\sqrt{2A}$.

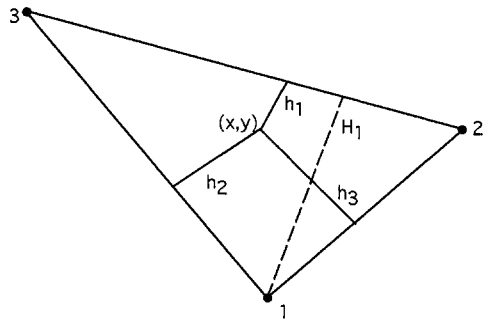


Figure 9.2: Normalized altitude coordinates $L_i = \frac{h_i(x,y)}{H_i}$.

On the triangle we define three basis functions ϕ_l which are linear in (x, y):

$$\phi_l(x, y) = \alpha_l + \beta_l x + \gamma_l y \tag{9.9}$$

etc. Each basis function ϕ_l is constrained to have unit value at node l and to vanish at the other nodes. These three constraints determine the constants α_l, β_l, γ_l. For ϕ_1, we have

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \tag{9.10}$$

Inversion of 9.10 gives

$$\phi_1 = \frac{[(x_2 y_3 - x_3 y_2) + x\Delta y_1 - y\Delta x_1]}{2A} \tag{9.11}$$

Cyclic permutation gives the other two bases

$$\phi_2 = \frac{[(x_3y_1 - x_1y_3) + x\Delta y_2 - y\Delta x_2]}{2A} \tag{9.12}$$

$$\phi_3 = \frac{[(x_1y_2 - x_2y_1) + x\Delta y_3 - y\Delta x_3]}{2A} \tag{9.13}$$

These functions provide a unique linear interpolation for any function f in terms of its 3 nodal values f_i:

$$f(x, y) = \sum_1^3 f_l\phi_l(x, y) \tag{9.14}$$

From the global perspective, $i = IN(E, l)$ is identical to local node number l on element E, and therefore the local function ϕ_l is a part of a global function ϕ_i. ϕ_i will be defined analogously on all elements which share node i. Otherwise, $\phi_i = 0$ by definition. ϕ_i is therefore a global function (*i.e.* defined over the entire domain) but with very limited, local support in the immediate 2-D neighborhood of node i. The definitions guarantee that ϕ_i will be continuous across adjacent element boundaries, and therefore any function interpolated "globally" (*i.e.* beyond a single element) as

$$f(x, y) = \sum_{i=1}^N f_i\phi_i(x, y) \tag{9.15}$$

will be continuous and piecewise linear in (x, y).

Differentiation

Derivatives of these basis functions with respect to (x, y) are readily obtained from (9.11):

$$\frac{\partial\phi_l}{\partial x} = \frac{\Delta y_l}{2A} \tag{9.16}$$

$$\frac{\partial\phi_l}{\partial y} = -\frac{\Delta x_l}{2A} \tag{9.17}$$

It is apparent that derivatives will be constants over each element, and therefore will be discontinuous at element boundaries. The same properties will pertain to derivatives of any function expanded in the basis – for example with f defined as in (9.14) above, we have for any single element

$$\frac{\partial f}{\partial x} = \sum_1^3 f_l\frac{\partial\phi_l}{\partial x} = \sum_1^3 f_l\frac{\Delta y_l}{2A}$$

$$\frac{\partial f}{\partial y} = \sum_1^3 f_l\frac{\partial\phi_l}{\partial y} = -\sum_1^3 f_l\frac{\Delta x_l}{2A} \tag{9.18}$$

The global representations are

$$\frac{\partial f}{\partial x} = \sum_{i=1}^{N_n} f_i\frac{\partial\phi_i}{\partial x}$$

$$\frac{\partial f}{\partial y} = \sum_{i=1}^{N_n} f_i\frac{\partial\phi_i}{\partial y} \tag{9.19}$$

Therefore we have a 2-D, C^0 linear element.

Integration

As introduced above, all FEM quantities are integrals; and the fundamental integration task is limited to performing integrals of the basis functions and the data over a single triangular element.

The linear triangles are unusual in that many applications can be created with exact, closed-form integration. The Table 9.1 gives the results for several common integrations.

In general, element integrals are evaluated numerically using approximate quadrature rules defined on triangular domains. These are given in the next section in connection with higher-order triangular bases.

9.2 Example: Helmholtz Equation on Linear Triangles

As an example, consider the Galerkin treatment of the Helmholtz equation $\nabla^2 u + fu = g$. In section 7.3 we introduced the weak form at equation(7.40); in Galerkin form, it is

$$-\langle \nabla u, \nabla \phi_i \rangle + \langle fu, \phi_i \rangle = \langle g, W_i \rangle \doteq \oint \mathbf{n} \cdot W_i \nabla u \, ds \tag{9.20}$$

Its discrete form $[A]\{u\} = \{B\}$ is, from equation(7.47) using Galerkin,

$$A_{ij} = -\langle \nabla \phi_j, \nabla \phi_i \rangle + \langle f\phi_j, \phi_i \rangle \tag{9.21}$$

$$b_i = \langle g, \phi_i \rangle - \oint \frac{\partial u}{\partial n} \phi_i \, ds \tag{9.22}$$

Implementation on linear triangles requires that we evaluate the element-level version of the discrete system. Table 9.1 provides all the information we require for exact integration. The resulting 3×3 element system is

$$\begin{bmatrix} \left[-\frac{\Delta y_1 \Delta y_1 + \Delta x_1 \Delta x_1}{4A} + \frac{fA}{6}\right] & \left[-\frac{\Delta y_1 \Delta y_2 + \Delta x_1 \Delta x_2}{4A} + \frac{fA}{12}\right] & \left[-\frac{\Delta y_1 \Delta y_3 + \Delta x_1 \Delta x_3}{4A} + \frac{fA}{12}\right] \\ \left[-\frac{\Delta y_2 \Delta y_1 + \Delta x_2 \Delta x_1}{4A} + \frac{fA}{12}\right] & \left[-\frac{\Delta y_2 \Delta y_2 + \Delta x_2 \Delta x_2}{4A} + \frac{fA}{6}\right] & \left[-\frac{\Delta y_2 \Delta y_3 + \Delta x_2 \Delta x_3}{4A} + \frac{fA}{12}\right] \\ \left[-\frac{\Delta y_3 \Delta y_1 + \Delta x_3 \Delta x_1}{4A} + \frac{fA}{12}\right] & \left[-\frac{\Delta y_3 \Delta y_2 + \Delta x_3 \Delta x_2}{4A} + \frac{fA}{12}\right] & \left[-\frac{\Delta y_3 \Delta y_3 + \Delta x_3 \Delta x_3}{4A} + \frac{fA}{6}\right] \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$= \begin{Bmatrix} \left[\frac{gA}{3}\right] \\ \left[\frac{gA}{3}\right] \\ \left[\frac{gA}{3}\right] \end{Bmatrix} - \begin{Bmatrix} \oint_e \frac{\partial u}{\partial n} \phi_1 \, ds \\ \oint_e \frac{\partial u}{\partial n} \phi_2 \, ds \\ \oint_e \frac{\partial u}{\partial n} \phi_3 \, ds \end{Bmatrix} \tag{9.23}$$

for the case in which the coefficients f and g are constant on an element. Assembly of this element matrix would follow the generic procedure outlined above, with the Incidence List as the key:

$$A_e(i,j) \Rightarrow A(IN(E,i),\ IN(E,j)) \tag{9.24}$$

$$B_e(i) \Rightarrow b(IN(E,i)) \tag{9.25}$$

ϕ	$\frac{[(x_j y_k - x_k y_j) + x\Delta y_i - y\Delta x_i]}{2A}$
A	$\frac{1}{2}\sum_1^3 x_l \, \Delta y_l$
Δx_i	$x_j - x_k$
Δy_i	$y_j - y_k$
$\langle 1 \rangle$	A
$\langle \phi_i \rangle$	$\frac{A}{3}$
$\langle \phi_i \phi_i \rangle$	$\frac{A}{6}$
$\langle \phi_i \, \phi_j \rangle$ $(i \neq j)$	$\frac{A}{12}$
$\left\langle \phi_i^l \phi_j^m \phi_k^n \right\rangle$	$2A\left[\frac{l!m!n!}{(l+m+n+2)!}\right]$
$\frac{\partial \phi_i}{\partial x}$	$\frac{\Delta y_i}{2A}$
$\frac{\partial \phi_i}{\partial y}$	$-\frac{\Delta x_i}{2A}$
$\left\langle f(x, y) \frac{\partial \phi_i}{\partial x} \right\rangle$	$\frac{\Delta y_i}{2A} \langle f(x, y) \rangle$
$\left\langle f(x, y) \frac{\partial \phi_i}{\partial y} \right\rangle$	$-\frac{\Delta x_i}{2A} \langle f(x, y) \rangle$

Table 9.1: Integration formulas for Linear Triangles. The local indices (i, j, k) are numbered in counterclockwise order.

The boundary integrals in the right-hand side vector $\oint_e ( )ds$ are evaluated around the perimeter of an individual element and represent the natural (Type 2 or Neumann) conditions. On assembly, two integrations would be indicated over "interior" line segments shared by two elements. Because the normal direction is exactly opposite, and the exact solution has continuity of ∇u, these contributions would exactly cancel and are eliminated from consideration *a priori*. The surviving line segments comprise the global boundary of the domain, and are the vehicle for insertion of boundary condition data. This consideration allows us to ignore, at the element level, the boundary integrals over interior line segments, and consider only segments where there is a suitable boundary condition on the global boundary. Thus we may interpret the notation $\oint_e ( )ds$ as integration along non-internal element sides.

Suppose the coefficients f and g were represented instead as linear functions with nodal values f_i, g_i. The Laplacian contributions to the matrix $[A]$ would be unchanged; but the f contributions would become:

$$\begin{bmatrix} \left[\frac{(6f_1+2f_2+2f_3)A}{60}\right] & \left[\frac{(2f_1+2f_2+f_3)A}{60}\right] & \left[\frac{(2f_1+f_2+2f_3)A}{60}\right] \\ \left[\frac{(2f_1+2f_2+f_3)A}{60}\right] & \left[\frac{(2f_1+6f_2+2f_3)A}{60}\right] & \left[\frac{(f_1+2f_2+2f_3)A}{60}\right] \\ \left[\frac{(2f_1+f_2+2f_3)A}{60}\right] & \left[\frac{(f_1+2f_2+2f_3)A}{60}\right] & \left[\frac{(2f_1+2f_2+6f_3)A}{60}\right] \end{bmatrix} \tag{9.26}$$

and the right-hand side would be

$$\begin{Bmatrix} \left[\frac{(2g_1+g_2+g_3)A}{12}\right] \\ \left[\frac{(g_1+2g_2+g_3)A}{12}\right] \\ \left[\frac{(g_1+g_2+2g_3)A}{12}\right] \end{Bmatrix} - \begin{Bmatrix} \oint_e \frac{\partial u}{\partial n}\phi_1 \, ds \\ \oint_e \frac{\partial u}{\partial n}\phi_2 \, ds \\ \oint_e \frac{\partial u}{\partial n}\phi_3 \, ds \end{Bmatrix} \tag{9.27}$$

The local coefficient variation produces specific local coefficient averages in the discrete system.

9.3 Higher Order Triangular Elements

Local Coordinate System

In Figure 9.2 we introduced a natural local coordinate system, based on the normalized altitudes:

$$L_i(x, y) = \frac{h_i(x, y)}{H_i} \tag{9.28}$$

These coordinates are sometimes referred to as *area coordinates* because they have the alternate interpretation as the ratio of the subarea A_l, to the full area A of the triangle. (See Figure 9.3.) From this interpretation it is clear that $\sum_1^3 L_l = 1$.

Clearly, these local coordinates are linear in (x, y). By inspection, L_l is unity at node l and vanishes at the other nodes; it therefore satisfies all requirements for a linear ϕ_l and we have the alternate representation of the linear basis

$$\phi_1 = L_1 \qquad \phi_2 = L_2 \qquad \phi_3 = L_3 = 1 - L_1 - L_2 \tag{9.29}$$

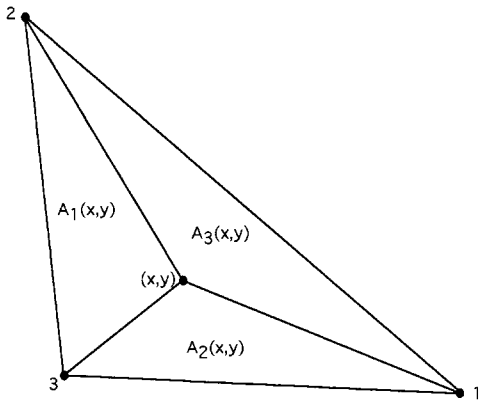


Figure 9.3: Local area coordinates for the triangle: $L_i = \frac{A_i(x,y)}{A}$.

Any two of the L_l constitute a valid local coordinate system. The mapping from local to global (x, y) coordinates:

$$x = \sum_{l=i,j,k} x_l \phi_l = \sum_{l=i,j,k} x_l L_l$$

$$y = \sum_{l=i,j,k} y_l \phi_l = \sum_{l=i,j,k} y_l L_l \tag{9.30}$$

is exact, given the linearity of the L and ϕ. In this way, any point in a triangle defined by local coordinates can be located uniquely in the global coordinate system.

We use the local system to define higher-order polynomial basis functions; and to define numerical quadrature points on a triangle.

Higher-Order Local Interpolation on Triangles

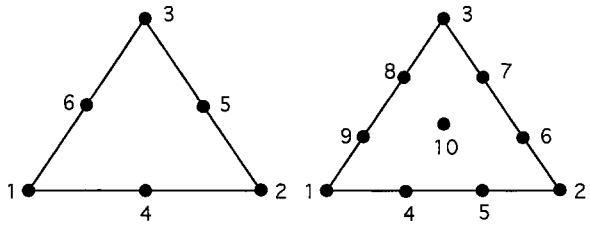


Figure 9.4: Quadratic (left) and Cubic triangular elements.

Higher-order bases are conveniently defined in the local coordinates – for example, the function. Figure 9.4 shows a quadratic element with three corner nodes and three midside nodes. Note the node numbering convention. Polynomial interpolation of the form

$$f(x) = a + bx + cy + dx^2 + exy + fy^2 \tag{9.31}$$

can be constructed by constraining matching nodal values at the six points. In particular, we can define six quadratic bases, one associated with each node, such that each is unity at its home node and vanishes at all other nodes. Since the local and global coordinate systems are linearly related,

either can be used, and the local system is far more convenient. Table 9.2 lists the six unique quadratics meeting these specifications, and is easily verified by inspection.

Cubic variation can be achieved in the form

$$f(x) = a + bx + cy + dx^2 + exy + fy^2 + gx^3 + hx^2y + pxy^2 + qy^3 \tag{9.32}$$

with an element with ten nodes as in Figure 9.4. Note that symmetry requires the same number of equidistant nodes on each side; otherwise neighboring elements will not conform. So cubic variation requires an additional node on the interior of the element. In Figure 9.4 we locate this node at the element center. Table 9.3 gives the cubic triangular bases, using the local numbering convention of Figure 9.4. As above, these are easily verified by inspection.

Both of these elements provide only C^0 continuity. Analogous to the 1-D case, we can define a triangular element with continuous first derivatives which will have cubic variation, and therefore 10 degrees of freedom as above. Each node on the element boundary will have 3 degrees of freedom – ϕ, $\frac{\partial\phi}{\partial x}$, and $\frac{\partial\phi}{\partial y}$. This element will therefore have a single midelement node with one degree of freedom in addition to the three corner nodes. Details of this and other higher-order elements may be found in standard references (*e.g.* [120], [52]).

Table 9.2: C^0 Quadratic Triangular Bases and their Derivatives.

	ϕ	$\frac{\partial\phi}{\partial L_1}$	$\frac{\partial\phi}{\partial L_2}$
Corner nodes ϕ_1	$L_1(2L_1 - 1)$	$4L_1 - 1$	0
ϕ_2	$L_2(2L_2 - 1)$	0	$4L_2 - 1$
ϕ_3	$L_3(2L_3 - 1)$	$-4L_3 + 1$	$-4L_3 + 1$
Midside nodes ϕ_4	$4L_1L_2$	$4L_2$	$4L_1$
ϕ_5	$4L_2L_3$	$-4L_2$	$4L_3 - 4L_2$
ϕ_6	$4L_3L_1$	$4L_3 - 4L_1$	$-4L_1$

Table 9.3: C^0 Cubic Triangular Bases and their Derivatives.

	ϕ	$\frac{\partial\phi}{\partial L_1}$	$\frac{\partial\phi}{\partial L_2}$
Corners ϕ_1	$\frac{1}{2}(3L_1-1)(3L_1-2)L_1$	$\frac{1}{2}[9L_1(3L_1-2)+2]$	0
ϕ_2	$\frac{1}{2}(3L_2-1)(3L_2-2)L_2$	0	$\frac{1}{2}[9L_2(3L_2-2)+2]$
ϕ_3	$\frac{1}{2}(3L_3-1)(3L_3-2)L_3$	$-\frac{1}{2}[9L_3(3L_3-2)+2]$	$-\frac{1}{2}[9L_3(3L_3-2)+2]$
Sides ϕ_4	$\frac{9}{2}L_1L_2(3L_1-1)$	$\frac{9}{2}(6L_1L_2-L_2)$	$\frac{9}{2}L_1(3L_1-1)$
ϕ_5	$\frac{9}{2}L_1L_2(3L_2-1)$	$\frac{9}{2}L_2(3L_2-1)$	$\frac{9}{2}(6L_2L_1-L_1)$
ϕ_6	$\frac{9}{2}L_2L_3(3L_2-1)$	$-\frac{9}{2}L_2(3L_2-1)$	$-\frac{9}{2}[L_2(3L_2-1)+L_3(6L_2-1)]$
ϕ_7	$\frac{9}{2}L_2L_3(3L_3-1)$	$-\frac{9}{2}[L_2(6L_3-1)]$	$\frac{9}{2}[L_3(3L_3-1)-L_2(6L_3-1)]$
ϕ_8	$\frac{9}{2}L_3L_1(3L_3-1)$	$\frac{9}{2}[L_3(3L_3-1)-L_1(6L_3-1)]$	$-\frac{9}{2}L_1(6L_3-1)$
ϕ_9	$\frac{9}{2}L_3L_1(3L_1-1)$	$\frac{9}{2}[L_3(6L_1-1)-L_1(3L_1-1)]$	$-\frac{9}{2}L_1(3L_1-1)]$
Center ϕ_{10}	$27L_1L_2L_3$	$27(L_2L_3-L_1L_2)$	$27(L_1L_3-L_1L_2)$

Differentiation

Differentiation of these bases requires some care. Basically we are interested in $\nabla\phi$ in the (x, y) system; but the bases can only be directly differentiated in the local system. We need to include the local-to-global coordinate transformation in a structured way.

Only two of the three local coordinates are independent, so we lose no generality by arbitrarily using only L_1 and L_2. Derivatives in this coordinate system are given by the chain rule:

$$\frac{\partial f}{\partial L_1} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial L_1} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial L_1} \tag{9.33}$$

and likewise for $\frac{\partial f}{\partial L_2}$. [1] These relationships are stated in matrix form as

$$\begin{Bmatrix} \frac{\partial f}{\partial L_1} \\ \\ \frac{\partial f}{\partial L_1} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{Bmatrix} \tag{9.34}$$

[1] Note that $\frac{\partial}{\partial L_1}$ means "holding L_2 constant", while $\frac{\partial}{\partial x}$ means "holding y constant", etc. The local differentiation also implies that $L_3 = 1 - L_1 - L_2$.

and the inverse relation

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial f}{\partial L_1} \\ \\ \frac{\partial f}{\partial L_2} \end{Bmatrix} \tag{9.35}$$

where the Jacobi matrix J is the generalization of its 1-D counterpart $\frac{dx}{d\xi}$ introduced in section 8.11:

$$J = \begin{bmatrix} \frac{\partial x}{\partial L_1} & \frac{\partial y}{\partial L_1} \\ \\ \frac{\partial x}{\partial L_2} & \frac{\partial y}{\partial L_2} \end{bmatrix} \tag{9.36}$$

The Jacobi Matrix can be evaluated anywhere on a given element. The linear local-to-global mapping introduced above (Equation 9.29)

$$x = \sum_1^3 x_l L_l = x_1 L_1 + x_2 L_2 + x_3(1 - L_1 - L_2)$$

$$y = \sum_1^3 y_l L_l = y_1 L_1 + y_2 L_2 + y_3(1 - L_1 - L_2) \tag{9.37}$$

is easily differentiated:

$$\frac{\partial x}{\partial L_1} = x_1 - x_3 \tag{9.38}$$

and so forth, giving a constant Jacobi matrix for a given element:

$$J = \begin{bmatrix} [x_1 - x_3] & [y_1 - y_3] \\ \\ [x_2 - x_3] & [y_2 - y_3] \end{bmatrix} \tag{9.39}$$

Numerical evaluation of J and its inverse is a simple matter on any triangle.

The bases are readily differentiated in the local coordinate system. For example, consider ϕ_4 on a quadratic element:

$$\frac{\partial \phi_4}{\partial L_1} = 4L_2 \quad \frac{\partial \phi_4}{\partial L_2} = 4L_1 \tag{9.40}$$

and we have

$$\begin{Bmatrix} \frac{\partial \phi_4}{\partial x} \\ \\ \frac{\partial \phi_4}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} 4L_2 \\ \\ 4L_1 \end{Bmatrix} \tag{9.41}$$

For ϕ_5 we have:

$$\frac{\partial \phi_5}{\partial L_1} = -4L_2 \quad \frac{\partial \phi_5}{\partial L_2} = 4L_3 - 4L_2 \tag{9.42}$$

giving

$$\begin{Bmatrix} \frac{\partial \phi_5}{\partial x} \\ \\ \frac{\partial \phi_5}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} -4L_2 \\ \\ 4L_3 - 4L_2 \end{Bmatrix} \tag{9.43}$$

(The reader is encouraged to verify the differentiation of ϕ_5, keeping in mind that L_1 and L_2 are independent variables; and that $L_3 = 1 - L_1 - L_2$.)

Tables 9.2 and 9.3 contain the required local derivatives of the bases for quadratic and cubic triangles. At any given point (L_1, L_2), their numerical evaluation is straightforward and their premultiplication by $[J]^{-1}$ completes the differentiation in the (x, y) system.

Below (section 9.4) we will introduce variation in $[J]$ within an element, which will be shown to greatly expand the capabilities of these elements with trivial computational expense.

Numerical Integration

The local coordinate system is also used to define numerical integration procedures in terms of quadrature points and weights. As introduced in the 1-D case, we approximate integration over the triangular element by a weighted sum of integrands evaluated at special points $(L_1, L_2)_q$ in the domain. In Table 9.4 we show quadratures with various degrees of polynomial precision, adapted from Cowper [25]. In each case the quadrature points are symmetrically arranged, so for example the 3-point formula of precision 2 involves all 3 permutations of the single point given as $(L_1, L_2, L_3) = (2/3, 1/6, 1/6)$. In Table 9.4 this is indicated as the multiplicity of a point. Also note that

$$\int\int_e dL_1\, dL_2 = \frac{1}{2} \tag{9.44}$$

and therefore the sum of the weights in every case is normalized here to $\frac{1}{2}$.

In our WR applications, we require integrals in the global (x, y) space:

$$< f >_e \equiv \int\int_e f(x, y)\, dx\, dy \tag{9.45}$$

and in the local L_1, L_2 system we have

$$dx\, dy = |J| dL_1\, dL_2 \tag{9.46}$$

and therefore our integrals must be evaluated as

$$< f >_e = \int\int_e f(L_1, L_2)|J| dL_1\, dL_2 \approx \sum_{q=1}^{N_q} (f\,|J|)_q\, W_q \tag{9.47}$$

where W_k is the weight associated with the point $(L_1, L_2)_q$; $(f\,|J|)_q$ is the integrand evaluated at that point; and N_q is the number of such points in an element. With the linearly-mapped triangle (equation 9.29) it is easy to confirm that

$$|J| = 2A \tag{9.48}$$

and that therefore the element integral of

$$< 1 >_e = |J| \sum_{q=1}^{N_q} W_q = A \tag{9.49}$$

confirming the normalization of these formulas.

Details of quadrature rules and their relative accuracy are recorded in several standard references on finite element methods, *e.g.* [120]; [52]; [101]. Integration over rectangular domains is common.

Table 9.4: Quadrature points and weights for triangles. The order of exact polynomial interpolation is indicated as N. Multiplicity $M > 1$ indicates multiple symmetric points. Adapted from [25].

N	W	L_1	L_2	L_3	M
2	0.16666667	0.66666667	0.16666667	0.16666667	3
2	0.16666667	0.50000000	0.50000000	0.00000000	3
3	-0.28125000 0.26041667	0.33333333 0.60000000	0.33333333 0.20000000	0.33333333 0.20000000	1 3
3	0.08333333	0.65902762	0.23193337	0.10903901	6
4	0.05497587 0.11169079	0.81684757 0.10810302	0.09157621 0.44594849	0.09157621 0.44594849	3 3
4	0.18750000 0.05208333	0.33333333 0.73671250	0.33333333 0.23793237	0.33333333 0.02535513	1 6
5	0.11250000 0.06296959 0.06619708	0.33333333 0.79742699 0.47014206	0.33333333 0.10128651 0.47014206	0.33333333 0.10128651 0.05971587	1 3 3
5	0.10297525 0.03184571	0.12494950 0.79711265	0.43752525 0.16540993	0.43752525 0.03747742	3 6
6	0.02542245 0.05839314 0.04142554	0.87382197 0.50142651 0.63650250	0.06308901 0.24928675 0.31035245	0.06308901 0.24928675 0.05314505	3 3 6

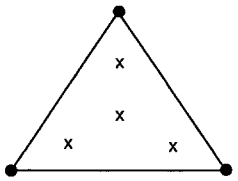


Figure 9.5: Four-point integration from Table 9.4, exact for order 3 polynomials in (L_1, L_2, L_3).

Triangular domains are treated by Cowper [25] (Table 9.4 herein), Lyness [83], and Berntsen and Espelid [11, 13, 12, 10].

Several quadrature formulas have been presented that can offer significant computational savings *e.g.* [37]. In particular, for linear triangles, exactly three quadrature points in each triangle, coinciding with the triangle vertices, has been extensively used [67]. The weights in this case are all equal:

$$(L_1, L_2, L_3)_q = (1,0,0),\ \ (0,1,0),\ \ (0,0,1) \tag{9.50}$$

$$W_q = \frac{A}{3},\ \frac{A}{3},\ \frac{A}{3} \tag{9.51}$$

We refer to this as *nodal quadrature.* It exactly integrates a linear function over the element. Note that in Table 9.4, quadratic precision is possible for three quadrature points; so there must be a compelling reason to favor this particular quadrature. (In the uses cited, the reason is matrix structure.)

This completes the element-level picture for any WR formulation involving C^0 triangular bases and their first derivatives. Everything (interpolation, differentiation, integration) is done in local coordinates involving simple polynomials. Operationally, one visits each quadrature point in succession, evaluating every WR integrand, multiplying it by $|J|W_q$ and adding it to the global system. Nesting this procedure inside a generic loop over all elements makes the assembly complete for an arbitrarily structured mesh of triangles.

9.4 Curved Triangular Elements and The Isoparametric Transformation

As a final refinement of the triangular element, we relax the constraint that elements have straight sides. The general situation is depicted in Figure 9.6, for an element with six points defining the curved sides. The key to this element is mapping it onto the generic triangle as shown, which in local terms is identical to the quadratic triangle defined above.

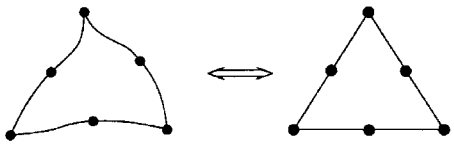


Figure 9.6: Isoparametric triangle.

Like other operations on these elements, the local coordinate system is the best starting point, and it is simplest to work from it, backwards to the global coordinate system. We define a suitable

local basis ψ for the mapping:

$$x(L_1, L_2) = \sum_l x_l \psi(L_1, L_2)_l$$
$$y(L_1, L_2) = \sum_l y_l \psi(L_1, L_2)_l \tag{9.52}$$

where the basis ψ is interpolating the global coordinates of the mapping points for a given element. In this form, we have the usual constraints that $\psi_l = 1$ at node l and zero at all other mapping nodes. Obviously, the quadratic bases defined above are the correct interpolants here. In addition, their continuity (C^0) guarantees that the shared, curved side of two adjacent elements will be congruent in the global (x, y) system. (See Figure 9.7.) So we see that the standard FE bases are capable of interpolating not only the dependent variables and coefficients of a problem, but the dependent variables as well!

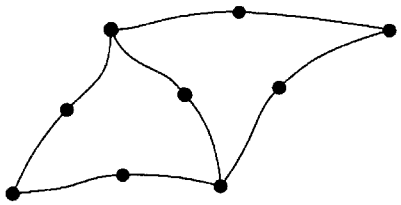


Figure 9.7: Congruence of 2 isoparametric triangles along a shared side.

The Jacobi transformation introduced earlier in section 9.3 needs to be upgraded to account for this enhancement. By definition, we have

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial f}{\partial L_1} \\ \\ \frac{\partial f}{\partial L_2} \end{Bmatrix} \tag{9.53}$$

$$J = \begin{bmatrix} \frac{\partial x}{\partial L_1} & \frac{\partial y}{\partial L_1} \\ \\ \frac{\partial x}{\partial L_2} & \frac{\partial y}{\partial L_2} \end{bmatrix} \tag{9.54}$$

With x and y interpolated as in (9.52) above, we have

$$\frac{\partial x}{\partial L_1} = \sum_l x_l \frac{\partial \psi_l}{\partial L_1}$$
$$\frac{\partial x}{\partial L_2} = \sum_l x_l \frac{\partial \psi_l}{\partial L_2} \tag{9.55}$$

and similarly for $\frac{\partial y}{\partial L_1}$, $\frac{\partial y}{\partial L_2}$. The required derivatives are all expressed in the local system and therefore their numerical evaluation at any (L_1, L_2) point is straightforward. Numerical evaluation of $[J]$ and its inverse, which now varies locally on the element, is readily obtained at any quadrature point. There is no further adjustment to the general development above.

The general form of integration introduced at (9.47) also remains unchanged:

$$< f >_e = \int\int_e f(L_1, L_2)|J| dL_1\, dL_2 \approx \sum_{q=1}^{N_q} (f\,|J|)_q\, W_q \tag{9.56}$$

The only complication is the extra local variation contributed to the integrand by $|J|$, which may require a higher-precision quadrature formula (*i.e.* more quadrature points).

The limited version of the Jacobi transformation used above, equation (9.30), can be seen to be the special case where ψ are the linear triangular bases. In general, we identify three cases:

- the mapping bases ψ are the same as the basis for the dependent variable ϕ. This is the *Isoparametric* case and is commonly used.
- ψ are of lower order than ϕ. This is the *Subparametric* case and it was used in the previous sections to develop the higher-order triangular bases. Like the Isoparametric case, this approach is common, especially the use of linear mapping for otherwise higher-order triangles.
- ψ are of higher order than ϕ. This is the *Superparametric* case.

9.5 Quadrilateral Elements

The most rudimentary bilinear element embodies the product of linear variation in x with linear variation in y:

$$\phi_i(x, y) = (a + bx)(c + dy) \tag{9.57}$$

on a square or rectangle aligned with the (x, y) axes.[2] Suitable bases are easily arranged as the product of separate 1-D Lagrange polynomials in x and y. If our goal were limited to this, we could stop here. However we wish to generalize this idea to allow simple representation of quadrilaterals which

- are not aligned with the global coordinate system;
- which are not rectangular;
- which have higher-order local interpolation; and
- which have curved sides.

We will introduce in a systematic way the standard four issues for the quadrilateral element family:

- the description of bases in local coordinates
- the mapping from local to global coordinates
- the procedure for differentiation in local coordinates
- the procedure for integration in local coordinates

We will concentrate on Lagrangian bases as in the 1-D elements. All the basic ideas were introduced in the previous discussion of triangular elements.

The Bilinear Element

In Figure 9.8 we introduce the simplest quadrilateral element in the local coordinate system (ξ, η). Both local coordinates are centered in the element and range from -1 to 1. The bilinear bases are the product of two 1-D Lagrange polynomials in ξ and η:

$$\phi_l(\xi, \eta) = \frac{1}{4}(1 + \xi_l\xi)(1 + \eta_l\eta) \tag{9.58}$$

and their local derivatives are

$$\frac{\partial \phi_l}{\partial \xi} = \frac{\xi_l}{4}(1 + \eta_l\eta)$$

[2]Note that the product term in xy creates quadratic variation on the diagonal of such an element; this complication was not built into the triangular elements.

$$\frac{\partial \phi_l}{\partial \eta} = \frac{\eta_l}{4}(1 + \xi_l \xi) \tag{9.59}$$

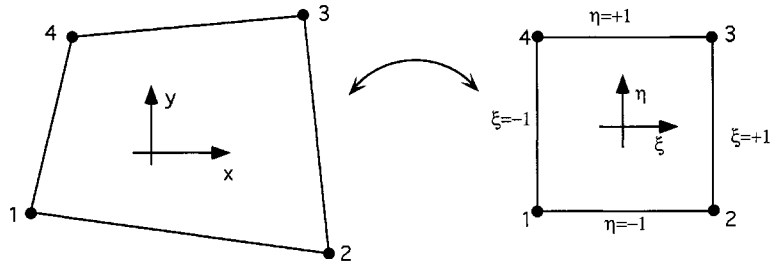


Figure 9.8: Local coordinate system for the bilinear quadrilateral element.

The mapping from local to global coordinates is conveniently stated as a bilinear transformation, which for this element is an Isoparametric mapping:

$$x(\xi, \eta) = \sum_l x_l \phi_l(\xi, \eta)$$

$$y(\xi, \eta) = \sum_l y_l \phi_l(\xi, \eta) \tag{9.60}$$

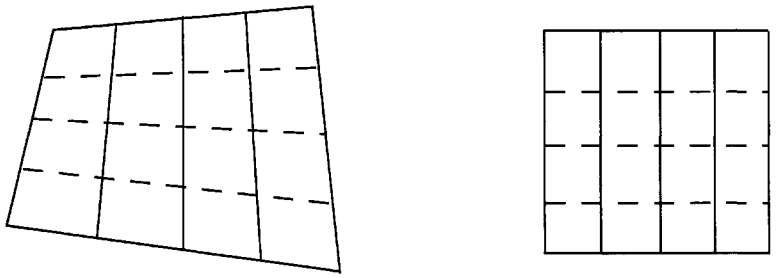


Figure 9.9: Lines of constant ξ and η as mapped into the (x, y) system.

Figure 9.9 illustrates the mapping of lines of constant ξ and η onto the global element. Along any line of constant ξ, we have

- all bases are linear in η; therefore
- x and y are linear in η; therefore
- $\frac{dx}{d\eta}$ and $\frac{dy}{d\eta}$ are constants; therefore
- $\frac{dy}{dx}$ is a constant

So we find that lines of constant ξ are straight lines in (x, y). The same is true for lines of constant η. Since the corners map exactly, we find that the bilinear mapping produces a straight-sided quadrilateral.

The Jacobi transformation is generic:

$$\begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{Bmatrix} \tag{9.61}$$

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} \tag{9.62}$$

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{9.63}$$

with J entries of the form

$$\frac{\partial x}{\partial \xi} = \sum_l x_l \frac{\partial \phi_l}{\partial \xi} \tag{9.64}$$

and so forth. Note that J in this simplest quadrilateral case varies over the element. [3]

Derivatives of the bases are thus readily evaluated at any point (ξ, η) by first evaluating $\frac{\partial \phi_l}{\partial \xi}$ and $\frac{\partial \phi_l}{\partial \eta}$; then evaluating $|J|$ and inverting it; then numerically evaluating the product (9.62).

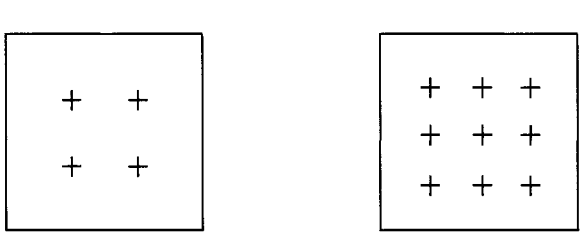


Figure 9.10: Illustrating 2-D Gauss-Legendre quadrature in local coordinates on the quadrilateral. 2×2 and 3×3 quadrature is shown.

Integration is achieved in the local coordinate system as the product of two 1-D integrations. Gauss-Legendre quadrature is commonly used, and the formulas in Table 8.9 are directly applicable. For example, 2×2 and 3×3 quadrature are illustrated in Figure 9.10, with Gauss points given in Tables 9.5 and 9.6. Tables and/or procedures for general $n \times n$ quadrature are readily generated from the 1-D rules. Remember that the 2-D weights are products of a pair of 1-D weights.

Higher-Order Quadrilateral Elements

The generalization of the above for higher-order polynomial variation is straightforward. In general, we may construct bi-quadratic, bi-cubic, and higher forms of elements by the same procedure of multiplying 1-D Lagrangian bases in ξ and η. The quadratic quadrilateral will have $3 \times 3 = 9$ nodes, with one located at the element center as illustrated in Figure 9.11. The bases and their local derivatives are given in Table 9.8. The cubic version will have 16 nodes, including 4 interior nodes, as in Figure 9.11. Tables 9.9 and 9.10 provide the necessary local description of these bases. The generalization to higher-order elements is straightforward and details can be found in standard sources.

Isoparametric Quadrilaterals

All of these higher-order quadrilaterals can be used with the simple bilinear mapping described above. This would be "subparametric" mapping, with straight-sided quadrilaterals in the global

[3] In the analogous linear triangle case $|J|$ was a constant; the variation here is created by the bilinear term $\xi\eta$ in the bases.

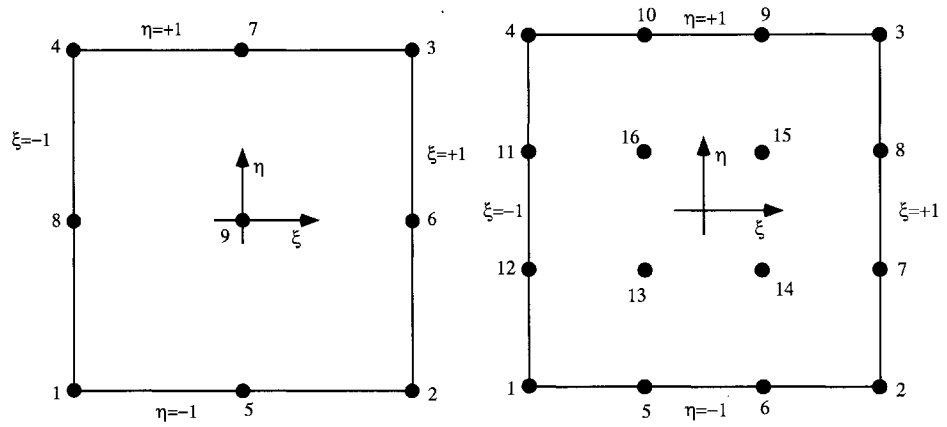


Figure 9.11: Lagrange bi-quadratic and bi-cubic quadrilateral elements.

system mapped to a square.

The isoparametric option provides curved sided "quadrilaterals" in the (x, y) system – objects with four piecewise continuous curves joining at 4 vertices. Figure 9.12 illustrates a contiguous pair of quadratic isoparametric quadrilaterals. The specific shape of any side is embedded mathematically in the standard local interpolation of x and y. For example, along the "bottom" side at $\eta = -1$, we have

$$x = \sum_l x_l \phi_l(\xi, -1)$$

$$y = \sum_l y_l \phi_l(\xi, -1) \tag{9.65}$$

Since only bases 1, 2, and 5 are nonzero along that side, we have a simple quadratic in ξ interpolating among the positions x_1, x_2, and x_5. The other 3 sides of the element are analogous. The C^0 property of the bases guarantees that the curved sides of adjacent elements will be congruent, as in Figure 9.12

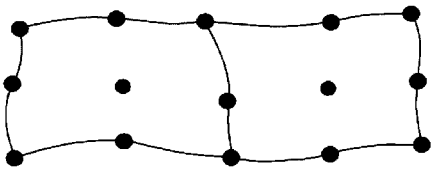


Figure 9.12: Isoparametric biquadratic elements, illustrating the conformance of the shared side.

On the element interior, the centerline $\eta = 0$ connects the midside and center nodes with a quadratic of the form

$$x = \sum_l x_l \phi_l(\xi, 0) \tag{9.66}$$

$$y = \sum_l y_l \phi_l(\xi, 0) \tag{9.67}$$

Only nodes 8, 9, and 6 will participate in this interpolation. Any line of constant η will be a

quadratic interpolation among this centerline and the top and bottom edges. Lines of constant ξ will map in an analogous manner.

The description of the bases for isoparametric elements is unchanged in the local coordinate system; and the general description of differentiation and integration involving the Jacobi transformation and 2-D Gaussian quadrature is also unchanged. The formulas described above are directly applicable and need not be repeated.

Table 9.5: 2×2 Gauss-Legendre quadrature, sufficient to integrate an integrand of order $\xi^3\eta^3$.

ξ_k	η_k	W_k
-0.57735027	-0.57735027	1.0
-0.57735027	0.57735027	1.0
0.57735027	-0.57735027	1.0
0.57735027	0.57735027	1.0

Table 9.6: 3×3 Gauss-Legendre quadrature, sufficient to integrate an integrand of order $\xi^5\eta^5$.

ξ_k	η_k	W_k
-0.77459667	-0.77459667	.30864197
-0.77459667	0	.49382716
-0.77459667	0.77459667	.30864197
0	-0.77459667	.49382716
0	0	.79012346
0	0.77459667	.49382716
0.77459667	-0.77459667	.30864197
0.77459667	0	.49382716
0.77459667	0.77459667	.30864197

Table 9.7: C^0 Bilinear Quadrilateral Bases and their Derivatives.

	ϕ	$\frac{\partial\phi}{\partial\xi}$	$\frac{\partial\phi}{\partial\eta}$
Corners $\xi_l = \pm 1;\ \eta_l = \pm 1$	$\frac{1}{4}(1+\xi_l\xi)(1+\eta_l\eta)$	$\frac{\xi_l}{4}(1+\eta_l\eta)$	$\frac{\eta_l}{4}(1+\xi_l\xi)$

Table 9.8: C^0 Quadratic Quadrilateral Bases and their Derivatives.

	ϕ	$\frac{\partial\phi}{\partial\xi}$	$\frac{\partial\phi}{\partial\eta}$
Corners $\xi_l = \pm 1;\ \eta_l = \pm 1$	$\frac{1}{4}\xi\xi_l(1+\xi_l\xi)\eta\eta_l(1+\eta_l\eta)$	$\frac{1}{4}\xi_l(1+2\xi_l\xi)\eta\eta_l(1+\eta_l\eta)$	$\frac{1}{4}\xi\xi_l(1+\xi_l\xi)\eta_l(1+2\eta_l\eta)$
Sides $\xi_l = 0;\ \eta_l = \pm 1$	$\frac{1}{2}(1-\xi^2)\eta\eta_l(1+\eta_l\eta)$	$\frac{1}{2}(-2\xi)\eta\eta_l(1+\eta_l\eta)$	$\frac{1}{2}(1-\xi^2)\eta_l(1+2\eta_l\eta)$
$\xi_l = \pm 1;\ \eta_l = 0$	$\frac{1}{2}(1-\eta^2)\xi\xi_l(1+\xi_l\xi)$	$\frac{1}{2}(1-\eta^2)\xi_l(1+2\xi_l\xi)$	$\frac{1}{2}(-2\eta)\xi\xi_l(1+\xi_l\xi)$
Center $\xi_l = 0;\ \eta_l = 0$	$(1-\xi^2)(1-\eta^2)$	$(-2\xi)(1-\eta^2)$	$(1-\xi^2)(-2\eta)$

Table 9.9: C^0 Cubic Quadrilateral Bases.

	ϕ
Corners $\xi_l = \pm 1;\ \eta_l = \pm 1$	$\frac{1}{256}(9\xi^2-1)(\xi_l\xi+1)(9\eta^2-1)(\eta_l\eta+1)$
Sides $\xi_l = \pm\frac{1}{3};\ \eta_l = \pm 1$	$\frac{9}{256}(9\eta^2-1)(\eta_l\eta+1)(1-\xi^2)(1+9\xi_l\xi)$
$\xi_l = \pm 1;\ \eta_l = \pm\frac{1}{3}$	$\frac{9}{256}(9\xi^2-1)(\xi_l\xi+1)(1-\eta^2)(1+9\eta_l\eta)$
Interior $\xi_l = \pm\frac{1}{3};\ \eta_l = \pm\frac{1}{3}$	$\frac{81}{256}(1-\xi^2)(1+9\xi_l\xi)(1-\eta^2)(1+9\eta_l\eta)$

Table 9.10: C^0 Cubic Quadrilateral Derivatives.

	$\frac{\partial \phi}{\partial \xi}$	$\frac{\partial \phi}{\partial \eta}$
Corners $\xi_l = \pm 1;\ \eta_l = \pm 1$	$\frac{1}{256}(27\xi^2\xi_l + 18\xi - \xi_l)(9\eta^2 - 1)(\eta_l\eta + 1)$	$\frac{1}{256}(9\xi^2 - 1)(\xi_l\xi + 1)(27\eta^2\eta_l + 18\eta - \eta_l)$
Sides $\xi_l = \pm\frac{1}{3};\ \eta_l = \pm 1$	$\frac{9}{256}(9\eta^2 - 1)(\eta_l\eta + 1)(-27\xi^2\xi_l - 2\xi + 9\xi_l)$	$\frac{9}{256}(27\eta^2\eta_l + 18\eta - \eta_l)(1 - \xi^2)(1 + 9\xi_l\xi)$
$\xi_l = \pm 1;\ \eta_l = \pm\frac{1}{3}$	$\frac{9}{256}(27\xi^2\xi_l + 18\xi - \xi_l)(1 - \eta^2)(1 + 9\eta_l\eta)$	$\frac{9}{256}(9\xi^2 - 1)(\xi_l\xi + 1)(-27\eta^2\eta_l - 2\eta + 9\eta_l)$
Interior $\xi_l = \pm\frac{1}{3};\ \eta_l = \pm\frac{1}{3}$	$\frac{81}{256}(-27\xi^2\xi_l - 2\xi + 9\xi_l)(1 - \eta^2)(1 + 9\eta_l\eta)$	$\frac{81}{256}(1 - \xi^2)(1 + 9\xi_l\xi)(-27\eta^2\eta_l - 2\eta + 9\eta_l)$

Chapter 10

Time-Dependent Problems

10.1 General Approach

The time domain enters physical problems in a fundamentally different way than the other spatial dimensions. As a result, it is treated differently in most FE studies. It would appear natural to simply add an extra dimension to the types of elements already defined, and to describe a time-dependent problem in terms of

$$u(x,\, y,\, z,\, t) = \sum_i u_i \phi_i(x,\, y,\, z,\, t) \tag{10.1}$$

where the ϕ_i constitute interpolants on "space-time elements", and t is added to the independent variable list in a routine manner. However there are few immediate advantages, as the time-domain presents itself in the form of initial-value problems rather than boundary-value problems associated with the other dimensions, and only in advanced problems is anything other than a perfectly regular temporal mesh, orthogonal to $(x,\, y,\, z)$, called for. So, it is nearly universal practice to separate space and time variation

$$u(x,\, y,\, z,\, t) = \sum_i u_i(t) \phi_i(x,\, y,\, z) \tag{10.2}$$

The ϕ_i here are regular, time-invariant FE bases interpolating *nodal histories* $u_i(t)$; and the $u_i(t)$ are the state variables of a dynamical system.[1]

10.2 Lumped and Discrete Systems

We will distinguish between two forms of FE system:

- the *lumped system*, where elliptic dimensions have been discretized by the WR methods described earlier, but the time domain is left continuous. This will be a system of coupled Ordinary Differential Equations in time.
- the *discrete system*, wherein the time domain is also discretized. The mathematical form will be difference equations describing time- and space-discrete processes.

[1] Exceptions to this general rule of time-invariant bases are called for in free- and moving-boundary problems, front-tracking problems, etc. In those problems, time-varying bases are uniquely capable of adjusting the spatial discretization in response to steep gradients or moving boundaries.

Implied is the need to appeal to an additional discretization principle for the time domain. In general we will utilize the standard set of techniques for integrating coupled ODE's as initial-value problems.

10.3 Example: Diffusion Equation

Consider the diffusion equation

$$\frac{\partial u}{\partial t} - \nabla \cdot D\nabla u = r \tag{10.3}$$

and its Galerkin weak form

$$\left\langle \frac{\partial u}{\partial t}\phi_i \right\rangle + \langle D\nabla u \cdot \nabla \phi_i \rangle - \oint D\nabla u \cdot \hat{\mathbf{n}}\phi_i \, ds = \langle r\phi_i \rangle \tag{10.4}$$

Here we have used the identity (see Appendix)

$$\langle (\nabla \cdot \mathbf{v})\, \phi_i \rangle + \langle \mathbf{v} \cdot \nabla \phi_i \rangle = \oint \mathbf{v} \cdot \hat{\mathbf{n}}\phi_i \, ds \tag{10.5}$$

Expanding u in the basis as in (10.2) above, we have spatial differentiation as before

$$\nabla u = \sum_i u_i \nabla \phi_i \tag{10.6}$$

and temporal differentiation gives us

$$\frac{\partial u}{\partial t} = \sum_i \frac{du_i}{dt}\phi_i \tag{10.7}$$

Substituting these quantities into the weak form provides

$$\sum_j \frac{du_j}{dt} \langle \phi_j \phi_i \rangle + \sum_j u_j \langle D\nabla \phi_j \cdot \nabla \phi_i \rangle = \oint D\nabla u \cdot \hat{\mathbf{n}}\phi_i \, ds + \langle r\phi_i \rangle \tag{10.8}$$

or in matrix form,

$$[M]\left\{\frac{du}{dt}\right\} + [K]\{u\} = \{R\} \tag{10.9}$$

with

$$M_{ij} = \langle \phi_j \phi_i \rangle \tag{10.10}$$

$$K_{ij} = \langle D\nabla \phi_j \cdot \nabla \phi_i \rangle \tag{10.11}$$

$$R_i = \oint D\nabla u \cdot \hat{\mathbf{n}}\, ds + \langle r\phi_i \rangle \tag{10.12}$$

This is the *lumped system* approximation to the diffusion equation. The assembly of the matrices $[M]$ (the "mass matrix") and $[K]$ (the "stiffness matrix"), and the right-hand side vector (including natural boundary condition information plus the forcing term r), are exactly as discussed in steady-state problems. We concentrate on the discretization of the time domain.

The simplest integration of the lumped system would involve two time-levels, t^k and t^{k+1}. Integrating from t^k to t^{k+1} we have

$$[M]\left\{u^{k+1}\right\} = [M]\left\{u^k\right\} + \int_k^{k+1} \left(-[K]\{u\} + \{R\}\right) dt \tag{10.13}$$

The temporal integral may be approximated in terms of its value at t^k and t^{k+1}:

$$\int_k^{k+1} g\,dt = \Delta t\left(\theta g^{k+1} + (1-\theta)g^k\right) \tag{10.14}$$

$\theta = 0$, 1, and 1/2 are the standard forward Euler, backward Euler, and Trapezoidal Rule integrations. Pursuing this gives the final *discrete system* representation:

$$\left[[M] + \theta\Delta t[K]\right]\left\{u^{k+1}\right\} = \left[[M] - (1-\theta)\Delta t[K]\right]\left\{u^k\right\} + \int_k^{k+1}\{R\}\,dt \tag{10.15}$$

Assuming the availability of initial values for $\left\{u^k\right\}$, we may distill the discrete system to a repetitive application of

$$[A]\left\{u^{k+1}\right\} = [B]\left\{u^k\right\} + \left\{c^{k+1/2}\right\} \tag{10.16}$$

with time-invariant matrices

$$A_{ij} = \langle\phi_j\phi_i + \theta\Delta t\, D\nabla\phi_j\cdot\nabla\phi_i\rangle \tag{10.17}$$

$$B_{ij} = \langle\phi_j\phi_i - (1-\theta)\Delta t\, D\nabla\phi_j\cdot\nabla\phi_i\rangle \tag{10.18}$$

and time-integrated forcing comprising the source term r plus natural boundary conditions:

$$c_i^{k+1/2} = \int_k^{k+1}\left(\oint D\nabla u\cdot\hat{\mathbf{n}}\phi_i\,ds + \langle r\phi_i\rangle\right)dt \tag{10.19}$$

Assembly of $[A]$, $[B]$ and $\left\{c^{k+1/2}\right\}$ involves standard application of the FE methodology developed for time-independent problems. For example, implementation on a linear triangular mesh, with D constant on an element and exact integration as in Table 9.1 leads to the following element-level matrices:[2]

$$A_{i,j} = \frac{A_e}{12}(1+\delta_{ij}) + \theta D\Delta t\frac{(\Delta x_i\Delta x_j + \Delta y_i\Delta y_j)}{4A_e} \tag{10.20}$$

$$B_{i,j} = \frac{A_e}{12}(1+\delta_{ij}) - (1-\theta)\,D\Delta t\frac{(\Delta x_i\Delta x_j + \Delta y_i\Delta y_j)}{4A_e} \tag{10.21}$$

Because $[A]$ and $[B]$ are time-invariant, they may be assembled, factored, etc once at the beginning of a simulation. Each time step is then a simple assembly of the time-variable forcing vector, plus manipulation of known matrices. On the linear triangles, this would be, for constant r at the element level,

$$c_i^{k+1} = \int_k^{k+1}\left(\frac{rA_e}{3} + \oint_e D\nabla u\cdot\hat{\mathbf{n}}\phi_i\,ds\right)dt \tag{10.22}$$

or for linear r at the element level,

$$c_i^{k+1} = \int_k^{k+1}\left((r_1+r_2+r_3+r_i)\frac{A_e}{12} + \oint_e D\nabla u\cdot\hat{\mathbf{n}}\phi_i\,ds\right)dt \tag{10.23}$$

The repetitive solution of (10.16) formally requires the computation, storage and multiplication of $[A]^{-1}$. If approached directly, this would require full matrix storage ($O(N^2)$ words for a mesh of

[2] δ_{ij} is the Kronecker delta: $\delta_{ij} = 1$ if $i = j$, 0 otherwise; and A_e is the element area.

N nodes) and excessive runtime ($O(N^3)$ operations for full matrix multiplication). It is therefore essential for practical problems to take advantage of sparse matrix storage and solution techniques.

For 2-D applications, the use of direct LR decomposition has proven to be especially useful. In this method, the matrix $[A]$ is factored into left and right triangular factors

$$[A] = [L][R] \tag{10.24}$$

Now solution of a triangular system, say

$$[L]\{y\} = \{z\} \tag{10.25}$$

is a straightforward matter because the first equation has only one unknown, and solution can progress downward in a sequential manner. The same is true for equations involving R, except the solution begins with the last equation and progresses backward. In either case, the inverse solution

$$\{y\} = [L]^{-1}\{z\} \tag{10.26}$$

is achieved without direct calculation of the inverse. Use of these ideas for the matrix equation $[A]\{u\} = \{z\}$ is achieved in three steps:

1) compute $[L]$ and $[R]$ from $[A]$
2) solve $[L]\{y\} = \{z\}$ for the intermediate vector $\{y\}$
3) solve $[R]\{u\} = \{y\}$ for $\{u\}$

Step 1 can be done once at the beginning of a simulation and stored; steps 2 and 3 need to be repeated for each right-side vector.

Now the factorization of a banded matrix with bandwidth N_b requires $O(NN_b^2)$ operations, as opposed to $O(N^3)$ operations for inversion; and steps 2 and 3 above require $O(NN_b)$ operations, as opposed to $O(N^2)$ for multiplication by a full inverse matrix. For 2-D applications this makes LR decomposition extremely effective. Further, $[L]$ and $[R]$ are also banded, with the same bandwidth as $[A]$. So, $[L]$ and $[R]$ can be stored with only $N * N_b$ entries. For a symmetric matrix, $[L] = [R]^T$.

The product $[B]\left\{u^k\right\}$ on the right-hand side of (10.16) is readily implemented in any sparse matrix storage scheme (*i.e.* one in which only the nonzero coefficients are stored).

In 3-D the storage requirement of LR decomposition becomes uneconomical for large problems – the minimum achievable bandwidth is impractically large. (See the discussion at Section 8.9.) In these and very large 2-D applications, it is desirable to use iterative sparse matrix solution methods. These are fundamentally arranged so that only sparse matrix multiplication is needed for each iteration. There is a large front of activity in this general area.

10.4 Example: Advection-Diffusion Equation

As an extension to the previous example, add an advective term to equation (10.3):

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \nabla \cdot D\nabla u = r \tag{10.27}$$

The Galerkin weak form has the additional term $\langle \mathbf{v} \cdot \nabla u \phi_i \rangle$:

$$\left\langle \frac{\partial u}{\partial t} \phi_i \right\rangle + \langle \mathbf{v} \cdot \nabla u \phi_i \rangle + \langle D\nabla u \cdot \nabla \phi_i \rangle - \oint D\nabla u \cdot \hat{\mathbf{n}}\, ds = \langle r\phi_i \rangle \tag{10.28}$$

Expanding u in the basis gives us

$$\sum_j \frac{du_j}{dt} \langle \phi_j \phi_i \rangle \sum_j u_j \langle (\mathbf{v} \cdot \nabla \phi_j) \phi_i \rangle + \sum_j u_j \langle D \nabla \phi_j \cdot \nabla \phi_i \rangle = \oint D \nabla u \cdot \hat{\mathbf{n}} \, ds + \langle r \phi_i \rangle \tag{10.29}$$

and the lumped system is now

$$[M] \left\{ \frac{du}{dt} \right\} + [K] \{u\} = \{R\} \tag{10.30}$$

with

$$\begin{aligned} M_{ij} &= \langle \phi_j \phi_i \rangle & (10.31) \\ K_{ij} &= \langle (\mathbf{v} \cdot \nabla \phi_j) \phi_i \rangle + \langle D \nabla \phi_j \cdot \nabla \phi_i \rangle & (10.32) \\ R_i &= \oint D \nabla u \cdot \hat{\mathbf{n}} \, ds + \langle r \phi_i \rangle & (10.33) \end{aligned}$$

The only change from the diffusion equation is the additional entry in $[K]$ for the advective term.

Following the same temporal discretization, the discrete system would again be

$$[A] \left\{ u^{k+1} \right\} = [B] \left\{ u^k \right\} + \left\{ c^{k+1/2} \right\} \tag{10.34}$$

with

$$\begin{aligned} A_{ij} &= \langle \phi_j \phi_i + \theta \Delta t \, (\mathbf{v} \cdot \nabla \phi_j) \phi_i + \theta \Delta t \, D \nabla \phi_j \cdot \nabla \phi_i \rangle & (10.35) \\ B_{ij} &= \langle \phi_j \phi_i - (1-\theta) \Delta t \, (\mathbf{v} \cdot \nabla \phi_j) \phi_i - (1-\theta) \Delta t \, D \nabla \phi_j \cdot \nabla \phi_i \rangle & (10.36) \\ c_i^{k+1/2} &= \int_k^{k+1} \left(\oint D \nabla u \cdot \hat{\mathbf{n}} \phi_i \, ds + \langle r \phi_i \rangle \right) dt & (10.37) \end{aligned}$$

The time-integrated forcing c is unchanged from the diffusion example. Matrices $[A]$ and $[B]$ for a linear triangular element would be modified to include the advective term:

$$A_{i,j} = \frac{A_e}{12} (1 + \delta_{ij}) + \theta \Delta t \frac{(v_x \Delta y_j - v_y \Delta x_j)}{6} + \theta D \Delta t \frac{(\Delta x_i \Delta x_j + \Delta y_i \Delta y_j)}{4 A_e} \tag{10.38}$$

$$B_{i,j} = \frac{A_e}{12} (1 + \delta_{ij}) - (1-\theta) \Delta t \frac{(v_x \Delta y_j - v_y \Delta x_j)}{6} - (1-\theta) D \Delta t \frac{(\Delta x_i \Delta x_j + \Delta y_i \Delta y_j)}{4 A_e} \tag{10.39}$$

As in the diffusion example, we have assumed constant coefficients over an element.

10.5 Example: Wave Equation

Consider the wave equation

$$\frac{\partial^2 u}{\partial t^2} - \nabla \cdot C^2 \nabla u = r \tag{10.40}$$

and its Galerkin weak form

$$\left\langle \frac{\partial^2 u}{\partial t^2} \phi_i \right\rangle + \left\langle C^2 \nabla u \cdot \nabla \phi_i \right\rangle - \oint C^2 \nabla u \cdot \hat{\mathbf{n}} \phi_i \, ds = \langle r \phi_i \rangle \tag{10.41}$$

The progression to the lumped system is exactly analogous to that for the diffusion equation, except for the order of temporal differentiation. The resulting lumped system is:

$$\sum_j \frac{d^2 u_j}{dt^2} \langle \phi_j \phi_i \rangle + \sum_j u_j \left\langle C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle = \oint C^2 \nabla u \cdot \hat{\mathbf{n}} \phi_i \, ds + \langle r \phi_i \rangle \tag{10.42}$$

or in matrix form,

$$[M] \left\{ \frac{d^2 u}{dt^2} \right\} + [K] \{u\} = \{R\} \tag{10.43}$$

with

$$M_{ij} = \langle \phi_j \phi_i \rangle \tag{10.44}$$

$$K_{ij} = \left\langle C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.45}$$

$$R_i = \oint C^2 \nabla u \cdot \hat{\mathbf{n}} \phi_i \, ds + \langle r \phi_i \rangle \tag{10.46}$$

The mass and stiffness matrices $[M]$ and $[K]$ are identical to those already described for the diffusion equation; the temporal discretization requires a different approach.

The second derivative in time demands at least three time levels in discrete form. Here we will use a conventional finite difference approach with equally-spaced time levels $k+1$, k, and $k-1$:

$$\frac{d^2 u_i}{dt^2} \approx \frac{u_i^{k+1} - 2u_i^k + u_i^{k-1}}{\Delta t^2} \tag{10.47}$$

for the time derivative, and for the term $[K]\{u\}$, we will use a generalized average centered at time k:

$$u \approx \frac{\theta}{2} u_i^{k+1} + (1-\theta) u_i^k + \frac{\theta}{2} u_i^{k-1} \tag{10.48}$$

Use of these approximations gives

$$[M] \left\{u^{k+1}\right\} - 2 [M] \left\{u^k\right\} + [M] \left\{u^{k-1}\right\}$$

$$+\Delta t^2 \frac{\theta}{2} [K] \left\{u^{k+1}\right\} + \Delta t^2 (1-\theta) [K] \left\{u^k\right\} + \Delta t^2 \frac{\theta}{2} [K] \left\{u^{k-1}\right\} = \Delta t^2 \{R\} \tag{10.49}$$

Rearrangement gives the final discrete system:

$$\left[[M] + \frac{\theta}{2} \Delta t^2 [K]\right] \left\{u^{k+1}\right\} = \left[2[M] - (1-\theta) \Delta t^2 [K]\right] \left\{u^k\right\}$$

$$- \left[[M] + \frac{\theta}{2} \Delta t^2 [K]\right] \left\{u^{k-1}\right\} + \Delta t^2 \{R\} \tag{10.50}$$

The discrete system entails repetitive solution of

$$[A] \left\{u^{k+1}\right\} = [B] \left\{u^k\right\} + [C] \left\{u^{k-1}\right\} + \Delta t^2 \left\{R^k\right\} \tag{10.51}$$

with time-invariant matrices

$$A_{ij} = \left\langle \phi_j \phi_i + \frac{\theta}{2} \Delta t^2 \, C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.52}$$

$$B_{ij} = \left\langle 2\phi_j \phi_i - (1-\theta) \Delta t^2 \, C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.53}$$

$$C_{ij} = \left\langle -\phi_j \phi_i - \frac{\theta}{2} \Delta t^2 \, C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.54}$$

and so forth. Note that here we require initial values for $\left\{u^k\right\}$ *and* $\left\{u^{k-1}\right\}$.

10.6 Example: Telegraph Equation

Adding a loss term to the wave equation gives us the telegraph equation:

$$\frac{\partial^2 u}{\partial t^2} + \tau \frac{\partial u}{\partial t} - \nabla \cdot C^2 \nabla u = r \tag{10.55}$$

The loss term will cause systematic adjustments in the Galerkin weak form and the lumped and discrete systems. The weak form is:

$$\left\langle \frac{\partial^2 u}{\partial t^2} \phi_i \right\rangle + \left\langle \tau \frac{\partial u}{\partial t} \phi_i \right\rangle + \left\langle C^2 \nabla u \cdot \nabla \phi_i \right\rangle - \oint C^2 \nabla u \cdot \hat{\mathbf{n}} \, ds = \langle r \phi_i \rangle \tag{10.56}$$

and the lumped system is:

$$\sum_j \frac{d^2 u_j}{dt^2} \langle \phi_j \phi_i \rangle + \sum_j \frac{du_j}{dt} \langle \tau \phi_j \phi_i \rangle + \sum_j u_j \left\langle C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle = \oint C^2 \nabla u \cdot \hat{\mathbf{n}} \, ds + \langle r \phi_i \rangle \tag{10.57}$$

or in matrix form,

$$[M] \left\{ \frac{d^2 u}{dt^2} \right\} + [T] \left\{ \frac{du}{dt} \right\} + [K] \{u\} = \{R\} \tag{10.58}$$

with $[M]$ and $[K]$ and $\{R\}$ as in the wave equation example, and

$$T_{ij} = \langle \tau \phi_j \phi_i \rangle \tag{10.59}$$

The loss term requires a temporal discretization; for example

$$\frac{du_i}{dt} \approx \frac{u_i^{k+1} - u_i^{k-1}}{2\Delta t} \tag{10.60}$$

Adding this to the wave equation development gives us

$$\begin{aligned} &[M] \left\{ u^{k+1} \right\} - 2 [M] \left\{ u^k \right\} + [M] \left\{ u^{k-1} \right\} + \frac{\Delta t}{2} [T] \left\{ u^{k+1} \right\} - \frac{\Delta t}{2} [T] \left\{ u^{k-1} \right\} \\ &+ \Delta t^2 \frac{\theta}{2} [K] \left\{ u^{k+1} \right\} + \Delta t^2 (1-\theta) [K] \left\{ u^k \right\} + \Delta t^2 \frac{\theta}{2} [K] \left\{ u^{k-1} \right\} = \Delta t^2 \{R\} \end{aligned} \tag{10.61}$$

Rearrangement gives the final discrete system:

$$\begin{aligned} &\left[[M] + \frac{\Delta t}{2} [T] + \frac{\theta}{2} \Delta t^2 [K] \right] \left\{ u^{k+1} \right\} \\ &= \left[2[M] - (1-\theta) \Delta t^2 [K] \right] \left\{ u^k \right\} + \left[-[M] + \frac{\Delta t}{2} [T] - \frac{\theta}{2} \Delta t^2 [K] \right] \left\{ u^{k-1} \right\} + \Delta t^2 \{R\} \end{aligned} \tag{10.62}$$

As in the wave equation, this discrete system entails repetitive solution of

$$[A] \left\{ u^{k+1} \right\} = [B] \left\{ u^k \right\} + [C] \left\{ u^{k-1} \right\} + \Delta t^2 \left\{ R^k \right\} \tag{10.63}$$

with time-invariant matrices

$$A_{ij} = \left\langle \phi_j \phi_i \left(1 + \frac{\tau \Delta t}{2} \right) + \frac{\theta}{2} \Delta t^2 \, C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.64}$$

$$B_{ij} = \left\langle 2\phi_j \phi_i - (1-\theta) \Delta t^2 \, C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.65}$$

$$C_{ij} = \left\langle -\phi_j \phi_i \left(1 - \frac{\tau \Delta t}{2} \right) - \frac{\theta}{2} \Delta t^2 \, C^2 \nabla \phi_j \cdot \nabla \phi_i \right\rangle \tag{10.66}$$

Chapter 11

Vector Problems

11.1 Introduction

In previous sections, we have concentrated solely on *scalar problems i.e.* problems in which the unknown field is described in terms of a single scalar variable. Many classical problems of mathematical physics are posed in terms of *vector fields*, where the unknown field comprises more than one scalar variable, for example a displacement vector in solid mechanics, a force field in classical electromagnetics, or a velocity field in fluid mechanics. These will be treated by example herein, in order to introduce some added features of common approaches. This field is broad, complex, and highly discipline-specific. The material here is strictly introductory. We will stick to 2-D problems.

In general there are two approaches to Finite Element discretization. In the first, we simply utilize the customary *scalar bases* $\phi_j(x,y)$ as in scalar problems; and approximate a vector field $\mathbf{V}$ in terms of unknown *vectors* $\mathbf{V}_j$:

$$\mathbf{V}(x,y) \simeq \sum_j \mathbf{V}_j \phi_j(x,y) \tag{11.1}$$

Alternatively, we may invent *vector bases* $\mathbf{\Phi}_j(x,y)$ and express a vector field as

$$\mathbf{V}(x,y) \simeq \sum_j V_j \mathbf{\Phi}_j(x,y) \tag{11.2}$$

In the case of scalar bases, we allow the most general vector field expansion within the limits of common local polynomial interpolation. The vector bases are typically constrained to have certain desirable vector properties *a priori*, for example zero divergence, integral properties or geometric constraints. These would be selected to represent specific physical features of a given problem and can be highly discipline-specific.

11.2 Gradient of a Scalar

Fundamental to many of the formulations leading to scalar PDE's is a gradient-flux relationship of the form

$$\mathbf{q} = -\nabla\psi \tag{11.3}$$

where ψ is a scalar potential. Coupled with a divergence criterion,

$$\nabla \cdot \mathbf{q} = -\sigma \tag{11.4}$$

we obtain the familiar Poisson equation

$$\nabla \cdot \nabla \psi = \sigma \tag{11.5}$$

Generalizations and extensions of this idea abound in classical mathematical physics. We have already studied methods for computing the scalar potential ψ; here we look at computation of the flux vector $\mathbf{q}$, once ψ is known. This is the simplest vector problem.

Why is this a problem at all? Since ψ is already expressed as a unique continuous and differentiable function $\psi(x,y) = \sum \psi_j \phi_j(x,y)$ then its derivative is also available and computable everywhere, using the same general apparatus used to assemble the computation of ψ. In fact the simple solution

$$\mathbf{q}(x,y) = -\sum_j \psi_j \nabla \phi_j(x,y) \tag{11.6}$$

is a workable one, easy to compute and relatively care-free. Of course, this is an inherently weak estimate for $\mathbf{q}$, on three grounds. First, the estimate will be a lower order polynomial than $\psi(x,y)$. For example, on C^o elements, $\mathbf{q}$ will not be unique along element sides; it will change abruptly in magnitude and/or direction as one moves from element to element. On common linear triangles, this approximation produces a $\mathbf{q}$ field which is piecewise constant – one constant vector per element, with no interpolating polynomial. Second, we lack any direct method for enforcing Neumann boundary conditions beyond their already weak enforcement on the ψ field. And third, simple differentiation of a numerical field is susceptible to precision loss when ψ is highly resolved. This is an old problem in numerical analysis – essentially, as resolution increases, derivative estimates lose precision before the field does. So, we seek a remedy to these problems – we want to control the basis for $\mathbf{q}$; look for an enforcement mechanism for Neumann boundary data; and integrate $\mathbf{q} = -\nabla\psi$ to avoid amplifying roundoff error.

The example below uses the Galerkin approach to achieve these goals.

Galerkin Form

The weak form will be a MWR statement of 11.4 with respect to a set of scalar weights ϕ_i:

$$\langle \mathbf{q}, \phi_i \rangle = -\langle \nabla\psi, \phi_i \rangle \tag{11.7}$$

Expanding $\mathbf{q}$ in the scalar basis ϕ,

$$\mathbf{q}(x,y) = \sum_j \mathbf{q}_j \phi_j(x,y) \tag{11.8}$$

we obtain the Galerkin approximation

$$\sum_j \mathbf{q}_j \langle \phi_j, \phi_i \rangle = -\langle \nabla\psi, \phi_i \rangle \tag{11.9}$$

which has scalar components

$$\sum_j q_{xj} \langle \phi_j, \phi_i \rangle = -\hat{\mathbf{x}} \cdot \langle \nabla\psi, \phi_i \rangle = -\left\langle \frac{\partial \psi}{\partial x} \phi_i \right\rangle \tag{11.10}$$

$$\sum_j q_{yj} \langle \phi_j, \phi_i \rangle = -\hat{\mathbf{y}} \cdot \langle \nabla\psi, \phi_i \rangle = -\left\langle \frac{\partial \psi}{\partial y} \phi_i \right\rangle \tag{11.11}$$

These may be solved for nodal values of q_{xj} and q_{yj}. By inspection we can see that this procedure will smooth out the discontinuities in $\nabla\psi$ by integrating the element-level derivatives over several elements. The mass matrix $\langle\phi_j, \phi_i\rangle$ on the left-hand side provides an additional smoothing influence.

It is convenient to reexpress these Galerkin approximations as a 2-D vector equation

$$\sum_j [\mathbf{K}_{ji}]\, \mathbf{q}_j^{xy} = -\mathbf{R}_i^{xy} \tag{11.12}$$

with

$$[\mathbf{K}_{ji}] = \begin{bmatrix} \langle\phi_j, \phi_i\rangle & 0 \\ 0 & \langle\phi_j, \phi_i\rangle \end{bmatrix} = \langle\phi_j, \phi_i\rangle\, [\,\mathbf{I}\,] \tag{11.13}$$

$$\mathbf{R}_i^{xy} = \begin{Bmatrix} \left\langle \frac{\partial\psi}{\partial x}, \phi_i \right\rangle \\ \left\langle \frac{\partial\psi}{\partial y}, \phi_i \right\rangle \end{Bmatrix} \tag{11.14}$$

$$\mathbf{q}_j^{xy} = \begin{Bmatrix} q_j^x \\ q_j^y \end{Bmatrix} \tag{11.15}$$

This is the i^{th} Galerkin gradient flux relation in the (x, y) system.

Natural Local Coordinate Systems and Neumann Boundaries

Boundary conditions on $\mathbf{q}$ naturally occur in terms of the Neumann boundary condition, $\hat{\mathbf{n}} \cdot \mathbf{q} = q^n$, where n is the coordinate normal to the boundary and $q^n \simeq -\frac{\partial\psi}{\partial n}$ is the Neumann data. Suppose there are nodal values q_i^n specified along a Neumann boundary. Then we need a constraint of the form

$$\hat{\mathbf{n}} \cdot \mathbf{q}_i = q_i^n \tag{11.16}$$

or in terms of q_i^x and q_i^y,

$$(\hat{\mathbf{n}} \cdot \hat{\mathbf{x}}) q_i^x + (\hat{\mathbf{n}} \cdot \hat{\mathbf{y}}) q_i^y = q_i^n \tag{11.17}$$

Adding a constraint requires sacrificing one of the Galerkin equations. Intuitively, we expect that $\nabla\psi$ is the weakest (in terms of discretization error) in the direction normal to a boundary; and therefore the normal component of 11.9 could be replaced by 11.16. Essentially, we are using Neumann data as a strong condition on the gradient.

This procedure can be automated easily. First, any vector $\mathbf{V}$ expressed in (x, y) can be projected into any local (n, s) coordinate system by the rotation:

$$\begin{Bmatrix} V^n \\ V^s \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} V^x \\ V^y \end{Bmatrix} \tag{11.18}$$

where θ is the angle between the x-axis and the n-axis.[1] The inverse relationship is

$$\begin{Bmatrix} V^x \\ V^y \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} V^n \\ V^s \end{Bmatrix} \tag{11.19}$$

Equivalently,

$$\mathbf{V}^{ns} = [\mathbf{A}]\mathbf{V}^{xy} \tag{11.20}$$

[1]Both coordinate systems are presumed to be right-handed; and θ is positive in the same sense, *i.e.* CCW.

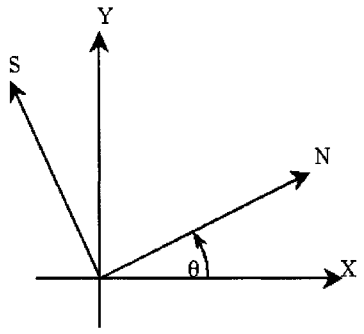


Figure 11.1: Coordinate rotation conventions.

$$\mathbf{V}^{xy} = [\mathbf{A}]^{-1}\mathbf{V}^{ns} = [\mathbf{A}]^{T}\mathbf{V}^{ns} \tag{11.21}$$

The rotation matrices have, *a priori*, the orthogonality property $[\mathbf{A}]^T = [\mathbf{A}]^{-1}$. ($[\mathbf{A}]^T$ indicates the transpose of $[\mathbf{A}]$.) Figure 11.1 illustrates the geometry convention.

We presume the existence of distinct local normal directions $\hat{\mathbf{n}}_i$, (equivalently, θ_i), associated with boundary nodes.[2] There will thus be a 2×2 rotation matrix $[\mathbf{A}_i]$ associated with each boundary node. For generality, let $[\mathbf{A}_i] = [\mathbf{I}]$ for interior nodes.

To convert the Galerkin equations 11.12 to the (n, s) system, premultiply by $[\mathbf{A}_i]$:

$$\sum_j [\mathbf{A}_i]\,[\mathbf{K}_{ji}]\,\mathbf{q}_j^{xy} = -\,[\mathbf{A}_i]\,\mathbf{R}_i^{xy} = -\mathbf{R}_i^{ns} \tag{11.22}$$

This expresses the i^{th} Galerkin gradient flux relations in the (n, s) system. But we still have the unknown vectors $\mathbf{q}_j^{xy}$ in the x, y system. These can be eliminated in favor of $\mathbf{q}_j^{ns}$ by substituting $\mathbf{q}_j^{xy} = [\mathbf{A}_j]^*\mathbf{q}_j^{ns}$ as in equation 11.21:

$$\sum_j [\mathbf{A}_i]\,[\mathbf{K}_{ji}]\,[\mathbf{A}_j]^T\,\mathbf{q}_j^{ns} = -\,[\mathbf{A}_i]\,\mathbf{R}_i^{xy} = -\mathbf{R}_i^{ns} \tag{11.23}$$

Dropping the superscripts in favor of a font change, we have the gradient flux equation weighted by ϕ_i, and expressed in completely local coordinates:

$$\sum_j [\mathcal{K}_{ji}]\,\mathcal{Q}_j^{ns} = -\mathcal{R}_i^{ns} \tag{11.24}$$

Assembling these gives the matrix equation

$$[\mathcal{K}]\,\mathcal{Q} = -\mathcal{R} \tag{11.25}$$

with $[\mathcal{K}]$ comprising 2×2 submatrices $[\mathcal{K}_{ij}]$:

$$[\mathcal{K}_{ij}] = [\mathbf{A}_i]\,[\mathbf{K}_{ij}]\,[\mathbf{A}_j]^T \tag{11.26}$$

[2]For a discussion of the discretization of $\hat{\mathbf{n}}_i$, see [107].

and

$$\mathcal{Q}_j = [\mathbf{A}_j]\,\mathbf{q}_j \tag{11.27}$$

$$\mathcal{R}_i = [\mathbf{A}_i]\,\mathbf{R}_i \tag{11.28}$$

$[\mathcal{K}]$ is an orthogonal transformation of $[\mathbf{K}]$. It preserves several useful properties of $[\mathbf{K}]$, including positive-definiteness and symmetry if they apply to $[\mathbf{K}]$.

Effectively we have interleaved the n- and s- components of the gradient relationship WR_1^n, WR_1^s, WR_2^n, WR_2^s, etc. Likewise, the n- and s - components of $\mathbf{q}$ are interleaved q_1^n, q_1^s , q_2^n, q_2^s, etc.; and the same for $\mathbf{R}$. Row $2i-1$ is the normal Galerkin approximation weighted by ϕ_i; row $2i$ is the tangential Galerkin approximation weighted by ϕ_i. Similarly, columns $2i-1$ and $2i$ correspond respectively to the normal and tangential unknowns at node j.

Note that $[\mathcal{K}_{ii}] = [\mathbf{K}_{ii}]$ (the special case $i = j$), *i.e.* the rotation $[\mathbf{A}_i]\,[\mathbf{K}_{ii}]\,[\mathbf{A}_i]^T$ has no effect due to the orthogonality of $[\mathbf{A}_i]$ and the fact that $[\mathbf{K}_{ii}]$ is a diagonal matrix of the form $\langle\phi_i, \phi_i\rangle\,[\mathbf{I}]$.

We are now set up to enforce strong (Dirichlet) constraints on q^n, preserving only the tangential component of the gradient flux relationship along a Neumann boundary. Following the solution for $\mathcal{Q}$, it is a simple matter to recover $\mathbf{q}$ in the (x, y) system in a node-by-node fashion:

$$\mathbf{q}_j = [\mathbf{A}_j]^T\,\mathcal{Q}_j \tag{11.29}$$

Dirichlet Boundaries

On boundaries where ψ is known, there is no prior information about q^n. As it stands, the above procedure will produce weak approximations for q^n because it is inherently biased toward the inside of the boundary where $\nabla\psi$ is defined. We can improve on this.

Assume a Poisson equation governs (equation 11.5) in Galerkin form

$$-\langle\nabla\psi, \nabla\phi_i\rangle = \langle\sigma\phi_i\rangle - \oint \frac{\partial\psi}{\partial n}\,\phi_i\,ds \tag{11.30}$$

The familiar practice in obtaining ψ is to "discard the Galerkin equation weighted with ϕ_i" when node i lies on a Dirichlet boundary – in favor of strong specification of ψ_i. This conveniently removed the unknown flux $\partial\phi/\partial n$ from consideration along that boundary. But that is exactly the information we would like to have now. So, collect all the "unused" Galerkin equations and "solve" for $q^n = -\partial\phi/\partial n$:

$$\oint q^n\,\phi_i\,ds = -\langle\nabla\psi, \nabla\phi_i\rangle - \langle\sigma\phi_i\rangle \tag{11.31}$$

Having already solved for ψ, everything on the RHS here is known. The LHS presents the boundary function q^n which may be expressed in the basis

$$q^n(s) = \sum_j q_j^n \phi_j(s) \tag{11.32}$$

and we acquire the boundary flux relations

$$\sum_j M_{ji} q_j^n = -\langle\nabla\psi, \nabla\phi_i\rangle - \langle\sigma\phi_i\rangle \tag{11.33}$$

for each node i on the Dirichlet boundary, with

$$M_{ji} = \oint \phi_j \, \phi_i \, ds \tag{11.34}$$

This equation is stronger than the normal Galerkin gradient-flux equation. Scrapping the latter in favor of 11.33 amounts to a sort of Neumann or weak constraint on q^n along boundaries where Dirichlet data is enforced on ψ.

11.3 Elasticity

Consider the classical case of linear elasticity, in which we seek to know the stress distribution in a plane continuum in terms of the stress tensor $[\sigma(x, y)]$, in response to a known body force $\beta(x, y)$:

$$\nabla \cdot [\sigma] = \beta \tag{11.35}$$

Here $[\sigma(x, y)]$ is a tensor (matrix) (a vector of vectors); its scalar entries σ_{ij} represent the force per unit area in direction i, exerted on a surface normal to direction j. Its divergence is a vector with x and y components indicating unbalanced stress in the x- and y- directions. Hence equation 11.35 represents Newton's second law for a solid without acceleration. In 2-D, we may reduce $[\sigma]$ to its two diagonal components σ_{xx}, σ_{yy}, which represent normal stress in the x and y directions, and the two off-diagonal components τ_{xy} and τ_{yx}, which represent tangential or shear stresses. The two shears are required to be equal; so we have exactly three components of stress: σ_{xx}, σ_{yy}, and τ. In terms of these, we may write the two scalar parts of 11.35. For the x component (the force balance in the x direction), we have

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau}{\partial y} = \beta_x \tag{11.36}$$

and for the y component,

$$\frac{\partial \tau}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} = \beta_y \tag{11.37}$$

Weak Form

The weak form of 11.35, with scalar weighting functions $\phi_i(x, y)$, is

$$\langle \nabla \cdot [\sigma], \phi_i \rangle = \langle \beta, \phi_i \rangle \tag{11.38}$$

and if we integrate the divergence term by parts, we have

$$\langle [\sigma], \nabla \phi_i \rangle = - \langle \beta, \phi_i \rangle + \oint [\sigma] \cdot \hat{\mathbf{n}} \, ds \tag{11.39}$$

The scalar components of this are: in the x direction,

$$\left\langle \sigma_{xx}, \frac{\partial \phi_i}{\partial x} \right\rangle + \left\langle \tau, \frac{\partial \phi_i}{\partial y} \right\rangle = \hat{\mathbf{x}} \cdot \mathbf{R_i} \tag{11.40}$$

and in the y direction,

$$\left\langle \tau, \frac{\partial \phi_i}{\partial x} \right\rangle + \left\langle \sigma_{yy}, \frac{\partial \phi_i}{\partial y} \right\rangle = \hat{\mathbf{y}} \cdot \mathbf{R_i} \tag{11.41}$$

where $\mathbf{R_i}$ is the right hand side vector which includes forces exerted on the boundary surface ($[\sigma]\cdot\hat{\mathbf{n}}$) and forces exerted on the interior of the domain (β):

$$\mathbf{R_i} = -\langle\beta,\phi_i\rangle + \oint [\sigma]\cdot\hat{\mathbf{n}}\phi_i \, ds \tag{11.42}$$

The quantity $\hat{\mathbf{x}}\cdot\mathbf{R_i}$ is the applied force in the $\mathbf{x}$ direction:

$$\hat{\mathbf{x}}\cdot\mathbf{R_i} = -\langle\hat{\mathbf{x}}\cdot\beta,\phi_i\rangle + \oint \hat{\mathbf{x}}\cdot[\sigma]\cdot\hat{\mathbf{n}}\phi_i \, ds \tag{11.43}$$

and similarly for $\hat{\mathbf{y}}\cdot\mathbf{R_i}$.

Constitutive Relations

Next we introduce the displacement field vector $\mathbf{D}$ with scalar components U, V in the x and y directions, respectively. It is asserted through a blend of observation and theory that the stress tensor components are completely described by the derivatives of $\mathbf{D}$, *i.e.* the strain tensor. Two standard cases are common, plane stress and plane strain. The constitutive relations for these cases are listed below. In each case there are two parameters E (Young's Modulus) and ν (Poisson ratio) which are considered known properties of the elastic medium.

Plane Stress:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau \end{Bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}\begin{Bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial V}{\partial y} \\ \frac{\partial U}{\partial y}+\frac{\partial V}{\partial x} \end{Bmatrix} \tag{11.44}$$

Plane Strain:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau \end{Bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}\begin{Bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial V}{\partial y} \\ \frac{\partial U}{\partial y}+\frac{\partial V}{\partial x} \end{Bmatrix} \tag{11.45}$$

Insertion of either of these into equations (11.40, 11.41) produces the weak forms of the elasticity equations in terms of displacement. For Plane Stress, we have respectively the x- and y-direction force balances:

$$\left\langle \frac{E}{1-\nu^2}\left(\frac{\partial U}{\partial x}+\nu\frac{\partial V}{\partial y}\right), \frac{\partial\phi_i}{\partial x}\right\rangle + \left\langle \frac{E}{2(1+\nu)}\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right), \frac{\partial\phi_i}{\partial y}\right\rangle = \hat{\mathbf{x}}\cdot\mathbf{R_i} \tag{11.46}$$

$$\left\langle \frac{E}{2(1+\nu)}\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right), \frac{\partial\phi_i}{\partial x}\right\rangle + \left\langle \frac{E}{1-\nu^2}\left(\frac{\partial V}{\partial y}+\nu\frac{\partial U}{\partial x}\right), \frac{\partial\phi_i}{\partial y}\right\rangle = \hat{\mathbf{y}}\cdot\mathbf{R_i} \tag{11.47}$$

Equivalently, collecting U and V terms together:

$$\left\langle \frac{E}{1-\nu^2}\frac{\partial U}{\partial x}, \frac{\partial\phi_i}{\partial x}\right\rangle + \left\langle \frac{E}{2(1+\nu)}\frac{\partial U}{\partial y}, \frac{\partial\phi_i}{\partial y}\right\rangle + \left\langle \frac{E}{2(1+\nu)}\frac{\partial V}{\partial x}, \frac{\partial\phi_i}{\partial y}\right\rangle + \left\langle \frac{\nu E}{1-\nu^2}\frac{\partial V}{\partial y}, \frac{\partial\phi_i}{\partial x}\right\rangle = \hat{\mathbf{x}}\cdot\mathbf{R_i} \tag{11.48}$$

$$\left\langle \frac{E}{2(1+\nu)}\frac{\partial U}{\partial y}, \frac{\partial\phi_i}{\partial x}\right\rangle + \left\langle \frac{\nu E}{(1-\nu^2)}\frac{\partial U}{\partial x}, \frac{\partial\phi_i}{\partial y}\right\rangle + \left\langle \frac{E}{2(1+\nu)}\frac{\partial V}{\partial x}\frac{\partial\phi_i}{\partial x}, \right\rangle + \left\langle \frac{E}{1-\nu^2}\frac{\partial V}{\partial y}, \frac{\partial\phi_i}{\partial y}\right\rangle = \hat{\mathbf{y}}\cdot\mathbf{R_i} \tag{11.49}$$

Galerkin Approximation

Finally, we will expand U and V in the basis ϕ to get to the Galerkin approximations:

$$U(x,y) = \sum_j U_j \phi_j(x,y) \tag{11.50}$$

$$V(x,y) = \sum_j V_j \phi_j(x,y) \tag{11.51}$$

This will lead to a matrix formulation of the system in terms of the unknown displacement $\mathbf{D}$ in response to the known forcing $\mathbf{R}$, with a system matrix $[K]$ comprising integrals of the basis functions, their derivatives, and the material properties:

$$[K]\{\mathbf{D}\} = \{\mathbf{R}\} \tag{11.52}$$

Now we need to declare some organization in this. First, arrange the WR equations in the following order: WR_{1x}, WR_{1y}, WR_{2x}, WR_{2y}, etc. That is, interleave the x and y force balances on alternating rows of the matrix equation. Similarly, interleave the x and y components of $\mathbf{D}$ and the right-hand side $\mathbf{R}$ in the same way. From equations 11.48 and 11.49 above, we get

$$\sum_j [\mathbf{K}_{ji}]\{\mathbf{D}_j\} = \{\mathbf{R}_i\} \tag{11.53}$$

with $[\mathbf{K}_{ji}]$ a 2×2 matrix of coefficients

$$[\mathbf{K}_{ji}] = \begin{bmatrix} \left\langle \frac{E}{1-\nu^2}\frac{\partial\phi_j}{\partial x}, \frac{\partial\phi_i}{\partial x}\right\rangle + \left\langle \frac{E}{2(1+\nu)}\frac{\partial\phi_j}{\partial y}, \frac{\partial\phi_i}{\partial y}\right\rangle & \left\langle \frac{E}{2(1+\nu)}\frac{\partial\phi_j}{\partial x}, \frac{\partial\phi_i}{\partial y}\right\rangle + \left\langle \frac{\nu E}{1-\nu^2}\frac{\partial\phi_j}{\partial y}, \frac{\partial\phi_i}{\partial x}\right\rangle \\ \left\langle \frac{E}{2(1+\nu)}\frac{\partial\phi_j}{\partial y}, \frac{\partial\phi_i}{\partial x}\right\rangle + \left\langle \frac{\nu E}{(1-\nu^2)}\frac{\partial\phi_j}{\partial x}, \frac{\partial\phi_i}{\partial y}\right\rangle & \left\langle \frac{E}{2(1+\nu)}\frac{\partial\phi_j}{\partial x}\frac{\partial\phi_i}{\partial x}, \right\rangle + \left\langle \frac{E}{1-\nu^2}\frac{\partial\phi_j}{\partial y}, \frac{\partial\phi_i}{\partial y}\right\rangle \end{bmatrix} \tag{11.54}$$

and the two-dimensional vectors associated with nodal displacement $\mathbf{D}$ and forcing $\mathbf{R}$:

$$\mathbf{D}_j = \begin{Bmatrix} U_j \\ V_j \end{Bmatrix} \tag{11.55}$$

$$\mathbf{R}_i = \begin{Bmatrix} R_i^x \\ R_i^y \end{Bmatrix} \tag{11.56}$$

Natural Local Coordinate Systems

Proceeding as in the case of the Gradient calculation above, we introduce nodal coordinate rotations $[A_i]$, and use these to transform Galerkin equations 11.53 into local (n,s) coordinate systems:

$$\sum_j [\mathcal{K}_{ji}]\{\mathcal{D}_j\} = \{\mathcal{R}_i\} \tag{11.57}$$

with

$$[\mathcal{K}_{ji}] = [\mathbf{A_i}][\mathbf{K_{ij}}][\mathbf{A_j}]^T \tag{11.58}$$

$$\{\mathcal{D}\}_i = [\mathbf{A_j}]\{\mathbf{D}\}_i \tag{11.59}$$

and

$$\{\mathcal{R}\}_i = [\mathbf{A_i}]\{\mathbf{R}\}_i \tag{11.60}$$

This entire set of equations is now represented in completely local coordinates:

$$[\mathcal{K}]\{\mathcal{D}\} = \{\mathcal{R}\} \tag{11.61}$$

This system is now ready for the application of boundary conditions in natural local coordinates. Essentially we need two pieces of information (Normal, Tangential) at all boundary nodes. The natural classification is Type I: displacement specified (Dirichlet); and Type II: stress specified (Neumann). Various mixtures of these in the normal and tangential directions are natural. For example, a boundary with no normal displacement and no tangential stress is a common idealization: a rigid, lubricated wall.

The Type II boundary is the "natural" one; the boundary integral in $\mathbf{R}$ is the vehicle for the stress data. On the Type I boundary, the procedure is familiar – strong enforcement of the Dirichlet displacement BC's instead of the associated Galerkin equations. The latter are then the vehicle for computing the unknown boundary stress via the boundary integral in $\mathbf{R}$.

Once the displacements are known, interior stress can be computed by differentiation of the constitutive relations (*e.g.* equations 11.44 or 11.45). The general Galerkin approach outlined above (in the section about the computation of gradients) is applicable. The Neumann data on the displacement problem (either specified originally, or derived as above) may be used as Dirichlet boundary data for the stress calculation.

References – Solid Mechanics

The literature on solid mechanics applications of FEM is vast. This is one of the earliest and most successful FEM applications. Today, engineers worldwide are able to solve 3-D vector problems on realistic, complex geometry, routinely. Standard references are to Zienkiewicz and co-workers [120, 121, 122, 123]. Segerlind [101] provides a valuable expository treatment. These works are assembled here for convenience.

- Segerlind, L.J. Applied Finite Element Analysis. Wiley, 1984.
- Zienkiewicz, O.C. The Finite Element Method in Engineering Science. McGraw-Hill, third edition, 1986.
- Zienkiewicz, O.C. and R.L. Taylor.The Finite Element Method. McGraw-Hill, fourth edition, 1987.
- Zienkiewicz, O.C. and R.L. Taylor. The Finite Element Method: Volume 1, The Basis. Butterworth-Heinemann, 2000.
- Zienkiewicz, O.C. and R.L. Taylor. The Finite Element Method: Volume 2, Solid Mechanics. Butterworth-Heinemann, 2000.

11.4 Electromagnetics

Here we look at the classic Maxwell equations describing vector fields $\mathbf{E}$ and $\mathbf{H}$ describing force fields experienced by electrically-charged particles at rest ($\mathbf{E}$) and moving ($\mathbf{H}$). Because these equations are posed in the frequency domain, all field quantities are complex quantities with both

amplitude and phase, at frequency ω. The quantity ι is introduced as $\sqrt{-1}$. The formulation here is in terms of general complex fields and coefficients. Implementation in a language which supports complex data declarations and elementary complex operations is assumed.

Of the many formulations for Maxwell's equations, we concentrate on the use of C^o elements, in 2-D. The basic formulation is in terms of continuous vector and scalar potentials, with a Lorentz gauge, following that given in Paulsen *et al.* [91]. The alternative vector bases are explored in *e.g.* Barton and Cendes [8].

Governing Equations

The classic Maxwell equations in primitive form are

$$\nabla \times \mathbf{E} = \iota\omega\mu\mathbf{H} \tag{11.62}$$

$$\nabla \cdot \epsilon\mathbf{E} = \rho \tag{11.63}$$

$$\nabla \times \mathbf{H} = \mathbf{J} - \iota\omega\epsilon\mathbf{E} \tag{11.64}$$

$$\nabla \cdot \mu\mathbf{H} = 0 \tag{11.65}$$

Boundary conditions are:

$$\hat{\mathbf{n}} \times \mathbf{E} = \mathbf{M_s} \quad on\,\Gamma_1 \tag{11.66}$$

$$\hat{\mathbf{n}} \times \mathbf{H} = -\mathbf{J_s} \quad on\,\Gamma_2 \tag{11.67}$$

For compatibility we require continuity among the source terms ($\mathbf{J}$, ρ):

$$\nabla \cdot \mathbf{J} = \iota\omega\rho \tag{11.68}$$

We may eliminate $\mathbf{H}$ between equations 11.62 and 11.64 and obtain the Helmholtz equation in $\mathbf{E}$:

$$\nabla \times \left(\frac{1}{\iota\omega\mu}\nabla \times \mathbf{E}\right) + \iota\omega\epsilon\mathbf{E} = \mathbf{J} \tag{11.69}$$

Physical conditions dictate discontinuities in $\mathbf{E}$ and $\mathbf{H}$ at interfaces where material properties change abruptly. Since this is a common occurrence in problems of practical concern, it presents important issues in selection of proper bases for these fields.

Potentials and Gauge

As in Boyse *et al.* [14] we introduce the scalar and vector potentials:

$$\mathbf{E} = \iota\omega\mathbf{A} - \nabla\Phi \tag{11.70}$$

$$\mathbf{H} = \frac{1}{\mu}\nabla \times \mathbf{A} \tag{11.71}$$

A "Gauge condition" specifying the divergence of $\mathbf{A}$ is needed to make the the potentials unique. We use a Lorentz Gauge (#2 in [14]):

$$\nabla \cdot \mathbf{A} = \iota\omega\epsilon\mu\Phi \tag{11.72}$$

This selection of Gauge results in continuity of both potentials $\mathbf{A}$ and Φ for heterogeneous media, notably where material properties change abruptly. As a result, these fields may be approximated in common C^o scalar bases.

Helmholtz Equations in the Potentials

Substituting the potentials for $\mathbf{E}$ in 11.69, and utilizing the fact that $\nabla \times \nabla f = 0$ for any function f, gives us

$$\nabla \times \left(\frac{1}{\iota\omega\mu}\nabla \times \iota\omega\mathbf{A}\right) + \iota\omega\epsilon\left(\iota\omega\mathbf{A} - \nabla\Phi\right) = \mathbf{J} \tag{11.73}$$

The last term on the LHS is further processed by substituting the Gauge condition as follows:

$$-\iota\omega\epsilon\nabla\Phi = -\nabla\left(\iota\omega\epsilon\Phi\right) + \Phi\nabla\left(\iota\omega\epsilon\right) \tag{11.74}$$

$$= -\nabla\left(\frac{1}{\mu}\nabla\cdot\mathbf{A}\right) + \Phi\nabla\left(\iota\omega\epsilon\right) \tag{11.75}$$

From here we find the final form of the *Helmholtz equation in* $\mathbf{A}$:

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \mathbf{A}\right) - \nabla\left(\frac{1}{\mu}\nabla\cdot\mathbf{A}\right) - \omega^2\epsilon\mathbf{A} + \iota\omega\Phi\nabla\epsilon = \mathbf{J} \tag{11.76}$$

Similarly, substituting the potentials for $\mathbf{E}$ into equation 11.63 gives us

$$\nabla\cdot\left(\iota\omega\epsilon\mathbf{A} - \epsilon\nabla\Phi\right) = \rho \tag{11.77}$$

Using the chain rule on the first term gives

$$\nabla\left(\iota\omega\epsilon\right)\cdot\mathbf{A} + \iota\omega\epsilon\nabla\cdot\mathbf{A} - \nabla\cdot\left(\epsilon\nabla\Phi\right) = \rho \tag{11.78}$$

Finally, insertion of the Gauge condition for $\nabla\cdot\mathbf{A}$ gives us the *Helmholtz equation in* Φ:

$$\nabla\cdot\epsilon\nabla\Phi + \omega^2\epsilon^2\mu\Phi - \iota\omega\mathbf{A}\cdot\nabla\epsilon = -\rho \tag{11.79}$$

The two Helmholtz equations in $\mathbf{A}$ and Φ will be solved on simple C^o elements with real, scalar bases.

The *boundary conditions* for these Helmholtz operators are as established in [14, 48]:

On Γ_1:

$$\hat{\mathbf{n}}\times\left(\iota\omega\mathbf{A} - \nabla\Phi\right) = \mathbf{M_s} \tag{11.80}$$

$$\Phi = \textit{arbitrary} \tag{11.81}$$

$$\nabla\cdot\mathbf{A} = \iota\omega\epsilon\mu\Phi \tag{11.82}$$

On Γ_2:

$$\hat{\mathbf{n}}\times\left(\frac{1}{\mu}\nabla\times\mathbf{A}\right) = -\mathbf{J_s} \tag{11.83}$$

$$\frac{\partial\Phi}{\partial n} = \textit{arbitrary} \tag{11.84}$$

$$\hat{\mathbf{n}}\cdot\epsilon(\iota\omega\mathbf{A} - \nabla\Phi) = \frac{1}{\iota\omega}\left(\hat{\mathbf{n}}\cdot\mathbf{J} - \nabla_s\cdot\mathbf{J_s}\right) \tag{11.85}$$

Together these guarantee that the solution of the ($\mathbf{A}$, Φ) Helmholtz system also satisfies the original primitive system. The arbitrary variation of Φ or $\frac{\partial\Phi}{\partial n}$ reflects an arbitrary (but necessary) allocation of the boundary data for $\hat{\mathbf{n}}\times\mathbf{E}$ or $\hat{\mathbf{n}}\cdot\mathbf{E}$ among the potentials when the boundary and the data are both smooth. Without this arbitrary allocation, the solution is indeterminate with a multiplicity of ($\mathbf{A}$, Φ) solutions equivalent to the same $\mathbf{E}$.

Weak Form

We use the weak version of equations (11.76, 11.79):[3]

$$\left\langle \left(\frac{1}{\mu}\nabla \times \mathbf{A}\right) \times \nabla\phi_i \right\rangle + \left\langle \left(\frac{1}{\mu}\nabla \cdot \mathbf{A}\right) \nabla\phi_i \right\rangle - < \omega^2\epsilon\mathbf{A}\phi_i > - < \iota\omega\epsilon\nabla(\Phi\phi_i) >=$$

$$\langle \mathbf{J}, \phi_i \rangle - \oint \hat{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \mathbf{A}\right) \phi_i \, ds + \oint \hat{\mathbf{n}} \left(\frac{1}{\mu}\nabla \cdot \mathbf{A} - \iota\omega\epsilon\Phi\right) \phi_i \, ds \tag{11.87}$$

$$< \nabla\phi_i \cdot \epsilon\nabla\Phi > - < \omega^2\epsilon^2\mu\Phi\phi_i > - < \iota\omega\epsilon\nabla \cdot (\phi_i\mathbf{A}) >= \langle \rho, \phi_i \rangle - \oint \hat{\mathbf{n}} \cdot \epsilon(\iota\omega\mathbf{A} - \nabla\Phi)\phi_i \, ds \tag{11.88}$$

where ϕ_i is a scalar basis associated with common C^0 finite elements. Expansion of $\mathbf{A}$ and Φ in the basis ϕ_i as in Paulsen *et al.* (1992) leads to the symmetric algebraic system $\sum_j \mathbf{K}_{ij}\mathbf{F}_j = \mathbf{R}_i$. In 2-D we have

$$\mathbf{K}_{ij} \begin{bmatrix} \langle \frac{1}{\mu}\nabla\phi_j \cdot \nabla\phi_i \rangle - \langle \omega^2\epsilon\phi_i\phi_j \rangle & \langle \frac{1}{\mu}(-\frac{\partial\phi_j}{\partial x}\frac{\partial\phi_i}{\partial y} + \frac{\partial\phi_j}{\partial y}\frac{\partial\phi_i}{\partial x}) \rangle & \langle -\iota\omega\epsilon(\phi_i\frac{\partial\phi_j}{\partial x} + \phi_j\frac{\partial\phi_i}{\partial x}) \rangle \\ \langle \frac{1}{\mu}(-\frac{\partial\phi_j}{\partial y}\frac{\partial\phi_i}{\partial x} + \frac{\partial\phi_j}{\partial x}\frac{\partial\phi_i}{\partial y}) \rangle & \langle \frac{1}{\mu}\nabla\phi_j \cdot \nabla\phi_i \rangle - \langle \omega^2\epsilon\phi_i\phi_j \rangle & \langle -\iota\omega\epsilon(\phi_i\frac{\partial\phi_j}{\partial y} + \phi_j\frac{\partial\phi_i}{\partial y}) \rangle \\ -\langle \iota\omega\epsilon(\phi_j\frac{\partial\phi_i}{\partial x} + \phi_i\frac{\partial\phi_j}{\partial x}) \rangle & -\langle \iota\omega\epsilon(\phi_j\frac{\partial\phi_i}{\partial y} + \phi_i\frac{\partial\phi_j}{\partial y}) \rangle & \langle \epsilon\nabla\phi_j \cdot \nabla\phi_i \rangle - \omega^2\epsilon^2\mu\phi_i\phi_j \rangle \end{bmatrix} \tag{11.89}$$

$\mathbf{F}_j = \{A_j^x, A_j^y, \Phi\}^*$, and $\mathbf{R}_i$ is the corresponding set of boundary and domain integrals of the forcing, weighted with ϕ_i:

$$\mathbf{R}_i = \left\{ \begin{array}{c} \hat{\mathbf{x}} \cdot \left(- \oint \hat{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \mathbf{A}\right) \phi_i \, ds + \oint \hat{\mathbf{n}} \left(\frac{1}{\mu}\nabla \cdot \mathbf{A} - \iota\omega\epsilon\Phi\right) \phi_i \, ds + \langle \mathbf{J}\phi_i \rangle \right) \\ \hat{\mathbf{y}} \cdot \left(- \oint \hat{\mathbf{n}} \times \left(\frac{1}{\mu}\nabla \times \mathbf{A}\right) \phi_i \, ds + \oint \hat{\mathbf{n}} \left(\frac{1}{\mu}\nabla \cdot \mathbf{A} - \iota\omega\epsilon\Phi\right) \phi_i \, ds + \langle \mathbf{J}\phi_i \rangle \right) \\ - \oint \hat{\mathbf{n}} \cdot \epsilon(\iota\omega\mathbf{A} - \nabla\Phi)\phi_i \, ds + \langle \rho\phi_i \rangle \end{array} \right\} \tag{11.90}$$

Notice that the $\oint$ terms contain all natural quantities: $\hat{\mathbf{n}} \times \mathbf{H}$, the Gauge, and $\hat{\mathbf{n}} \cdot \epsilon\mathbf{E}$.

This system is rotated into local (n, s) coordinates after assembly, as described above. The relevant rotation matrix $[\mathbf{A}_i]$ here is

$$[\mathbf{A}_i] = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11.91}$$

A remarkable feature of this is the fact that all off-diagonal terms in $[\mathbf{K}]$ vanish identically for i, j on the interior – *i.e.* for equations completely removed from any boundary. This leaves a simple Galerkin-Helmholtz equation in all three variables separately. At the boundaries, the off-diagonal terms are nonzero; there, they taylor the Helmholtz relations to the natural boundary integrals. This permits an enormous efficiency in assembly time as well as storage. As stated above, $[\mathbf{K}]$ is also symmetric.

[3]The first two terms on the LHS of 11.87 have been integrated by parts (see Appendix). The last term on the LHS is:

$$\begin{aligned} \langle \iota\omega\Phi\nabla\epsilon, \phi_i \rangle &= \langle \nabla(\iota\omega\Phi\phi_i\epsilon) \rangle - \langle \epsilon\nabla(\iota\omega\Phi\phi_i) \rangle \\ &= \oint \hat{\mathbf{n}}(\iota\omega\Phi\phi_i\epsilon) \, ds - \langle \iota\omega\epsilon\nabla(\Phi\phi_i) \rangle \end{aligned} \tag{11.86}$$

which completes the Gauge condition in the $\oint$ term. Similar treatment of 11.79 produces the two parts of $\hat{\mathbf{n}} \cdot \mathbf{E}$ in the RHS $\oint$ term in 11.88.

Boundary Conditions

Boundary conditions are implemented on straight or smoothly-curved boundaries as in [91]:

- **Type I Boundary:** On Γ_1, (11.80) is satisfied by setting $\Phi = 0$ (arbitrarily) and $\hat{\mathbf{n}} \times \mathbf{A} = \frac{\mathbf{M}_s}{\iota\omega}$. Accordingly, (11.88) and the tangential components of (11.87) are not enforced. The normal component of (11.87) is enforced with boundary integrals naturally zero.
- **Type II Boundary:** On Γ_2, $\hat{\mathbf{n}} \cdot \mathbf{A}$ is dictated by (11.85) with the arbitrary assumption that $\frac{\partial \Phi}{\partial n} = 0$. Accordingly, the normal component of (11.87) is not enforced. The rest of the Galerkin equations are forced with natural boundary data, as in (11.83, 11.85).

Sharp corners (either physical corners, or poorly-resolved curvature) present special problems (see [15]).

Reconstructing E and H

Once $\mathbf{A}$ and Φ are known, we may compute the primitive fields $\mathbf{E}$ and $\mathbf{H}$ by differentiation:

$$\mathbf{E} = \iota\omega\mathbf{A} - \nabla\Phi \tag{11.92}$$

$$\mathbf{H} = \frac{1}{\mu}\nabla \times \mathbf{A} \tag{11.93}$$

The simplest strategy is direct differentiation. For example, point values of $\mathbf{E}$ at element centroids or Gauss points provides a workable result; it will be discontinuous at all element boundaries for C^o elements. A refined set of fields can be obtained by projecting these discontinuous functions onto the continuous basis ϕ by a Galerkin method:

$$\sum_j < \phi_i, \phi_j > \mathbf{E}_j = < (\iota\omega\mathbf{A} - \nabla\Phi), \phi_i > \tag{11.94}$$

In doing this one needs to be careful to enforce the physical discontinuity in $\mathbf{E}$ at the junction of distinct materials. The reader is referred to the literature cited for details.

References - E&M

The literature elaborates the many complexities associated with the Maxwell fields and their representation on FEM bases. The reader is referred in particular the the work of Boyse [14, 17, 16, 15], Cendes [8], Jiang [48], Lynch [78, 79, 76, 71, 77], Paulsen [91, 95, 92, 94, 93], and Yuan [118, 117].

The formulation described here employs scalar bases; it was introduced and tested by Boyse *et al.* (1992) and Paulsen *et al.* (1992). Corroborating support for the boundary condition treatment is given by Jiang *et al.* (1996). Alternative fomulations using vector bases are common; see for example Barton and Cendes (1987).

Below we list some of these works for convenience.

- Barton, M.L. and Z.J. Cendes, "New vector finite elements for three-dimensional magnetic field computation". *J. Appl. Phys.* **61**(8):3919-3921, 1987

- Boyse WE, Lynch DR, Paulsen KD and GN Minerbo, "Nodal based finite element modeling of Maxwell's equations in three dimensions," *IEEE Trans Antennas and Propagat* **40**:642-651, 1992
- WE Boyse and KD Paulsen, "Accurate Solutions of Maxwell's equations around PEC corners and highly curved surfaces using nodal finite elements," *IEEE Trans Antennas and Propagation* **45**(12):1758-1767, 1997
- Jiang, B-n, J. Wu, LA Povinelli, "The origin of spurious solutions in computational electromagnetics", *J. Comput. Phys* **125**: 104-123, 1996
- Lynch DR, Paulsen KD, and JW Strohbehn, "Hybrid element method for unbounded problems in hyperthermia," *Int. J. for Num. Meths. in Engg.* **23**:1915-1937, 1986
- Lynch DR, KD Paulsen and WE Boyse, "Synthesis of vector parasites in finite element Maxwell solutions," *IEEE Trans Microwave Theory and Techniques* **41**(8):1439-1448, 1993
- Paulsen, KD, WE Boyse, and DR Lynch, "Continuous potential Maxwell solutions on nodal-based finite elements," *IEEE Trans Antennas Propagat* **40**: 1192-1200, 1992
- Yuan X., Lynch DR, and KD Paulsen, "Importance of normal field continuity in inhomogeneous scattering calculations," *IEEE Trans. Microwave Theory and Techniques* **39**:638-642, 1991

11.5 Fluid Mechanics with Mixed Interpolation

Here we describe an approach (one of many) to discretizing two-dimensional fluid flows. The set of time-dependent, viscous Navier-Stokes equations at low Reynolds number is used. The solution scheme is the Galerkin finite element method, incorporating "mixed interpolation" with scalar C^o bases.

Governing equations

The governing equations comprise the scalar continuity equation and the two scalar components of the momentum equation:

Continuity Equation:

$$\frac{1}{b}\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{V} = 0 \tag{11.95}$$

X-Momentum Equation:

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} - \nabla \cdot \left(\frac{1}{Re}\nabla u\right) = 0 \tag{11.96}$$

Y-Momentum Equation:

$$\frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} - \nabla \cdot \left(\frac{1}{Re}\nabla v\right) = 0 \tag{11.97}$$

where p is the pressure, u and v are the x and y-components of the velocity $\mathbf{V}$, and b is the bulk modulus (a compressibility factor). The Reynolds number is defined as $Re = l_c v_c/\nu$, where l_c and v_c are the characteristic length and fluid speed of the system, and ν is the kinematic viscosity of the fluid. Nonlinear advective terms are ignored here.

Bases and Weights

The three time-dependent scalar fields (p, u, v) are discretized with finite elements in space, but left continuous in time for present purposes:

$$f(x, y, t) \simeq \sum_j f_j(t)\phi_j(x, y) \tag{11.98}$$

We will illustrate *mixed interpolation* (Taylor and Hughes [104], pp. 98–102). In this formulation, the spatial basis for pressure is of lower polynomial degree (linear) than that for velocity (quadratic). Both bases are scalar C^o functions. The pressure basis is used as the weighting set for the Continuity equation. Similarly, the velocity basis forms the weighting set for the momentum equation. So this is a Galerkin method – "mixed Galerkin".

Mixed Elements

A nine-node Lagrangian quadrilateral element is selected for both scalar components of the velocity field. The pressure field is expressed on the same geometric elements, but with only bilinear interpolation among the four corner nodes. A schematic of this curved quadrilateral element and its transformation from the global coordinates (x, y) to the standard local element coordinates (ξ, η) is presented in Figure 11.2. The mapping is isoparametric in the quadratic basis. Note the local node numbering convention – local nodes 1 through 4 are the corners. The Table which follows details the local variable numbering and the degrees of freedom at each node.

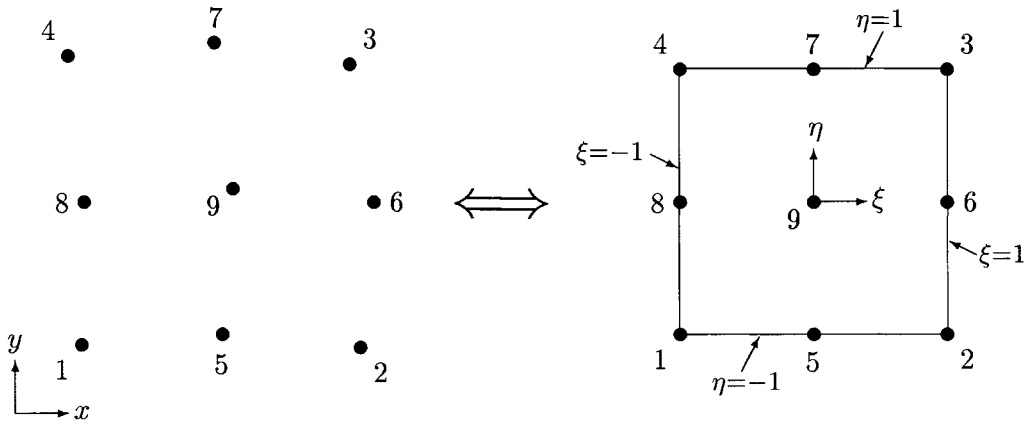


Figure 11.2: Isoparametric transformation of a nine-node quadrilateral element

Local node number	Variables present	Degrees of freedom
1, 2, 3, 4	p, u, v	3
5, 6, 7, 8, 9	u, v	2

$$Total\ degrees\ of\ freedom = 22$$

The pressure, p, is expanded in 2D bilinear bases $\psi(\xi, \eta)$. The two scalar components of velocity, (u, v), are expressed in terms of 2D biquadratic bases, $\varphi(\xi, \eta)$:

$$p(x, y, t) \simeq \sum_{j=1}^{4} p_j(t)\psi_j(\xi, \eta)$$

$$u(x,y,t) \simeq \sum_{j=1}^{9} u_j(t)\varphi_j(\xi,\eta)$$
$$v(x,y,t) \simeq \sum_{j=1}^{9} v_j(t)\varphi_j(\xi,\eta),$$

The bilinear (ψ) and biquadratic (φ) interpolation functions are as given in earlier chapters. The standard isoparametric mapping is in terms of φ:

$$x = \sum_{j=1}^{9} x_j\varphi_j(\xi,\eta) \qquad y = \sum_{j=1}^{9} y_j\varphi_j(\xi,\eta) \tag{11.99}$$

Weak Form

The weak form of the continuity equation is obtained with integration by parts of the divergence term:

$$\langle \frac{1}{b}\frac{\partial p}{\partial t}, \psi_i\rangle - \langle \mathbf{V}\cdot\nabla\psi_i\rangle = -\oint \mathbf{V}\cdot\hat{\mathbf{n}}\ \psi_i\ ds \tag{11.100}$$

This exposes the natural boundary integral of the flow normal to the boundary. The weak forms of the momentum equations are obtained with integration by parts of the Laplacian, which moves a viscous stress term into the boundary integral:

$$\langle \frac{\partial u}{\partial t}, \varphi_i\rangle + \langle \frac{\partial p}{\partial x}, \varphi_i\rangle + \langle \frac{1}{Re}\nabla u, \nabla\varphi_i\rangle = \oint \frac{1}{Re}\frac{\partial u}{\partial n}\ \varphi_i\ ds \tag{11.101}$$

$$\langle \frac{\partial v}{\partial t}, \varphi_i\rangle + \langle \frac{\partial p}{\partial y}, \varphi_i\rangle + \langle \frac{1}{Re}\nabla v, \nabla\varphi_i\rangle = \oint \frac{1}{Re}\frac{\partial v}{\partial n}\ \varphi_i\ ds \tag{11.102}$$

Note: an alternate strategy here would be to integrate the pressure gradient terms by parts, too. That would add a boundary integral of the pressure force normal to the boundary, supplementing the tangential viscous stress term already there.

Galerkin Equations

Expanding p in its basis ψ, and $\mathbf{V}$ in φ, we obtain the Galerkin system:

$$\sum_j \left(\frac{\partial p_j}{\partial t}\langle \frac{1}{b}\psi_j, \psi_i\rangle - u_j\langle \varphi_j, \frac{\partial\psi_i}{\partial x}\rangle - v_j\langle \varphi_j, \frac{\partial\psi_i}{\partial y}\rangle \right) = -\oint \mathbf{V}\cdot\hat{\mathbf{n}}\ \psi_i\ ds \tag{11.103}$$

$$\sum_j \left(\frac{\partial u_j}{\partial t}\langle \varphi_j, \varphi_i\rangle + p_j\langle \frac{\partial\psi_j}{\partial x}, \varphi_i\rangle + u_j\langle \frac{1}{Re}\nabla\varphi_j, \nabla\varphi_i\rangle \right) = \oint \frac{1}{Re}\frac{\partial u}{\partial n}\ \varphi_i\ ds \tag{11.104}$$

$$\sum_j \left(\frac{\partial v_j}{\partial t}\langle \varphi_j, \varphi_i\rangle + p_j\langle \frac{\partial\psi_j}{\partial y}, \varphi_i\rangle + v_j\langle \frac{1}{Re}\nabla\varphi_j, \nabla\varphi_i\rangle \right) = \oint \frac{1}{Re}\frac{\partial v}{\partial n}\ \varphi_i\ ds \tag{11.105}$$

If we collect the nodal variables and equations, we have

$$\sum_j \left(\mathbf{M}_{ij}\frac{d\mathcal{U}_j}{dt} + \mathbf{K}_{ij}\mathcal{U}_j \right) = \mathcal{R}_i \tag{11.106}$$

with

$$\mathbf{M}_{ij} = \begin{bmatrix} \langle \frac{1}{b}\psi_j, \psi_i \rangle & 0 & 0 \\ 0 & \langle \varphi_j, \varphi_i \rangle & 0 \\ 0 & 0 & \langle \varphi_j, \varphi_i \rangle \end{bmatrix} \tag{11.107}$$

$$\mathbf{K}_{ij} = \begin{bmatrix} 0 & -\langle \varphi_j \frac{\partial \psi_i}{\partial x} \rangle & -\langle \varphi_j \frac{\partial \psi_i}{\partial y} \rangle \\ \langle \varphi_j \frac{\partial \psi_i}{\partial x} \rangle & \langle \frac{1}{Re} \nabla \varphi_j, \nabla \varphi_i \rangle & 0 \\ \langle \varphi_j \frac{\partial \psi_i}{\partial y} \rangle & 0 & \langle \frac{1}{Re} \nabla \varphi_j, \nabla \varphi_i \rangle \end{bmatrix} \tag{11.108}$$

$$\mathcal{U}_j = \begin{Bmatrix} p_j \\ u_j \\ v_j \end{Bmatrix} \qquad \mathcal{R}_i = \begin{Bmatrix} -\oint \mathbf{V} \cdot \hat{\mathbf{n}}\ \psi_i\ ds \\ \oint \frac{1}{Re} \frac{\partial u}{\partial n}\ \varphi_i\ ds \\ \oint \frac{1}{Re} \frac{\partial v}{\partial n}\ \varphi_i\ ds \end{Bmatrix} \tag{11.109}$$

It needs to be understood that M_{ij} and K_{ij} lack their first row (column) when i (j) is not a corner node. Similar adjustments are implied for $\mathcal{U}_j$ and $\mathcal{R}_i$. The optional rows are associated with the continuity equation; the optional columns are associated with the pressure variable.

Numbering Convention

The way the equations and the field variables are ordered governs the way the rows and column entries of each element matrix are filled, and the way the global matrix is assembled. A workable convention is:

- variable order: $(p), u, v$
- equation order: (continuity), x-momentum, y-momentum

The following pointer arrays are useful in implementing this scheme:

`INC(L,I)` = Incidence list of each element `L`. `I` is the local node number, `INC` is the global node number. This is the customary element incidence list.

`NDOF(I)` = Number of degrees of freedom at each node

`NLOC(I)` = Location of the first degree of freedom at each node in the global field variable vector. For an arbitrary nine-node quadrilateral element, at a corner node `I` which has three degrees of freedom, `NLOC(I)` is the location of p_i in the global solution vector; and at a midside or midelement node `J`, which has two degrees of freedom, `NLOC(J)` is the location of u_j in the global solution vector. `NLOC` also applies to row and column numbering in all Galerkin matrices.

These arrays define the mapping of matrix entries from local elemental addresses to global addresses. For example, in element 199, local node 2 (a corner): the u_2 location is column NLOC(IN(199,2))+1; and row NLOC(IN(199,2))+2 is the y-momentum equation, weighted with local basis 2.

Coordinate Rotation

It is desirable to rotate this system into local (n, s) coordinates after assembly, as described above. Similar to the rotation in the electromagnetic case, the relevant rotation matrix $[\mathbf{A}_i]$ here is

$$[\mathbf{A}_i] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & \sin\theta_i \\ 0 & -\sin\theta_i & \cos\theta_i \end{bmatrix} \tag{11.110}$$

for cases where i is a corner node. As in the other vector problems, the intent of the rotation is to render the unknowns, the momentum equations, and the natural boundary conditions in natural local coordinates.

References: Fluid Mechanics

Taylor and Hughes [104] is an early classic in this important field. For the most comprehensive contemporary views, see Gresho and Sani [39] and Zienkiewicz and Taylor [124]. For a specific implementation of the method described here, see Ip and Lynch [46].

- Gresho, P.M. and R.L. Sani. *Incompressible Flow and the Finite Element Method*, Wiley, Chichester, 1998.
- Ip, J.T.C. and D.R. Lynch. Finite element solution of the two-dimensional incompressible Navier-Stokes equations with mixed interpolation. Report NML-91-1, Numerical Methods Laboratory, January 1991.
- Taylor, C., and T. G. Hughes. *Finite Element Programming of the Navier-Stokes Equations*, Pineridge Press, Swansea, 1981.
- Zienkiewicz, O. C. and R.L. Taylor, *The Finite Element Method: Volume 3, Fluid Dynamics* Butterworth-Heinemann, 2000.

11.6 Oceanic Tides

The shallow water equations describe tidal motions in the ocean. These motions are essentially 2-D and exist at discrete frequencies determined by gravity and inertial forces on a planetary scale. The resultant motions are modulated by the geometry of ocean basins and by frictional losses at the basin edges (over the continental shelves). The linearized equations of motion are

$$\iota\omega\zeta + \nabla \cdot (h\mathbf{V}) = 0 \tag{11.111}$$

$$\iota\omega\mathbf{V} + \mathbf{f} \times \mathbf{V} + g\nabla\zeta + \tau\mathbf{V} = 0 \tag{11.112}$$

with

ζ the ocean surface height above mean sea level
$\mathbf{V}$ the horizontal fluid velocity with scalar components u, v
ω the frequency of the motion
$\mathbf{f} = f\hat{\mathbf{z}}$ the Coriolis parameter
g gravity
τ a friction parameter

h the ocean depth
$\iota = \sqrt{-1}$

There are three unknown scalar fields, $\zeta(x,y)$, $u(x,y)$, and $v(x,y)$. These are complex amplitudes of the motion at the given frequency ω. Each will be expressed in the same real scalar basis $\phi(x,y)$ with complex nodal amplitudes ζ_i, etc. Implementation of the FEM algebra in a language which supports complex arrays and their manipulation is assumed.

The velocity may be eliminated among these to produce a Helmholtz equation in ζ. First, express the momentum equation in scalar form

$$\iota\omega u - fv + g\frac{\partial\zeta}{\partial x} + \tau u = 0 \tag{11.113}$$

$$\iota\omega v + fu + g\frac{\partial\zeta}{\partial y} + \tau v = 0 \tag{11.114}$$

$$\begin{bmatrix} \iota\omega+\tau & -f \\ f & \iota\omega+\tau \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} = -\begin{Bmatrix} g\frac{\partial\zeta}{\partial x} \\ g\frac{\partial\zeta}{\partial y} \end{Bmatrix} \tag{11.115}$$

Inverting this we obtain

$$\begin{Bmatrix} hu \\ hv \end{Bmatrix} = -\frac{1}{(\iota\omega+\tau)^2+f^2} \begin{bmatrix} \iota\omega+\tau & f \\ -f & \iota\omega+\tau \end{bmatrix} \begin{Bmatrix} gh\frac{\partial\zeta}{\partial x} \\ gh\frac{\partial\zeta}{\partial y} \end{Bmatrix} \tag{11.116}$$

or, in vector form,

$$h\mathbf{V} = -\frac{(\iota\omega+\tau)gh\nabla\zeta \; - \mathbf{f}\times gh\nabla\zeta}{(\iota\omega+\tau)^2+f^2} \tag{11.117}$$

Eliminating $\mathbf{V}$ from equation 11.111 we get

$$\iota\omega\zeta - \nabla\cdot\left(\frac{(\iota\omega+\tau)gh\nabla\zeta \; - \mathbf{f}\times gh\nabla\zeta}{(\iota\omega+\tau)^2+f^2}\right) = 0 \tag{11.118}$$

Weak Form and Galerkin Helmholtz Equation

The weak form of 11.111 is

$$\langle \iota\omega\zeta, \phi_i\rangle - \langle h\mathbf{V}, \nabla\phi_i\rangle = -\oint (h\mathbf{V})\cdot\hat{\mathbf{n}}\;\phi_i\;ds \tag{11.119}$$

or, from 11.118 after substitution for $h\mathbf{V}$:

$$\langle \iota\omega\zeta, \phi_i\rangle + \left\langle \left(\frac{(\iota\omega+\tau)gh\nabla\zeta \; - \mathbf{f}\times gh\nabla\zeta}{(\iota\omega+\tau)^2+f^2}\right), \nabla\phi_i \right\rangle = -\oint (h\mathbf{V})\cdot\hat{\mathbf{n}}\;\phi_i\;ds \tag{11.120}$$

In both of these forms we have integrated the divergence term by parts, exposing the boundary integral of the fluid flow normal to the boundary. This is the vehicle for enforcing Neumann boundary conditions with this natural boundary data. Expressing ζ in the FE basis ϕ,

$$\zeta(x,y) \simeq \sum_j \zeta_j\phi_j(x,y) \tag{11.121}$$

we obtain the Galerkin Helmholtz equation for tides:

$$\sum_i K_{ij}\zeta_j = R_i \tag{11.122}$$

with stiffness matrix $[K]$:

$$K_{ij} = \langle \iota\omega\phi_j, \phi_i \rangle + \left\langle \left(\frac{(\iota\omega+\tau)gh\nabla\phi_j - \mathbf{f}\times gh\nabla\phi_j}{(\iota\omega+\tau)^2+f^2} \right), \nabla\phi_i \right\rangle \tag{11.123}$$

and R is the natural boundary integral containing the water transport across the boundary:

$$R_i = -\oint (h\mathbf{V})\cdot\hat{\mathbf{n}}\ \phi_i\ ds \tag{11.124}$$

From the triple product identities we have

$$(\mathbf{f}\times\nabla\phi_j)\cdot\nabla\phi_i = \mathbf{f}\cdot(\nabla\phi_j\times\nabla\phi_i) \tag{11.125}$$

and an equivalent expression for $[K]$ is

$$K_{ij} = \langle \iota\omega\phi_j, \phi_i \rangle + \left\langle \frac{(\iota\omega+\tau)gh\nabla\phi_j\cdot\nabla\phi_i}{(\iota\omega+\tau)^2+f^2} \right\rangle - \left\langle \frac{fgh\nabla\phi_j\times\nabla\phi_i}{(\iota\omega+\tau)^2+f^2} \right\rangle \tag{11.126}$$

This Galerkin-Helmholtz equation can be solved for ζ alone, subject to Dirichlet (ζ known) or Neumann ($\mathbf{V}\cdot\hat{\mathbf{n}}$ known) BC's. The "unused" Galerkin equations associated with Dirichlet bases may be used to derive the equivalent Neumann data along Dirichlet boundaries.

Velocity Solution

The velocity $\mathbf{V}$ may be obtained from the momentum equation 11.115 once ζ is known. This involves differentiation of ζ. Point differentiation is possible and workable, if one accepts a discontinuous velocity solution. The Galerkin form of 11.115 is

$$\sum_j \langle (\iota\omega+\tau)\phi_j, \phi_i\rangle u_j - \sum_j \langle f\phi_j, \phi_i\rangle v_j = -\langle g\frac{\partial\zeta}{\partial x}, \phi_i\rangle \tag{11.127}$$

$$\sum_j \langle f\phi_j, \phi_i\rangle u_j + \sum_j \langle (\iota\omega+\tau)\phi_j, \phi_i\rangle v_j = -\langle g\frac{\partial\zeta}{\partial y}, \phi_i\rangle \tag{11.128}$$

or, in matrix form,

$$\sum_j [M_{ij}]\,\mathbf{V}_j = \mathbf{R}_i \tag{11.129}$$

with

$$M_{ij} = \begin{bmatrix} \langle (\iota\omega+\tau)\phi_j, \phi_i\rangle & \langle -f\phi_j, \phi_i\rangle \\ \langle f\phi_j, \phi_i\rangle & \langle (\iota\omega+\tau)\phi_j, \phi_i\rangle \end{bmatrix} \tag{11.130}$$

$$\mathbf{V}_j = \begin{Bmatrix} u \\ v \end{Bmatrix} \qquad \mathbf{R}_i = \begin{Bmatrix} -\langle g\frac{\partial\zeta}{\partial x}, \phi_i\rangle \\ -\langle g\frac{\partial\zeta}{\partial y}, \phi_i\rangle \end{Bmatrix} \tag{11.131}$$

This system may now be rotated into the local (n, s) coordinate systems as in the previous section ("Gradient of a Scalar"). Strong enforcement of boundary conditions on $\mathbf{V}\cdot\hat{\mathbf{n}}$ is appropriate, using the same Neumann data which went into the $\oint$ terms in equation 11.120. BC's on $\mathbf{V}\cdot\hat{\mathbf{n}}$ may also be derived along Dirichlet (ζ known) boundaries, as suggested above. This entire procedure is a generalization of that described above for computing the simple gradient of a scalar.

References - Oceanic Tides

Several groups have generally used this basic framework. Notable are the contributions from Le Provost [54, 55, 106], Lynch [63, 66], Pearson [96], Platzman [98], Walters [108, 109], Westerink [111], and Winter [47] and their co-workers. The method has been extended to 3-D tides (the third dimension is diffusive) in [38, 62, 82, 73, 88]. The idea also forms the basis of a "detiding" or inverse model approach [75]. Throughout there is the use of this method for the steady-state ($\omega = 0$) limit (*e.g.* [82].)

Below we list, for convenience, some of these works.

- Greenberg, D., F.E. Werner, D.R. Lynch. 1998. A diagnostic finite-element ocean circulation model in spherical-polar coordinates. *J Atmospheric and Oceanic Technology* **15**:942-958.
- Le Provost, C. and A. Poncet. 1978. Finite element method for spectral modelling of tides. *Int. J. for Num. Meths. in Engg.* **12**: 853-871.
- Lynch, D.R. 1985. Mass balance in shallow water simulations. *Comm. in Applied Numerical Methods* **1**:153-158.
- Lynch, D.R. and F.E. Werner. 1987. Three-dimensional hydrodynamics on finite elements. Part I: Linearized Harmonic Model. *Int. J. for Numerical Methods in Fluids.* **7**, 871-909.
- Jamart, B.M. and D.F. Winter. 1980. Finite element computation of the barotropic tides in Knight Inlet, British Columbia. *In:* H.J. Freeland, D.M. Farmer, C.D. Levings (eds): *Fjord Oceanography*, Plenum Press, New York. pp283-289.
- Pearson, C.E. and D.F. Winter. 1977. On the calculation of tidal currents in homogeneous estuaries. *J. Phys. Oceanogr.*, **7**, 520-531.
- Platzman, G.W. 1978. Normal modes of the world ocean. Part I. Design of a finite element barotropic model. *J. Phys. Oceanogr.*, **8**, 323-343.
- Walters, R.A. 1987. A model for tides and currents in the English Channel and southern North Sea. *Adv. in Water Res.*, **10**, 138-148.
- Westerink, J.J., K.D. Stolzenbach, and J.J. Connor. 1989. General spectral computations of the nonlinear shallow water tidal interactions within the Bight of Abaco. *J. Phys. Oceanogr.* **19**: 1348-1371.

Chapter 12

Numerical Analysis

All finite element methods lead to discrete algebraic representations – difference equations – which approximate the continuum problem. In this chapter we explore some of the properties of these discrete approximations. Fundamentally, we are interested in two aspects: *accuracy* and *stability*. Our general approach will be to assume an unbounded, uniform domain of simple elements, such that a generic interior WR equation can be considered and solved. This will restrict our attention to simple linear problems with constant coefficients, where closed-form solutions are available.

12.1 Elliptic Equations: Galerkin on 1-D Linear Elements

Laplace Equation on 1-D Linear Elements

We consider the simple equation

$$\frac{d^2u}{dx^2} = 0 \tag{12.1}$$

We will discretize this using Galerkin on an infinite array of linear elements of length h, with nodes numbered naturally in increasing order i, $i+1$, etc. The results of previous sections for any interior node are readily adapted:

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0 \tag{12.2}$$

This is obviously identical to the standard second-order finite difference representation.

The analytic solution permits solutions of the form $u = a + bx$. It is clear that the discrete system also has the same solution; and therefore we find perfect correspondence with the continuum in this simplest case. Fundamentally, the exact solution is contained in the linear basis.

Advective-Diffusive Equation on 1-D Linear Elements

Here we have the steady-state version of this equation:

$$D\frac{d^2u}{dx^2} - V\frac{du}{dx} = 0 \tag{12.3}$$

which supports solutions of the form $u = e^{rx}$ with two roots: $r = 0$ and $r = \frac{V}{D}$. The case $r = 0$ is just the solution u =constant.

The Galerkin version is:

$$D\left(\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}\right)-V\left(\frac{u_{i+1}-u_{i-1}}{2h}\right)=0 \tag{12.4}$$

Again, this is indistinguishable from a standard centered finite difference equation. The equivalent representation

$$u_{i+1}(1-\frac{P_e}{2})-2u_i+u_{i-1}(1+\frac{P_e}{2})=0 \tag{12.5}$$

highlights the essential role of the Peclet number $P_e \equiv \frac{Vh}{D}$ which is the dimensionless measure of the mesh size for this problem. The difference equations support solutions of the form $u=e^{rx}$, but the roots r will not be identical to the analytic ones. It is convenient to define the factor γ as the ratio of adjacent solutions

$$\gamma=\frac{u_{i+1}}{u_i}=e^{rh} \tag{12.6}$$

and we will compare the Galerkin version γ_g with its analytic counterpart γ_a.

Substitution of (12.6) into (12.5) gives us the quadratic equation for the Galerkin version γ_g:

$$\gamma_g^2(1-\frac{P_e}{2})-2\gamma_g+(1+\frac{P_e}{2})=0 \tag{12.7}$$

There are two roots:

$$\gamma_g=\frac{1\pm\frac{P_e}{2}}{1-\frac{P_e}{2}} \tag{12.8}$$

Clearly, one root is unity and a perfect replica of the continuum. The other root,

$$\gamma_g=\frac{1+\frac{P_e}{2}}{1-\frac{P_e}{2}} \tag{12.9}$$

is an approximation to the continuum. Approximating it as a Taylor series we have

$$\begin{aligned}\gamma_g &= \left(1+\frac{P_e}{2}\right)\left(1+\left(\frac{P_e}{2}\right)+\left(\frac{P_e}{2}\right)^2+\left(\frac{P_e}{2}\right)^3+\ldots\right)\\ &= 1+P_e+\frac{1}{2}{P_e}^2+\frac{1}{4}{P_e}^3+\ldots\end{aligned} \tag{12.10}$$

The Taylor series for the analytic value $\gamma_a \equiv e^{\frac{Vh}{D}} \equiv e^{P_e}$ is

$$\gamma_a=1+P_e+\frac{1}{2!}{P_e}^2+\frac{1}{3!}{P_e}^3+\ldots \tag{12.11}$$

and we see that γ_g is correct to order ${P_e}^3$.

One needs to be careful that this "third order" accuracy is an intrinsic or *per step* measure, describing the variation accumulated over one mesh spacing. As h decreases, it gives an artificial picture of convergence. A more meaningful measure is the variation accumulated over a fixed distance $L=nh$ wherein n increases as h decreases. For both analytic and numerical, this is $\gamma^n=\gamma^{\frac{L}{h}}$ and an appropriate measure of precision is the ratio of numerical to analytic solution at a fixed distance from a given node:

$$\Gamma=\left(\frac{\gamma_g}{\gamma_a}\right)^n=\left(\frac{\gamma_g}{\gamma_a}\right)^{\frac{L}{h}} \tag{12.12}$$

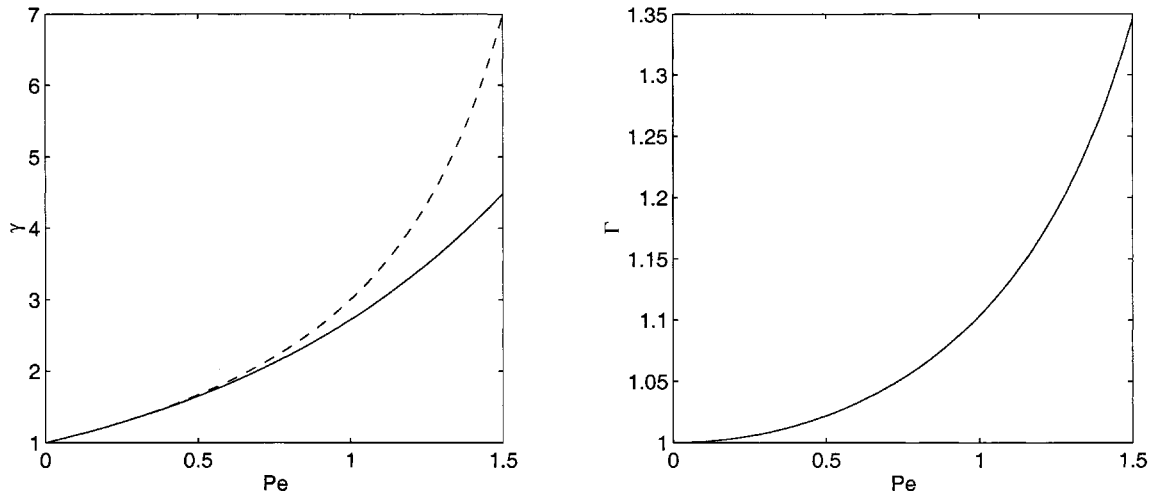


Figure 12.1: γ (left) and Γ (right) *vs.* Pe. In the left panel, the solid line is the analytical result; the dash line is the Galerkin approximation.

A natural definition of L is the e-folding length $\frac{D}{V}$, so we have $\gamma_a^{\frac{L}{h}} = e^1$ and

$$\Gamma = \frac{\gamma_g^{\frac{1}{P_e}}}{e} \tag{12.13}$$

Figure 12.1 shows the behaviour of Γ versus P_e. At $P_e = 1$ the error is approximately 10%.

An additional feature of this numerical solution is the loss of monotonicity for large Peclet numbers. From equation (12.6) it is easy to see that γ will become negative for $P_e > 2$. Under these conditions, this mode of the Galerkin solution oscillates in sign from one node to the next. This behaviour is completely spurious, *i.e.* it has no equivalent in the continuum solution. The situation is depicted in Figure 12.2, which shows both analytic and Galerkin solutions to a simple two-point boundary value. The failure to properly resolve the natural length scale $\frac{Vh}{D}$ is combined here with a difference operator which supports this oscillatory mode of solution. Both Figures 12.1 and 12.2 support the conclusion that a small Peclet number is a critical ingredient for accurate simulation with this discretization. (At vanishingly small P_e, we revert to the previous case ($V = 0$); in Figure 12.2 we would have the perfect representation of a straight line connecting the boundary values.)

Helmholtz Equation on 1-D Linear Elements

Next we consider the Helmholtz Equation:

$$\frac{d^2u}{dx^2} + k^2u = 0 \tag{12.14}$$

with $k^2 > 0$. Its homogeneous response is of the form $e^{\pm jkx}$ ($j \equiv \sqrt{-1}$.) These two solutions are forward- and backward-propagating waves of length $L = 2\pi/k$, each with constant amplitude. Two

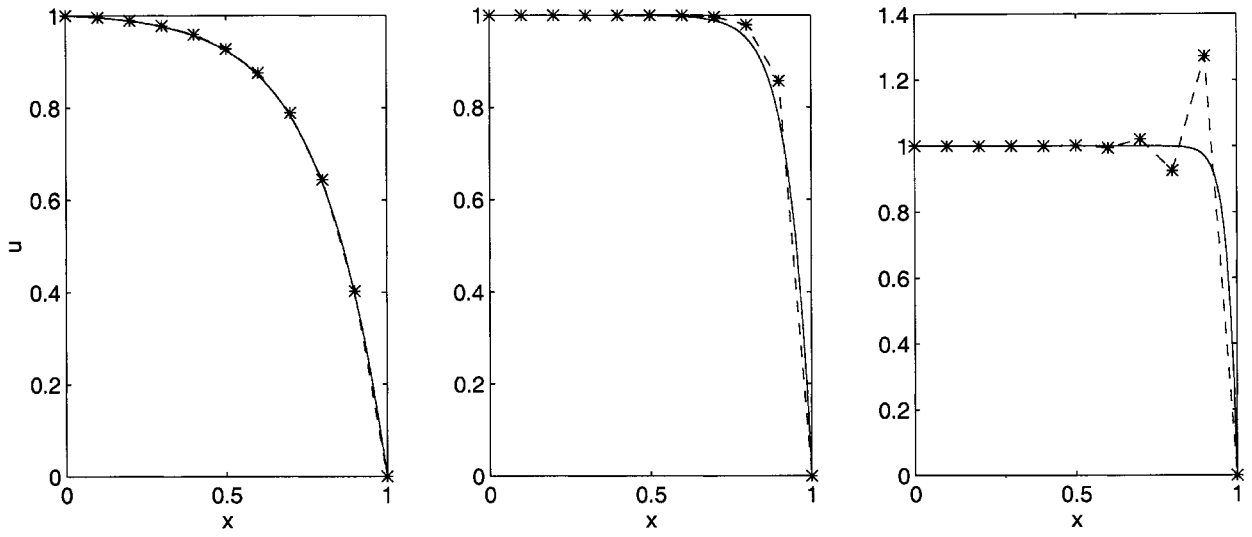


Figure 12.2: Two-point boundary value problem for $P_e = 0.5$ (left), 1.5 (center), and 3.5 (right). The solid line is the exact solution; the dash line is the Galerkin solution with nodal points highlighted by $*$.

boundary conditions determine their amplitudes, provided that the system length is not a multiple of L.

The Galerkin equations are:

$$(u_{i+1} - 2u_i + u_{i-1}) + \frac{K^2}{6}(u_{i+1} + 4u_i + u_{i-1}) = 0 \tag{12.15}$$

with $K^2 \equiv k^2h^2$. As in the previous section, we seek a solution of the form

$$u_{i+1} = \gamma u_i \tag{12.16}$$

which leads us to the quadratic equation

$$\gamma^2\left[1 + \frac{K^2}{6}\right] + \gamma\left[-2 + 4\frac{K^2}{6}\right] + \left[1 + \frac{K^2}{6}\right] = 0 \tag{12.17}$$

The solution is

$$\gamma_g = \frac{\left[1 - 2\frac{K^2}{6}\right]}{\left[1 + \frac{K^2}{6}\right]} \pm j\sqrt{1 - \left(\frac{\left[1 - 2\frac{K^2}{6}\right]}{\left[1 + \frac{K^2}{6}\right]}\right)^2} \tag{12.18}$$

(the subscript g is introduced here to distinguish the *Galerkin* γ from its analytic counterpart.) For small K^2 (specifically, $K^2 < 12$), we have $|\gamma| = 1$ and recreate the analytic structure of forward- and backward-propagating waves with constant amplitude.

To compare numerical and analytic solutions, we compute

$$\Gamma = \left(\frac{\gamma_g}{\gamma_a}\right)^n \tag{12.19}$$

with $nh = L = \frac{2\pi}{k}$, the analytic wavelength. Thus we have

$$n = \frac{2\pi}{K} \tag{12.20}$$

This is a measure of the buildup of numerical error over a characteristic length of the domain. By definition, the analytic solution completes one wavelength over this length, so $\gamma_a^n = 1$. Also, since $|\gamma_g| = 1$, we have $|\Gamma| = 1$ and the error will lie in the phase of the numerical solution, not its amplitude. The argument (phase) of Γ is then the measure of fidelity. Combining these facts, we have

$$\Gamma = (\gamma_g)^{\frac{2\pi}{K}} \tag{12.21}$$

which will have perfect amplitude = 1 and argument

$$\arg(\Gamma) = \frac{2\pi}{K} \arg(\gamma_g) \tag{12.22}$$

A perfect solution would have $\arg(\Gamma) = 0$ (or 2π). Figure 12.3 shows $\arg(\Gamma)/2\pi$ versus K^2. $K^2 = 1$ produces phase error of order 0.08π, *i.e.* roughly 4% error.

An alternate measure of the same error is the ratio of numerical wavelength to its analytic counterpart. Numerically, m mesh spacings are equal to one wavelength:

$$m \cdot \arg(\gamma_g) = 2\pi \tag{12.23}$$

while analytically, n spacings are needed. Thus, the wavelength ratio is

$$\frac{m}{n} = \frac{K}{arg(\gamma_g)} \tag{12.24}$$

This ratio is plotted in Figure 12.3. The plot is the inverse of the relative phase plot (compare equations 12.24 and 12.22). The numerical wavelength is always larger than the analytic, consistent with the negative phase error.

Poisson Equation on 1-D Linear Elements

Next we consider the inhomogeneous equation

$$\frac{d^2u}{dx^2} = f \tag{12.25}$$

It is natural to study this equation in terms of the Fourier decomposition of the forcing function f:

$$f(x) = \sum_{\sigma} F_\sigma e^{j\sigma x} \tag{12.26}$$

where j is the imaginary unit $\sqrt{-1}$ and σ is a real-valued wavenumber corresponding to a wave of length $2\pi/\sigma$. Since everything is linear, we may consider each Fourier mode separately. The solution for a single mode is readily expressed as

$$u(x) = U_\sigma e^{j\sigma x} \tag{12.27}$$

and direct substitution gives us

$$U_\sigma = \frac{F_\sigma}{(j\sigma)^2} \tag{12.28}$$

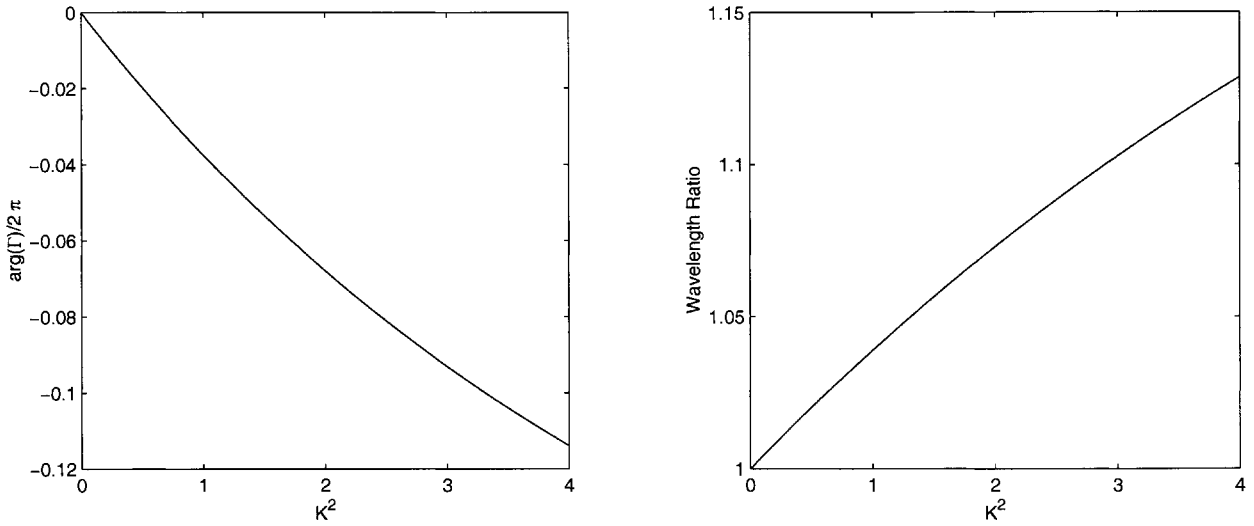


Figure 12.3: Homogeneous Helmholtz solution. On the left, the phase error is plotted, normalized by 2π. On the right, the ratio of numerical to analytic wavelength is plotted.

The complete solution is the synthesis of the modes

$$u(x) = \sum_{\sigma} U_\sigma e^{j\sigma x} = \sum_{\sigma} \frac{F_\sigma}{(j\sigma)^2} e^{j\sigma x} \tag{12.29}$$

This is a classic result; the Galerkin system has an analogous structure.

The Galerkin version of the Poisson equation on uniform 1-D linear elements of length h is

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \frac{f_{i+1} + 4f_i + f_{i-1}}{6} \tag{12.30}$$

As in the analytic solution, we will consider the Fourier decomposition of $f(x)$, sampled at the nodes. Its spectrum is therefore terminated at the Nyquist point:

$$0 \leq \sigma h \leq \pi \tag{12.31}$$

Again, the difference equations are linear so we consider only a single mode σ, and we have the discrete representation

$$f_i = F_\sigma e^{j\sigma x_i} \tag{12.32}$$

The solution $u(x)$ will be of the same form

$$u_i = U_\sigma e^{j\sigma x_i} \tag{12.33}$$

Nodal values of u may be conveniently expressed in terms of the shift factor $e^{j\sigma h}$:

$$\begin{aligned} u_i &= U_\sigma e^{j\sigma x_i} \\ u_{i+1} &= U_\sigma e^{j\sigma x_i} e^{j\sigma h} = u_i e^{j\sigma h} \\ u_{i-1} &= U_\sigma e^{j\sigma x_i} e^{-j\sigma h} = u_i e^{-j\sigma h} \end{aligned} \tag{12.34}$$

and therefore the difference operator $[u_{i+1} - 2u_i + u_{i-1}]$ can be condensed to

$$u_{i+1} - 2u_i + u_{i-1} = u_i \left(e^{j\sigma h} - 2 + e^{-j\sigma h}\right) = u_i \left(2\cos(\sigma h) - 2\right) \tag{12.35}$$

and with a little rearrangement we arrive at the result

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = (j\sigma)^2 u_i \left[\frac{\sin\left(\frac{S}{2}\right)}{\frac{S}{2}}\right]^2 \tag{12.36}$$

where we have introduced the dimensionless mesh length S

$$S \equiv \sigma h \equiv \frac{2\pi h}{\lambda} \tag{12.37}$$

where λ is the wavelength. The factor $C(S) \equiv \left[\frac{\sin\left(\frac{S}{2}\right)}{\frac{S}{2}}\right]$ contains all of the discretization effect. Clearly, as $S \to 0$, *i.e.* as the mesh becomes fine relative to the wavelength of the forcing, $C \to 1$ and the Galerkin second derivative is perfect. For finite S, we have an error in the derivative which is quantified by $|C^2 - 1|$.

The same procedure leads to the compact expression for the spatial average of f:

$$\frac{f_{i+1} + 4f_i + f_{i-1}}{6} = f_i \left[\frac{e^{j\sigma h} + 4 + e^{-j\sigma h}}{6}\right] = f_i \left[\frac{4 + 2\cos(S)}{6}\right] = A(S) f_i \tag{12.38}$$

where $A(S) \equiv \left[\frac{4+2\cos(S)}{6}\right]$ is an averaging operator. Like C, $A \to 1$ as $S \to 0$. Use of these discretization factors in the Galerkin system above gives us the compact representation

$$(j\sigma)^2 C^2 U_\sigma = A F_\sigma \tag{12.39}$$

and the solution is analogous to the continuum:

$$U_\sigma = \frac{F_\sigma}{(j\sigma)^2}\left[\frac{A}{C^2}\right] \tag{12.40}$$

The discrepancy between Galerkin and analytic solutions is clearly the discretization factor $\left[\frac{A}{C^2}\right]$, for which the value unity represents perfection. In Figure 12.4 we plot it as a function of S.

The expected behaviour is apparent: high fidelity for low values of S (*i.e.* well-resolved Fourier modes) which degrades as the discretization becomes coarse. Example values of the error $\left[\frac{C^2}{A}\right] - 1$ are given in Table 12.1 and confirm second-order convergence in S. The conventional rule of thumb – 10 nodes per wavelength – produces entry-level accuracy of order 3%.

For problems forced by several modes, the synthesis

$$u_i = \sum_\sigma \frac{F_\sigma}{(j\sigma)^2}\left[\frac{A}{C^2}\right] e^{j\sigma x_i} \tag{12.41}$$

will introduce a distortion unless f is properly resolved by the mesh (*i.e.* S is sufficiently small for all constituents σ in the forcing function f).

On a finite-length mesh, this solution would be supplemented by a homogeneous solution (derived above) which would be fit to boundary conditions at the two endpoints.

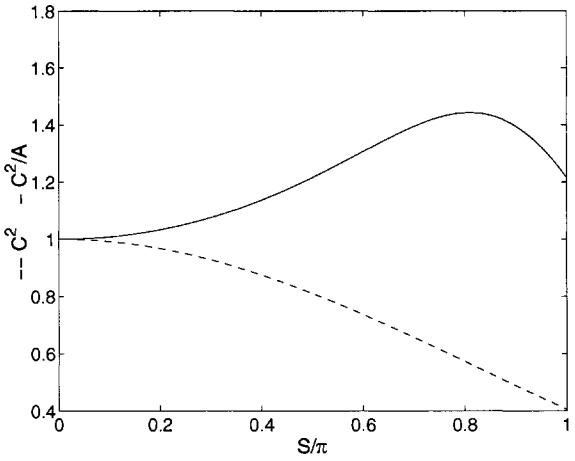


Figure 12.4: Discretization factors for the Galerkin Poisson equation, as a function of $\sigma h/\pi$. The solid line is C^2/A; the dash line is C^2.

S/π	.05	.1	.2	.4
ϵ	.00206	.00825	.03331	.13703
λ/h	40	20	10	5

Table 12.1: Discretization error $\frac{C^2}{A} - 1$ versus dimensionless wavenumber $S = \sigma h$ for the Galerkin Poisson equation on 1-D linear elements. λ/h is the number of nodes per wavelength.

Inhomogeneous Helmholtz Equation on 1-D Linear Elements

Next consider the inhomogeneous Helmholtz equation

$$\frac{d^2u}{dx^2} + k^2 u = f \tag{12.42}$$

Proceeding as above, we examine Fourier modes of the form $u(x) = U_\sigma e^{j\sigma x}$ and analogously for $f(x)$. The result is similar to Poisson equation:

$$\left[(j\sigma^2) + k^2\right] U_\sigma = F_\sigma \tag{12.43}$$

$$U_\sigma = \frac{F_\sigma}{[(j\sigma)^2 + k^2]} \tag{12.44}$$

The addition of the k^2 term introduces resonance in the response at

$$\sigma^{*2} = k^2 \tag{12.45}$$

and the response for σ^2 close to k^2 will be enhanced relative to the limiting Poisson case ($\sigma^2 >> k^2$).

Again, the Galerkin system has an analogous structure. On uniform 1-D linear elements of length h we have

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + k^2 \left[\frac{u_{i+1} + 4u_i + u_{i-1}}{6}\right] = \frac{f_{i+1} + 4f_i + f_{i-1}}{6} \tag{12.46}$$

Following the same procedure as in the Poisson case, for a single Fourier mode $u_i = U_\sigma e^{j\sigma x_i}$ we obtain

$$\left[(j\sigma)^2 C^2 + k^2 A\right] U_\sigma = A F_\sigma \tag{12.47}$$

$$U_\sigma = \frac{F_\sigma}{\left[\frac{C^2}{A}(j\sigma)^2 + k^2\right]} \tag{12.48}$$

The Galerkin resonance has shifted by the familiar factor C^2/A:

$$\sigma^{*2} = \frac{k^2}{C^2/A} \tag{12.49}$$

and this is illustrated in Figure 12.5, where both analytic and Galerkin responses are plotted versus S. It is obvious that "large" values of $K \equiv kh$ will seriously compromise the solution; and the practical rule emerges that the resonance wavenumber must be well resolved in the same sense as S, roughly 0.4π or less to provide entry-level accuracy. In Figure 12.6 we re-examine the plot of analytic and Galerkin responses in the well-resolved range of both S and K.

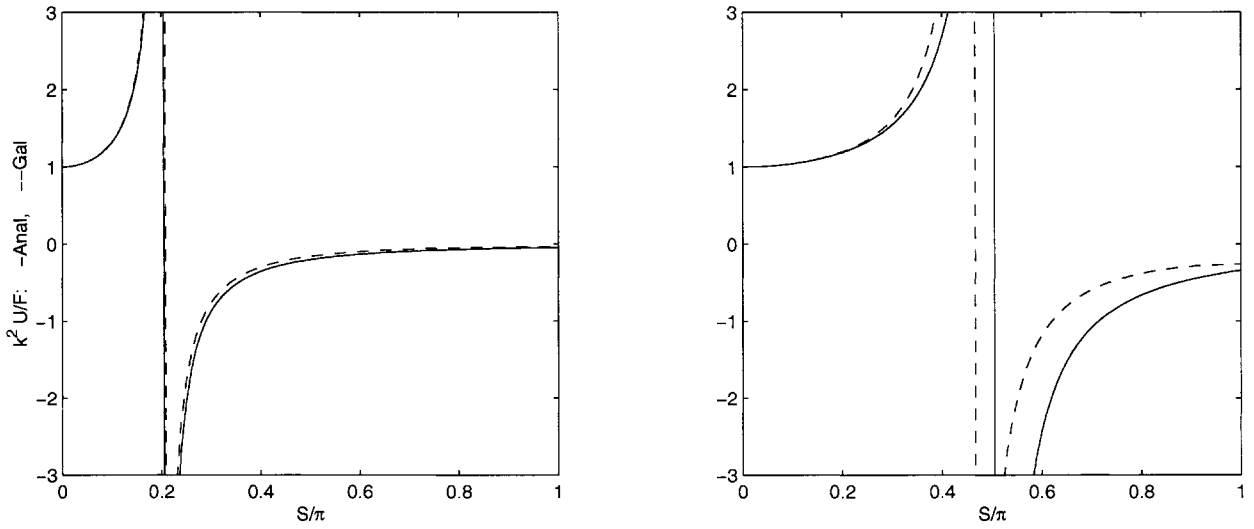


Figure 12.5: Normalized Helmholtz response $U_\sigma / \frac{F_\sigma}{k^2}$ *vs* S/π for Analytic (solid line) and Galerkin solutions. In the left panel $K = .2\pi$; in the right, $K = .5\pi$.

In Figure 12.7 we plot the resonance wavenumber S versus K for analytic and Galerkin systems. It is evident that the Galerkin solution systematically underestimates the resonance, and that the gap grows with increasing K, reaching approximately $\Delta S^* = .02\pi$ at $K = .4\pi$. This creates a gap in the spectrum where the solution is simply unrealistic – the analytic response has passed through resonance and changed phase, while the Galerkin solution is still slightly below resonance and in phase. Measures of error are meaningless in this gap.

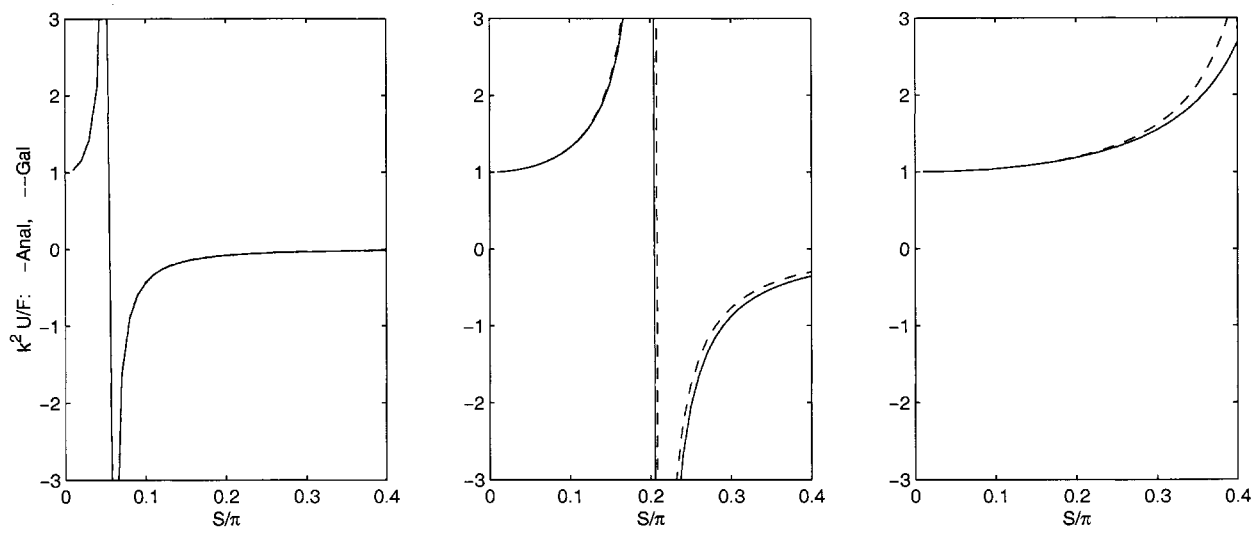


Figure 12.6: Same as Figure 12.5 but the well-resolved range of S is plotted, for $K/\pi = .05$, .20 and .50.

In Figure 12.8, we plot the difference between the Galerkin and analytic responses, omitting the gap in the spectrum described above. It is clear that the error near resonance is the dominant error and can be significant – *e.g.* .10 or more over a significant part of the well-resolved spectrum – in the two larger values of K shown. Since these plots are dimensionless, normalized by the response at $S = 0$, this is equivalent to a 10% error.

On a finite-length mesh, this solution would be supplemented by a homogeneous solution (derived above) which would be fit to boundary conditions at the two endpoints.

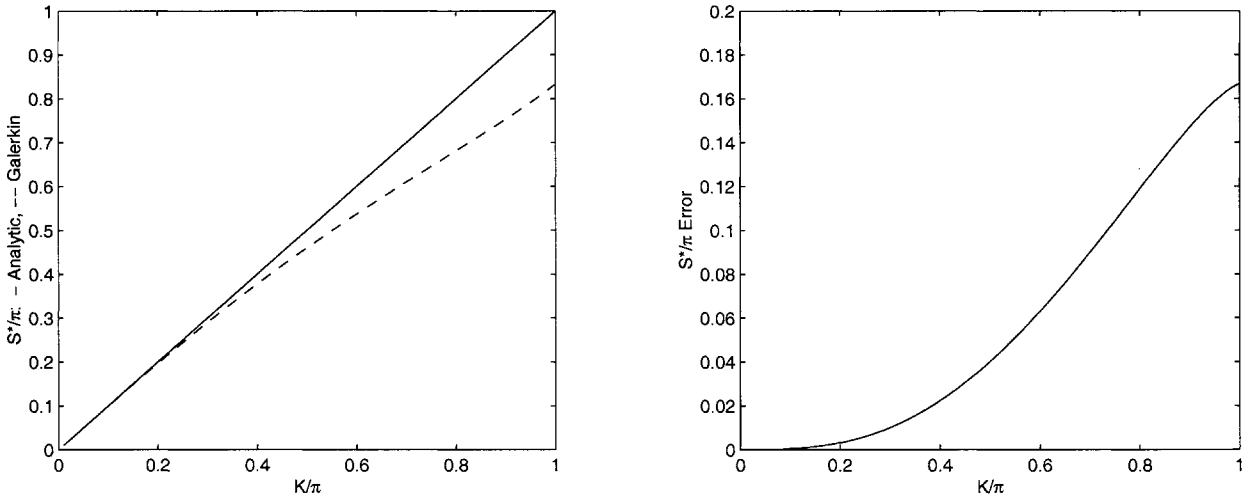


Figure 12.7: Resonance S^* *vs.* K for analytic and Galerkin solutions (left); and their difference (right). Note both S and K axes are scaled by π.

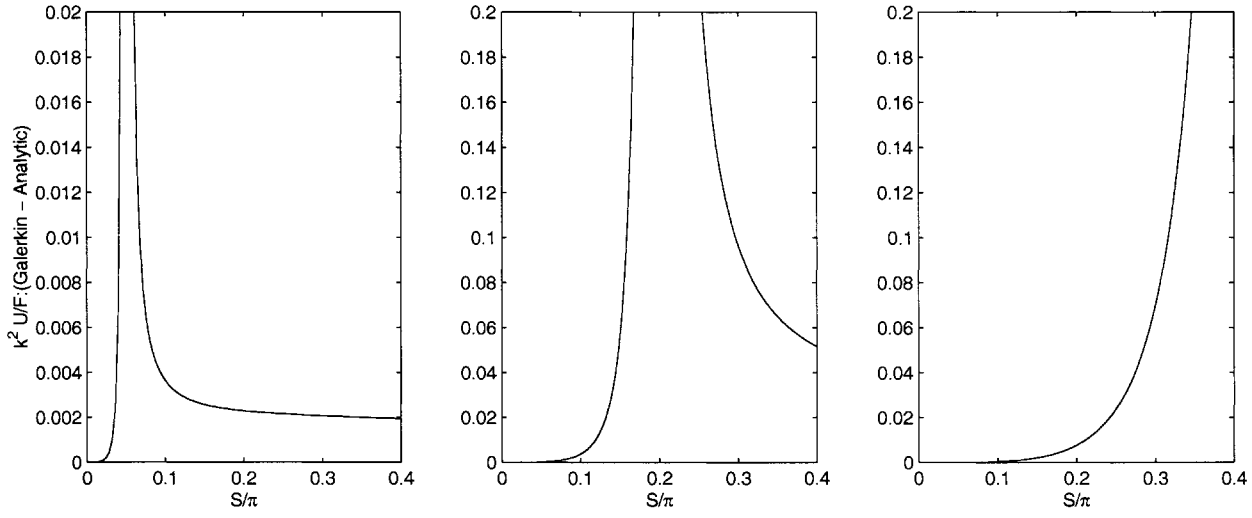


Figure 12.8: Error in the normalized response curves $U_\sigma / \frac{F_\sigma}{k^2}$ depicted in Figure 12.6. $K/\pi = .05$, .20 and .50 from left to right. Note that the y-axis in the leftmost plot is scaled by an additional factor of 10.

12.2 Fourier Transforms for Difference Expressions

In the previous two sections we utilized Fourier Analysis to compare solutions to discretized PDE's with their analytic counterparts. We assumed a solution of the form $u(x) = U_\sigma e^{j\sigma x}$, with $j = \sqrt{-1}$ and real wavenumber σ; and sought relations among the PDE coefficients, σ, and the mesh spacing $h \equiv \Delta x$. Required for this analysis is constant PDE coefficients and a uniform, unbounded mesh - *i.e.* boundaries at infinity. The essential idea is that differential operators have analytic Fourier transforms – *e.g.*

$$\frac{d^2u}{dx^2} \rightarrow (j\sigma)^2 u \tag{12.50}$$

and their discrete counterparts have similar transforms

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \rightarrow (j\sigma)^2 C^2 u_i \tag{12.51}$$

with the discretization effects concentrated in the discretization factor C. This idea provides powerful insight; our purpose here is to organize and package the essential relations for general use. We stick to the simplest finite elements – Galerkin on uniform, linear elements. For generality and comparative value, we provide also the common finite difference discretizations (second-order, centered) on the same mesh of equidistant grid points. Both 1- and 2-D transforms are provided.

1-D Transforms

Table 12.3 provides standard difference operators for the FE and FD approximations mentioned. For the FD cases, we use conventional second-order centered differences. For the FE cases, we use Galerkin on 1-D, linear elements, weighted with basis function i. Equal node spacing $h = \Delta x$ is assumed. In the FE cases, the difference expressions have been normalized by the factor h. The node numbering system is natural, ordered monotonically with increasing x. The FD and FE expressions are essentially the same, with the exception of the averaging which is introduced in one term.

For a solution of the form

$$u(x) = U_\sigma e^{j\sigma x} \tag{12.52}$$

the solutions at adjacent nodes are shifted by the exponential factor

$$u(x + h) = u(x) e^{j\sigma h} \tag{12.53}$$

$$u_{i+1} = u_i e^{jS} \tag{12.54}$$

with $S \equiv \sigma h$. Use of this shift factor allows us to condense the difference equations in Table 12.3. For example, the second derivative expression reduces to

$$\begin{aligned} \frac{1}{h^2}(u_{i+1} - 2u_i + u_{i-1}) &= \frac{1}{h^2}(e^{jS} - 2 + e^{-jS})u_i = \frac{2}{h^2}(cos(S) - 1)u_i \\ &= (j\sigma)^2 \left[\frac{sin(S/2)}{S/2}\right]^2 u_i \end{aligned} \tag{12.55}$$

Defining the discretization factor $C(S) \equiv [\frac{sin(S/2)}{S/2}]$, we have the Fourier transform

$$\frac{d^2u}{dx^2}|_i \rightarrow \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \rightarrow (j\sigma)^2 C^2 u_i \tag{12.56}$$

It is useful to note that C depends only on the dimensionless mesh spacing $S = \sigma h$. Since $\sigma L = 2\pi$, with L the wavelength of the Fourier mode being considered, we have

$$S = 2\pi \frac{h}{L} \tag{12.57}$$

i.e. S is proportional to the resolution of wavelength L. In the limit of high resolution, $S \to 0$ and all factors A, B and C approach unity. The limit of low resolution is at the Nyquist point $L_N = 2h$, or $S_N = \pi$. Wavelengths smaller than L_N do not exist on a discrete grid; they are indistinguishable from their higher-wavelength aliases with $L > L_N$. (The reader is referred to any text on the Discrete Fourier Transform.)

Similar discretization factors may be developed for the other terms in Table 12.3. For the centered first derivative, we have

$$\frac{1}{2h}\left(u_{i+1} - u_{i-1}\right) = \frac{1}{2h}\left(e^{jS} - e^{-jS}\right) u_i = j\sigma \left[\frac{\sin(S)}{S}\right] \tag{12.58}$$

Thus with $B(S) \equiv \left[\frac{\sin(S)}{S}\right]$, we have

$$\frac{du}{dx}|_i \to \frac{(u_{i+1} - u_{i-1})}{2h} \to j\sigma B u_i \tag{12.59}$$

Finally, the averaging operator $A(S) = \left[\frac{4+2\cos(S)}{6}\right]$ may be obtained in similar fashion:

$$\frac{1}{6}\left(u_{i+1} + 4u_i + u_{i-1}\right) = \frac{1}{6}\left(e^{jS} + 4 + e^{-jS}\right) u_i = \left[\frac{4 + 2\cos(S)}{6}\right] u_i \tag{12.60}$$

The discretization factors $A(S)$, $B(S)$ and $C(S)$ are listed in Table 12.4. They support the simple Fourier Transforms listed in table 12.5.

Recall that $S \equiv \sigma h \equiv 2\pi h/\lambda$, with λ the wavelength of the Fourier mode. Limits on S are $0 \leq S \leq \pi$, with the upper limit representing the Nyquist cutoff *i.e.* the shortest possible wavelength resolvable on a discrete grid, $\lambda_N = 2h$. Table 12.2 shows some practical values of S; $S/\pi = .2$ represents entry-level resolution at 10 elements per wavelength.

Figure 12.9 is a plot of the discretization factors A, B, and C^2 versus S. Note that at the Nyquist point, the first derivative fails, $B \to 0$ as $S \to \pi$. This reflects the practical fact that the first derivative expression is a centered first difference, *i.e.* a leapfrog expression which cannot distinguish a Fourier mode of length $2h$ from a constant. Also note that the ratio B/A is very nearly unity for well-resolved modes; indicating that the averaging operator effectively compensates for the discretization error in the first derivative in this range.

2-D Transforms

The two-dimensional extension is straightforward. Table 12.6 shows the discrete operators. The averaging effect of the FE method is more pronounced than in 1-D. The solution is expressed in terms of x- and y-dimension wavenumbers σ, γ:

$$u(x, y) = U_{\sigma\gamma} e^{j(\sigma x + \gamma y)} \tag{12.61}$$

S/π	.05	.1	.2	.4	1.0
$(S/\pi)^2$	.0025	.01	.04	.16	1.0
λ/h	40	20	10	5	2

Table 12.2: Dimensionless wavenumber $S = \sigma h$ for 1-D Fourier analysis, and corresponding value of the number of elements per wavelength λ/h.

	FD	FE
$\frac{\partial^2\phi}{\partial x^2}\|_i$	$\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{h^2}$	$\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{h^2}$
$\frac{\partial\phi}{\partial x}\|_i$	$\frac{\phi_{i+1}-\phi_{i-1}}{2h}$	$\frac{\phi_{i+1}-\phi_{i-1}}{2h}$
$\phi\|_i$	ϕ_i	$\frac{\phi_{i+1}+4\phi_i+\phi_{i-1}}{6}$

Table 12.3: 1-D Difference Operators for FD and FE Approximations.

$$A = \frac{4+2\,cos(S)}{6}$$

$$B = \frac{sin(S)}{S}$$

$$C = \frac{sin(S/2)}{(S/2)}$$

Table 12.4: Definition of discretization factors for solutions of the form $\phi = \phi_i\, e^{j(\sigma x)}$. $\Delta x = h$ is the mesh spacing; $S \equiv \sigma h$.

	FD	FE
$\frac{\partial^2\phi}{\partial x^2}\|$	$-C^2\sigma^2\phi_i$	$-C^2\sigma^2\phi_{ij}$
$\frac{\partial\phi}{\partial x}\|_i$	$jB\sigma\phi_i$	$jB\sigma\phi_i$
$\phi\|_i$	ϕ_i	$A\phi_i$

Table 12.5: Difference operators as in Table 12.3 for $\phi = \phi_i\, e^{j(\sigma x)}$. The discretization factors $A(S)$, $B(S)$ and $C(S)$ are defined in Table 12.4 and approach unity as the mesh is refined.

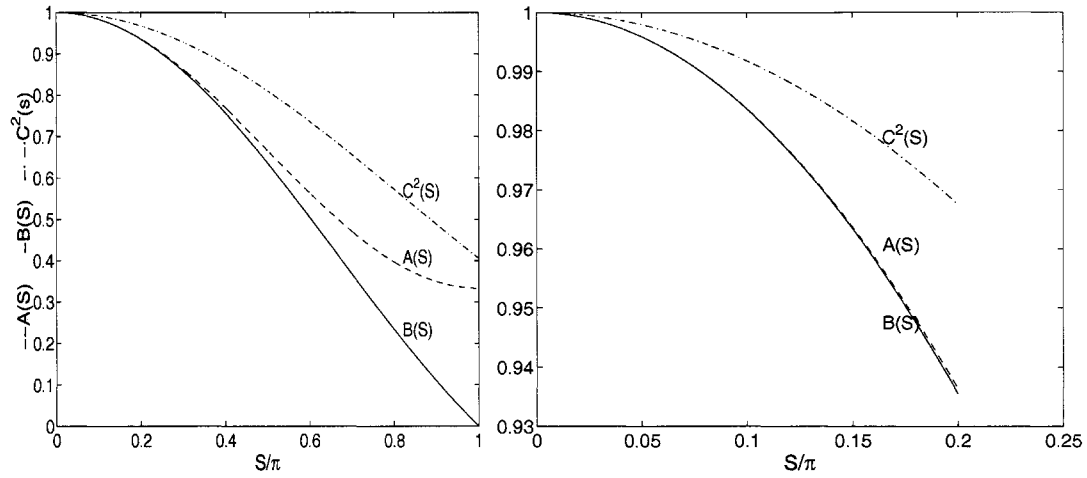


Figure 12.9: Discretization factors A, B, and C^2 versus dimensionless mesh spacing $S \equiv \sigma h \equiv 2\pi h/\lambda$. These represent Oth, first, and second derivative discretization effects for 1-D linear finite elements (Galerkin), for a Fourier mode of wavelength λ. In each case, perfection is unity. Left: full range terminating at the Nyquist point $\lambda = 2h$. Right: Well-resolved range terminating at $\lambda = 10h$.

Discretization factors related to each dimension appear in Table 12.7, and the 2-D Fourier transforms are listed in 12.8. These are the straightforward extension of the 1-D tables; the reader is encouraged to derive them.

Notice that the Finite Difference results are identical to the linear Galerkin results in this analysis, with the simple substitution $A = 1$.

	FD	FE
$\frac{\partial^2\phi}{\partial x^2}\|_{ij}$	$\frac{\phi_{i+1,j}-2\phi_{i,j}+\phi_{i-1,j}}{h^2}$	$\frac{1}{6}(\frac{\phi_{i+1,j+1}-2\phi_{i,j+1}+\phi_{i-1,j+1}}{h^2})$ $+\frac{4}{6}(\frac{\phi_{i+1,j}-2\phi_{i,j}+\phi_{i-1,j}}{h^2})$ $+\frac{1}{6}(\frac{\phi_{i+1,j-1}-2\phi_{i,j-1}+\phi_{i-1,j-1}}{h^2})$
$\frac{\partial^2\phi}{\partial y^2}\|_{ij}$	$\frac{\phi_{i,j+1}-2\phi_{i,j}+\phi_{i,j-1}}{h^2}$	$\frac{1}{6}(\frac{\phi_{i+1,j+1}-2\phi_{i+1,j}+\phi_{i+1,j-1}}{h^2})$ $+\frac{4}{6}(\frac{\phi_{i,j+1}-2\phi_{i,j}+\phi_{i,j-1}}{h^2})$ $+\frac{1}{6}(\frac{\phi_{i-1,j+1}-2\phi_{i-1,j}+\phi_{i-1,j-1}}{h^2})$
$\frac{\partial^2\phi}{\partial x\partial y}\|_{ij}$	$\frac{\phi_{i+1,j+1}-\phi_{i-1,j+1}-\phi_{i+1,j-1}+\phi_{i-1,j-1}}{4h^2}$	$\frac{\phi_{i+1,j+1}-\phi_{i-1,j+1}-\phi_{i+1,j-1}+\phi_{i-1,j-1}}{4h^2}$
$\frac{\partial\phi}{\partial x}\|_{ij}$	$\frac{\phi_{i+1,j}-\phi_{i-1,j}}{2h}$	$\frac{1}{6}(\frac{\phi_{i+1,j+1}-\phi_{i-1,j+1}}{2h})$ $+\frac{4}{6}(\frac{\phi_{i+1,j}-\phi_{i-1,j}}{2h})$ $+\frac{1}{6}(\frac{\phi_{i+1,j-1}-\phi_{i-1,j-1}}{2h})$
$\frac{\partial\phi}{\partial y}\|_{ij}$	$\frac{\phi_{i,j+1}-\phi_{i,j-1}}{2h}$	$\frac{1}{6}(\frac{\phi_{i+1,j+1}-\phi_{i+1,j-1}}{2h})$ $+\frac{4}{6}(\frac{\phi_{i,j+1}-\phi_{i,j-1}}{2h})$ $+\frac{1}{6}(\frac{\phi_{i-1,j+1}-\phi_{i-1,j-1}}{2h})$
$\phi\|_{ij}$	ϕ_{ij}	$\frac{1}{6}(\frac{\phi_{i+1,j+1}+4\phi_{i,j+1}+\phi_{i-1,j+1}}{6})$ $+\frac{4}{6}(\frac{\phi_{i+1,j}+4\phi_{i,j}+\phi_{i-1,j}}{6})$ $+\frac{1}{6}(\frac{\phi_{i+1,j-1}+4\phi_{i,j-1}+\phi_{i-1,j-1}}{6})$

Table 12.6: 2-D Difference Operators for FD and FE Approximations.

$A_x = \frac{4+2\,cos(S)}{6}$		$A_y = \frac{4+2\,cos(G)}{6}$	
$B_x = \frac{sin(S)}{S}$		$B_y = \frac{sin(G)}{G}$	
$C_x = \frac{sin(S/2)}{(S/2)}$		$C_y = \frac{sin(G/2)}{(G/2)}$	

Table 12.7: Definition of discretization factors for solutions of the form $\phi = \phi_{ij}\, e^{j(\sigma x+\gamma y)}$. $\Delta x = \Delta y = h$ is the mesh spacing; $S \equiv \sigma h$; $G \equiv \gamma h$.

	FD	FE
$\frac{\partial^2\phi}{\partial x^2}\|_{ij}$	$-C_x^2\sigma^2\phi_{ij}$	$-A_yC_x^2\sigma^2\phi_{ij}$
$\frac{\partial^2\phi}{\partial y^2}\|_{ij}$	$-C_y^2\gamma^2\phi_{ij}$	$-A_xC_y^2\gamma^2\phi_{ij}$
$\frac{\partial^2\phi}{\partial x\partial y}\|_{ij}$	$-B_xB_y\sigma\gamma\phi_{ij}$	$-B_xB_y\sigma\gamma\phi_{ij}$
$\frac{\partial\phi}{\partial x}\|_{ij}$	$jB_x\sigma\phi_{ij}$	$jA_yB_x\sigma\phi_{ij}$
$\frac{\partial\phi}{\partial y}\|_{ij}$	$jB_y\gamma\phi_{ij}$	$jA_xB_y\gamma\phi_{ij}$
$\phi\|_{ij}$	ϕ_{ij}	$A_xA_y\phi_{ij}$

Table 12.8: Difference operators as in Table 12.6 for $\phi = \phi_{ij}\, e^{i(\sigma x+\gamma y)}$. The discretization factors A, B and C are defined in Table 12.7 and approach unity as the mesh is refined.

12.3 2-D Elliptic Equations

Laplace Equation on Bilinear Rectangles

In two dimensions, the Laplace Equation is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{12.62}$$

For Fourier modes of the form

$$u(x,y) = U_{\sigma\gamma} e^{j(\sigma x + \gamma y)} \tag{12.63}$$

we have the dispersion relationship

$$\sigma^2 + \gamma^2 = 0 \tag{12.64}$$

We imagine a boundary along the x-axis with Dirichlet boundary condition characterized by a Fourier series with real wavenumbers $\sigma \equiv 2\pi/\lambda$. The problem then is to find the Fourier spectrum in the y-direction, given σ. The dispersion relationship is readily rearranged:

$$j\gamma = \pm\sigma \tag{12.65}$$

and reveals the familiar result that a periodic mode in the x dimension (σ real) must be accompanied by an exponential decay mode in the y dimension (γ imaginary). It is convenient to multiply both terms by h^2:

$$S^2 + G^2 = 0 \tag{12.66}$$

These relations characterize the analytic solution.

The Galerkin approximation on linear elements is

$$A_y C_x^2 \sigma^2 + A_x C_y^2 \gamma^2 = 0 \tag{12.67}$$

or equivalently

$$\frac{C_x^2}{A_x} S^2 + \frac{C_y^2}{A_y} G^2 = 0 \tag{12.68}$$

The factor $\frac{C^2}{A}$ is plotted above in Figure 12.4. Over the well-resolved range it is close to unity and rises monotonically with S or G, reaching a roughly 3% error at $\lambda = 10h$ (table 12.1).

Figure 12.10 plots the dispersion relation for both analytic and numerical cases. The fidelity is qualitatively and quantitatively excellent over the well-resolved range. The curvature in this Figure indicates that for a given real σ, ($S^2 \geq 0$), the numerical damping errs on the high side relative to the analytic. The roles of S and G are symmetric.

This numerical relationship is monotonic over the whole range of $0 < S < 2\pi/3$ *i.e.* for $\lambda \geq 3h$. In this range, G is strictly imaginary. For larger S ($\lambda < 3h$), G becomes complex with real part $= \pi$, indicating that at these very coarse values of S, the accompanying G represents an oscillatory mode in the y-direction with $\lambda = 2h$. The imaginary part of G for these modes represents very high damping, with $e-$folding lengths of less than one mesh spacing from $S = 2\pi/3$ all the way to the Nyquist point $S = \pi$. These very poorly-resolved modes are trapped close to their origin, as for example at a boundary where a Dirichlet condition is enforced.

In Figure 12.11 the imaginary part of G is plotted for all real values of $S > 0$. The cutoff point is clearly visible at $S = 2\pi/3$. The e-folding length of the G mode is $1/IM(G)$, in units of h.

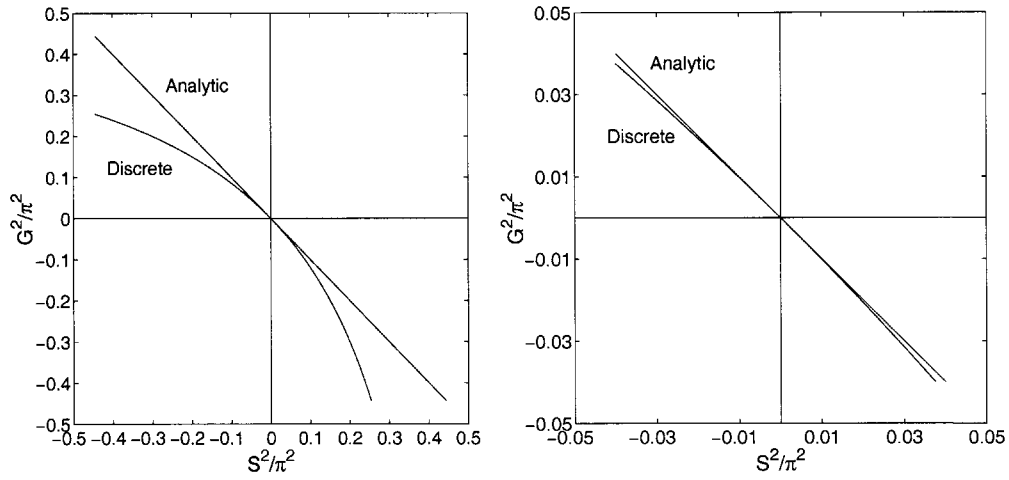


Figure 12.10: Dispersion relation for the 2-D Laplace equation in analytic and numerical form. Left: Coarse range of (S^2, G^2) terminating at the cutoff point $\sigma h = 2\pi/3$ *i.e.* $\lambda = 3h$. Over this range, G is purely imaginary indicating a monotonic exponential decay in the y-direction. Right: Well-resolved range terminating at $\lambda = 10h$.

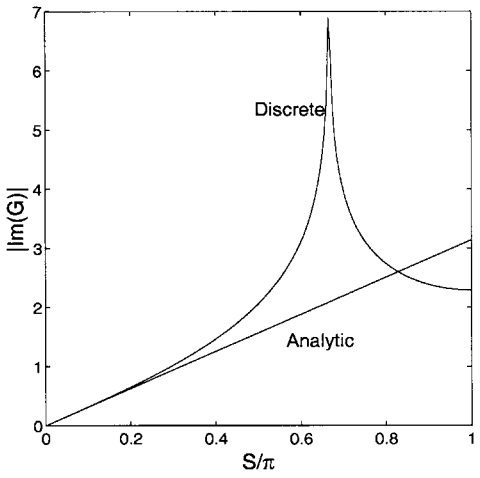


Figure 12.11: 2-D Laplace equation: Imaginary part of G versus S. (The $e-$folding length is $1/Im(G)$, in units of h.) Below $S = 2\pi/3$, $Re(G) = 0$, indicating monotonic decay as in the analytic solution. For higher values of S, $Re(G) = \pi$, indicating an oscillatory mode of length $2h$. This plot shows that for a Fourier mode in the x-direction with reasonable resolution, the corresponding y-mode is qualitatively correct and quantitatively accurate; while very poorly resolved x-modes are trapped close to their source by high damping rates in the y-direction and with node-to-node oscillations for the worst cases $\lambda < 3h$.

Helmholtz Equation on Bilinear Rectangles

In two dimensions, the Helmholtz Equation is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + k^2 u = 0 \tag{12.69}$$

The wavenumber k is often a scaled excitation frequency in many applications; it sets an internal length scale for the problem, $l \equiv 2\pi/k$, a feature missing in the Laplace equation. Thus a numerical solution must resolve this length scale as well as any introduced at boundaries.

As in the 2-D Laplace analysis, we imagine a boundary along the x-axis characterized by a Fourier series with wavenumbers $\sigma \equiv 2\pi/\lambda$. The problem then is to find the Fourier spectrum in the y-direction, given k^2 and σ. We restrict this analysis to real, positive values of k^2 and σ.

For Fourier modes of the form

$$u(x,y) = U_{\sigma\gamma} e^{j(\sigma x + \gamma y)} c \tag{12.70}$$

we have the dispersion relationship

$$\sigma^2 + \gamma^2 = k^2 \tag{12.71}$$

or equivalently

$$j\gamma = \pm j\sqrt{k^2 - \sigma^2}, \qquad k^2 > \sigma^2 \tag{12.72}$$

$$j\gamma = \pm\sqrt{\sigma^2 - k^2}, \qquad \sigma^2 > k^2 \tag{12.73}$$

For a given k^2, long-wavelength periodic modes in the x dimension (small σ^2) will be accompanied by undamped wave propagation in the y-direction. Short-wavelength x-modes (large σ^2) will be accompanied by pure exponential decay in the y-direction. The cutoff point is $\sigma = k$ or, in terms of the internal length scale, $l = \lambda$.

The Galerkin approximation on linear elements is

$$A_y C_x^2 \sigma^2 + A_x C_y^2 \gamma^2 = k^2 A_x A_y \tag{12.74}$$

or equivalently

$$\frac{C_x^2}{A_x} S^2 + \frac{C_y^2}{A_y} G^2 = K^2 \tag{12.75}$$

with $K^2 \equiv k^2 h^2$.

The solution for G^2 is plotted in Figure 12.12, for an intermediate value of $K^2 = .04$. As in the analytic solution, we have real, positive G^2 for small S^2, reverting to real, negative G^2 near the analytic crossover point $S^2 = K^2$. For larger values of S^2, G^2 remains negative and real as in the analytic solution. Near $S = 2\pi/3$ there is a transformation to complex G^2, qualitatively departing from the analytic behaviour as in the case of the 2D Laplace equation above. These very poorly resolved modes have $Re(G) = \pi$ *i.e.* their wavelengths are all equal to $2h$. Their imaginary parts all exceed unity, indicating e-folding lengths of less than one grid spacing. Qualitatively, we have fidelity to the analytic solution up to this break point: wave-like behaviour over the long-wavelength end of the σ spectrum, exponential decay over the short-wave end, and a good approximation of the crossover point $S^2 = K^2$ separating these two extremes. Quantitatively, in the well-resolved

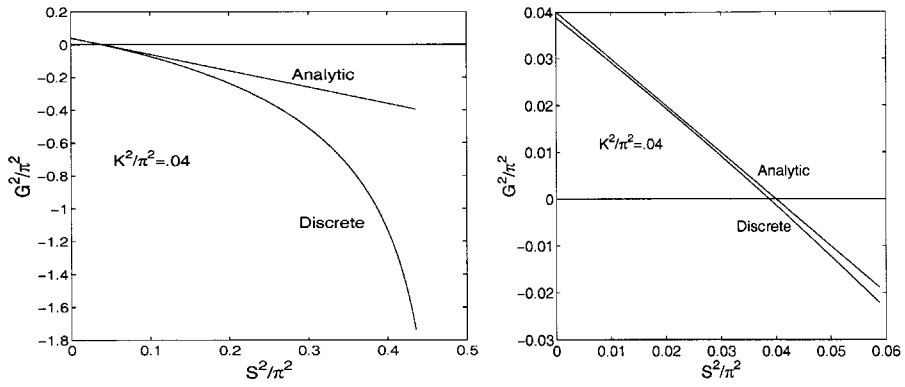


Figure 12.12: Dispersion relation for the 2-D Helmholtz equation with $K^2/\pi^2 = .04$. Left: Coarse range S^2, G^2 terminating near the cutoff point $\sim \sigma h = 2\pi/3$ *i.e.* $\lambda = 3h$. Right: Well-resolved range.

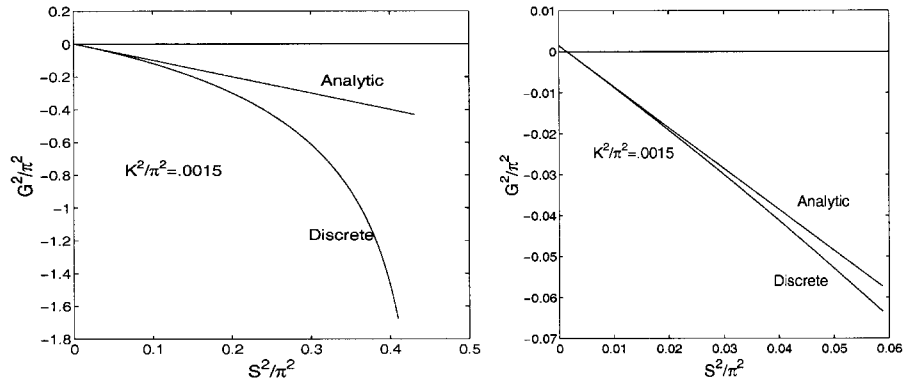


Figure 12.13: Dispersion relation for the 2-D Helmholtz equation with $K^2/\pi^2 = .0015$. Same setup as Figure 12.12.

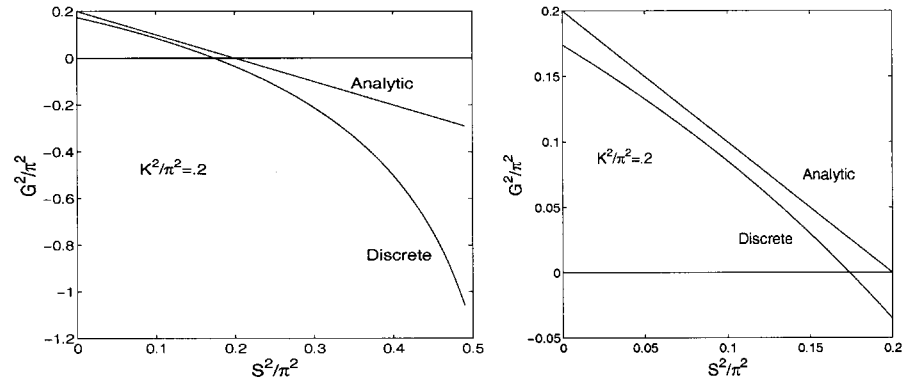


Figure 12.14: Dispersion relation for the 2-D Helmholtz equation with $K^2/\pi^2 = .2$. Same setup as Figure 12.12.

range we have excellent agreement with the analytic solution. Generally, where exponential decay is present, it is overestimated by the numerical solution. At the very coarse end of resolution, S modes are strongly trapped near their source.

In Figure 12.13 we show the case of high internal resolution, $K^2/\pi^2 = .0015$. This case has the same general features, and is close to the Laplace equation except at very high resolution (near the origin of the plot).

In Figure 12.14 we show the case $K^2/\pi^2 = .2$. This represents coarse resolution of the internal length scale, with $l/h \approx 4.5$. The dispersion relation is qualitatively the same as the previous cases. There is an overall loss of accuracy, however, which is of order 10% as apparent in the plots. Essentially, a coarse value of K^2 requires comparable coarseness in either G^2 or S^2, or both. The transformation to complex G^2 occurs around $S^2/\pi^2 = .55$, beyond which G represents oscillatory modes with $\lambda = 2h$ and very strong damping as in the previous cases.

12.4 Diffusion Equation

Here we have

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \tag{12.76}$$

We seek solutions of the form

$$u(x,t) = U_\sigma e^{\alpha t} e^{j\sigma x} \tag{12.77}$$

The exact solution requires

$$\alpha = -D\sigma^2 \tag{12.78}$$

i.e.

$$-\frac{\alpha}{D\sigma^2} = 1 \tag{12.79}$$

indicating that all modes will decay, with more rapid decay for short wavelength modes (large σ) than for long wavelength modes (small σ). This expresses the smoothing effect of the diffusion operator.

Consider first the discrete form of the x-derivative, leaving the time derivative in its continuous form. On 1-D linear finite elements we have the system of ODE's

$$\frac{\partial}{\partial t}\left(\frac{u_{i+1} + 4u_i + u_{i-1}}{6}\right) = D\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right) \tag{12.80}$$

Substituting 12.77, we obtain

$$A\alpha = -DC^2\sigma^2 \tag{12.81}$$

or

$$-\frac{\alpha}{D\sigma^2} = \frac{C^2}{A} \tag{12.82}$$

The discretization factor $\frac{C^2}{A}$ approaches unity at small S, so the numerical and analytic dispersion relations converge there. For finite S, $(\frac{C^2}{A} - 1)$ quantifies the discretization error. This was studied earlier (figure 12.4 and Table 12.1). We have quantitative agreement to about 3% for reasonable resolution, and qualitative agreement over the whole range of possible S. The numerical solution consistently overestimates the magnitude of α.

Next we discretize the time-domain. We will use a standard two-level timestepping method, with k indicating time level:

$$\frac{du}{dt}^k \rightarrow \frac{u^{k+1} - u^k}{\Delta t} \tag{12.83}$$

on the left side, and

$$u^{k+\theta} \rightarrow \theta u^{k+1} + (1-\theta)u^k \tag{12.84}$$

on the right side. The result, after Fourier transformation, is

$$A\frac{(\beta - 1)}{\Delta t} = -D[\theta\beta + (1-\theta)]C^2\sigma^2 \tag{12.85}$$

in which $\beta \equiv e^{\alpha\Delta t}$ is the ratio of the solution at adjacent time levels:

$$\beta \equiv e^{\alpha\Delta t} = u_i^{k+1}/u_i^k \tag{12.86}$$

Rearranging terms we have

$$(\beta - 1) = -\frac{D\Delta t}{h^2}[\theta\beta + (1-\theta)]\frac{C^2}{A}S^2 \tag{12.87}$$

Introducing the dimensionless timestep $r \equiv \frac{D\Delta t}{h^2}$, we have

$$(\beta - 1) = -r[\theta\beta + (1-\theta)]\frac{C^2}{A}S^2 \tag{12.88}$$

and finally

$$\beta(1 + r\theta\frac{C^2S^2}{A}) = (1 - (1-\theta)r\frac{C^2S^2}{A}) \tag{12.89}$$

$$\beta = \frac{(1 - (1-\theta)r\frac{C^2S^2}{A})}{(1 + r\theta\frac{C^2S^2}{A})} = 1 - \frac{r\frac{C^2S^2}{A}}{(1 + r\theta\frac{C^2S^2}{A})} \tag{12.90}$$

β is the ratio of solutions at adjacent points in time, and this expression contains a wealth of information. Below we examine it for stability, ($|\beta| \leq 1$) monotonicity, ($\beta \geq 0$) and accuracy (fidelity between β and its analytic counterpart).

We will restrict our analysis to $0 \leq \theta \leq 1$, and to real, non-negative wavenumbers $0 \leq S \leq \pi$. C^2 and A are therefore real and positive, as is the dimensionless timestep r.

Stability ($-1 \leq \beta \leq 1$)

- $\beta \leq 1$ for all r and $S \geq 0$. The limiting case $\beta = 1$ occurs as $S \rightarrow 0$, as in the analytic solution.
- The condition $-1 \leq \beta$ controls stability. This requires

$$\frac{r\frac{C^2S^2}{A}}{(1 + r\theta\frac{C^2S^2}{A})} \leq 2 \tag{12.91}$$

i.e.

$$r\frac{C^2S^2}{A}(1 - 2\theta) \leq 2 \tag{12.92}$$

This is always met if θ exceeds $1/2$, irrespective to the size of r or S. Thus we have unconditional stability for $\theta \geq 1/2$.

- For smaller θ, we have

$$r\frac{C^2S^2}{A} \leq \frac{2}{(1-2\theta)} \tag{12.93}$$

For a given r and θ, the worst case is the highest wavenumber $S = \pi$, for which $C^2S^2 = 4$ and $A = 1/3$. In that case, we have

$$r \leq \frac{1}{6(1-2\theta)} \tag{12.94}$$

Any practical calculation will have information at all wavenumbers, due to noise in initial conditions, data, and the accumulation of roundoff error. Thus this limitation on stability for the shortest waves $S = \pi$ governs even in otherwise well-resolved calculations, since tiny amounts of noise will be amplified and ultimately overwhelm the calculation if condition 12.94 is violated.

The summary result for stability is:

- $\theta \geq 1/2$: unconditional stability
- $\theta < 1/2$: $r \leq \frac{1}{6(1-2\theta)}$, limited by $S = \pi$

Monotonicity ($0 \leq \beta$)

Assuming stability, we are also concerned with the condition that solutions do not "wiggle" in time, which occurs when β is negative. This would be a qualitative departure from the analytic solution. The condition for a monotone solution is $\beta \geq 0$, and from 12.90 we have

$$\frac{r\frac{C^2S^2}{A}}{(1+r\theta\frac{C^2S^2}{A})} \leq 1 \tag{12.95}$$

$$r\frac{C^2S^2}{A}(1-\theta) \leq 1 \tag{12.96}$$

The case $\theta = 1$ is special in that it preserves monotonicity for all S, irrespective of the size of the timestep r. For a given r and $\theta < 1$, the largest wavenumbers S will be at risk of losing monotonicity. The worst case is $S = \pi$, for which we have $C^2S^2 = 4$ and $A = 1/3$:

$$r \leq \frac{1}{12}\frac{1}{(1-\theta)} \tag{12.97}$$

Since we are assuming stability, the poorly resolved modes will decay and there should be no mechanism for them to achieve significant amplitude. Therefore their nonmonotonicity may not be a practical problem. The threshold for reasonable resolution is $\lambda = 10h$ *i.e.* $S = .2\pi$. For this value of S, we have $C^2S^2/A \approx .40$. Thus,

$$r \leq 2.5\frac{1}{(1-\theta)} \tag{12.98}$$

should provide monotone behaviour for reasonably-well-resolved modes. This may be a more practical criterion for discretization.

The summary result for monotonicity is:

- $\theta = 1$: unconditional monotonicity
- $\theta < 1$: $r \leq \frac{1}{12(1-\theta)}$ gives monotonicity for the entire spectrum $0 \leq S \leq \pi$. For higher values of r, monotonicity is lost, beginning with the shortest modes $S = \pi$.
- $\theta < 1$: $r \leq \frac{2.5}{(1-\theta)}$ gives monotonicity for the well-resolved part of the spectrum, $0 \leq S \leq .2\pi$

Figures 12.15 and 12.16 illustrate these properties of stability and monotonicity.

Accuracy

The above analyses of β are useful in a qualitative sense and predict the conditions under which important departures from the analytic behaviour occur. For a more quantitative analysis we examine the ratio of β to the analytic value β_a:

$$\frac{\beta}{\beta_a} = \frac{\beta}{e^{-D\sigma^2\Delta t}} = \frac{\beta}{e^{-rS^2}} \tag{12.99}$$

This ratio is the ratio of numerical to analytic solution after one time step, starting from the same initial conditions. A good method would have this ratio close to unity. However, as Δt is refined, β converges to its analytic counterpart and therefore the information in this measure is lost. As a remedy, we examine the ratio of numerical to analytic solution after a characteristic time has passed:

$$\tau = N\Delta t \tag{12.100}$$

and examine the "Propagation Factor" T:

$$T = [\frac{\beta}{\beta_a}]^N = [\frac{\beta}{e^{-rS^2}}]^N \tag{12.101}$$

In this way, as Δt is decreased, N is increased, partially offsetting the per-step convergence by requiring more steps. We take τ to be the e-folding time of the analytic solution for a given mode[1]:

$$\tau = \frac{1}{D\sigma^2} \tag{12.102}$$

and thus

$$N = \frac{1}{D\Delta t\sigma^2} = \frac{1}{rS^2} \tag{12.103}$$

Finally,

$$T = \frac{\beta^{\frac{1}{rS^2}}}{e^{-1}} \tag{12.104}$$

This is a useful measure of accuracy for stable, monotone modes, with $T = 1$ indicating perfection.

Figure 12.17 illustrates the use of T.

Leapfrog Time-Stepping

As a complement to the 2-level timestepping scheme studied above, consider the explicit, 3-level Leapfrog scheme with

$$\frac{du^k}{dt} \rightarrow \frac{u^{k+1} - u^{k-1}}{2\Delta t} \tag{12.105}$$

[1]Note that τ depends on σ; large-σ modes decay more quickly.

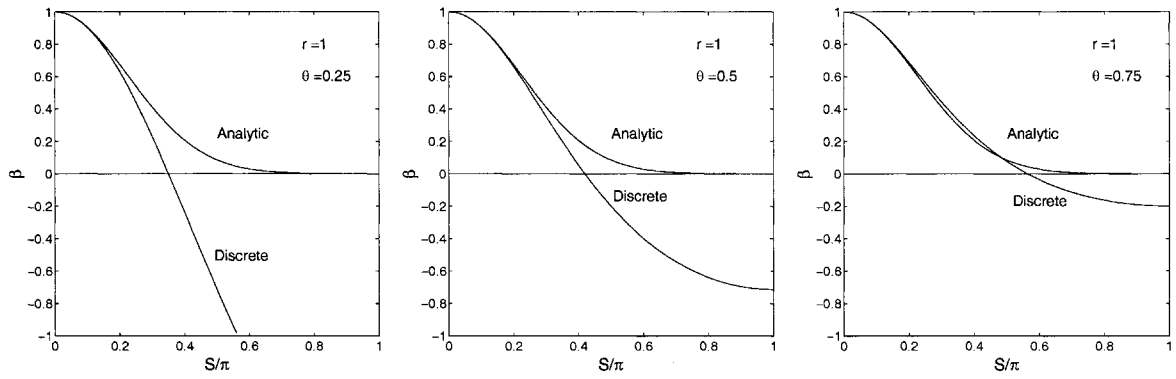


Figure 12.15: $\beta \equiv e^{\alpha \Delta t}$ versus S for the diffusion equation, with dimensionless timestep $r = 1$. β represents the per-timestep growth rate of a Fourier mode with wavenumber S. There are three zones. 1) Stable, monotone modes have $0 \leq \beta \leq 1$; this characterizes the analytic solution. 2) Stable, nonmonotone modes $-1 \leq \beta \leq 0$; these modes change sign every timestep and have no analytic counterpart. 3) Unstable modes, $\beta \leq -1$; these modes can be seeded by roundoff error, data noise, etc. and grow without bound.

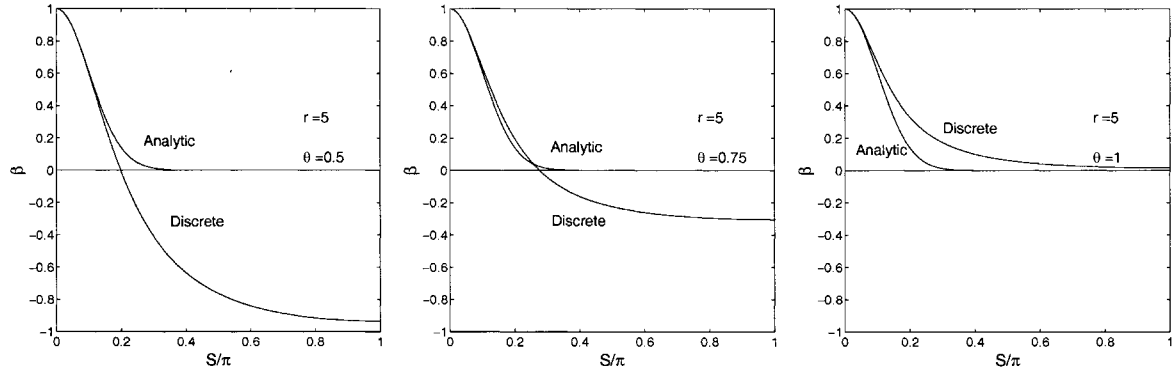


Figure 12.16: Same as Figure 12.15, but with $r = 5$.

$$\frac{u^{k+1} - u^{k-1}}{2\Delta t} = D\frac{d^2u^k}{dx^2} \tag{12.106}$$

Introduction of 12.77 leads to

$$A\frac{\beta^2 - 1}{2\Delta t} = -DC^2\sigma^2\beta \tag{12.107}$$

As above, $\beta \equiv e^{\alpha\Delta t} \equiv u_i^{k+1}/u_i^k$ is the ratio of the solution at adjacent time levels. Rearranging terms we have

$$(\beta^2 - 1) = -\left[\frac{2D\Delta t}{h^2}\frac{C^2S^2}{A}\right]\beta \tag{12.108}$$

Introducing the dimensionless timestep $r \equiv \frac{D\Delta t}{h^2}$, we have the quadratic equation

$$\beta^2 + \left[2r\frac{C^2S^2}{A}\right]\beta - 1 = 0 \tag{12.109}$$

There are 2 roots:

$$\beta = -\left[r\frac{C^2S^2}{A}\right] \pm \sqrt{\left[r\frac{C^2S^2}{A}\right]^2 + 1} \tag{12.110}$$

Since $\left[r\frac{C^2S^2}{A}\right] \geq 0$, the negative option always represents an unstable oscillatory mode

$$\beta^- \leq -1 \tag{12.111}$$

irrespective of timestep or wavenumber. This scheme is unconditionally unstable.

Because the positive option is stable, monotonic, and a good approximant to the analytic solution, this scheme could be used if an effective filter could be devised to prevent the growth of the unstable modes which have no analytic counterpart. This however would require the introduction of an additional formal algebraic operation which would have to be analyzed.

3-level Implicit Time-Stepping

A generalization of the leapfrog scheme is

$$\frac{u^{k+1} - u^{k-1}}{2\Delta t} = D\left[\frac{\theta}{2}\frac{d^2u}{dx^2}^{k+1} + (1-\theta)\frac{d^2u}{dx^2}^{k} + \frac{\theta}{2}\frac{d^2u}{dx^2}^{k-1}\right] \tag{12.112}$$

Here the approximation to the spatial derivative is centered and implicit, requiring matrix factorization for the unknowns at time $k+1$. The leapfrog case studied above is recovered here with $\theta = 0$. Following the above development, with $R \equiv r\frac{C^2S^2}{A}$, we obtain

$$\beta^2 - 1 = -2R\left[\frac{\theta}{2}(\beta^2 + 1) + (1-\theta)\beta\right] \tag{12.113}$$

$$[1 + R\theta]\beta^2 + [2R(1-\theta)]\beta + [-1 + R\theta] = 0 \tag{12.114}$$

The roots are

$$\beta = \frac{-R(1-\theta) \pm \sqrt{[R(1-\theta)]^2 - [1 + R\theta][-1 + R\theta]}}{1 + R\theta} \tag{12.115}$$

and a little rearrangement produces

$$\beta = \frac{-R(1-\theta) \pm \sqrt{1 + R^2[1-2\theta]}}{1 + R\theta} \tag{12.116}$$

Stability for these roots is most easily obtained by applying the general result for quadratic roots in the appendix. The stability condition is

$$\theta > 1/2 \tag{12.117}$$

irrespective of timestep or wavenumber. This is consistent with the leapfrog findings above – unconditional instability for $\theta = 0$.

We restrict our attention to the range $1/2 \leq \theta \leq 1$. The $-\sqrt{}$ option for β represents a spurious temporal mode which is generally negative, representing time-oscillations with period $2\Delta t$, for all wavelengths irrespective of resolution. The $+\sqrt{}$ option mimics the analytic solution for good resolution. Its monotonicity requires β real and in the range $0 \leq \beta \leq 1$. β will become complex when $R^2[2\theta - 1]$ exceeds unity; so real-valued β requires

$$R^2 \leq \frac{1}{[2\theta - 1]} \tag{12.118}$$

Assuming this, $\beta \geq 0$ requires

$$\sqrt{1 - R^2[2\theta - 1]} \geq R(1-\theta) \tag{12.119}$$

For $\theta \leq 1$, the monotonicity requirement is

$$R \leq 1/\theta \tag{12.120}$$

This requirement is more strict than the non-complex requirement (12.118)

Summarizing, this scheme has two temporal modes associated with each S. Both are unconditionally stable for $\theta > 1/2$, and unconditionally unstable otherwise. In the range $1/2 \leq \theta \leq 1$, one of the two modes will generally be nonmonotonic and spurious for all wavelengths, while the other is realistic and *monotonic* provided that

$$R < \frac{1}{\theta} \tag{12.121}$$

Since $R \equiv r\frac{C^2S^2}{A}$, the worst case is the shortest-wavelength modes for which $\frac{C^2S^2}{A} = 12$. Thus, we expect *non-monotonic* behaviour to infect the realistic modes of the solution when

$$r > \frac{1}{12\theta} \tag{12.122}$$

The loss of monotonicity will be initiated at the short-wavelength end of the Fourier spectrum and progress to longer-wavelength modes with rising r. At the threshold of reasonable resolution, $S = .2\pi$ ($\lambda = 10h$), we have $\frac{C^2S^2}{A} \approx .4$, and these modes will *lose monotonicity* for

$$r > \frac{2.5}{\theta} \tag{12.123}$$

12.5 Explicit Wave Equation

In one spatial dimension, we have the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{12.124}$$

with C^2 real and positive. For solutions of the form 12.77, we have the analytic dispersion relation

$$\alpha^2 = -c^2\sigma^2 \tag{12.125}$$

$$\alpha = \pm jc\sigma \tag{12.126}$$

indicating that all real wavenumber modes propagate with wavespeed c:

$$u(x,t) = U_\sigma e^{\alpha t} e^{j\sigma x} = U_\sigma e^{j\sigma(x \pm ct)} \tag{12.127}$$

As in the diffusion equation analysis, it is instructive to discretize only the x-derivative, leaving the time derivative continuous. On 1-D linear finite elements we have the system of ODE's

$$\frac{d^2}{dt^2}\left(\frac{u_{i+1} + 4u_i + u_{i-1}}{6}\right) = c^2\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right) \tag{12.128}$$

Substituting 12.77, we obtain

$$A\alpha^2 = -c^2C^2\sigma^2 \tag{12.129}$$

or

$$-\frac{\alpha^2}{c^2\sigma^2} = \frac{C^2}{A} \tag{12.130}$$

The discretization factor $\frac{C^2}{A}$ again appears as the effect of the spatial discretization. $(\frac{C^2}{A} - 1)$ quantifies the discretization error. This was studied earlier (figure 12.4 and Table 12.1). We have quantitative agreement to about 3% for reasonable resolution, and because $\frac{C^2}{A}$ is real and positive, we have qualitative agreement over the whole range of possible S; $|\alpha|$ rises monotonically with $|\sigma|$.

Next we discretize the time-domain. Because this equation is second-order in time, we must use at least three discrete time levels. Consider the centered, explicit approximation

$$\frac{u^{k+1} - 2u^k + u^{k-1}}{\Delta t^2} = c^2 \frac{\partial^2 u^k}{dx^2} \tag{12.131}$$

which is in some ways analogous to the leapfrog method examined above. Finite element discretization and Fourier transformation yields

$$A\frac{(\beta^2 - 2\beta + 1)}{\Delta t^2} = -c^2C^2\sigma^2\beta \tag{12.132}$$

$$(\beta^2 - 2\beta + 1) = -\frac{c^2\Delta t^2}{h^2}\frac{C^2S^2}{A}\beta \tag{12.133}$$

As above, $\beta \equiv e^{\alpha\Delta t}$ is the ratio of the solution at adjacent time levels. Rearranging terms we have

$$\beta^2 - \beta\left[2 - \mathrm{Co}^2\frac{C^2S^2}{A}\right] + 1 = 0 \tag{12.134}$$

with Co the Courant number,

$$\mathsf{Co} \equiv \frac{c\Delta t}{h} \tag{12.135}$$

The roots of this equation are

$$\beta = [1 - \mathsf{Co}^2 \frac{C^2 S^2}{2A}] \pm j\sqrt{1 - \left[1 - \mathsf{Co}^2 \frac{C^2 S^2}{2A}\right]^2} \tag{12.136}$$

Stability

For $\left[1 - \mathsf{Co}^2 \frac{C^2 S^2}{2A}\right]^2 < 1$, β is complex and $|\beta| = 1$, indicating a propagating mode with undamped amplitude – neutral stability, qualitatively faithful to the analytic solution. For $\left[1 - \mathsf{Co}^2 \frac{C^2 S^2}{2A}\right]^2 > 1$, β is real and $\beta^- < -1$ *i.e.* that root is unstable. So the stability criterion is

$$\left[1 - \mathsf{Co}^2 \frac{C^2 S^2}{2A}\right] > -1 \tag{12.137}$$

$$\left[\mathsf{Co}^2 \frac{C^2 S^2}{2A}\right] < 2 \tag{12.138}$$

$$\left[\mathsf{Co}^2\right] < \frac{4A}{C^2 S^2} \tag{12.139}$$

The worst case is $S = \pi$, in which case $C^2 S^2 = 4$ and $A = 1/3$. So the stability criterion is

$$\mathsf{Co}^2 \equiv \frac{c^2 \Delta t^2}{h^2} \leq \frac{1}{3} \tag{12.140}$$

Accuracy

As for the diffusion equation, we examine the propagation factor, the ratio of numerical to analytic solution after N time steps, sufficient to advance the solution one characteristic time $\tau = N\Delta t$:

$$T = [\frac{\beta}{\beta_a}]^N \tag{12.141}$$

Here we take τ to be the time for analytic propagation of one wavelength:

$$\tau = \frac{2\pi}{\sigma c} \tag{12.142}$$

$$N = \frac{2\pi}{\sigma c \Delta t} = \frac{2\pi}{\mathsf{Co} S} \tag{12.143}$$

For this undamped wave equation, this choice of τ results in $\beta_a^N = 1$, so the expression for T is especially simple:

$$T = \beta^N \tag{12.144}$$

An accurate method would have T close to unity. Since for this method, $|\beta| = 1$, we have $|T| = 1$. But since T is complex, its argument (phase) will contain discretization error. $arg(T)/2\pi - 1$ is a normalized measure of phase error due to discretization. This is displayed in Figure 12.18. For well-resolved modes ($S/\pi \leq .2$ *i.e.* . $\lambda \geq 10h$), the phase error is within about 2% for all stable Co.

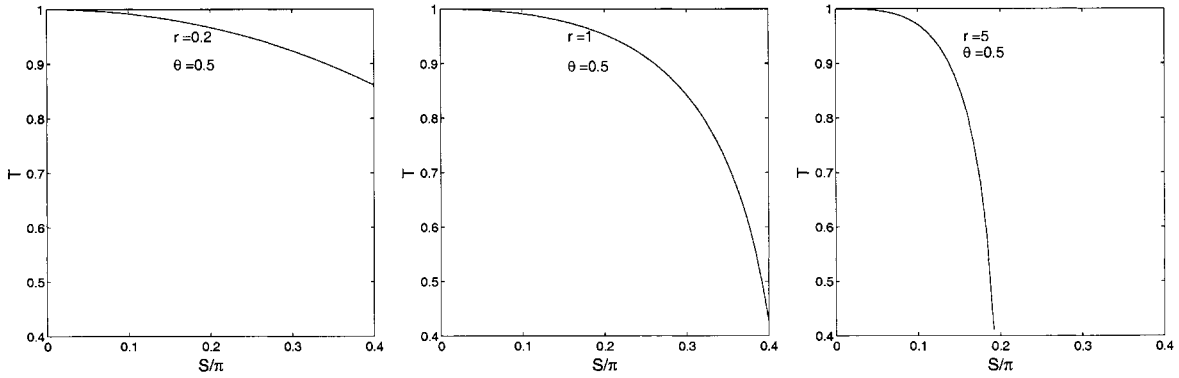


Figure 12.17: Plots of the Propagation Factor T for the 2-level-in-time diffusion equation, over the well-resolved range of S for several values of dimensionless timestep r.

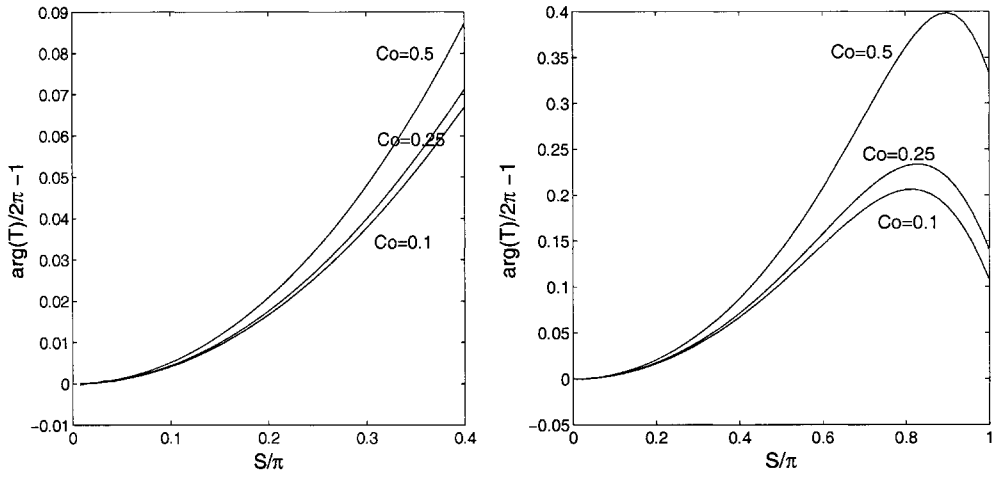


Figure 12.18: Plots of the Propagation Factor phase error for the explicit wave equation. Left: well-resolved range of S; right, full range. Co is the Courant number, $\frac{c\Delta t}{h}$.

12.6 Implicit Wave Equation

The Courant number restriction limits the time step size for the explicit wave equation. Consider the centered-in-time implicit version:

$$\frac{u^{k+1} - 2u^k + u^{k-1}}{\Delta t^2} = c^2 \left[\frac{\theta}{2} \frac{\partial^2 u}{dx^2}^{k+1} + (1-\theta) \frac{\partial^2 u}{dx^2}^{k} + \frac{\theta}{2} \frac{\partial^2 u}{dx^2}^{k-1} \right] \tag{12.145}$$

This will require a matrix factorization and solution for the unknowns at time level $k+1$. Finite element discretization and Fourier transformation yields

$$A \frac{(\beta^2 - 2\beta + 1)}{\Delta t^2} = -c^2 C^2 \sigma^2 \left[\frac{\theta}{2}(\beta^2 + 1) + (1-\theta)\beta \right] \tag{12.146}$$

$$(\beta^2 - 2\beta + 1) = -\mathsf{Co}^2 \frac{C^2 S^2}{A} \left[\frac{\theta}{2}(\beta^2 + 1) + (1-\theta)\beta \right] \tag{12.147}$$

As above, $\beta \equiv e^{\alpha \Delta t}$ is the ratio of the solution at adjacent time levels; and $\mathsf{Co} \equiv \frac{c\Delta t}{h}$ is the Courant number. Temporarily denoting $\chi = \mathsf{Co}^2 \frac{C^2 S^2}{A}$, we have

$$(\beta^2 - 2\beta + 1) = -\chi \left[\frac{\theta}{2}(\beta^2 + 1) + (1-\theta)\beta \right] \tag{12.148}$$

and rearranging terms we obtain

$$(1 + \frac{\theta}{2}\chi)\beta^2 + \beta[-2 + (1-\theta)\chi] + (1 + \frac{\theta}{2}\chi) = 0 \tag{12.149}$$

Solving for β, we obtain

$$\beta = \left[\frac{1 - \frac{(1-\theta)}{2}\chi}{1 + \frac{\theta}{2}\chi} \right] \pm j \sqrt{1 - \left[\frac{1 - \frac{(1-\theta)}{2}\chi}{1 + \frac{\theta}{2}\chi} \right]^2} \tag{12.150}$$

Stability

Provided the term under the radical is real, we have $|\beta| = 1$ and thus a neutrally stable representation of undamped propagating modes. Otherwise, inspection of 12.150 shows that $|\beta|$ exceeds unity. So stability requires

$$-1 \leq \left[\frac{1 - \frac{(1-\theta)}{2}\chi}{1 + \frac{\theta}{2}\chi} \right] \leq 1 \tag{12.151}$$

Since χ and θ are real and positive, the lower limit governs and the stability criterion is:

$$-1 \leq \left[\frac{1 - \frac{(1-\theta)}{2}\chi}{1 + \frac{\theta}{2}\chi} \right] \tag{12.152}$$

$$(1 - 2\theta)\chi \leq 4 \tag{12.153}$$

If θ exceeds $1/2$, then this is always true and we have unconditional stability. Otherwise, the stability is conditional:

$$\chi \leq \frac{4}{(1-2\theta)} \tag{12.154}$$

With $\chi = \mathsf{Co}^2 \frac{C^2S^2}{A}$, this is

$$\mathsf{Co}^2 \frac{C^2S^2}{A} \leq \frac{4}{(1-2\theta)} \tag{12.155}$$

The worst case is for $S = \pi$, when $\frac{C^2S^2}{A} = 12$. Thus we have the requirement

$$\mathsf{Co}^2 \leq \frac{1}{3(1-2\theta)} \quad , \quad \theta < \frac{1}{2} \tag{12.156}$$

and unconditional stability for $\theta > \frac{1}{2}$. As in previous analyses, the stability is dictated by the shortest wavelength modes, which are presumed to be present due to accumulated roundoff error and noisy data.

Accuracy

We quantify accuracy as in the explicit wave equation case, with the Propagation Factor $T = \beta^N$ with $N = 2\pi/\mathsf{Co}S$. As before, for stable modes we have $|\beta| = 1$ and thus any discretization error will be expressed in the phase of T. Figure 12.19 displays $arg(T)/2\pi - 1$ for some representative parameters. The small range of Co is the same used in Figure 12.18 for the explicit case. Relative to that case, the phase error has been reversed in sign. The larger range of Co begins to show significant phase errors – for example, for $\mathsf{Co} = 2$, the error approaches $-\pi$ for $S \approx .4$, indicating that these modes are propagating with only about 50% of their analytic wavespeed.

This is potentially misleading. Notice that since the wavespeed c is the ratio of the wavelength λ to the period P, the Courant number is the ratio of spatial resolution λ/h to temporal resolution $P/\Delta t$:

$$\mathsf{Co} = \frac{c\Delta t}{h} = \frac{\Delta t}{P} \cdot \frac{\lambda}{h} \tag{12.157}$$

This value of $S = .4$ represents $\lambda = 5h$. With $\mathsf{Co} = 1$ we have the same temporal resolution *i.e.* 5 Δt per period; and at $\mathsf{Co} = 2$ we have only $2.5\Delta t$ per period. To expect accurate propagation phasing under these coarse resolutions would be unrealistic. A more reasonable scenario for an implicit method would be the case where geometric constraints require unusually high spatial resolution. A small Courant number as required by the explicit method would then demand similarly fine resolution in time, just to maintain stability. A larger Courant number would be desireable in this situation, hence the potential value of the implicit method. Following this mesh refinement scenario, as one jumps from $\mathsf{Co} = .5$, explicit to $\mathsf{Co} = 2$, implicit in Figures 12.18 and 12.19, a comparable shift by a factor of 4 is appropriate in S/π, say from .2 to .05. This significantly alters the interpretation of these Figures.

12.7 Advection Equation

The advection equation in one space dimension is

$$\frac{\partial u}{\partial t} + V\frac{\partial u}{\partial x} = 0 \tag{12.158}$$

For solutions of the form $u(x,t) = U_\sigma e^{\alpha t} e^{j\sigma x}$, we have the analytic dispersion relation

$$\alpha = -j\sigma V \tag{12.159}$$

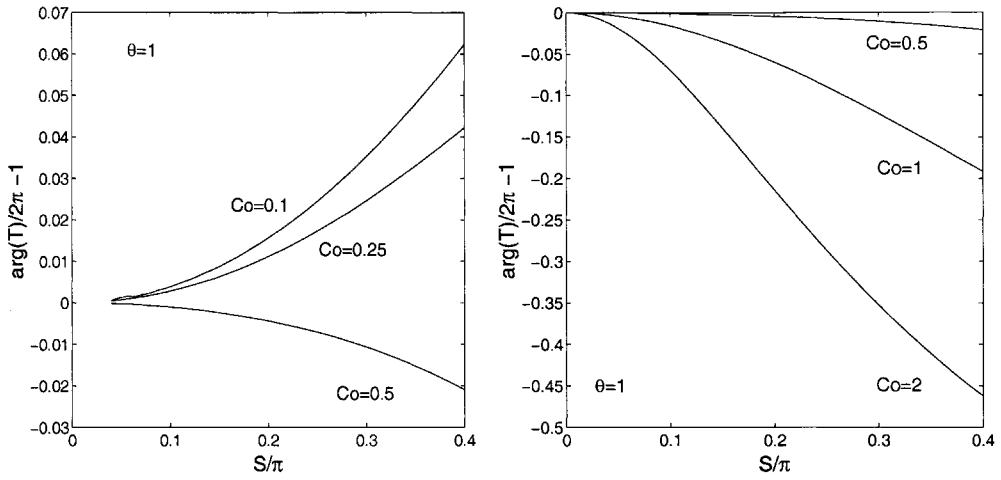


Figure 12.19: Plots of the Propagation Factor phase error for the implicit wave equation, $\theta = 1$. Left: small Co; right, large Co.

and solutions of the form $U_\sigma e^{j\sigma(x-Vt)}$ result, with period $P = \frac{2\pi}{\sigma V}$ and wavelength $\lambda = \frac{2\pi}{\sigma}$. The ratio β of $u(x, t + \Delta t)$ to $u(x, t)$ is

$$\beta_a = e^{\alpha \Delta t} = e^{-j\sigma V \Delta t} \tag{12.160}$$

with the subscript a indicating the analytic solution. Introducing a characteristic length h, we have in dimensionless form

$$\beta_a = e^{-jSK} \tag{12.161}$$

with $S = \sigma h$ as usual, and K the advective Courant Number

$$K \equiv \frac{V \Delta t}{h} \tag{12.162}$$

The number of time steps N required for the propagation of one wavelength is given by $N\Delta t = P$:

$$N = \frac{2\pi}{SK} \tag{12.163}$$

and $\beta_a^N = 1$.

Euler Advection

The simplest Euler time-discretization is

$$\frac{u^{k+1} - u^k}{\Delta t} + V \frac{\partial u^k}{\partial x} = 0 \tag{12.164}$$

Its Galerkin/Fourier form is

$$A(\beta - 1) + jSBK = 0 \tag{12.165}$$

and thus

$$\beta = 1 - j \frac{SB}{A} K \tag{12.166}$$

Since S, B, A, and K are real, we have a complex number with magnitude > 1:

$$|\beta|^2 = 1 + [\frac{SB}{A}K]^2 \tag{12.167}$$

This scheme is therefore unconditionally unstable.

Two-Level Implicit Advection

The general two-time-level scheme is

$$\frac{u^{k+1} - u^k}{\Delta t} + \theta V \frac{\partial u}{\partial x}^{k+1} + (1-\theta) V \frac{\partial u}{\partial x}^{k} = 0 \tag{12.168}$$

Following the usual path, the Galerkin-Fourier result is

$$A(\beta - 1) + jSBK[\theta\beta + (1-\theta)] = 0 \tag{12.169}$$

and we have

$$\beta = \frac{1 - (1-\theta) j \frac{SB}{A} K}{1 + \theta j \frac{SB}{A} K} \tag{12.170}$$

$$|\beta|^2 = \frac{1 + (1-\theta)^2 [\frac{SB}{A}K]^2}{1 + \theta^2 [\frac{SB}{A}K]^2} \tag{12.171}$$

Stability is guaranteed for $\theta \geq 1/2$.

For a measure of accuracy, we examine the propagation factor T,

$$T = (\frac{\beta}{\beta_a})^N = \beta^{\frac{2\pi}{SK}} \tag{12.172}$$

In the case $\theta = 1/2$, we have $|\beta| = 1$ and therefore $|T| = 1$, giving us perfect amplitude preservation. Discretization error for this case is confined to the phase of T.

In Figure 12.20 we display the normalized phase error $\frac{arg(T)}{-2\pi} - 1$, for $\theta = 1/2$. ($arg(T) = N \cdot arg(\beta)$.) A negative error in this context indicates that the numerical waves propagate more slowly than their analytic counterparts. This is the case for all waves in Figure 12.20, and the error grows with increasing S (decreasing wavelength). Note that an error of $-.1$ on this plot means that the numerical wave speed is 90% of the analytic; such a mode will be 180° out of phase in a spatial sense after propagating only 5 wavelengths. For $K = 2$, for example, numerical wave speeds vary by about 10% over the well-resolved range $0 < S < .2\pi$. Thus a well-resolved spatial signal will undergo increasing distortion as a simulation progresses, with the lower-wavenumber components propagating at close to the analytic speed, and the higher-wavenumber components (around $\lambda = 10h$) slowed down and thoroughly bogused. This effect is more pronounced at large K.

Leapfrog Advection

Finally, consider the Leapfrog scheme

$$\frac{u^{k+1} - u^{k-1}}{2\Delta t} + V \frac{\partial u}{\partial x}^{k} = 0 \tag{12.173}$$

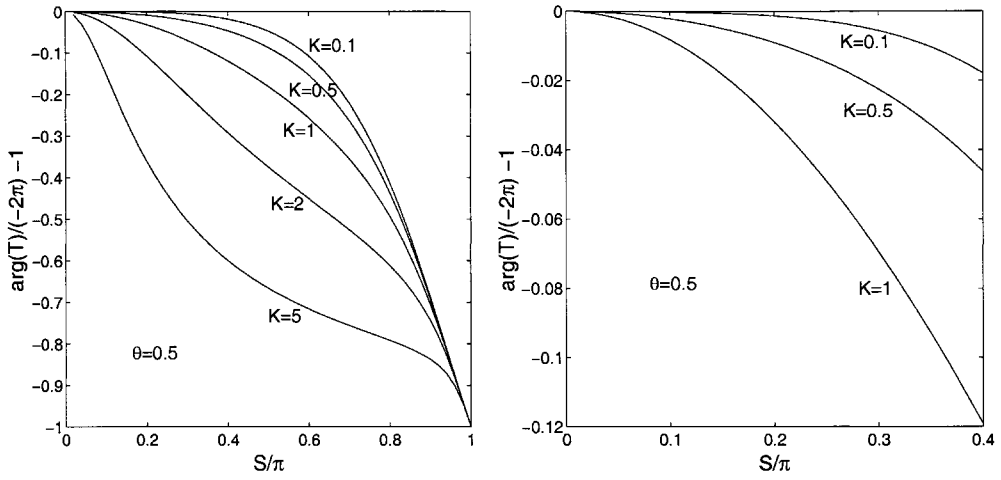


Figure 12.20: Normalized phase error, $\frac{arg(T)}{-2\pi} - 1$, for the two-level advection equation with $\theta = 1/2$. K is the advective Courant number, $V\Delta t/h$.

Its Galerkin/Fourier form is

$$A(\beta^2 - 1) + 2jSBK\beta = 0 \tag{12.174}$$

$$\beta^2 + [2j\frac{SB}{A}K]\beta - 1 = 0 \tag{12.175}$$

with S, B, A, and K real and positive. There are two roots:

$$\beta = -[j\frac{SB}{A}K] \pm \sqrt{1 - [\frac{SB}{A}K]^2} \tag{12.176}$$

If $\frac{SB}{A}K \leq 1$ then $|\beta|^2 = 1$ and this scheme is neutrally stable. Otherwise,

$$\beta = j\Big(-[\frac{SB}{A}K] \pm \sqrt{[\frac{SB}{A}K]^2 - 1}\Big) \tag{12.177}$$

Since by hypothesis $[\frac{SB}{A}K] > 1$, the negative option is unstable. The condition for stability is therefore

$$\frac{SB}{A}K \leq 1 \tag{12.178}$$

The limiting case is at the maximum of $\frac{SB}{A}$, which occurs at $S = 2\pi/3$ *i.e.* $\lambda = 3h$. For this mode, $\frac{SB}{A} = \sqrt{3}$ and thus the stability constraint for the leapfrog method is

$$K = \frac{V\Delta t}{h} \leq \frac{1}{\sqrt{3}} = .577 \tag{12.179}$$

By introducing a third level in time, the leapfrog scheme necessarily has two temporal modes β for each S, equation (12.176). In the stable range, the negative option has negative real part. In the limit of small K this mode has a temporal period of $2\Delta t$. It is therefore the parasitic or

spurious mode; the positive option corresponds to the analytic solution. This may be confirmed by Taylor Series expansions of β_a and β^+:

$$\beta_a = e^{-jSK} = 1 - jSK - \frac{(SK)^2}{2} + j\frac{(SK)^3}{6} + \ \ldots \tag{12.180}$$

$$\beta^+ = -j\frac{SB}{A}K + \left(1 - \frac{1}{2}[\frac{SB}{A}K]^2 - \frac{1}{8}[\frac{SB}{A}K]^4 + \ \ldots\right) \tag{12.181}$$

$$\beta^- = -j\frac{SB}{A}K - \left(1 - \frac{1}{2}[\frac{SB}{A}K]^2 - \frac{1}{8}[\frac{SB}{A}K]^4 + \ \ldots\right) \tag{12.182}$$

For high resolution, $S/A \to 1$ and the agreement between β_a and β^+ is correct to order $[SK]^3$. β^- has no analytic counterpart.

Control of the spurious mode β^- is a central challenge for this scheme. If that can be met, the accuracy of β^+ is measured by its propagation factor. Since $|\beta| = 1$, then also $|T| = 1$ and the discretization error is concentrated in the phase of T. As in the 2-level scheme, we consider the normalized phase error $arg(T)/(-2\pi) - 1$ with $arg(T) = Narg(\beta)$. This is plotted in Figure 12.21.

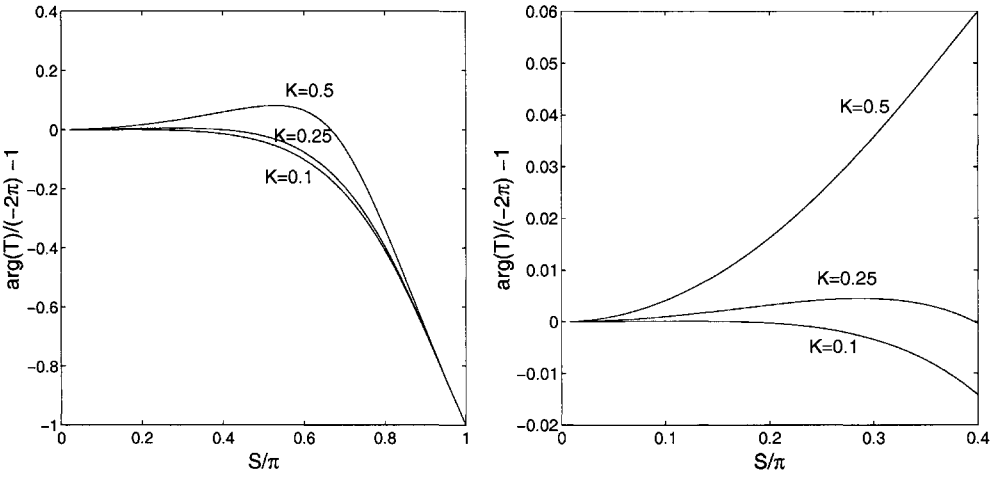


Figure 12.21: Normalized phase error, $\frac{arg(T)}{-2\pi} - 1$, for the leapfrog advection equation. T is based on β^+, equation 12.176. K is the advective Courant number, $V\Delta t/h$.

12.8 Advective-Diffusive Equation

Next we add a diffusion term to equation 12.158:

$$\frac{\partial u}{\partial t} + V\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2} = 0 \tag{12.183}$$

Assuming solutions of the form $u(x,t) = U_\sigma e^{\alpha t} e^{j\sigma x}$, the analytic dispersion relation is

$$\alpha = -j\sigma V - \sigma^2 D \tag{12.184}$$

and solutions of the form $U_\sigma e^{-\sigma^2 Dt} e^{j\sigma(x-Vt)}$ result. We have wave propagation at uniform speed V as in pure advection, accompanied by exponential decay at the rate $-\sigma^2 D$ as in pure diffusion. The ratio β of $u(x, t+\Delta t)$ to $u(x,t)$ is

$$\beta_a = e^{\alpha \Delta t} = e^{-\sigma^2 D \Delta t} e^{-j\sigma V \Delta t} = e^{-rS^2} e^{-jSK} \tag{12.185}$$

with $S \equiv \sigma h$ the dimensionless wavenumber, $K \equiv \frac{V\Delta t}{h}$ the advective Courant number as in the advection discussion, and $r \equiv \frac{D\Delta t}{h^2}$ the dimensionless timestep as in the diffusion discussion. It is useful to introduce a Peclet number,

$$P_e \equiv \frac{Vh}{D} = \frac{K}{r} \tag{12.186}$$

which is a measure of the relative strength of advection to diffusion. An equivalent expression for β_a is

$$\beta_a = e^{-rS^2} e^{-jrSP_e} \tag{12.187}$$

Euler

For Euler time-stepping we have the explicit time-discrete equation

$$\frac{u^{k+1} - u^k}{\Delta t} + \left[V \frac{\partial u}{\partial x} - D \frac{\partial^2 u}{\partial x^2} \right]^k = 0 \tag{12.188}$$

The Galerkin/Fourier version is

$$A\left(\frac{\beta - 1}{\Delta t}\right) + \left[Vj\sigma B + D\sigma^2 C^2 \right] = 0 \tag{12.189}$$

Solving for β, we have

$$\beta = 1 - \left[jK \frac{SB}{A} + r \frac{S^2 C^2}{A} \right] \tag{12.190}$$

or, with $K = rP_e$,

$$\beta = 1 - r \left[jP_e \frac{SB}{A} + \frac{S^2 C^2}{A} \right] \tag{12.191}$$

We know from the steady-state form of this equation (section 12.1) that $P_e > 2$ produces wrinkled steady solutions. We also know from studies of the pure diffusion equation (section 12.4) that $r > 1/6$ will be unstable for the shortest waves $S = \pi$, and this will govern here since the Fourier factor B representing the first derivative vanishes at $S = \pi$. Finally, it is evident that the imaginary part of β will exceed unity when $K\frac{SB}{A} > 1$, and so this clearly bounds the region of stability. The worst case here is at the maximum of $\frac{SB}{A}$, which occurs at $S = 2\pi/3$, *i.e.* $\lambda = 3h$, and this leads to the constraint $K < 1/\sqrt{3}$. So we restrict our analysis to

$$P_e < 2 \qquad r < 1/6 \qquad K < 1/\sqrt{3} \tag{12.192}$$

Since $K = rP_e$, the first two bounds imply the third. These are approximate guidelines.

From (12.191) we have the magnitude of β:

$$|\beta|^2 = \left[1 - r \frac{S^2 C^2}{A} \right]^2 + \left[rP_e \frac{SB}{A} \right]^2 \tag{12.193}$$

and from the triangle inequality, we have

$$|\beta| \leq \left|1 - r\frac{S^2C^2}{A}\right| + \left|rP_e\frac{SB}{A}\right| \tag{12.194}$$

The approximate rules obtained above can be seen to be operative here – the restriction on r keeps the first term below unity; the restriction on $K = rPe$ keeps the second term below unity.

2-Level Implicit

A more general 2-level scheme is

$$\frac{u^{k+1} - u^k}{\Delta t} + \theta\left[V\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2}\right]^{k+1} + (1-\theta)\left[V\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2}\right]^{k} = 0 \tag{12.195}$$

The Galerkin/Fourier version is

$$A\left(\frac{\beta - 1}{\Delta t}\right) + \left[Vj\sigma B + D\sigma^2C^2\right]\left[\theta\beta + (1-\theta)\right] = 0 \tag{12.196}$$

and solving for β we obtain

$$\beta = \frac{1 - r(1-\theta)\left[jP_e\frac{SB}{A} + \frac{S^2C^2}{A}\right]}{1 + r\theta\left[jP_e\frac{SB}{A} + \frac{S^2C^2}{A}\right]} \tag{12.197}$$

This scheme is unconditionally stable for $\theta \geq 1/2$. In Figure 12.22 we display the zone of stability in (r, Pe) space, for various values of $\theta < 1/2$. This plot was obtained by fixing r and θ and determining by direct evaluation the maximum stable Pe for which all S are stable. Generally, there is a decrease in allowable Pe with r, and an abrupt transition to unconditional stability at the pure diffusion limit $r = \frac{1}{6(1-2\theta)}$. The case $\theta = 0$ confirms the approximate rules arrived at above. If one stays within the $Pe \leq 2$ guideline suggested above, it is evident that the diffusion term alone governs stability.

The right panel in Figure 12.22 is the same function expressed in (r, K) space. It is useful to keep in mind that both r and K are proportional to Δt; otherwise it is possible to misinterpret the decrease in allowable K at low r.

As a measure of accuracy, we look at the propagation factor T

$$T = \left(\frac{\beta}{\beta_a}\right)^N \tag{12.198}$$

with N defined as in the pure advection case to be the number of time steps needed to propagate one wavelength, analytically:

$$N = \frac{2\pi}{SK} \tag{12.199}$$

With this definition, we have

$$T = \left(\frac{\beta}{e^{-rS^2}}\right)^N \tag{12.200}$$

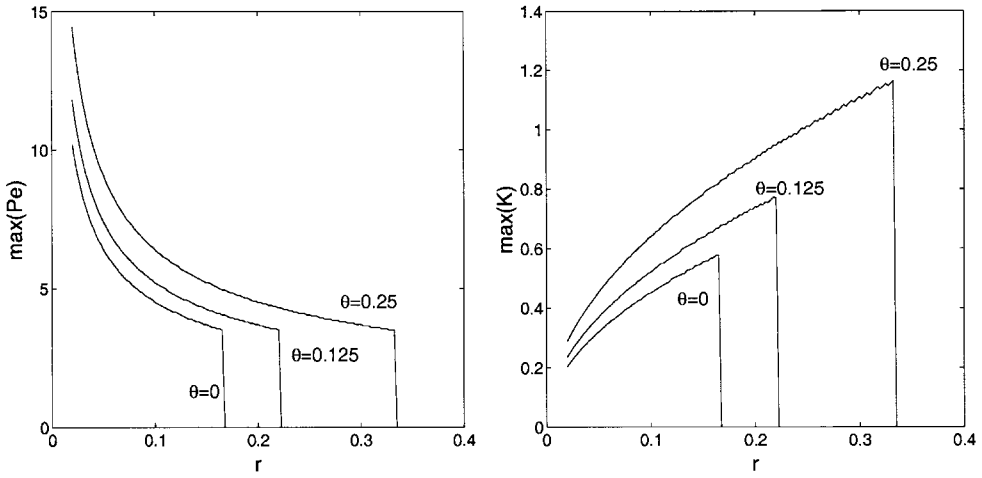


Figure 12.22: Stability envelopes for the 2-level advective diffusive equation. Left: maximum stable Pe is plotted as a function of r, for three values of θ. Combinations of (r, Pe) below/to left of the curve are stable at the indicated value of θ. Right: the same function displayed in (r, K) space ($K = rPe$).

and measures of amplitude and phase are

$$|T| = \left| \frac{\beta}{e^{-rS^2}} \right|^N \qquad arg(T) = N \cdot arg(\beta) \tag{12.201}$$

For a perfect scheme, $|T| = 1$ and $arg(T) = 2\pi$. Figures 12.23 and 12.24 display the explicit case $\theta = 0$; Figures 12.25 and 12.26 display the centered implicit case $\theta = 0.5$. At $r = .1$, the phase error is improved in the implicit scheme by a factor of 10, reflecting the second-order temporal truncation error for $\theta = .5$.

A more appropriate evaluation for the implicit scheme would involve a larger timestep r. Figure 12.27 shows the effect of increasing both r and K by a factor of 10.

It is interesting to put these last calculations in perspective, since the errors in $|T|$ are severe. For $r = 1$, $K = 2.5$, we have $Pe = 2.5$, in the range of qualitatively poor steady state solutions. At $S = .4\pi$, $\lambda = 5h$ and $N = 2$, *i.e.* it takes only 2 timesteps to propagate one wavelength. This is very coarse temporal and spatial resolution, and while the scheme is stable, its accuracy is poor.

Leapfrog

The leapfrog Advective-Diffusive equation is

$$\frac{u^{k+1} - u^{k-1}}{2\Delta t} + \left[V \frac{\partial u}{\partial x} - D \frac{\partial^2 u}{\partial x^2} \right]^k = 0 \tag{12.202}$$

The Galerkin/Fourier version is

$$A \left(\frac{\beta^2 - 1}{2\Delta t} \right) + \left[V j \sigma B + D \sigma^2 C^2 \right] \beta = 0 \tag{12.203}$$

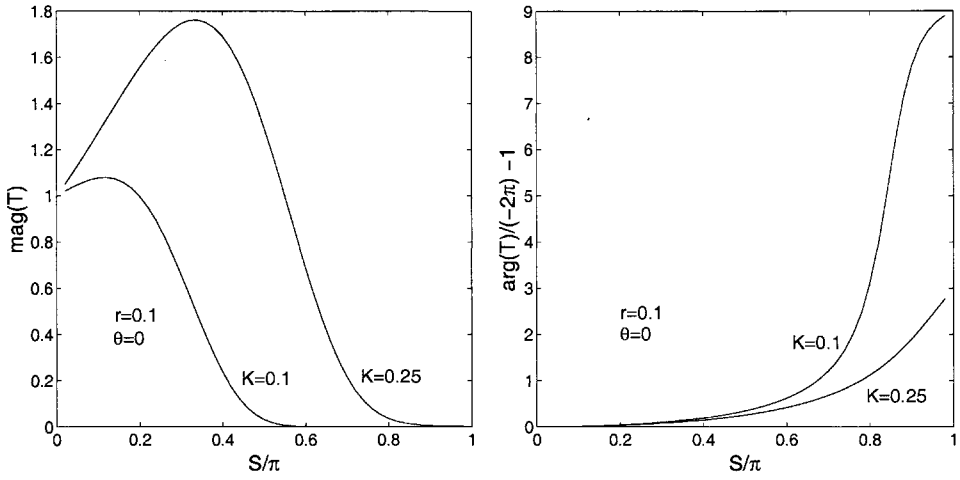


Figure 12.23: Propagation Factor accuracy for the 2-level explicit advective diffusive equation, $\theta = 0$. Left: $|T|$; perfection is unity. Right: normalized phase error.

$$\beta^2 + 2\beta\left[jKS\frac{B}{A} + rS^2\frac{C^2}{A}\right] - 1 = 0 \tag{12.204}$$

This scheme is unconditionally unstable. At $S = \pi$, $B = 0$ and the advection term is null, leaving only diffusion dynamics. We know from section 12.4 that leapfrog diffusion is unconditionally unstable for this mode. That instability governs the advective-diffusive equation also.

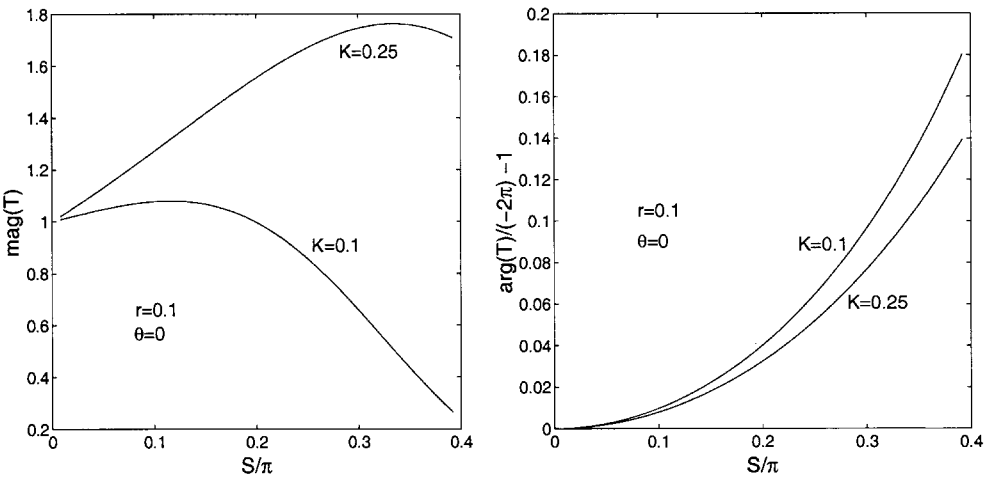


Figure 12.24: Propagation Factor accuracy for the 2-level explicit advective diffusive equation, $\theta = 0$. Same as Figure 12.23 except the well-resolved range of S is displayed.

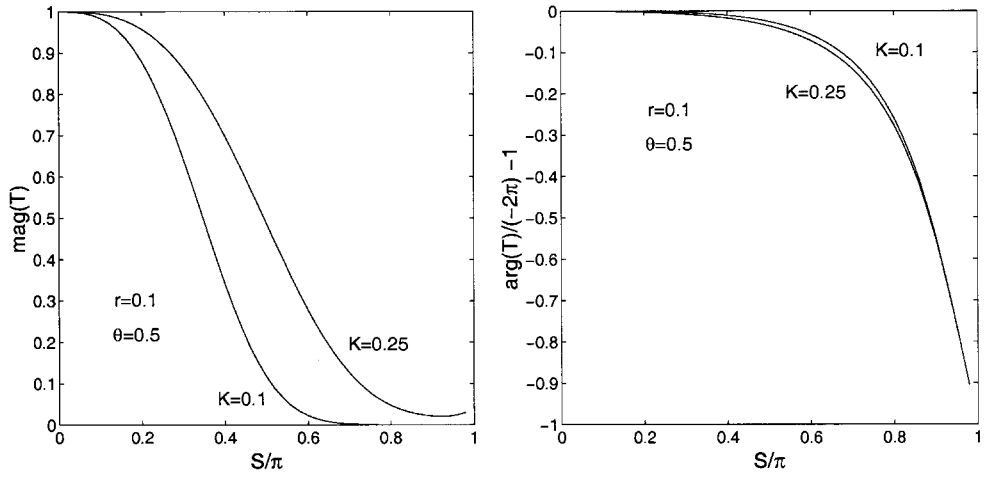


Figure 12.25: Propagation Factor accuracy for the 2-level implicit advective diffusive equation, $\theta = 0.5$. Left: $|T|$; perfection is unity. Right: normalized phase error.

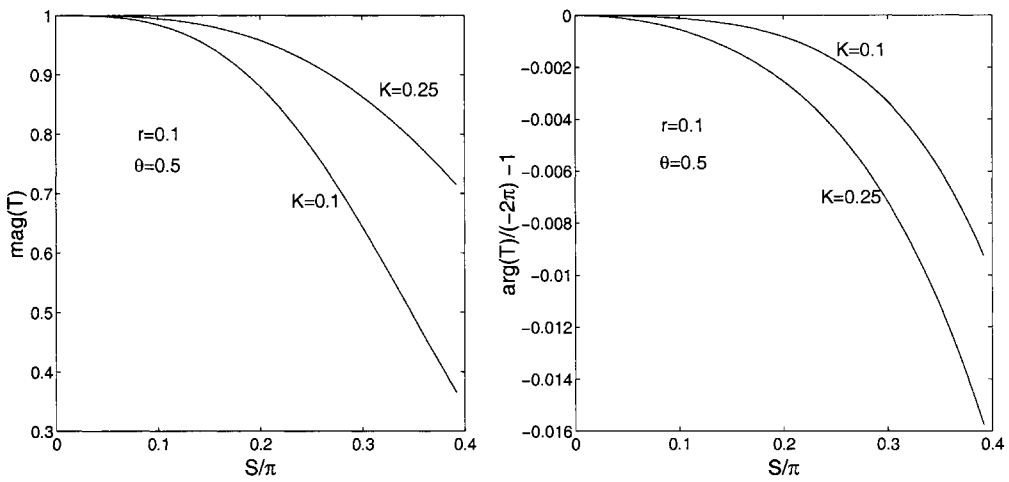


Figure 12.26: Propagation Factor accuracy for the 2-level implicit advective diffusive equation, $\theta = 0.5$. Same as Figure 12.25 except the well-resolved range of S is displayed.

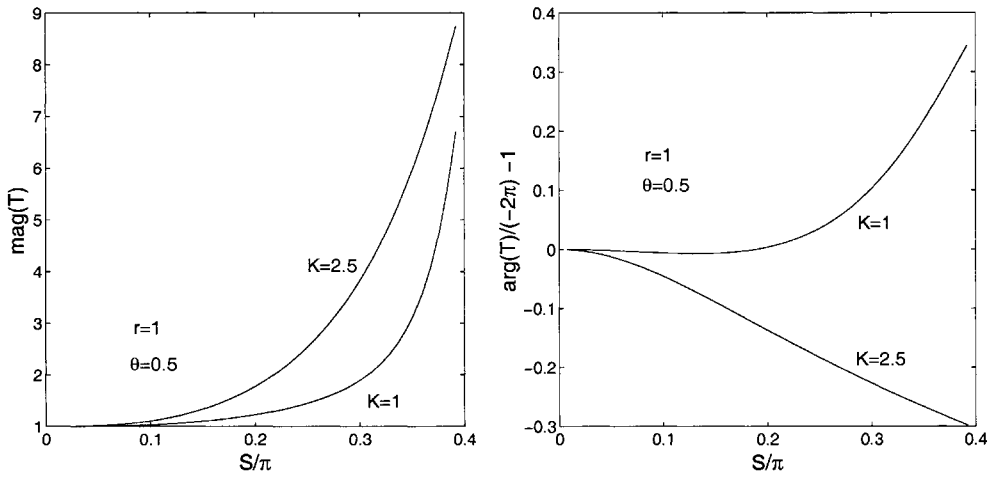


Figure 12.27: Propagation Factor for centered implicit scheme. Same as Figure 12.26 but with Δt increased by a factor of 10.

Part III

Inverse Methods

Chapter 13

Inverse Noise, SVD, and Linear Least Squares

So far we have concentrated on conventional problem formulations: a PDE is defined by natural principles, the necessary and sufficient boundary and initial conditions are given from either data or hypothesis, and there is a unique solution. The job has been to define discrete approximations which have analogous properties – *i.e.* there is a set of necessary and sufficient information (ICs, BCs, forcing, parameters) which, once specified, guarantees a computable and unique solution. If the numerical method is good, that discrete solution will mimic the true solution in calculus and, by extension, in nature. Numerical Analysis reveals the degree of agreement.

Many important problems can be successfully and precisely posed in this way. But in fact, practical obstacles frequently prevent the assembly of the data required for a unique solution. Some data may be missing altogether; while the rest is certainly known only imperfectly. Further, we may be given some of the *answer*, *i.e.* observations of the field quantity which is to be known; we then expect a numerical solution to reproduce the observations. Of course the observations are incomplete (why else would we be resorting to computation?) and are certainly imprecise at some level. So we are faced with the problem of *estimating* the imperfectly-known but necessary BC's and forcing, and thence *estimating* the unknown field which is itself partially but imprecisely known. This is the sense of mathematical inversion – the need to work a problem *backwards*, with the objective of estimating both inputs and outputs. The extreme of this situation is the case where abundant data describe the response or system output, and the inputs – BC's, IC's, forcing, parameters – are the basic subject of investigation.

In this section we explore some of the basic ideas which are useful in inversion. We presume the *forward* problem is a well-posed FEM statement based on a differential equation, with well-known requirements for the necessary/sufficient data support. Our emphasis will be on formulations and algorithms which exploit properties of such systems: well-conditioned problems with square, nonsingular matrices which can be assumed to be sparse and/or banded.

13.1 Matrix Inversion and Inverse Noise

For starters, we have been solving the matrix inversion problem all along:[1]

$$[K]\{u\} = \{b\} \qquad \Leftrightarrow \qquad \{u\} = \left[K^{-1}\right]\{b\} \tag{13.1}$$

Both of these have been treated as equivalent so far – essentially, we assume that $[K^{-1}]$ exists, can be computed, and that multiplication by either $[K]$ or $[K^{-1}]$ produces orderly results. When $[K]$ is obtained by a valid FD or FE method, with proper forcing and BC enforcement, then that is a consequence. By definition, $[K]$ is square, and of full rank (all of its rows are linearly independent). Furthermore, $[K^{-1}]$ will generally be a full matrix even though $[K]$ is commonly sparse and banded. This is a consequence of the fact that each result u_j is influenced by each of the forcings b_i.

The right-hand side $\{b\}$ in all such cases represents the sum of boundary condition values and forcing on the interior of the domain. What if $\{b\}$ is known only imprecisely?

Mean and Variability.

First, distinguish between mean and actual values of $\{b\}$:

$$\{b\} = \{\overline{b}\} + \{\tilde{b}\} \tag{13.2}$$

with $\{\overline{b}\}$ the "mean" or "true" value of $\{b\}$, which is assumed known; and $\{\tilde{b}\}$ a perturbation, representing an unknown part due to measurement noise, uncertainty in a prediction context, or the variation among an ensemble of similar right-hand sides. We refer to $\{\tilde{b}\}$ as the system *Noise*. A similar decomposition of $\{u\}$ is useful:

$$\{u\} = \{\overline{u}\} + \{\tilde{u}\} \tag{13.3}$$

and we have the twin relations

$$[K]\{\overline{u}\} = \{\overline{b}\} \qquad \Leftrightarrow \qquad \{\overline{u}\} = \left[K^{-1}\right]\{\overline{b}\} \tag{13.4}$$

$$[K]\{\tilde{u}\} = \{\tilde{b}\} \qquad \Leftrightarrow \qquad \{\tilde{u}\} = \left[K^{-1}\right]\{\tilde{b}\} \tag{13.5}$$

The mean response $\{\overline{u}\}$ is computable from $\{\overline{b}\}$; $\{\tilde{u}\}$ is the *Inverse Noise* and is linear in the noise $\{\tilde{b}\}$. Note that if the overbar indicates an ensemble average, then as a matter of definition,

$$\overline{\{\tilde{u}\}} = 0 \qquad \textit{and} \qquad \overline{\{\tilde{b}\}} = 0 \tag{13.6}$$

Covariance.

The statistics of the inverse noise are of fundamental importance. By definition the mean noise is zero. What about its covariance – *i.e.* the expected value of $\tilde{u}_i^2$ and of all the various products $\tilde{u}_i\tilde{u}_j$? These are all the entries in the matrix $\{\tilde{u}\}\{\tilde{u}\}^T$. Define the covariance matrix as

$$[Cov(\tilde{u})] \equiv \overline{\{\tilde{u}\}\{\tilde{u}\}^T} \tag{13.7}$$

[1] Throughout we assume $\{b\}$, $\{x\}$, etc. are *column* vectors; their transposes $\{b\}^T$, $\{x\}^T$ are *row* vectors.

For any realization of $\{\tilde{b}\}$ we have

$$\{\tilde{u}\} = \left[K^{-1}\right]\{\tilde{b}\} \tag{13.8}$$

and therefore[2]

$$\{\tilde{u}\}\{\tilde{u}\}^T = \left[K^{-1}\right]\{\tilde{b}\}\{\tilde{b}\}^T\left[K^{-T}\right] \tag{13.9}$$

It follows that

$$\overline{\{\tilde{u}\}\{\tilde{u}\}^T} = \left[K^{-1}\right]\overline{\{\tilde{b}\}\{\tilde{b}\}^T}\left[K^{-T}\right] \tag{13.10}$$

i.e. ,

$$[Cov(\tilde{u})] = \left[K^{-1}\right]\left[Cov(\tilde{b})\right]\left[K^{-T}\right] \tag{13.11}$$

In the simple case of uncorrelated noise, we have

$$\left[Cov(\tilde{b})\right] = \sigma^2[I] \tag{13.12}$$

with σ^2 the noise variance. This produces inverse noise

$$[Cov(\tilde{u})] = \sigma^2\left[K^{-1}\right]\left[K^{-T}\right] \tag{13.13}$$

We see in this case that $\left[K^{-1}\right]\left[K^{-T}\right]$ is a *noise filter* - either a suppressor or an amplifier. An undesirable situation would have the latter property.

The RHS vector $\{\tilde{b}\}$ represents the system forcing – for PDE's, it includes the sum of boundary condition forcing and interior forcing:

$$\{\tilde{b}\} = \{\tilde{b}\}_D + \{\tilde{b}\}_N + \{\tilde{b}\}_I \tag{13.14}$$

with subscripts D, N, I indicating Dirichlet and Neumann boundary condition variability, and Interior forcing variability. Each realization of the inverse noise is additive:

$$\begin{aligned} \{\tilde{u}\} &= \left[K^{-1}\right]\{\tilde{b}\}_D + \left[K^{-1}\right]\{\tilde{b}\}_N + \left[K^{-1}\right]\{\tilde{b}\}_I && (13.15)\\ &= \{\tilde{u}\}_D + \{\tilde{u}\}_N + \{\tilde{u}\}_I && (13.16) \end{aligned}$$

The variability of $\{\tilde{b}\}$ is

$$\{\tilde{b}\}\{\tilde{b}\}^T = \left\{\{\tilde{b}\}_D + \{\tilde{b}\}_N + \{\tilde{b}\}_I\right\}\left\{\{\tilde{b}\}_D + \{\tilde{b}\}_N + \{\tilde{b}\}_I\right\}^T \tag{13.17}$$

If the 3 sources of variability are statistically independent, then the noise covariance is additive:

$$\left[Cov(\tilde{b})\right] = \left[Cov(\tilde{b})\right]_D + \left[Cov(\tilde{b})\right]_N + \left[Cov(\tilde{b})\right]_I \tag{13.18}$$

as is the inverse noise covariance:

$$\begin{aligned} [Cov(\tilde{u})] &= \left[K^{-1}\right]\left[Cov(\tilde{b})\right]\left[K^{-T}\right] && (13.19)\\ &= [Cov(\tilde{u})]_D + [Cov(\tilde{u})]_N + [Cov(\tilde{u})]_I && (13.20) \end{aligned}$$

Otherwise, the three sources are correlated and we need additional terms in the covariance among them; the separation of the variability becomes less useful.

[2] $\left[K^{-T}\right]$ indicates the inverse transpose. Recall that inversion and transposition are commutative.

Variance.

The diagonals of the covariance matrix are the variances, *i.e.* the expected squared values, of the individual members of the vector involved. The sum of the diagonals of a matrix is its *Trace*; so the Trace of $[Cov(\tilde{u})]$ is a convenient scalar measure of the variability of $\{u\}$. It is otherwise expressed as $\overline{\{\tilde{u}\}^T\{\tilde{u}\}}$, or $\sum_i \overline{\tilde{u}_i^2}$. Divided by N, it is the mean squared size of $\{\tilde{u}\}$. We define variance of a vector $\tilde{u}$ as the trace of its covariance:

$$Var(\tilde{u}) = \overline{\{\tilde{u}\}^T\{\tilde{u}\}} = Tr\,[Cov(\tilde{u})] = \sum_i \overline{\tilde{u}_i^2} \tag{13.21}$$

This scalar metric is the simple addition of the variances of all the individual $\tilde{u}_i$, without regard for their covariance. The observations above about additive noise covariance pertain here as well.

For the inversion considered here,

$$[K]\{\tilde{u}\} = \{\tilde{b}\} \qquad \Leftrightarrow \qquad \{\tilde{u}\} = \left[K^{-1}\right]\{\tilde{b}\} \tag{13.22}$$

we have

$$Var(\tilde{u}) = \overline{\{\tilde{b}\}^T\,[K^{-T}]\,[K^{-1}]\,\{\tilde{b}\}} \tag{13.23}$$

Noise Models.

It is useful to pose analytic models of input variability, or "noise models". An example is the distance-based form

$$\overline{\tilde{b}_i\tilde{b}_j} = \sigma^2(1 + r_{ij}/l)e^{-r_{ij}/l} \tag{13.24}$$

with r_{ij} the separation distance between locations (nodes) i and j, σ the scale of the variability, and l a correlation length scale. Many such forms have been posed as models of noise generated by various processes, using combinations of exponential decay, polynomial dependence, and periodicity. This two-parameter model has zero gradient, maximum value σ^2 at $r = 0$, and monotonic decay with r. (See Figure 13.1.) At a separation of one correlation length, this covariance decays to about 75% of its peak value; at two correlation lengths, 40% of peak value; at four lengths, 10%.

Covariance matrices formed from these functions will be symmetric as required, with a strong diagonal structure. They will be full matrices, with some very small numbers, unless something is done to eliminate the tiny covariance among very distant points. The inverse noise

$$[Cov(\tilde{u})] = \left[K^{-1}\right]\left[Cov(\tilde{b})\right]\left[K^{-T}\right] \tag{13.25}$$

will be a full matrix anyway, as $\left[K^{-1}\right]$ is a full matrix as assumed herein. The diagonals of $[Cov(\tilde{u})]$ are the expected or mean values of $\tilde{u}^2$. They represent limits within which nodal values of u can be known, due to similar limits in the specification of the RHS. A map of these diagonals plotted as contours of nodal values on a finite element mesh constitutes a map of inverse noise or *imprecision.* (More later on this.)

Example: Suppose we have a system where node 2 contains a point source of strength $1 \pm .5$; node 3 is on a Neumann boundary with $\oint \frac{\partial u}{\partial n}\phi_i ds = .5 \pm .3$; and nodes 4, 5, and 6 are uniformly-spaced Dirichlet boundary nodes with values $.6 \pm .2$. The Dirichlet data is correlated with length

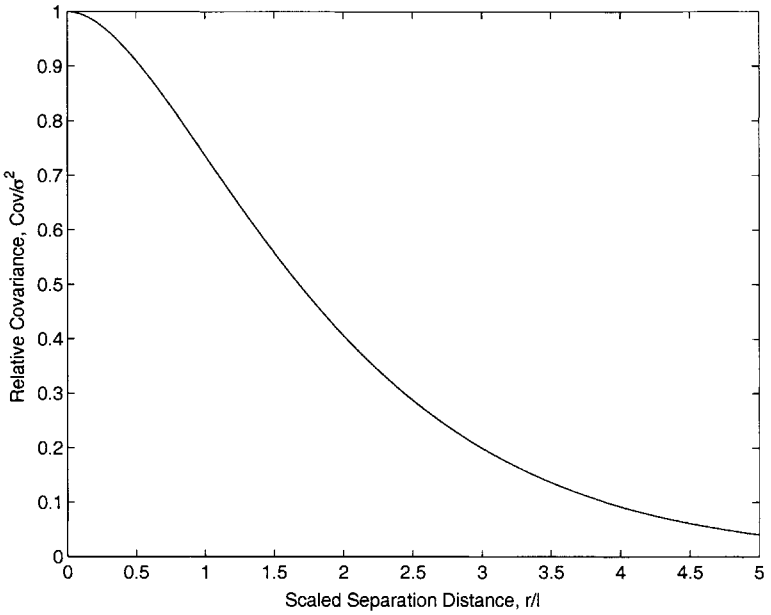


Figure 13.1: Distance-based covariance of the form $\sigma^2(1 + r/l)e^{-r/l}$, (equation 13.24).

scale = 1 element size and the covariance model of equation 13.24. The rest of the forcing is completely independent. The right hand sides are

$$\left\{\bar{b}\right\}_I = \begin{Bmatrix} 0 \\ 1.0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{Bmatrix} \quad \{\bar{b}\}_N = \begin{Bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{Bmatrix} \quad \{\bar{b}\}_D = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0.6 \\ 0.6 \\ 0.6 \\ 0 \\ \vdots \end{Bmatrix} \tag{13.26}$$

and the covariances are

$$\left[Cov(\tilde{b})\right]_I = (0.5)^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{13.27}$$

$$\left[Cov(\tilde{b})\right]_N = (0.3)^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{13.28}$$

$$\left[Cov(\tilde{b})\right]_D = (0.2)^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & a & b & 0 & \cdots \\ 0 & 0 & 0 & a & 1 & a & 0 & \cdots \\ 0 & 0 & 0 & b & a & 1 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{13.29}$$

with $a = 2e^{-1}$ and $b = 3e^{-2}$.

EigenTheory

Returning to the question of whether matrix inversion is a noise amplifier: consider the special case where $[K]$ is real, square $(N \times N)$, and symmetric. Then there exist N independent vectors $\{V\}_i$, each of which solves the *eigenvalue problem*:

$$[K]\{V\}_i = \lambda_i \{V\}_i \tag{13.30}$$

with a real eigenvalue λ_i accompanying each eigenvector $\{V\}_i$. The $\{V\}_i$ are orthogonal – $\{V\}_i^T\{V\}_j = 0$ for all $i \neq j$; and we can easily normalize them so that $\{V\}_i^T\{V\}_i = 1$ for all i. The collection of λ_i is the *spectrum* of $[K]$.

Let the square matrix $[V]$ have columns $\{V\}_j$. The basic eigenvector relation is

$$[K][V] = [V][diag(\lambda)] \tag{13.31}$$

Since $[V]$ is orthonormal, we have

$$[V]^T[V] = [I] \qquad [V][V]^T = [I] \tag{13.32}$$

so that postmultiplying 13.31 by $[V]^T$ we have the *eigenvalue decomposition* of $[K]$:

$$[K] = [V][diag(\lambda)][V]^T \tag{13.33}$$

If $[K]$ is square and symmetric, the above decomposition exists and all $\lambda_i \geq 0$. If $[K]$ is singular, then one or more of the λ_i vanishes. Otherwise, we have a unique inverse

$$[K]^{-1} = [V]\left[diag\left(\frac{1}{\lambda}\right)\right][V]^T \tag{13.34}$$

and for $[K]\{u\} = \{b\}$ we have

$$\{u\} = [V]\left[diag\left(\frac{1}{\lambda}\right)\right][V]^T\{b\} = \sum_i \{V\}_i \frac{\{V\}_i \cdot \{b\}}{\lambda_i} \tag{13.35}$$

Now the vectors $\{V\}_i$ comprise a complete basis for any N-dimensional vector. And, they are the *natural* basis for $\{u\}$ and $\{b\}$:

$$\{u\} = [V]\{c\} \qquad \Leftrightarrow \qquad \{c\} = [V]^T\{u\} \tag{13.36}$$

$$\{b\} = [V]\{d\} \qquad \Leftrightarrow \qquad \{d\} = [V]^T\{b\} \tag{13.37}$$

Here, $\{c\}$ is the projection of $\{u\}$ onto V-space; $\{d\}$ is the projection of $\{b\}$.[3] Premultiplying 13.35 by $[V]^T$, and substituting 13.36 and 13.37, we have

$$\{c\} = [diag(\lambda)]^{-1}\{d\} \tag{13.40}$$

or for all i,

$$c_i = \frac{1}{\lambda_i} d_i \tag{13.41}$$

Effectively: in V space, $\frac{1}{\lambda}$ achieves inversion of $[K]$. If any λ_i is zero, then the inversion is undefined unless we are lucky and $\{b\}$ happens to be orthogonal to $\{V\}_i$.[4] If λ_i is very small, then noise projected onto $\{V\}_i$ is amplified by $\frac{1}{\lambda_i}$. Viewed in the original coordinate system (equation 13.35), inversion goes like this: $\{b\}$ is projected onto V-space; multiplied by $\frac{1}{\lambda}$; and the results projected back into $\{u\}$.

So here is a criterion for noise amplification: small eigenvalues of the system matrix $[K]$. If some λ's are small, the inversion will amplify components of $\{b\}$ selectively – specifically, that part of $\{b\}$ which projects onto (is parallel to) the associated $\{V\}_i$.

The *Condition Number* κ is defined as the ratio of the largest to the smallest λ_i, in absolute value. If κ is of order 1, then noise in $\{b\}$ will be passed into $\{u\}$ relatively undistorted. If $\kappa >> 1$, then the noise is filtered and selectively amplified. This situation is a criterion for near-singularity, *i.e.* as one or more $\lambda \to 0$, and $\kappa \to \infty$.

The covariance of $\{u\}$ and $\{c\}$ are

$$[Cov(u)] = [V]\left[diag\left(\frac{1}{\lambda}\right)\right][V]^T[Cov(b)][V]\left[diag\left(\frac{1}{\lambda}\right)\right][V]^T \tag{13.42}$$

$$[Cov(c)] = \left[diag\left(\frac{1}{\lambda}\right)\right][Cov(d)]\left[diag\left(\frac{1}{\lambda}\right)\right] \tag{13.43}$$

[3] Equivalently, in terms of the individual vectors $\{V\}_i$, we have

$$\{u\} = \sum_i c_i\{V\}_i \qquad \Leftrightarrow \qquad c_i = \{V\}_i^T\{u\} \tag{13.38}$$

$$\{b\} = \sum_i d_i\{V\}_i \qquad \Leftrightarrow \qquad d_i = \{V\}_i^T\{b\} \tag{13.39}$$

[4] In that case, there are multiple solutions.

If $[Cov(b)] = \sigma^2 [I]$, then $[Cov(d)] = \sigma^2 [I]$ and

$$[Cov(u)] = \sigma^2 [V] \left[diag \left(\frac{1}{\lambda} \right)^2 \right] [V]^T \tag{13.44}$$

$$[Cov(c)] = \sigma^2 \left[diag \left(\frac{1}{\lambda} \right)^2 \right] \tag{13.45}$$

with individual entries

$$\overline{u_i u_j} = \sigma^2 \sum_k \frac{V_{ik} V_{jk}}{\lambda_k^2} \tag{13.46}$$

As described above, the diagonals of the covariance matrices are the variances, *i.e.* the expected squared values, of the individual members of the vector involved. The sum of the diagonals, *i.e.* the *Trace* of $[Cov(u)]$, is a convenient scalar measure of variability. It is most simply expressed as $Var(u) = \sum_i \overline{u_i^2}$. Divided by N, it is the mean squared size of $\{u\}$. Some useful relations are

$$Var(u) = Var(c) \quad ; \quad Var(b) = Var(d) \tag{13.47}$$

$$Var(u) = \sum_i \overline{\left(\frac{d_i}{\lambda_i} \right)^2} \tag{13.48}$$

$$Var(u) \geq \max_i \overline{\left(\frac{d_i}{\lambda_i} \right)^2} \tag{13.49}$$

$$Var(u) \leq \sum_i \overline{d_i^2} \sum_j \frac{1}{\lambda_j^2} = Var(b) \sum_j \frac{1}{\lambda_j^2} \tag{13.50}$$

From 13.49 we see that only one bad (small) λ can spoil the whole inversion, by ruining its precision. The only (unlikely) exception would be if the associated d_i is *guaranteed* to be comparably small. Of course that puts restrictions on the allowable right-hand sides $\{b\}$.

If as above we have the simple situation $[Cov(b)] = \sigma^2 [I]$, then the individual variances $Var(b_i) = Var(d_i) = \sigma^2$ and we have

$$Var(u) = \sigma^2 \sum_k \frac{1}{\lambda_k^2} \tag{13.51}$$

$$Var(u_i) = \sigma^2 \sum_k \left(\frac{V_{ik}}{\lambda_k} \right)^2 \tag{13.52}$$

The above theory applies only when $[K]$ is square and symmetric. Fortunately, its generalization to the nonsymmetric and nonsquare case is available – the Singular Value Decomposition. This is reviewed in the next section.

13.2 The Singular Value Decomposition

Now for the general case: the matrix equation

$$[K] \{u\} = \{b\} \tag{13.53}$$

with $[K]$ nonsquare, dimension $m \times n$ matrix[5]. We will assume that and $m \geq n$. The simple inverse $[K]^{-1}$ is undefined here, as is a unique solution in general. But it is meaningful to ask in what sense we might find $\{u\}$ which satisfies 13.53 in some sense, as a linear operation on $\{b\}$:

$$\{u\} = [K]^{-1}\{b\} \tag{13.54}$$

Essentially, we are looking for a definition of the inverse of a general nonsquare matrix.

SVD Basics

Any matrix $[K]$ may be factored as

$$[K] = [U][diag(\omega)][V]^T \tag{13.55}$$

and the following properties pertain:

- $[K]$ and $[U]$ dimensioned $m \times n$
- $[diag(\omega)]$ and $[V]$ dimensioned $n \times n$
- Columns of $[U]$ are orthogonal, $[U]^T[U] = [I]$
- Columns of $[V]$ (rows of $[V]^T$) are orthogonal, $[V]^T[V] = [I]$
- Since $[V]$ is square, its rows are also orthogonal: $[V][V]^T = [I]$
- If $m = n$ then $[U]$ is square and its rows are also orthogonal: $[U][U]^T = [I]$
- All $\omega_i \geq 0$

Because the columns of $[U]$ and $[V]$, $\{U\}_i$ and $\{V\}_i$, are of prime importance, it is useful to write the SVD 13.55 in greater detail as

$$[K] = \begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \uparrow & \uparrow & \uparrow & & \uparrow \\ U_1 & U_2 & U_3 & \cdots & U_n \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \omega_1 & 0 & 0 & \cdots & 0 \\ 0 & \omega_2 & 0 & \cdots & 0 \\ 0 & 0 & \omega_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \omega_n \end{bmatrix} \begin{bmatrix} \cdots & \leftarrow & V_1 & \rightarrow & \cdots \\ \cdots & \leftarrow & V_2 & \rightarrow & \cdots \\ \cdots & \leftarrow & V_3 & \rightarrow & \cdots \\ & & \vdots & & \\ \cdots & \leftarrow & V_n & \rightarrow & \cdots \end{bmatrix} \tag{13.56}$$

(The layout is meant to reinforce the condition that $m \geq n$.) $\{U\}_i$ comprise the Left Singular Vectors of $[K]$; $\{V\}_i$ are its Right Singular Vectors; ω_i are its Singular Values. The vectors $\{V\}_i$ are a complete, orthogonal basis for any n-dimensional vector. In the square case ($m = n$), the set of $\{U\}_i$ is a distinct but still complete basis in n-space. It is conventional to order the ω in order by size, beginning with the largest. The same reordering is required of the Singular Vectors. The condition number κ of the matrix $[K]$ is defined as the ratio of the largest to the smallest Singular Value: $\kappa \equiv \omega_1/\omega_n$.

From equation 13.55 and the orthogonality of $[V]$, we have

$$[K][V] = [U][diag(\omega)] \tag{13.57}$$

or, for each i,

$$[K]\{V\}_i = \omega_i \{U\}_i \tag{13.58}$$

Essentially, $[K]$ maps $\{V\}_i$ onto $\{U\}_i$ with scaling ω_i.

[5] m rows, n columns

The Square, Nonsingular Case

In the square case, $m = n$, we have the inverse

$$[K]^{-1} = [V]\left[diag(\frac{1}{\omega})\right][U]^T \tag{13.59}$$

which is defined *if all of the* $\omega_i > 0$. Otherwise, $[K]$ is singular and its inverse is undefined. For the nonsingular case,

$$[K]\{u\} = \{b\} \qquad \Leftrightarrow \qquad \{u\} = [V]\left[diag(\frac{1}{\omega})\right][U]^T\{b\} \tag{13.60}$$

and we see that

- the columns of $[V]$ constitute a natural basis for $\{u\}$. The projection of $\{u\}$ onto $[V]$-space is $\{c\}$:

$$\{c\} = [V]^T\{u\} \qquad \Leftrightarrow \qquad [V]\{c\} = \{u\} \tag{13.61}$$

- the columns of $[U]$ are the natural basis for $\{b\}$. The projection of $\{b\}$ onto $[U]$-space is $\{d\}$:

$$\{d\} = [U]^T\{b\} \qquad \Leftrightarrow \qquad [U]\{d\} = \{b\} \tag{13.62}$$

  (the right half of this pair *requires* that $m = n$).

- inversion comprises projecting $\{b\}$ onto $[U]$-space; modulation by the diagonal matrix $[\frac{1}{\omega}]$; and reassembly of $\{u\}$ in $[V]$-space:

$$c_i = \frac{d_i}{\omega_i} \qquad \Leftrightarrow \qquad \{u\} = \sum_i \{V\}_i \frac{\{U\}_i \cdot \{b\}}{\omega_i} \tag{13.63}$$

When $[K]$ is square and nonsingular, this is the unique solution. When $[K]$ is symmetric, $[U] = [V]$ and we recover the special eigenvalue theory described in the previous section with $\omega \to \lambda$.

Relative to noise in the $\{b\}$ vector, we have the standard result

$$[Cov(u)] = \left[K^{-1}\right][Cov(b)]\left[K^{-T}\right] \tag{13.64}$$

and from 13.59 we have

$$[Cov(u)] = [V]\left[diag(\frac{1}{\omega})\right][U]^T[Cov(b)][U]\left[diag(\frac{1}{\omega})\right]\left[V^T\right] \tag{13.65}$$

$$[Cov(c)] = \left[diag(\frac{1}{\omega})\right][Cov(d)]\left[diag(\frac{1}{\omega})\right] \tag{13.66}$$

and for the Variance,

$$Var(u) = Var(c) \quad ; \quad Var(b) = Var(d) \tag{13.67}$$

$$Var(u) = \sum_i \left(\frac{d_i}{\omega_i}\right)^2 \tag{13.68}$$

$$\max_i \overline{\left(\frac{d_i}{\omega_i}\right)^2} \leq Var(u) \leq Var(b) \sum_j \frac{1}{\omega_j^2} \tag{13.69}$$

If the forcing covariance is homogeneous and uncorrelated, *i.e.* $[Cov(b)] = \sigma^2 [I] = [Cov(d)]$, then we have

$$[Cov(u)] = \sigma^2 [V] \left[diag(\frac{1}{\omega^2})\right] \left[V^T\right] \tag{13.70}$$

$$[Cov(c)] = \sigma^2 \left[diag(\frac{1}{\omega^2})\right] \tag{13.71}$$

$$Var(u) = \sigma^2 \sum_k \frac{1}{\omega_k^2} \tag{13.72}$$

$$Var(u_i) = \sigma^2 \sum_k \left(\frac{V_{ik}}{\omega_k}\right)^2 \tag{13.73}$$

These results essentially replicate those in the preceeding (EigenTheory) section, with the understanding that $\{b\}$ is projected onto $[U]$ rather than $[V]$. The take-home message is that noise present in $\{U\}_i$ (the forcing) will show up in the answer in terms of $\{V\}_i$, amplified by $\frac{1}{\omega_i}$. Thus small ω_i constitute noise amplifiers relative to larger ones. The measure of this noise distortion is the condition number.

The Square, Singular Case

Now suppose that one of the singular values, ω_N, is exactly zero. The general SVD inversion stated in the form

$$c_i = \frac{d_i}{\omega_i} \quad \Leftrightarrow \quad \{u\} = \sum_i \{V\}_i \frac{\{U\}_i \cdot \{b\}}{\omega_i} \tag{13.74}$$

with $d_i \equiv \{U\}_i \cdot \{b\}$. It is clear that we have a problem. There are two situations to note, based on whether the forcing $\{b\}$ projects onto $\{U\}_N$ or not.

Consider first the special case where $\{b\}$ is orthogonal to $\{U\}_N$: $\{b\} \cdot \{U\}_N = 0$ (*i.e.* , $d_N = 0$). A complete and exact solution is

$$\{u\} = \sum_i^{N-1} \{V\}_i \frac{\{U\}_i \cdot \{b\}}{\omega_i} \tag{13.75}$$

that is, the final term involving $\{V\}_N$ is left off since it is not needed to satisfy $[K]\{u\} = \{b\}$. However this solution is not unique. By definition we have

$$[K]\{V\}_i = \omega_i \{U\}_i \tag{13.76}$$

for all i, and we also have $\omega_N = 0$, such that

$$[K]\{V\}_N = 0 \tag{13.77}$$

Thus we can add an arbitrary amount of $\{V\}_N$ to the answer without affecting the right hand side in the least. So we have a family of solutions with parameter α:

$$\{u\} = \sum_i^{N-1} \{V\}_i \frac{\{U\}_i \cdot \{b\}}{\omega_i} + \alpha \{V\}_N = \sum_i^{N-1} \{V\}_i \frac{d_i}{\omega_i} + \alpha \{V\}_N \tag{13.78}$$

and each of them satisfies $[K]\{u\} = \{b\}$ exactly.

We need a criterion for selecting α. The variance of $\{u\}$ can be a criterion here:

$$Var(u) = \sum_{i}^{N-1} \left(\frac{\{U\}_i \cdot \{b\}}{\omega_i} \right)^2 + \alpha^2 \tag{13.79}$$

By setting α to zero, we achieve the minimum variance, exact solution. Operationally, this amounts to setting $\frac{1}{\omega_N}$ to zero rather than infinity, thereby avoiding an inadvertant division by zero.

The other case is more likely, wherein $d_N \equiv \{U\}_i \cdot \{b\}$ is nonzero, either by definition or due to imperfect precision. We still have the same options: either include an arbitrary amount of $\{V\}_N$ in the solution, knowing that it can contribute nothing to the right-hand side, or to remove it. In either case, we have a nonzero residual in the governing equation $[K]\{u\} - \{b\} \equiv \{r\} \neq 0$, since $[K]$ cannot produce any $\{U\}_N$ which is present in $\{b\}$. Expressing u in the basis $[V]$:

$$\{u\} = \sum_{i}^{N-1} c_i \{V\}_i + \alpha \{V\}_N \tag{13.80}$$

substituting into $[K]\{u\} = \{b\} = \sum_i^N d_i \{U\}_i$, and using the fact that $[K]\{V\}_i = \omega_i \{U\}_i$, we obtain the residual

$$\{r\} = \sum_{i}^{N-1} c_i \omega_i \{U\}_i - \sum_{i}^{N} d_i \{U\}_i \tag{13.81}$$

Since $\omega_N = 0$, we have

$$\{r\} = \sum_{i}^{N-1} (c_i \omega_i - d_i) \{U\}_i - d_N \{U\}_N \tag{13.82}$$

The variance of the residual is

$$Var(r) = \sum_{i}^{N-1} (c_i \omega_i - d_i)^2 + d_N^2 \tag{13.83}$$

Clearly, the minimum variance solution is gotten by setting

$$c_i = \frac{d_i}{\omega_i} \tag{13.84}$$

and the parameter α is left undetermined as above. The choice $\alpha = 0$ adds nothing to the residual and achieves the minimum norm solution as before.

Operationally, the rule is the same, and remarkably simple. If $\omega_N = 0$, treat it as if it were infinite. The result will be the Minimum Variance solution which also achieves the Minimum Variance residual. In special cases where d_N is luckily zero, the solution will be exact. Otherwise, the residual will reside entirely in $\{U\}_N$, and nothing can be done about it. In all cases, the solution will have zero projection along the direction $\{V\}_N$.

Put in other words: if $\omega_N = 0$, keep $\{V\}_N$ out of the solution – it is undetermined – and ignore the presence of $\{U\}_N$ in the forcing – there is no way to reduce its presence in the residual.

The Square, Nearly-Singular Case

Next suppose that one of the ω_N is nearly zero – *e.g.* near the limit of machine precision. This case is the practical one, since the exact occurrence of $\omega_i = 0$ is highly unlikely for big matrices on a finite machine.

We have a choice: include ω_N in the computation, or treat it as if it were zero. In the former case, we retain a full $N-$ dimensional basis for $\{b\}$ and $\{u\}$; but the inversion becomes a noise amplifier. In the latter case, we avoid the amplified noise at the expense of forbidding the solution to ever have any $\{V\}_N$ content; any forcing in the vector $\{U\}_N$ is ignored. If $\{U\}_N$ is an important mode of the forcing, then we are in trouble; noise in its specification will overwhelm other modes of the solution. If it is not, then we are justified somewhat in ignoring it. Important physical questions concern the existence of such a poorly conditioned mode.

The practical solution is to consider as zero all ω_i which fail to meet a precision criterion based on condition number – essentially insisting on a cutoff on condition number by treating the smallest ω_N as if they were exactly zero. In effect, we eliminate the possibility of $\{V\}_i$ in the solution for those smallest ω's. If by chance there is a tiny bit of $\{U\}_N$ in the forcing, then we ignore it rather than amplify it.

The Over-Determined Case

This is the case $m > n$, *i.e.* where there are more equations than unknowns. Assuming all equations are independent, we generally have no hope of a perfect solution. There will always be a nonzero residual $\{r\} = [K]\{u\} - \{b\}$. The SVD is still defined for this case, and the procedure outlined above gives the smallest residual *i.e.* minimizes $Var(r)$.

The SVD is defined as in equation 13.55:

$$[K] = [U]\,[diag(\omega)]\,[V]^T \tag{13.85}$$

with the attendant properties as stated at equation 13.55. Note, however, that here $m > n$, so $[U]$ is not square and its rows are not orthogonal: $[U][U]^T \neq [I]$.

The columns of $[V]$ constitute a complete, orthonormal basis for $\{u\}$.

$$\{c\} = [V]^T\{u\} \qquad \Leftrightarrow \qquad [V]\{c\} = \{u\} \tag{13.86}$$

and the columns of $[U]$ constitute a natural, orthonormal basis for $\{b\}$. However, since $m > n$, this basis is *incomplete*:

$$\{d\} = [U]^T\{b\} \qquad \Leftrightarrow \qquad [U]\{d\} = \{b\} - \{b'\} \tag{13.87}$$

with $\{b'\}$ lying outside the $[U]$ space:

$$[U]^T\{b'\} = 0 \tag{13.88}$$

As in the general case, $[K]$ maps $\{V_i\}$ onto $\{U_i\}$ with scaling ω_i:

$$[K]\{V_i\} = \omega_i\{U_i\} \tag{13.89}$$

Since $\{u\}$ is completely contained within $[V]$, then $[K]\{u\}$ is completely contained within $[U]$. Therefore $\{b'\}$ cannot be reached by any $\{u\}$. For the most general solution $\{u\} = \sum_{i=1}^{n} c_i\{V_i\}$,

we have the residual

$$\{r\} = \sum_{i=1}^{n}(c_i\omega_i - d_i)\{U_i\} - \{b'\} \tag{13.90}$$

Since all vectors $\{U_i\}$ and $\{b'\}$ are orthogonal, we have

$$Var(r) = \sum_{i}^{n}(c_i\omega_i - d_i)^2 + Var(b') \tag{13.91}$$

Therefore the solution

$$c_i = \frac{d_i}{\omega_i} \qquad \Leftrightarrow \qquad \{u\} = \sum_i \{V\}_i \frac{\{U\}_i \cdot \{b\}}{\omega_i} \tag{13.92}$$

is the Minimum Variance solution.

Because of this property, the SVD solution is frequently invoked in Least-Squares problems. Provided the equations are scaled properly, this is one obvious path to the minimum variance estimate for linear systems.

If ω_i is too small, it can be treated as if it were zero, as in the square cases. One avoids the creation of some inverse noise, by leaving it untouched in the solution residual $\{b'\}$.

The Under-Determined Case

This is the case $m < n$, where we have more unknowns than equations. Normally this will give us an exact solution with $n - m$ arbitrary parameters (assuming that the equations are all independent). The SVD theory above can be applied by augmenting the system with empty equations of the form $0 \cdot u = 0$, such that $m = n$. We will generate $n - m$ modes with $\omega_i = 0$, which are readily handled as above. So we need no special theory for this case. The standard SVD solution will have zero residual; and, among that subset of solutions, $Var(u)$ will be minimized.

SVD Covariance

In every case we have the same general procedure: eliminate all singular and nearly-singular modes ($\omega_i \approx 0$) from the calculations. The result is a reduced-rank system which prevents noise creation by leaving it essentially uninverted, when to do otherwise would be to greatly amplify it. The Rank R is defined as the number of active modes – the number of *practically* nonzero ω.

The covariance formula for the answer u is given above. **For the special case** $Cov(b) = \sigma^2[I]$, *i.e.* **uncorrelated, homogeneous noise** of size σ^2, individual entries in $Cov(u)$ are

$$\overline{u_i u_j} = \sigma^2 \sum_{k=1}^{R} \frac{V_{ik}V_{jk}}{\omega_k^2} \tag{13.93}$$

The effect of reducing the rank R is obvious here. Overall, $Var(u)$ is similarly reduced:

$$Var(u) = \sum_{i=1}^{N} \overline{u_i u_i} = \sigma^2 \sum_{k=1}^{R}\sum_{i=1}^{n} \frac{V_{ik}V_{ik}}{\omega_k^2} = \sigma^2 \sum_{k=1}^{R} \frac{1}{\omega_k^2} \tag{13.94}$$

Since in this case we have $Var(b) = m\sigma^2$ and $Var(b') = (m-R)\sigma^2$, therefore the residual variance is

$$Var(r) = Var(b') = (m-R)\sigma^2 \tag{13.95}$$

Clearly, increasing the rank R reduces $Var(r)$, transferring it to $Var(u)$ with multiplier $\frac{1}{\omega}$.

SVD References

The reader is referred to more fundamental expositions of the SVD, for example that provided by Golub and Van Loan [35]. Press *et al.* [99] offer an excellent practical account. Both of these stress practical computability concerns. The public software implementation in LAPACK [3] is recommended.

13.3 Linear Least Squares and the Normal Equations

Here we summarize the standard formulation of the Linear Least Squares problem in terms of the Normal Equations. Given an overdetermined system of linear equations (more independent equations than unknowns), this classic formulation expresses the first-order conditions for the minimum of a quadratic norm of the misfit. The Normal Equations are linear in the unknowns. The Generalized Least Squares (GLS) approach at the end of this section provides the key operational extension to the underdetermined case. GLS is also the link to SVD theory. A good reference for LLS is Seber [100].

Quadratic Forms and Gradient

First consider the general quadratic norm of a vector $\{x\}$:

$$Q = \{x\}^T [W] \{x\} = \sum_i \sum_j x_i W_{ij} x_j \tag{13.96}$$

The derivative with respect to an individual x_k is

$$\frac{\partial Q}{\partial x_k} = \sum_j W_{kj} x_j + \sum_i x_i W_{ik} \tag{13.97}$$

There are two parts: the dot product of x with row k and with column k of $[W]$. Pictorially, this is

$$\begin{aligned} \frac{\partial Q}{\partial x_k} &= \begin{bmatrix} \leftarrow & W_{kj} & \rightarrow \end{bmatrix} \begin{Bmatrix} \uparrow \\ x \\ \downarrow \end{Bmatrix} + \begin{Bmatrix} \leftarrow & x & \rightarrow \end{Bmatrix} \begin{bmatrix} \uparrow \\ W_{ik} \\ \downarrow \end{bmatrix} \\ &= \begin{bmatrix} \leftarrow & W_{kj} & \rightarrow \end{bmatrix} \begin{Bmatrix} \uparrow \\ x \\ \downarrow \end{Bmatrix} + \begin{bmatrix} \leftarrow & W_{kj}^T & \rightarrow \end{bmatrix} \begin{Bmatrix} \uparrow \\ x \\ \downarrow \end{Bmatrix} \end{aligned} \tag{13.98}$$

Assembling all the individual $\frac{\partial Q}{\partial x_k}$ into the gradient vector ∇Q, we have

$$\{\nabla Q\} = [W]\{x\} + \{x\}^T [W] \tag{13.99}$$

$$= \left([W] + [W]^T\right) \{x\} \tag{13.100}$$

In many practical contexts, $[W]$ will be symmetric. In that case we have

$$\{\nabla Q\} = 2\,[W]\,\{x\} \tag{13.101}$$

For the scalar product of $\{x\}$ with any vector $\{V\}$,

$$S = \{x\}^T\,\{V\} = \sum_i x_i V_i = \{V\}^T\,\{x\} \tag{13.102}$$

it is easy to confirm that $\frac{\partial S}{\partial x_i} = V_i$ and therefore

$$\{\nabla S\} = \{V\} \tag{13.103}$$

Ordinary Least Squares

Now we have the linear system[6]

$$[A]\,\{x\} = \{b\} \tag{13.104}$$

with $[A]$ nonsquare ($m \times n$, $m > n$) and nonsymmetric; $\{b\}$ known; and $\{x\}$ unknown. Assuming that $[A]$ has more than n independent rows, then the system is overdetermined; no solution exists which satisfies all the equations. Define the residual of the system as

$$\{r\} \equiv [A]\,\{x\} - \{b\} \tag{13.105}$$

and under these conditions, $\{r\}$ is necessarily nonzero. So it is reasonable to seek its minimum. First, simply work with its variance, and make this our metric Ω, which will be minimized:

$$\Omega = Var(r) = \{r\}^T\,\{r\} \tag{13.106}$$

Substituting for $\{r\}$ we obtain

$$\begin{aligned}
\Omega &= \Big([A]\,\{x\} - \{b\}\Big)^T\Big([A]\,\{x\} - \{b\}\Big) && (13.107)\\
&= \{x\}^T\,[A]^T\,[A]\,\{x\} - \{x\}^T\,[A]^T\,\{b\} - \{b\}^T\,[A]\,\{x\} + \{b\}^T\,\{b\} && (13.108)\\
&= \{x\}^T\,[A]^T\,[A]\,\{x\} - 2\,\{x\}^T\,[A]^T\,\{b\} + \{b\}^T\,\{b\} && (13.109)
\end{aligned}$$

The gradient of Ω is, using the previous results,

$$\Big\{\nabla\Omega\Big\} = \Big(\Big[[A]^T\,[A]\Big] + \Big[[A]^T\,[A]\Big]^T\Big)\Big\{x\Big\} - 2\Big[\ A\ \Big]^T\Big\{b\Big\} \tag{13.110}$$

Since $\Big[[A]^T\,[A]\Big]$ is symmetric,

$$\Big\{\nabla\Omega\Big\} = 2\Big[[A]^T\,[A]\Big]\Big\{x\Big\} - 2\Big[\ A\ \Big]^T\Big\{b\Big\} \tag{13.111}$$

The first-order conditions for minimizing Ω are the vanishing of all components of its gradient. So this leads to the **OLS Normal Equations** which define the solution with minimum residual variance:

$$\Big[[A]^T\,[A]\Big]\Big\{x\Big\} = \Big[\ A\ \Big]^T\Big\{b\Big\} \tag{13.112}$$

[6]Note here we treat a general matrix equation, and save the notation $[K]\,\{u\} = \{b\}$ to denote the well-posed FE or FD model.

Notice that effectively we have premultiplied the original nonsquare equation by $[A]^T$; the result is an $n \times n$ system with the formal solution obtained by inverting $[A]^T [A]$:

$$\left\{x\right\} = \left[[A]^T [A]\right]^{-1} \left[\begin{array}{c} \\ A \\ \\ \end{array}\right]^T \left\{b\right\} \tag{13.113}$$

Now the question turns to the conditioning of $[A]^T [A]$ and the existence of its inverse at all. In the context of experimental work, it is common to find that while all equations are independent, the normal equations are poorly conditioned and produce noisy results. An obvious strategy is to solve the normal equations using SVD and to control the rank to keep out modes of the solution which have small singular values. The discussion in the previous section details this strategy.

Of course, the direct solution of equation 13.104 by SVD is possible, without the intervening construction of the Normal Equations.

Weighted Least Squares

Next suppose that "all residuals are not equal". We can formally insert a weighting matrix $[W]$ into Ω:

$$\Omega' = \{r\}^T [W] \{r\} \tag{13.114}$$

At its most elementary, the diagonals of $[W]$ adjust for units, expected size of each residual, *etc.* But this quadratic norm offers the additional possibility of penalizing cross-products of the residuals $(r_i W_{ij} r_j)$. More fundamentally, we may be concerned with differences among the r_i and other linear combinations:

$$\{r'\} = [V] \{r\} \tag{13.115}$$

where the V_{ij} could be positive or negative. Then the variance of $\{r'\}$ is

$$\Omega' = Var(r') = \{r\}^T [V]^T [V] \{r\} \tag{13.116}$$

In this more general case, the weighting matrix $[W] = [V]^T [V]$ is symmetric, and $[V]$ expresses prior attitude about what is important in the residual $\{r\}$.

Following the procedure of the previous section, we can arrive at the Normal Equations for this case of Weighted Least Squares. *We will assume that* $[W]$ *is symmetric.* (It is also reasonable to assume that $[W]$ is nonsingular, but that is not needed yet.)

$$\Omega' = \Big([A]\{x\} - \{b\}\Big)^T [W] \Big([A]\{x\} - \{b\}\Big) \tag{13.117}$$

$$= \{x\}^T [A]^T [W] [A] \{x\} - 2\{x\}^T [A]^T [W] \{b\} + \{b\}^T [W] \{b\} \tag{13.118}$$

The gradient of Ω' is

$$\left\{\nabla\Omega'\right\} = 2\Big[[A]^T [W] [A]\Big]\left\{x\right\} - 2\Big[[A]^T [W]\Big]\left\{b\right\} \tag{13.119}$$

and the first-order conditions for minimizing Ω' are

$$\Big[[A]^T [W] [A]\Big]\left\{x\right\} = \Big[[A]^T [W]\Big]\left\{b\right\} \tag{13.120}$$

These are the **WLS Normal Equations.** As in the OLS version, we have squared up the system by premultiplication by $[A]^T [W]$; the end result is an $n \times n$ system with the formal solution obtained by inverting $[A]^T [W] [A]$. The previous discussion concerning conditioning of the OLS matrix (the case $[W] = [I]$), and the SVD solution, applies here too.

General Least Squares

Finally: in the most general case, we may express concern over both the residual $\{r\}$ and the answer $\{x\}$:

$$\Omega'' = \{r\}^T [W_r] \{r\} + \{x\}^T [W_x] \{x\} \quad (13.121)$$
$$= \Omega' + \{x\}^T [W_x] \{x\} \quad (13.122)$$

As above, the weight matrices $[W_r]$ and $[W_x]$ express unhappiness with the size and various aspects of the shape of the two vectors involved, relative to prior opinion, *i.e.* logically prior to the inversion. We assume symmetry for both matrices. The gradient is obtained by extension of $\{\nabla\Omega'\}$ from above:

$$\left\{\nabla\Omega''\right\} = \left\{\nabla\Omega'\right\} + 2\left[\ W_x\ \right]\left\{x\right\} \quad (13.123)$$

Setting the gradient to zero gives the **GLS Normal Equations**:

$$\left[[A]^T [W_r] [A] + [W_x]\right]\left\{x\right\} = \left[[A]^T [W_r]\right]\left\{b\right\} \quad (13.124)$$

If we contemplate the inversion of this system,

$$\left\{x\right\} = \left[[A]^T [W_r] [A] + [W_x]\right]^{-1}\left[[A]^T [W_r]\right]\left\{b\right\} \quad (13.125)$$

it becomes clear that $[W_x]$ can add desirable conditioning. Suppose, for example, that $[W_x] = \frac{1}{\sigma^2}[I]$ – expressing a simple preference for answers which are not big compared to σ. In this case, $[W_x]$ just beefs up the diagonal of the GLS system, presumably enhancing its invertability compared to the simpler cases which rely on the potentially ill-conditioned system matrix $[A]^T [A]$. It is therefore common in linear least squares problems to invoke $[W_x]$ as a **regularization** effect, *i.e.* as a way to avoid solutions which are big or noisy. This is seeking the same general outcome as provided by the direct SVD solution of equation 13.104, wherein the rank of the system is used to tune out such unwanted effects. In both cases we add a bias in favor of solutions which are smaller than might otherwise be produced.

Now all this makes the selection of the weight matrices seem arbitrary, based on convenience. There is however good statistical reasoning to support the following conclusion: *the weight matrix should be the inverse of the covariance* of the vector being estimated:

$$[W_x] = [Cov(x)]^{-1} \quad (13.126)$$
$$[W_r] = [Cov(r)]^{-1} \quad (13.127)$$

More precisely, these should be *prior estimates* of the covariances, *i.e.* without the benefit of knowing the Least-Squares result $\{x\}$. The Appendix provides some theoretical support for this; we will use this approach throughout. [7]

There is a second consideration concerning $[W_x]$ and GLS in general. In cases where OLS or WLS is *underdetermined* – $m < n$, *i.e.* fewer equations than unknowns – then GLS still provides a unique answer, and $[W]$ is the key to it. Consider the special case with rank $R = m < n$; all the

[7] It is reasonable to hope that the GLS matrix $[A]^T [W_r] [A] + [W_x]$ has an inverse. Certainly $[W_x]$ is, ideally, invertable, since its inverse is $[Cov(x)]$.

equations are independent. $[A]^T [A]$ necessarily has no inverse; we have multiple solutions which each produce $\{r\} = 0$; and OLS/WLS provides no choice among them. If $\{b\}$ is noisy, then we are *fitting noise* with potentially silly answers $\{x\}$. But with GLS, we add a second consideration to the objective Ω, in the term $\{x\}^T [W_x] \{x\}$ which penalizes the size and shape of $\{x\}$. Among the solutions with $\{r\} = 0$, the small, smooth ones are favored. The solution selected may well have nonzero $\{r\}$ if there is noise (always!). WLS achieves a balance between small $\{r\}$ and credible $\{x\}$. The balance achieved is implied in the details of $[W_x]$ and $[W_r]$.

This is strongly reminiscent of SVD solutions to underdetermined systems. Full-rank SVD drives $\{r\}$ to zero first, then works on minimizing $\{x\}$. Reducing the rank of the system below $R = m$, by condition number control, permits some nonzero $\{r\}$ to occur. Effects similar to WLS are achieved, although the details are different. But SVD has one scalar control, and therefore no analog of the flexibility introduced into GLS by the weight matrix $[W_x]$.

So, the GLS system is an important endpoint for Linear Least Squares. It provides much freedom in tayloring a problem in both over- and under-determined cases.

All of this suggests a joint strategy of first formulating the GLS normal equations, concentrating on the weight matrices $[W_x]$ and $[W_r]$ in the design of the inversion; and then using SVD to solve the resulting square system, using condition number control to avoid near-singularities and the resultant noise amplification in the normal equations.

Chapter 14

Fitting Models to Data

14.1 Inverting Data

Now we return to the general problem posed at the outset: using observations of a system to determine the necessary/sufficient forcing. We have the well-posed FD or FE model:

$$[K]\{u\} = \{b\} \tag{14.1}$$

with $\{b\}$ the unknown system forcing, $\{u\}$ the unknown response, and $[K]$ the known FEM system matrix. We assume the existence of $[K^{-1}]$.

The strategy:

- Measure $\{u\}$. We assume the measurements are incomplete (all u_i cannot be measured – why else would we be using a model?). We also assume that the measurements are not perfect.
- Deduce $\{b\}$ by making a least-squares fit of 14.1 to the data.
- Estimate the complete response $\{u\}$ implied by 14.1 and the estimated $\{b\}$. This includes the potential for "correcting" the measurements to account for measurement error.
- Estimate the uncertainty (inverse noise) associated with $\{u\}$ and $\{b\}$.
- Examine the misfit between the model $\{u\}$ and the data.

To do this we need *prior estimates* of the covariances of $\{u\}$, $\{b\}$, and the expected disagreement between model and data.

Model-Data Misfit

Define the **data** $\{d\}$ as the sum of the **sample** of $\{u\}$ plus a **model-data misfit** $\{\delta\}$

$$\{d\} = [S]\{u\} + \{\delta\} \tag{14.2}$$

Here $[S]$ is a nonsquare sampling matrix representing all the ways we could sample the model output $\{u\}$: direct sampling of nodal values, interpolation among them, differentiation, averaging, integrating, etc. – all of these are linear operations and representable as a linear sample. The misfit $\{\delta\}$ represents the sum of two effects: the measurement error, plus the discrepancy between model and reality ($\{u\}$ is the model response, while $\{d\}$ is obtained from nature). These two contributors to $\{\delta\}$ are indistinguishable here without more information.

As an example of the sampling matrix $[S]$, consider a measurement of the true field $\mu(x,y,z)$ at a point k, $(x,y,z)_k$, in nature. The standard FEM representation of the model field u is

$$u(x,y,z) = \sum_j u_j \phi_j(x,y,z) \tag{14.3}$$

and direct sampling at point k gives

$$u(x,y,z)_k = \sum_j u_j \phi_j(x,y,z)_k \tag{14.4}$$

and thus $S_{k,j} = \phi_j(x,y,z)_k$, with k the row index and j the column index. More generally, any **linear** sample $L_k(\mu)$ of the natural field μ

$$d_k = L_k(\mu) \tag{14.5}$$

can also be obtained from the FEM representation as

$$L_k(u) = \sum_i u_i L_k(\phi_i(x,y,z)) \tag{14.6}$$

and thus

$$S_{k,i} = L_k(\phi_i) \tag{14.7}$$

Essentially, $[S]$ is a matrix of samples of the FEM interpolants ϕ. Because of the local nature of these FEM interpolants, local sampling leads to a sparse sampling matrix $[S]$. We reiterate that $L(u)$ represents any linear sampling of a continuous field, including point sampling, averaging, differencing, differentiating, integrating, etc. Of course, each sample L_k has its own intrinsic sampling error (variance about the truth).

Now we have the two fundamental equations 14.2 and 14.1:

$$\{d\} = [S]\{u\} + \{\delta\} \qquad\qquad \{u\} = [K]^{-1}\{b\} \tag{14.8}$$

(Recall that $[K]^{-1}$ exists by definition, because the FEM statement is well-posed.) Before going further, we should de-mean everything. Introducing for each vector $\{u\}$, $\{b\}$ and $\{d\}$ a mean value and a perturbation,

$$\{u\} = \{\overline{u}\} + \{\hat{u}\} \qquad \{b\} = \{\overline{b}\} + \{\hat{b}\} \qquad \{d\} = \{\overline{d}\} + \{\hat{d}\} \tag{14.9}$$

we obtain for the mean[1]

$$\left\{\overline{d}\right\} = [S]\{\overline{u}\} \qquad\qquad \{\overline{u}\} = [K]^{-1}\left\{\overline{b}\right\} \tag{14.10}$$

and for the variability

$$\left\{\hat{d}\right\} = [S]\{\hat{u}\} + \{\delta\} \qquad\qquad \{\hat{u}\} = [K]^{-1}\left\{\hat{b}\right\} \tag{14.11}$$

Henceforth we deal with the perturbations only, assuming the means are known. We drop the notational distinction for cleanliness.

[1]Notice here that we are asserting that the FEM model mean is equal to the natural mean, *i.e.* there is no model bias.

Equations 14.11 are the fundamental ones. Eliminating $\{u\}$ among them we obtain the basic statement that the model-data misfit is linear in the data and in the unknowns $\{b\}$:

$$\{\delta\} = \{d\} - [S]\,[K]^{-1}\,\{b\} \tag{14.12}$$

Essentially, we want an estimate of $\{b\}$ such that

$$[S]\,[K]^{-1}\,\{b\} \simeq \{d\} \tag{14.13}$$

in some least-squares sense. Equivalently,

$$\{\delta\} \simeq 0 \tag{14.14}$$

Note here that $[S]\,[K]^{-1}$ is not square so no conventional inverse exists.

Direct Solution Strategies and Inverse Noise

In general, all solution strategies for equation 14.13 which we will consider will produce estimates of $\{b\}$ which are linear in the data:

$$\{b\} = [B]\,\{d\} \tag{14.15}$$

with $[B]$ the pseudo-inverse of $[S]\,[K]^{-1}$. Accordingly, the estimate of $\{u\}$ is also linear in the data:

$$\{u\} = [K]^{-1}\,\{b\} = \left[[K]^{-1}\,[B]\right]\{d\} \tag{14.16}$$

(No data, no $\{b\}$, no $\{u\}$ and no misfit. Ignorance is bliss!) Assuming some data, the inverse noise is obtained directly:

$$[Cov(b)] = [B]\,[Cov(d)]\,[B]^T \tag{14.17}$$

$$[Cov(u)] = [K]^{-1}\,[Cov(b)]\,[K]^{-T} = [K]^{-1}\,[B]\,[Cov(d)]\,[B]^T\,[K]^{-T} \tag{14.18}$$

Different solution strategies will have different linear estimators $[B]$. For example, straight solution of equation 14.13 via **SVD** requires only one parameter, the condition number cutoff. SVD will minimize $Var(\delta)$, driving it to zero if possible, and in that case minimizing $Var(b)$ as a secondary goal. The relations for $[B]$ are covered in the foregoing section in terms of the Singular Vectors and Values of the matrix $[S]\,[K]^{-1}$:

$$[B]_{SVD} = [V]\left[diag(\frac{1}{\omega})\right][U]^T \tag{14.19}$$

If **GLS** is used, the principle will be minimization of the composite criterion

$$\Omega'' = \{\delta\}^T\,[W_\delta]\,\{\delta\} + \{b\}^T\,[W_b]\,\{b\} \tag{14.20}$$

and the estimator $[B]$ is obtained from the normal equations:

$$[B]_{GLS} = \left(\left[SK^{-1}\right]^T\,[W_\delta]\left[SK^{-1}\right] + [W_b]\right)^{-1}\left[SK^{-1}\right]^T\,[W_\delta] \tag{14.21}$$

Here we need to introduce $[W_\delta]$ and $[W_b]$; ideally these are the inverse *prior covariances* of $\{\delta\}$ and $\{b\}$, respectively. The equations 14.17 and 14.18 above for $[Cov(b)]$ and $[Cov(u)]$ are *posterior* estimates *i.e.* they are posterior to the data and the inversion of it. [2]

[2]The prior and posterior relationships between $[Cov(u)]$ and $[Cov(b)]$ are identical, $[Cov(u)] = [K]^{-1}\,[Cov(b)]\,[K]^{-T}$, due to the direct correspondence between u and b: $\{u\} = [K]^{-1}\,\{b\}$, independent of the data.

More on the Model-Data Misfit

Above we asserted that $\{\delta\}$ comprises two parts, due to model and data imperfections. It is useful to develop this a little further.

Let $\{\mu\}$ be the Truth, *i.e.* the true state of nature[3]; and $\{u\}$ be the model version as above. Define the discrepancy as $\{\epsilon_m\}$, the "model error":

$$\{u\} = \{\mu\} + \{\epsilon_m\} \tag{14.22}$$

As above, define the data as the sum of a sample of Truth plus measurement error $\{\epsilon_d\}$ ("data error"; "measurement noise")

$$\{d\} = [S]\{\mu\} + \{\epsilon_d\} \tag{14.23}$$

The model-data misfit is now

$$\begin{aligned} \{d\} - [S]\{u\} &= [S]\{\mu\} + \{\epsilon_d\} - [S]\{\mu\} - [S]\{\epsilon_m\} \\ &= \{\epsilon_d\} - [S]\{\epsilon_m\} \end{aligned} \tag{14.24}$$

and so the misfit $\{\delta\}$ can be separated in principle into two parts representing a sampling of model imperfections, plus measurement noise:

$$\{\delta\} = \{\epsilon_d\} - [S]\{\epsilon_m\} \tag{14.25}$$

In other words, the misfit is the superposition of the effects of imperfect model and imperfect data. The superposition depends on linearity in the sampling and in the processes generating $\{\epsilon_m\}$ and $\{\epsilon_d\}$. But it is valid for nonlinear as well as linear models. Assuming the two sources are independent, and independent of the state of nature $\{\mu\}$, we have

$$[Cov(\delta)] = [S][Cov(\epsilon_m)][S]^T + [Cov(\epsilon_d)] \tag{14.26}$$

and

$$[Cov(d)] = [S][Cov(\mu)][S]^T + [Cov(\epsilon_d)] \tag{14.27}$$

The former is needed in specifying $[W_\delta]$; it combines the two sources of statistical misfit. The latter is needed in developing the posterior covariances $[Cov(b)]$ and $[Cov(u)]$, equations 14.17 and 14.18. It combines natural variability with measurement variability. If $\{u\}$ and $\{\epsilon_m\}$ are independent, then we have the further relation

$$[Cov(d)] = [S]\Big([Cov(u)] + [Cov(\epsilon_m)]\Big)[S]^T + [Cov(\epsilon_d)] \tag{14.28}$$

All of these additive covariance relations are intuitively sensible, and useful. Accounting for covariance among $\{u\}$, $\{\epsilon_m\}$, $\{\epsilon_d\}$, etc, is difficult except in special cases.

Note: $Cov(u)$ in 14.28 is defined in the absence of data. It is logically prior to the data. In 14.18 we have $Cov(u)$ posterior to the data – the variability of the estimate of u from the data. These two covariances are fundamentally different and it is important to distinguish them.

[3]Truth is by definition unknowable with certainty; it can only be estimated.

14.2 Constrained Minimization and Gradient Descent

In this section we describe the Generalized Least Squares problem for minimizing Model-Data misfit as one of *constrained minimization.* The classic notion of Lagrange Multipliers is introduced. The multipliers are additional mathematical variables – the *Adjoint Variables* – which figure prominently in iterative solution strategies by *Gradient Descent* toward the GLS minimum. This approach achieves the same solution as the GLS solution, but the path is different and we encounter the Adjoint Variables along the way.

Generalized Least Squares as Constrained Minimization

We are concerned with minimizing a general quadratic functional Ω which combines model-data misfit and solution:

$$\Omega = \{b\}^T [W_b] \{b\} + \{\delta\}^T [W_\delta] \{\delta\} \tag{14.29}$$

with $[W_b]$, $[W_\delta]$ symmetric positive definite; and subject to the constraints

$$[K]\{u\} = \{b\} \tag{14.30}$$

$$\{d\} - [S]\{u\} = \{\delta\} \tag{14.31}$$

Eliminating δ by substitution leaves us the minimization of

$$\Omega = \{b\}^T [W_b] \{b\} + \Big(\{d\} - [S]\{u\}\Big)^T [W_\delta] \Big(\{d\} - [S]\{u\}\Big) \tag{14.32}$$

subject to the remaining constraint

$$[K]\{u\} = \{b\} \tag{14.33}$$

This is the problem we will solve. It is posed as one of *constrained minimization.* We have control variables (unknowns) $\{b\}$ and $\{u\}$ to be manipulated, but they are not independent; the constraint 14.33 must be enforced among them. The classical approach is to construct an augmented quadratic form

$$\Omega^+ = \{b\}^T [W_b] \{b\} + \Big(\{d\} - [S]\{u\}\Big)^T [W_\delta] \Big(\{d\} - [S]\{u\}\Big) + \{\lambda\}^T \Big([K]\{u\} - \{b\}\Big) \tag{14.34}$$

which includes the constraints weighted by the *Lagrange Multipliers* $\{\lambda\}$. These supplement the list of unknowns. Intuitively, the minimum value of Ω^+ will occur with $[K]\{u\} = \{b\}$, in which case Ω and Ω^+ have the same value. The value of λ_i represents the resistance of the minimum to violation of the i^{th} constraint.

With the constraint embedded in Ω^+, we may proceed to describe the first-order conditions for its minimum. Specifically, the gradient of Ω^+ with respect to the three controls $\{b\}$, $\{u\}$, and $\{\lambda\}$, is

$$\left\{\frac{\partial \Omega^+}{\partial b}\right\} = 2[W_b]\{b\} - \{\lambda\} \tag{14.35}$$

$$\left\{\frac{\partial \Omega^+}{\partial u}\right\} = -2\left[S^T W_\delta\right]\{d\} + 2\left[S^T W_\delta S\right]\{u\} + [K]^T\{\lambda\} \tag{14.36}$$

$$\left\{\frac{\partial \Omega^+}{\partial \lambda}\right\} = [K]\{u\} - \{b\} \tag{14.37}$$

All components of this gradient must vanish. Before proceeding, let's demonstrate that the condition $\nabla\Omega^+ = 0$ produces the identical solution to that obtained with the normal equations. Solving 14.35 for $\{b\}$ and 14.37 for $\{u\}$ gives us

$$\{u\} = [K]^{-1}\{b\} \tag{14.38}$$

$$\{\lambda\} = 2[W_b]\{b\} \tag{14.39}$$

Substitution of these into 14.36 gives us the single equation for $\{b\}$:

$$-2\left[S^T W_\delta\right]\{d\} + 2\left[S^T W_\delta S K^{-1}\right]\{b\} + 2\left[K^T W_b\right]\{b\} = 0 \tag{14.40}$$

$$\left[\left[S^T W_\delta S K^{-1}\right] + \left[K^T W_b\right]\right]\{b\} = \left[S^T W_\delta\right]\{d\} \tag{14.41}$$

Premultiplying by $[K^{-1}]^T = [K^T]^{-1} = [K^{-T}]$ gives us

$$\left[K^{-T} S^T W_\delta S K^{-1} + W_b\right]\{b\} = \left[K^{-T} S^T W_\delta\right]\{d\} \tag{14.42}$$

which is identical to that derived above (equation 13.124) for GLS for $[S][K^{-1}]\{b\} = \{d\}$ (with $[A] \equiv [SK^{-1}]$, etc.). Thus we see that introduction of the Lagrange Multipliers does not change the basic answer – but it introduces some flexibility in the solution technique.

The Adjoint Method

Here we describe the solution of equations (14.35, 14.36, 14.37) by *iteration.* The method is frequently referred to as the "adjoint method". Rather than eliminating the adjoint variables $\{\lambda\}$ algebraically, they are retained and explicitly computed. They play a prominent role in the iteration. Instead of computing a large concatenation of matrices involving several full-matrix products and inversions, we concentrate on solving the easy subsets of the algebra iteratively.

The basic idea is expressed procedurally:

1. Guess $\{b\}$. At the start, this is the **Prior Estimate** of the basic unknown.

2. Solve 14.37 for $\{u\}$:

$$[K]\{u\} = \{b\} \tag{14.43}$$

   Since $[K]$ is a conventional FEM matrix, it can be counted on to be well-conditioned, and sparse. Its efficient and accurate solution is assumed. Its right-hand side is the most recent estimate of $\{b\}$. This system is referred to as the **Forward System** or Forward Model, since it represents the basic problem being posed. Here we solve it with the best available estimate of the unknown right-hand side.

3. Evaluate the model-data misfit $\{\delta\}$

$$\{\delta\} = \{d\} - [S]\{u\} \tag{14.44}$$

   (This step is optional, and can be accomplished implicitly in the assembly of the right-hand side in step 4.)

4. Solve 14.36 for $\{\lambda\}$:

$$[K]^T\{\lambda\} = 2\left[S^T W_\delta\right]\{d\} - 2\left[S^T W_\delta S\right]\{u\} \quad (14.45)$$

$$= 2\left[S^T W_\delta\right]\{\delta\} \quad (14.46)$$

Efficient and accurate solution of this system is assumed based on the properties of $[K]^T$ which mirror those of $[K]$. This system is referred to as the **Adjoint System** or Adjoint Model. It is structurally the transpose of the Forward System.

5. Evaluate the remaining first-order conditions – equation 14.35. This is the gradient of Ω^+ with respect to $\{b\}$:

$$\left\{\frac{\partial\Omega^+}{\partial b}\right\} = 2[W_b]\{b\} - \{\lambda\} \quad (14.47)$$

If this is zero, **STOP**. All the conditions for a minimum are satisfied. Otherwise, make an adjustment to $\{b\}$ and **REITERATE**, returning to step 2.

This procedure amounts to a search in the decision space $\{b\}$ for that location (set of unknown forcings) which minimizes Ω^+. Each step is intrinsically simple. The computationally intensive ones are steps 2 and 4, both of which involve repetitive solution of a sparse, well-conditioned matrix equation arising in FEM analysis. Since the matrices involved are $[K]$ and its transpose, a single factorization at the beginning of the iteration is desirable. For example, using LU decomposition,

$$[K] = [L][U] \quad (14.48)$$

$$[K]^T = [U]^T[L]^T \quad (14.49)$$

Similar efficiencies are available with other matrix factorization schemes. The computational requirements therefore scale as a single common FEM assembly and factorization at the start (necessary anyway!), plus two common FEM back-substitutions per iteration. Efficiency in the iteration is therefore critical. The key to this efficiency will be found in step 5, the computation of the update to the estimate of $\{b\}$. Below we discuss two common methods: the Steepest Descent Method and the Conjugate Gradient Method. Both share a two-part structure which relies on information about the nonzero gradient of Ω^+ obtained in step 5.

Gradient Descent

We are searching for the minimum of the scalar functional Ω^+, in the multidimensional space $\{b\}$. We assume the ability to evaluate Ω^+ and its gradient vector $\{\nabla\Omega^+\} = \{\frac{\partial\Omega^+}{\partial b}\}$ at any point. Assuming that $\nabla\Omega^+ \neq 0$, we require an update to the current best estimate of $\{b\}$ which makes Ω^+ get smaller.

Let the current position be $\{b\}^k$. Then

$$\{b\}_{k+1} = \{b\}_k + \{\Delta b\} \quad (14.50)$$

Furthermore, divide $\{\Delta b\}$ into direction $\{\partial b\}$ and magnitude α:

$$\{\Delta b\} = \alpha\{\partial b\} \quad (14.51)$$

(It is suggested that $\{\partial b\}$ be a unit vector, or otherwise scaled to some sensible size; but that is not necessary.) There are two questions: which direction to select, and how far to go in that direction.

We will address the latter question first. Given a direction $\{\partial b\}$, we are searching along a line in the $n-$ dimensional $\{b\}$- space. The idea is to minimize Ω^+ along this line, by selecting the optimal value of α. This is a nice subproblem which has a simple solution.

First define the effect of $\{\partial b\}$:

$$\{\partial u\} = \left[K^{-1}\right]\{\partial b\} \tag{14.52}$$

$$\{\partial \delta\} = -[S]\{\partial u\} \tag{14.53}$$

and notice that the algebra is linear in α:

$$\{\Delta b\} = \alpha\{\partial b\} \tag{14.54}$$

$$\{\Delta u\} = \left[K^{-1}\right]\{\Delta b\} = \alpha\{\partial u\} \tag{14.55}$$

$$\{\Delta \delta\} = \alpha\{\partial \delta\} \tag{14.56}$$

So the objective Ω^+ at the new position is

$$\Omega^+_{k+1} = \{\delta_k + \Delta\delta\}^T[W_\delta]\{\delta_k + \Delta\delta\} + \{b_k + \Delta b\}^T[W_b]\{b_k + \Delta b\} \tag{14.57}$$

$$= \Omega^+_k + \{\Delta\delta\}^T[W_\delta]\{\delta_k\} + \{\delta_k\}^T[W_\delta]\{\Delta\delta\} + \{\Delta\delta\}^T[W_\delta]\{\Delta\delta\} \tag{14.58}$$

$$+ \{\Delta b\}^T[W_b]\{b_k\} + \{b_k\}^T[W_b]\{\Delta b\} + \{\Delta b\}^T[W_b]\{\Delta b\} \tag{14.59}$$

$$\Delta\Omega^+ = 2\alpha\{\partial\delta\}^T[W_\delta]\{\delta_k\} + \alpha^2\{\partial\delta\}^T[W_\delta]\{\partial\delta\} \tag{14.60}$$

$$+ 2\alpha\{\partial b\}^T[W_b]\{b_k\} + \alpha^2\{\partial b\}^T[W_b]\{\partial b\} \tag{14.61}$$

(We have assumed that $[W_b]$ and $[W_\delta]$ are symmetric.) Now by differentiating relative to the scalar α we obtain the condition for the minimum along the line parallel to ∂b:

$$\alpha = -\frac{\{\partial b\}^T[W_b]\{b_k\} + \{\partial\delta\}^T[W_\delta]\{\delta_k\}}{\{\partial b\}^T[W_b]\{\partial b\} + \{\partial\delta\}^T[W_\delta]\{\partial\delta\}} \tag{14.62}$$

This equation is valid for α for any given direction $\{\partial b\}$. What is left is, how to select the direction.

Steepest Descent

This is the simplest gradient descent method. The direction $\{\partial b\}$ is chosen to be parallel to the negative gradient:

$$\{\partial b\} = -\left\{\frac{\partial\Omega^+}{\partial b}\right\} \tag{14.63}$$

or any normalization of the gradient. Since the gradient is the direction of maximum increase, then this is the *Method of Steepest Descent.* Coupled with equation 14.62, we have a method which selects the steepest direction down, and goes straight in that direction until it bottoms out. Then it changes direction. Although this is intuitively appealing, and simple, it has pitfalls and is usually avoided. There is no guarantee that the directions chosen will not repeat or nearly repeat themselves, resulting in *slow convergence.* The method has no memory of previous directions used. It will converge to the correct answer since in this case (Linear Least Squares) only a single minimum is possible. In general an infinite number of iterations is needed, as with most iterative methods.

Conjugate Gradient Descent

This method computes a sequence of gradients $\left\{\frac{\partial\Omega}{\partial b}\right\}_k = \{g\}_k$, and a sequence of directions $\{\partial b\}_k = \{h\}_k$. The sequences are computed as follows:

$$\{h\}_{k+1} = -\{g\}_{k+1} + \gamma_{k+1}\{h\}_k \tag{14.64}$$

The scalar parameter γ is recomputed at each step as

$$\gamma_{k+1} = \frac{\left(\{g\}_{k+1} - \{g\}_k\right)^T \{g\}_{k+1}}{\{g\}_k^T \{g\}_k} \tag{14.65}$$

This method may be started with $\{h\}_1 = -\{g\}_1$, *i.e.* using steepest descent for the first step. Note that once started, there is memory in the direction-setting. The step size α is optimized as in 14.62.

Details of this method are beyond our purpose here; see for example Golub and Van Loan, [35]. The method was initially developed as a direct solution strategy for linear systems; in the present context it converges to the exact solution in exactly n iterations, with the first iterations making the most progress. However the iterations are unstable in the presence of roundoff, so it is important to stop this method well short of n steps. In practice the Conjugate Gradient Descent can be significantly faster than Steepest Descent. The notation here follows that of Press *et al.* [99], except for a reversal in the sign of $\{g\}$.

Summary – Adjoint Method with Gradient Descent

In Figure 14.1 we summarize the iterative method outlined above. Each iteration consists of a forward and an adjoint model run; the objective is the minimization of Ω^+ which includes the quadratic norms of the Model-Data Misfit $\{\delta\}$ and the Forcing $\{b\}$. The gradient of Ω^+ is used to direct the next iteration.

Monte Carlo Variance Estimation – Inverse Noise

The Gradient Descent methods normally do not include an iterative estimate of $[Cov(u)]$. Since they converge to the same solution as GLS, the expressions for $[Cov(u)]$ from that development pertain here; they can be evaluated after the iteration has converged.

Iterative methods are frequently invoked when memory, run-time, or coding considerations are dominant. In particular, note that the method summarized in Figure 14.1 requires only factorization of the FEM matrices $[K]$ and its transpose, plus some more elementary operations, all of which are necessary in a standard 'forward model' run and its comparison with data. It may be constructed fairly simply, starting from a conventional 'forward model' environment. In these cases, the matrix formalism is likely to be awkward or unavailable for constructing the Covariances.

An alternative approach is to perform an ensemble of iterative inversions, each with a statistically valid perturbation added to the data. The ensemble of results $\{u\}$ can then be sampled for mean and variance. The mean is then the estimated answer (inverse of the data); and the variance is then the estimate of the diagonal of $[Cov(u)]$. For a sufficiently large ensemble, properly representing $[Cov(d)]$, these estimates are precise and equivalent to the algebraic expressions given above for GLS.

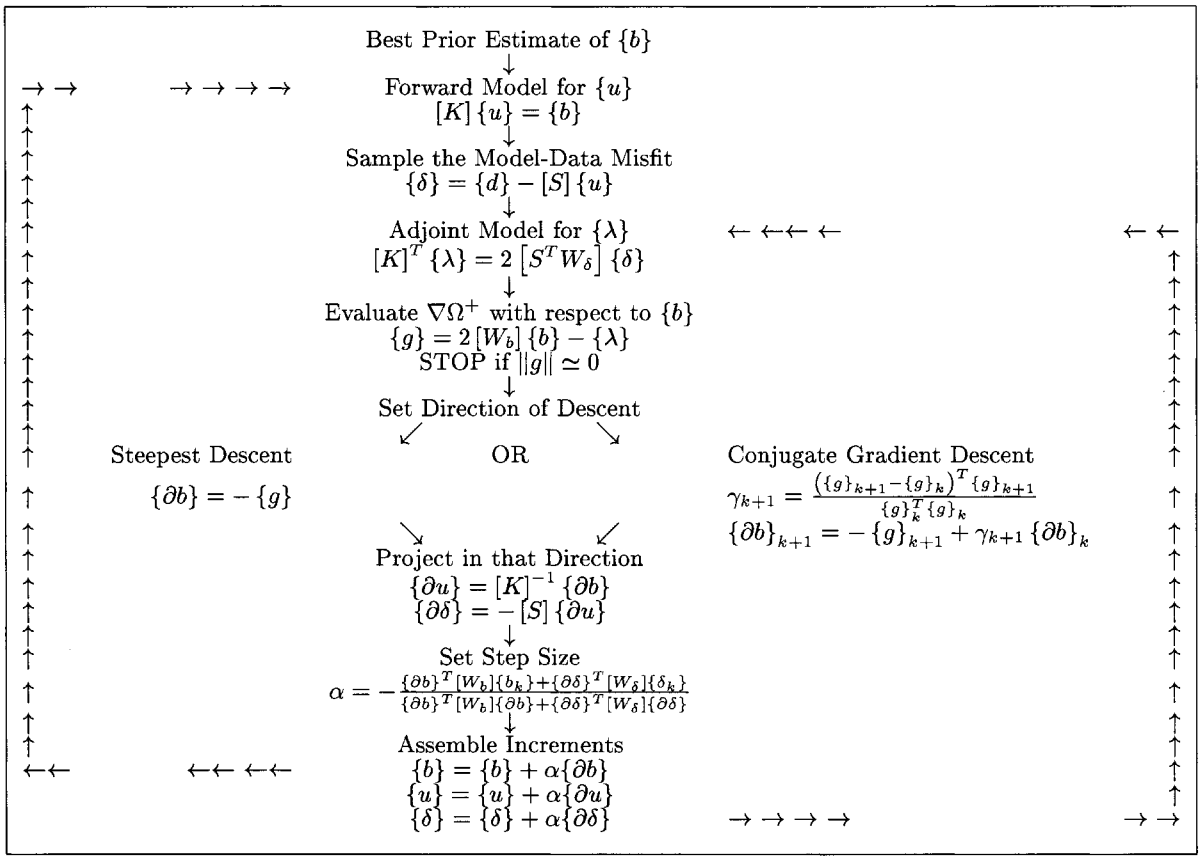


Figure 14.1: Adjoint method of iterative solution of the GLS Model-Data Misfit problem. There are two alternate return paths. On the left side, the forward model and misfit are recalculated once the full increment to $\{b\}$ is known. On the right side, linear superposition is used to project the new Misfit. The superposition saves one forward model run per iteration.

This "Monte Carlo" approach rests on generation of ensembles of randomly-perturbed data. When the data are statistically independent, this is straightforward and supported in most computational systems. When they covary, however, the problem becomes more sophisticated. A transformation method involving the Cholesky decomposition is described in the Appendix.

Much has been written about Monte Carlo methods; we refer the reader to the literature (*e.g.* Winston, [114]).

14.3 Inverting Data With Representers

A distinctly different approach to the GLS minimization of Model-Data Misfit is the Representer approach. We pose the same problem as above, in terms of Lagrange Multipliers. The first-order conditions for a minimum comprise the basic problem statement. The path to solution begins with the Adjoint equations, forced by a single misfit δ_i. With m data, exactly m such solutions (Representers) are needed to close the calculation in a final $m \times m$ matrix inversion. The result is a direct solution technique which is efficient when the number of data m is much less than the

number of unknowns n.

The equations and unknowns are the same as expressed in the first steps of the Adjoint method, Figure 14.1. Their order is rearranged here to suit the exposition:

$$
\begin{aligned}
[K]^T \{\lambda\} &= 2\left[S^T W_\delta\right] \{\delta\} && (14.66)\\
2\{b\} &= [W_b]^{-1} \{\lambda\} = [Cov(b)] \{\lambda\} && (14.67)\\
[K] \{u\} &= \{b\} && (14.68)\\
\{\delta\} &= \{d\} - [S] \{u\} && (14.69)
\end{aligned}
$$

Notice that here we have set $\{g\} = 0$ rather than using it to direct a gradient search. Also note that we have assumed that $[W_b] = [Cov(b)]^{-1}$. Significantly, this particular arrangement of the first-order minimum conditions requires no matrix inversion beyond the factorization of the FEM system matrix $[K]$, which is assumed to be routine here.

The Procedure

Construct a solution of 14.66 through 14.68, starting with *Unit Misfit* $\delta_i = 1$, $\delta_{j \neq i} = 0$. Call the resulting forward model solution $\{U\}_i$. Sample it; and call its sampling the Representer $\{r\}_i = [S] \{U\}_i$. Do this once for each datum, $i = 1 \rightarrow m$.

Now if the individual misfits δ_i were known, then by superposition we have

$$
\begin{aligned}
\{\delta\} &= \{d\} - \sum_i \delta_i \{r\}_i \\
&= \{d\} - [R] \{\delta\} && (14.70)
\end{aligned}
$$

where the Representer vectors $\{r\}_i$ are the *columns* of $[R]$. The basic idea is to use this to calculate $\{\delta\}$:

$$
[I + R] \{\delta\} = \{d\} \qquad (14.71)
$$

The matrix $[I + R]$ is square, $m \times m$, and presumably full. Its inversion closes the system, efficiently if $m \ll n$ and all data are independent.

With $\{\delta\}$ known, the solution $\{u\}$ can be synthesized from the unit response vectors $\{U\}_i$:

$$
\begin{aligned}
\{u\} &= \sum_i \delta_i \{U\}_i \\
&= [U] \{\delta\} && (14.72)
\end{aligned}
$$

where the vectors $\{U\}_i$ are the *columns* of $[U]$. Alternatively, if $[U]$ has not been saved, one can simply re-solve equations 14.66, 14.67, and 14.68, starting with the known $\{\delta\}$.

If m is the number of data, this procedure requires the construction of m representers, each of which requires a forward and an adjoint solution, a sampling of $\{u\}$, plus a computation of $\{b\}$. The latter is reduced to a simple matrix product because $[W_b]^{-1} = [Cov(b)]$, a prior input. Roughly, this is comparable to solving $2m$ forward problems, requiring only a single factorization of the FEM system $[K]$ at the start. The final step requires solving the dense $m \times m$ system once, followed by a pair of forward solutions to recover $\{b\}$ and $\{u\}$. So we see that for small m, the Representer method is very attractive.

Naturally, one needs to worry about the conditioning of the matrix $[I + R]$. Since this blends many features of the problem, caution is advised and inversion via SVD and inspection of the condition number is recommended.

Inverse Noise

The estimator for $\{u\}$ is linear in the data:

$$\{u\} = [U]\,[I + R]^{-1}\,\{d\} \tag{14.73}$$

and the covariance is

$$[Cov(u)] = [U]\,[I + R]^{-1}\,[Cov(d)]\,[I + R]^{-T}\,[U]^T \tag{14.74}$$

This particular expression is remarkably compact; and efficient to the extent that inversion of $[I + R]$ is efficient. Note the absence of the FEM matrix $[K]$ and its inverse here; and the sampling matrix $[S]$. They are implicit in the construction of $[R]$ and $[U]$.

14.4 Inverting Data with Unit Responses

Representers are effective when m (the # of data) is small relative to n (the number of unknowns). The procedure collects the calculations into a single dense $m \times m$ matrix expression of the model-data misfit. The result: a direct method which requires $2m$ forward model solutions, a single $m \times m$ inversion, and two more forward model solutions.

In the opposite case, when $n \ll m$, an analogous procedure is based on responses to unit forcing, with the system closing around the relation between $\{b\}$ and $\{\lambda\}$. The work involved is $2n + 1$ forward solutions, a single $n \times n$ inversion, and one final forward solution. As above we assume that each forward model solution is relatively easy and well-conditioned.

Procedure

The standard four equations are reexpressed in the most convenient order:

$$\begin{aligned}
[K]\,\{u\} &= \{b\} && (14.75)\\
\{\delta\} &= \{d\} - [S]\,\{u\} && (14.76)\\
[K]^T\,\{\lambda\} &= 2\left[S^T W_\delta\right]\{\delta\} && (14.77)\\
2\,\{b\} &= [Cov(b)]\,\{\lambda\} && (14.78)
\end{aligned}$$

First, construct n solutions of 14.75 and 14.76 with unit forcing, $b_i = 1$, $b_{j\neq i} = 0$. Sample each one using $[S]$; call those resulting vectors the unit predictions $\{P\}_i$. Now the misfit may be expressed as the superposition of the unit responses $\{P\}_i$, with unknown weights $b_i = 1$:

$$\{\delta\} = \{d\} - \sum_i b_i\,\{P\}_i \tag{14.79}$$

Next create $n+1$ forward solutions to 14.77, driven individually by $\{d\}$ and by the $\{P\}_i$. Call these solutions $\{\lambda\}_d$ and $\{\lambda\}_i$. With these in hand, $\{\lambda\}$ can be expressed as a linear combination of the unknown b_i:

$$\{\lambda\} = \{\lambda\}_d - \sum_i b_i \{\lambda\}_i = \{\lambda\}_d - [\Lambda]\{b\} \tag{14.80}$$

where the columns of $[\Lambda]$ are the individual responses $\{\lambda\}_i$ to unit forcing b_i. Equation 14.78 can now be assembled:

$$2\{b\} = [Cov(b)]\left(\{\lambda\}_d - [\Lambda]\{b\}\right) \tag{14.81}$$

and the final form is

$$\left(2[I] + [Cov(b)][\Lambda]\right)\{b\} = [Cov(b)]\{\lambda\}_d \tag{14.82}$$

Solution of this for $\{b\}$, plus a single forward solution to 14.75 for $\{u\}$, completes the procedure.

14.5 Summary: GLS Data Inversion

We have examined several approaches to inverting data. All share the same relations:

$$[K]\{u\} = \{b\} \qquad (Forward\ FEM\ Model) \tag{14.83}$$

$$\{\delta\} = \{d\} - [S]\{u\} \qquad (Model - Data\ Misfit) \tag{14.84}$$

and the same general objective, to minimize $\{\delta\}$ in some sense. By eliminating $\{u\}$, we have

$$\{\delta\} = \{d\} - [S][K]^{-1}\{b\} \tag{14.85}$$

and wanting the misfit $\{\delta\}$ to be small, we seek $\{b\}$ such that the sampled model fits the data:

$$\cdot [S][K]^{-1}\{b\} \approx \{d\} \tag{14.86}$$

$[S][K]^{-1}$ is generally not invertable; but we seek its pseudo-inversion. The residual of this equation is the model-data misfit.

The SVD is unique in that $Var(\delta)$ is minimized alone. Otherwise, the objective is to minimize Ω:

$$\Omega = \{b\}^T [W_b]\{b\} + \{\delta\}^T [W_\delta]\{\delta\} \tag{14.87}$$

In every case the control variables are the two arrays $\{b\}$ and $\{\delta\}$. The size of the system is n equations and m data. With the exception of SVD, all methods can be viewed as rearrangements of the same first-order conditions for the same extremum of Ω. We assume that $[K]$ is sparse, nonsingular and easily factorable, being a discrete, well-posed approximant to a PDE. Existence of its inverse is guaranteed.

- SVD is a direct method using factorization of $[S][K]^{-1}$. It is supported by a general theory which is a direct extension of EigenTheory. It highlights the role of small Singular Values as amplifying noise contained in the associated Singular Vectors.
- The direct GLS method uses the Generalized Normal Equations for equation 14.86. Inversion of a full, $n \times n$ matrix is required; there is no guarantee of its condition.

- An iterative method introduces Lagrange Multipliers as additional *adjoint variables*; and uses Gradient Descent in the decision space $\{b\}$. Each iteration requires a forward and an adjoint model run. Each iteration is comparable to two forward model runs with pre-computed and factored matrix $[K]$. Slow convergence can offset the speed per iteration.

- A direct method uses Representers (responses to unit misfits). It requires computation of m representers and inversion of a dense $m \times m$ matrix. It is desirable when $m \ll n$.

- A second direct method uses Impulse Responses (responses to unit forcing). It is analogous to the Representer approach and is attractive when $n \ll m$.

- Other iterative methods are not discussed but are desirable. In particular a Gradient Descent method in $\{\delta\}$ space might be efficient for $m \ll n$.

- All results are linear in the data:

$$\{b\} = [B]\{d\} \tag{14.88}$$
$$\{u\} = [K]^{-1}[B]\{d\} \tag{14.89}$$
$$\{\delta\} = \{d\} - [S][K]^{-1}[B]\{d\} \tag{14.90}$$

  Each method has a different approach to the practical computation of $[B]$; they are algebraically identical with the exception of SVD.

- The weight matrices are ideally equal to the *Prior Inverse Covariances*:

$$[W_b] = [Cov(b)]^{-1} \tag{14.91}$$
$$[W_\delta] = [Cov(\delta)]^{-1} \tag{14.92}$$

- The method of computing Posterior Covariances depends for convenience on the solution path chosen. All are consistent. For the Gradient Descent method, no direct formula is easily gotten. A Monte Carlo method for computing variance is likely to be preferred in this case, since these methods are most attractive when the forward model is easily solved but the algebra associated with direct solution is prohibitively large.

- Generally $[Cov(u)]$ (the inverse noise) is dependent on $[Cov(d)]$ but not on the data themselves.

14.6 Parameter Estimation

In this section we turn to the problem of estimating the parameters of a PDE and its discrete representation. By *parameters* we refer specifically to coefficients appearing in the governing PDE. Boundary conditions, initial conditions, *etc.* are excluded by this definition. We will employ a General Least Squares approach, as in the inverse problems described above. However, the parameter estimation problem is significantly different. Even for a linear PDE and linear discretization of same, the dependent variables $\{u\}$ are *nonlinear* in the parameters – by definition, the PDE involves products of its parameters with the dependent variables. And its discrete representation

$$[K]\{u\} = \{b\} \tag{14.93}$$

has unknown $[K]$, thus spoiling the algebraic linearity which would occur if only $\{u\}$ and $\{b\}$ were unknown. Here we assume that $\{u\}$ and $\{b\}$ are unknown as before, and we have an additional vector of unknowns $\{y\}$ which appear in the discretized PDE. They typically are the degrees of freedom involved in a discrete representation of continuous parameter field(s). Accordingly, $[K] = [K(y)]$, and we have the nonlinear forward problem

$$[K(y)]\{u\} = \{b\} \tag{14.94}$$

with unknowns $\{u\}$, $\{b\}$, $\{y\}$, and imperfect observations $\{d\}$ of the system state $\{u\}$.

$$\{d\} - [S]\{u\} = \{\delta\} \tag{14.95}$$

In a GLS sense, the model-data misfit $\{\delta\}$ is to be minimized. Solution by iteration is necessitated by the nonlinearity.

GLS Objective

We have the minimization of the GLS objective

$$\Omega = \{b\}^T [W_b]\{b\} + \{\delta\}^T [W_\delta]\{\delta\}^T + \{y\}^T [W_y]\{y\} \tag{14.96}$$

subject to the constraints 14.94 and 14.95 above. As usual $[W_b] = [Cov(b)]^{-1}$ and similarly for $[W_\delta]$ and $[W_y]$. To enforce the forward model constraint 14.94, we introduce the Lagrange Multipliers $\{\lambda\}$. The objective becomes minimization of the augmented Ω^{++}, which now involves terms in the misfit $\{\delta\}$, the forcing $\{b\}$, the parameters $\{y\}$, and $\{\lambda\}$:

$$\Omega^{++} = \{b\}^T [W_b]\{b\} + \{\delta\}^T [W_\delta]\{\delta\} + \{y\}^T [W_y]\{y\} + \{\lambda\}^T \Big([K]\{u\} - \{b\}\Big) \tag{14.97}$$

Eliminating δ by substitution of 14.95 leads us to the minimization of

$$\begin{aligned} \Omega^{++} = \{b\}^T [W_b]\{b\} \quad &+ \quad \Big(\{d\} - [S]\{u\}\Big)^T [W_\delta] \Big(\{d\} - [S]\{u\}\Big) \\ &+ \quad \{y\}^T [W_y]\{y\} + \{\lambda\}^T \Big([K]\{u\} - \{b\}\Big) \end{aligned} \tag{14.98}$$

with control variables $\{b\}$, $\{u\}$, $\{y\}$, and $\{\lambda\}$, and with the understanding that $\{\delta\}$ is a surrogate of $\{u\}$:

$$\{\delta\} \equiv \{d\} - [S]\{u\} \tag{14.99}$$

First-Order Conditions for GLS Extremum

With the constraint embedded in Ω^{++}, we may proceed to describe the first-order conditions for its minimum. Specifically, the gradient of Ω^{++} with respect to the four controls $\{b\}$, $\{u\}$, $\{\lambda\}$, and $\{y\}$ is

$$\left\{\frac{\partial \Omega^{++}}{\partial b}\right\} = 2[W_b]\{b\} - \{\lambda\} \tag{14.100}$$

$$\left\{\frac{\partial \Omega^{++}}{\partial u}\right\} = [K]^T\{\lambda\} - 2\left[S^T W_\delta\right]\{\delta\} \tag{14.101}$$

$$\left\{\frac{\partial\Omega^{++}}{\partial\lambda}\right\} = [K]\{u\} - \{b\} \tag{14.102}$$

$$\left\{\frac{\partial\Omega^{++}}{\partial y}\right\} = 2[W_y]\{y\} + \{\lambda\}^T\left[\frac{\partial K}{\partial y}\right]\{u\} \tag{14.103}$$

plus the surrogate relation

$$\{\delta\} \equiv \{d\} - [S]\{u\} \tag{14.104}$$

These are essentially unchanged from the comparable first-order conditions for estimating $\{b\}$, except for

- the dependence of $[K]$ everywhere on the unknown $\{y\}$; and
- the additional fourth gradient vector 14.103 relative to the parameters $\{y\}$.

If we freeze $\{y\}$, then 14.103 is irrelevant and the equations reduce to the fixed-parameter set previously studied. If the number of parameters is small, we may get away with nesting an estimation for $\{b\}$, with any of the above techniques, within an iteration for $\{y\}$, such that the newly-hatched gradient 14.103 is annihilated. This is particularly attractive when the number of data are small, in which case we could expect the Representer approach to be very effective. A good prior estimate of $\{y\}$ is a necessity.

Equation 14.103 is new. The first term is familiar by now; recall especially that $[W_y]^{-1} = [Cov(y)]$, and that this represents a *prior estimate*; *i.e.* it is independent of the solution. The second term deserves some special attention; its most general expansion is

$$\{\lambda\}^T\left[\frac{\partial K}{\partial y}\right]\{u\} = \sum_i\sum_j \lambda_i \frac{\partial K_{ij}}{\partial y_m} u_j \tag{14.105}$$

If $[K]$ is linear in the parameters $\{y\}$, then the matrices involved here are constants; however even in this case they present important requirements for storage, especially when the dimension of $\{y\}$ is high; and the assembly of 14.105 will be computationally intensive for the general case.

We will review this term before going further, to see if we can take advantage of the fact that $[K]$ is a standard FEM matrix representing familiar FEM constructions.

The Gradient in Parameter Space

Here it is convenient to reintroduce the FEM residual vector $\{r\}$:

$$\{r\} \equiv [K]\{u\} - \{b\} \tag{14.106}$$

and the final term in Ω^{++} (equation 14.98) is

$$\{\lambda\}^T\{r\} = \sum_i \lambda_i r_i \tag{14.107}$$

If each individual r_i is a MWR statement with weighting function $\phi_i(\mathbf{x})$, then this term is a MWR statement with weighting function $\lambda(\mathbf{x}) = \sum_i \lambda_i\phi_i(\mathbf{x})$. The gradient relative to y_m is

$$\frac{\partial}{\partial y_m}\left(\{\lambda\}^T\{r\}\right) = \sum_i \lambda_i \frac{\partial r_i}{\partial y_m} \tag{14.108}$$

Example. Consider the PDE

$$-\nabla \cdot (y \nabla u) = f \tag{14.109}$$

with Neumann boundary data q. Its MWR expression is

$$r_i = \langle y \nabla u \nabla \phi_i \rangle - \oint q \ \phi_i \ ds - \langle f \phi_i \rangle \tag{14.110}$$

If the parameter $y(\mathbf{x})$ is expressed in the basis $\psi(\mathbf{x})$,

$$y(\mathbf{x}) = \sum_m y_m \psi_m(\mathbf{x}) \tag{14.111}$$

then we have

$$\frac{\partial r_i}{\partial y_m} = \langle \psi_m \nabla u \nabla \phi_i \rangle \tag{14.112}$$

and the contribution to the gradient is

$$\frac{\partial}{\partial y_m} \left(\{\lambda\}^T \{r\} \right) = \langle \psi_m \nabla u \nabla \lambda \rangle \tag{14.113}$$

If λ and u are known, or presumed known in an iteration, then this gradient vector is readily evaluated using standard FEM assembly procedures, treating λ and u as data. Since we need to iterate on the parameter estimation problem *a priori*, with routine re-assembly of the basic FEM matrix $[K]$ in each iteration, only marginal additional work need be generated in the same loop to assemble the gradient.

From the more general perspective of equation 14.105, we have

$$K_{ij} = \langle y \nabla \phi_j \nabla \phi_i \rangle \tag{14.114}$$

$$\frac{\partial K_{ij}}{\partial y_m} = \langle \psi_m \nabla \phi_j \nabla \phi_i \rangle \tag{14.115}$$

which leads to the same result.

$$\{\lambda\}^T \left[\frac{\partial K}{\partial y} \right] \{u\} = \sum_i \sum_j \lambda_i \frac{\partial K_{ij}}{\partial y_m} u_j = \langle \psi_m \nabla u \nabla \lambda \rangle \tag{14.116}$$

For this case, then, we can pull together the complete specification of the gradient, from 14.103:

$$\left\{ \frac{\partial \Omega^{++}}{\partial y} \right\} = 2 \left[W_y \right] \{y\} + \langle \psi_m \nabla u \nabla \lambda \rangle \tag{14.117}$$

If Dirichlet BC's are involved, we need an extra measure of care here. Suppose, for example, that u_5 is a Dirichlet node, and row 5 of $[K]$ has been replaced with a direct specification of u_5 (unity on the diagonal, zero off-diagonal). In this case, r_5 is completely insensitive to any of the parameters y_m; so in the assembly of 14.113, λ_5 needs to be *temporarily* treated as if it were zero. (In general, $\lambda_5 \neq 0$, hence this is a procedural workaround. Other assembly procedures can be fashioned for the Dirichlet case.)

An Adjoint Method for Parameter Estimation

Here we sketch an Adjoint-based iterative method for satisfying the first-order conditions. The idea is a straight generalization of that expressed in section 14.2, for the estimation of $\{b\}$. Here we add gradient descent in the combined $\{b\}$, $\{y\}$ space. The basic idea is expressed procedurally:

1. **Prior Estimates** of $\{b\}, \{y\}$ are needed. These are the basic unknowns. At the start, a good guess is needed.
2. **Forward Model**: Assemble the forward model system $[K(y)]$ and $\left[\frac{\partial K}{\partial y}\right]$. Solve 14.102 for $\{u\}$:
$$[K]\{u\} = \{b\} \tag{14.118}$$
3. **Model-Data Misfit**: Evaluate $\{\delta\}$ from the new value of $\{u\}$ using equation 14.104
$$\{\delta\} = \{d\} - [S]\{u\} \tag{14.119}$$
4. **Adjoint Model**: Solve 14.101 for $\{\lambda\}$:
$$[K]^T\{\lambda\} = 2\left[S^T W_\delta\right]\{\delta\} \tag{14.120}$$
5. **Gradient Descent**: Evaluate the remaining first-order conditions – equations 14.100 and 14.103. These give the gradient of Ω^{++} with respect to $\{b\}$ and $\{y\}$
$$\left\{\frac{\partial \Omega^{++}}{\partial b}\right\} = 2[W_b]\{b\} - \{\lambda\} \tag{14.121}$$
$$\left\{\frac{\partial \Omega^{++}}{\partial y}\right\} = 2[W_y]\{y\} + \{\lambda\}^T\left[\frac{\partial K}{\partial y}\right]\{u\} \tag{14.122}$$
If these gradients are zero, **STOP**. All the conditions for a minimum are satisfied. Otherwise, make an adjustment to $\{b\}$ and $\{y\}$, and **REITERATE**, returning to step 2 (Forward model assembly and solution) above.

14.7 Summary – Terminology

Here is a general summary of terminology used so far in the discussion of fitting algebraic models to data:

- The finite element model is $[K]\{u\} = \{b\}$
- $[K]$ is the FEM matrix, sparse, well-conditioned
- $\{u\}$ is the field variable of interest
- $\{b\}$ is the unknown forcing vector (BC's + inhomogeneous term in the PDE)
- $\{r\}$ is the residual in the FEM equations, $[K]\{u\} - \{b\}$. It is normally zero here except during an iteration.

- $\{\mu\}$ is the true field variable in nature, unknown but approximated by $\{d\}$ and by $\{u\}$
- $\{d\}$ is the data, imperfect observations of $\{\mu\}$
- $\{\epsilon\}$ indicates an error *i.e.* a discrepancy between Truth and an Estimate. Since Truth is *a priori* unknowable except in a statistical sense, therefore errors are unknowable.
- $\{\epsilon_m\}$ is the model error: $\{u\} = \{\mu\} + \{\epsilon_m\}$
- $\{\epsilon_d\}$ is the observational error: $\{d\} = [S]\{\mu\} + \{\epsilon_d\}$
- $\{y\}$ is the vector of unknown model parameters: $[K] = [K(y)]$
- $[S]$ is a sampling matrix: $[S]\{u\}$ approximates the data
- $\{\delta\}$ is the model-data misfit: $\{\delta\} = \{d\} - [S]\{u\}$. In terms of the errors, $\{\delta\} = \{\epsilon_d\} - [S]\{\epsilon_m\}$ which is useful in constructing $[Cov(\delta)]$.
- $[W]$ indicates a weight matrix used in GLS minimization. Generally, for any variable $\{x\}$ being estimated, $[W_x] = [Cov(x)]^{-1}$.

Chapter 15

Dynamic Inversion

In the previous chapter we examined inversion of static or steady-state models. Here we examine models of time-dynamic processes. As in the steady case, we will restrict our analysis to models which are linear in the necessary forcing; the associated inverse problems will be linear in the data. The parameter estimation problem is fundamentally nonlinear even for a linear model in the conventional sense. In either case, most of the apparatus already introduced will be used here. In particular, iterative solutions will be especially attractive if they take advantage of efficient forward model solvers.

15.1 Parabolic Model: Advective-Diffusive Transport

Consider the linear transport equation for a single unknown $u(x, y, z, t)$:

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \nabla \cdot D \nabla u + \kappa u = \sigma \tag{15.1}$$

with parameters $\mathbf{v}, D$ (transport) and κ (first-order decay rate); and an exogenous source σ. For the time being we will assume that *the parameters are known perfectly*; and that the source σ is known imperfectly.

To complete the problem specification, we require boundary conditions and initial conditions. Both are assumed to be known imperfectly. In addition, we know the field $u(x, y, z)$ at a future time, the Terminal Condition, but only imperfectly. The conventional forward problem would have us integrate forward in time from assumed initial conditions, using best estimates of IC's, BC's, and sources, in the hope of hitting the terminal condition. By assuming that these (plus the parameters) are known, we specify a unique solution; and in a practical world, the hope of perfect success is nil. So conventional practice would ask for reasonable adjustments in the information which is the least well-known – IC's, BC's, and sources – in order to achieve a reasonable match with TC's. Inversion is the formalization of this procedure to produce optimal estimates of u as well as supplements to all information which is imperfectly known (IC's, BC's, TC's, and sources).[1]

[1]Notice here we are not treating IC's as observations, as was effectively the case in the static problems in the previous chapters. The forward problem requires *simultaneous* IC's , *i.e.* a *synoptic field* $u(x, y, z)$. In practice such a field observation would not be possible; so it is assumed here that IC's are obtained by a blend of observations and model calculations, with various levels of processing to obtain the field estimate. We also assume that such processing includes an estimate of the field covariance as well as the state itself. The same discussion pertains to TC's. Later we will explore this distinction.

For Euler time-stepping with Galerkin FEM, we would have:

$$\begin{aligned} M_{ij} &= \langle \phi_j \phi_i \rangle \\ A_{ij} &= \langle \phi_j \phi_i \rangle - \Delta t \langle \mathbf{v} \cdot \nabla \phi_j \ \phi_i + D \nabla \phi_j \cdot \nabla \phi_i + \kappa \ \phi_j \phi_i \rangle \\ b_{ik} &= \Delta t \left(\langle \sigma \ \phi_i \rangle + \oint D \frac{\partial u}{\partial n} \phi_i ds \right)^k \end{aligned}$$

For a more general implicit time-stepping scheme, we have:

$$\begin{aligned} M_{ij} &= \langle \phi_j \phi_i \rangle + \Delta t \langle \mathbf{v} \cdot \nabla \phi_j \ \phi_i + D \nabla \phi_j \cdot \nabla \phi_i + \kappa \ \phi_j \phi_i \rangle \theta \\ A_{ij} &= \langle \phi_j \phi_i \rangle - \Delta t \langle \mathbf{v} \cdot \nabla \phi_j \ \phi_i + D \nabla \phi_j \cdot \nabla \phi_i + \kappa \ \phi_j \phi_i \rangle (1 - \theta) \\ b_{ik} &= \Delta t \left(\langle \sigma \ \phi_i \rangle + \oint D \frac{\partial u}{\partial n} \phi_i ds \right)^{k+\theta} \end{aligned}$$

with ϕ the basis and θ a time-weighting parameter.

Table 15.1: FEM discretization of advective-diffusive-reactive equation; terms as in equations 15.2-15.4.

Forward Model in Discrete Form

The simulation model in discrete form is:

Initial Condition

$$\{u\}_0 = \{\tilde{u}\}_0 \tag{15.2}$$

Dynamic

$$[M] \{u\}_{k+1} = [A] \{u\}_k + \left\{\tilde{b}\right\}_k \qquad k = [0, n-1] \tag{15.3}$$

Terminal Condition

$$\{u\}_n \approx \{\tilde{u}\}_n \tag{15.4}$$

with

- $\{u\}_k$: Vector of Concentrations at time k (state variables in forward model)
- $\{\tilde{u}\}_0$, $\{\tilde{u}\}_n$: Initial/Terminal Conditions (Prior estimate)
- $[M]$, $[A]$: Physical Transport Matrices (advection + dispersion + decay, storage)
- $\left\{\tilde{b}\right\}_k$: Source + Inhomogeneous Boundary Conditions (Prior estimate)

(Notice that the tilde notation indicates a best prior estimate.) A typical FEM discretization is given in Table 15.1.

In normal forward or "open-loop" mode, integration from IC's (equation 15.2) generates disagreements with terminal conditions (equation 15.4). The goal is to reduce the disagreement by augmenting the source terms and/or the initial conditions *optimally*. Accordingly we rewrite the forward model with three explicit sources of error or uncertainty in the simulation:

Initial Condition

$$\{u\}_0 = \{\tilde{u}\}_0 + \{\alpha\} \tag{15.5}$$

Dynamic

$$[M]\{u\}_{k+1} = [A]\{u\}_k + \left\{\tilde{b}\right\}_k + \{\rho\} \qquad k = [0, n-1] \tag{15.6}$$

Terminal Condition

$$\{u\}_n = \{\tilde{u}\}_n + \{\delta\} \tag{15.7}$$

The vectors $\{\alpha\}$, $\{\delta\}$, and $\{\rho\}$ are unknown; but there is a prior estimate of their covariances $[Cov(\alpha)]$, $[Cov(\delta)]$, and $[Cov(\rho)]$. $\{\rho\}$ is here assumed to be constant through time (but spatially varying *i.e.* all scalar entries in the vector are independent). Time-variation in $\{\rho\}$ is an interesting extension; we ignore it here to start.

Objective and First-Order Conditions

We seek a generalized least squares (GLS) fit to the Initial and Terminal Conditions:

$$\text{Minimize } \Omega = \tfrac{1}{2}\left(\{\alpha\}^T[W_\alpha]\{\alpha\} + \{\rho\}^T[W_\rho]\{\rho\} + \{\delta\}^T[W_\delta]\{\delta\}\right)$$

subject to the forward model constraints, with $[W_\alpha] = [Cov(\alpha)]^{-1}$ etc. The controls available are $\{\alpha\}$, $\{\delta\}$, and $\{\rho\}$, in addition to the basic unknown solution $\{u\}_k$. To handle the constraints, we introduce the Lagrange Multipliers $\{\lambda\}$ and minimize the functional Z:

$$\begin{aligned}
Z &= \frac{1}{2}\left(\{\alpha\}^T[W_\alpha]\{\alpha\} + \{\rho\}^T[W_\rho]\{\rho\} + \{\delta\}^T[W_\delta]\{\delta\}\right) \\
&+ \left\{\tilde{\lambda}\right\}_0^T(\{u\}_0 - \{\tilde{u}\}_0 - \{\alpha\}) \\
&+ \left\{\tilde{\lambda}\right\}_n^T(\{u\}_n - \{\tilde{u}\}_n - \{\delta\}) \\
&+ \{\lambda\}_1^T\left([M]\{u\}_1 - [A]\{u\}_0 - \left\{\tilde{b}\right\}_0 - \{\rho\}\right) \\
&+ \{\lambda\}_2^T\left([M]\{u\}_2 - [A]\{u\}_1 - \left\{\tilde{b}\right\}_1 - \{\rho\}\right) \\
&+ \vdots \\
&+ \{\lambda\}_k^T\left([M]\{u\}_k - [A]\{u\}_{k-1} - \left\{\tilde{b}\right\}_{k-1} - \{\rho\}\right) \\
&+ \vdots \\
&+ \{\lambda\}_n^T\left([M]\{u\}_n - [A]\{u\}_{n-1} - \left\{\tilde{b}\right\}_{n-1} - \{\rho\}\right)
\end{aligned}$$

The gradient of Z is:

$$\left\{\frac{\partial Z}{\partial \rho}\right\} = [W_\rho]\{\rho\} - \sum_{k=1}^{n}\{\lambda\}_k \tag{15.8}$$

$$\left\{\frac{\partial Z}{\partial \alpha}\right\} = [W_\alpha]\{\alpha\} - \left\{\tilde{\lambda}\right\}_0 \tag{15.9}$$

$$\left\{\frac{\partial Z}{\partial \delta}\right\} = [W_\delta]\{\delta\} - \left\{\tilde{\lambda}\right\}_n \tag{15.10}$$

$$\left\{\frac{\partial Z}{\partial u}\right\}_0 = \left\{\tilde{\lambda}\right\}_0 - [A]^T \{\lambda\}_1 \tag{15.11}$$

$$\left\{\frac{\partial Z}{\partial u}\right\}_k = [M]^T \{\lambda\}_k - [A]^T \{\lambda\}_{k+1} \qquad k = [1, n-1] \tag{15.12}$$

$$\left\{\frac{\partial Z}{\partial u}\right\}_n = [M]^T \{\lambda\}_n + \left\{\tilde{\lambda}\right\}_n \tag{15.13}$$

$$\left\{\frac{\partial Z}{\partial \lambda}\right\} = \textit{Forward Model} \tag{15.14}$$

Setting these to zero gives the necessary conditions for a minimum. $\left\{\frac{\partial Z}{\partial \lambda}\right\} = 0$ recovers the **forward model**, equations 15.5, 15.6, 15.7. $\left\{\frac{\partial Z}{\partial u}\right\} = 0$ gives the **adjoint model.** The derivatives with respect to the perturbations $\{\alpha\}$, $\{\delta\}$, and $\{\rho\}$, give the relationships among the two models.

Adjoint Model

From the first-order conditions $\left\{\frac{\partial Z}{\partial u}\right\} = 0$, we have the relationships among the Lagrange Multipliers. This is the adjoint model:

Initial Condition

$$\left\{\tilde{\lambda}\right\}_0 - [A]^T \{\lambda\}_1 = 0 \tag{15.15}$$

Dynamic

$$[M]^T \{\lambda\}_k - [A]^T \{\lambda\}_{k+1} = 0 \qquad k = [1, n-1] \tag{15.16}$$

Terminal Condition

$$[M]^T \{\lambda\}_n = -\left\{\tilde{\lambda}\right\}_n \tag{15.17}$$

The relationships between the forward and adjoint variables are gotten from the final gradients relative to $\{\alpha\}$, $\{\delta\}$, and $\{\rho\}$:

$$[W_\alpha]\{\alpha\} = \left\{\tilde{\lambda}\right\}_0 \tag{15.18}$$

$$[W_\delta]\{\delta\} = \left\{\tilde{\lambda}\right\}_n \tag{15.19}$$

$$[W_\rho]\{\rho\} = \sum_{k=1}^{n} \{\lambda\}_k \tag{15.20}$$

Recall that $[W_\rho]^{-1} = [Cov(\rho)]$; so these relations are readily solved in either direction.

Note the adjoint structure is the dual of the forward model:

- The adjoint variables are the Lagrange Multipliers associated with the forward model dynamics
- Integration is backward in time from the T.C.
- Advection is reversed in $[A]^T$
- Adjoint boundary conditions are the homogeneous form of the forward model

- The forward model produces the terminal misfit $\{\delta\}$. This forces the adjoint model, $\left\{\tilde{\lambda}\right\}_n = [W_\delta]\{\delta\}$. No $\{\delta\}$, no adjoint solution needed.
- The adjoint solution produces:
  - the supplement to the source terms: $\rho \sim \sum \lambda_k$
  - the initial condition supplement $\alpha \sim \tilde{\lambda}_0$

We see that the adjoint model inherits its general structure and properties from the forward model. If the forward model can be assembled and easily solved, we can expect the same from the adjoint model.

Direct Solution – An Elliptic Problem in Time

It is useful first to condense the system a little by eliminating the intermediary variables $\{\alpha\}$, $\{\rho\}$, $\{\delta\}$, leaving only the prime (u) and dual (λ) variables:

Forward Model

$$\{u\}_0 = \{\tilde{u}\}_0 + [Cov(\alpha)]\left\{\tilde{\lambda}\right\}_0 \tag{15.21}$$

$$[M]\{u\}_{k+1} = [A]\{u\}_k + \{b\}_k + [Cov(\rho)]\sum_{k=1}^{n}\{\lambda\}_k \qquad k=[0,n-1] \tag{15.22}$$

Adjoint Model

$$\left\{\tilde{\lambda}\right\}_0 - [A]^T\{\lambda\}_1 = 0 \tag{15.23}$$

$$[M]^T\{\lambda\}_k - [A]^T\{\lambda\}_{k+1} = 0 \qquad k=[1,n-1] \tag{15.24}$$

$$[M]^T\{\lambda\}_n = -[W_\delta](\{u\}_n - \{\tilde{u}\}_n) \tag{15.25}$$

The combined system is formally a two-point (in time), elliptic boundary value problem, subject to constraints on initial and terminal conditions. To see this, we eliminate $\{\lambda\}$ to obtain:

$$[M]\{u\}_{k+1} - [M+A]\{u\}_k + [A]\{u\}_{k-1} = \{b\}_k - \{b\}_{k-1} \qquad k=[1,n-1] \tag{15.26}$$

with boundary conditions[2]

$$[M]\{u\}_1 - [A]\{u\}_0 = \{b\}_0 - [Cov(\rho)]\left[M^T - A^T\right]^{-1}\{c\} \tag{15.27}$$

$$[M]\{u\}_n - [A]\{u\}_{n-1} = \{b\}_{n-1} - [Cov(\rho)]\left[M^T - A^T\right]^{-1}\{c\} \tag{15.28}$$

The quantity $\{c\}$ driving these BC's is a composite of initial and terminal misfits; it is the same at both ends:

$$\{c\} = [W_\alpha]\Big(\{u\}_0 - \{\tilde{u}\}_0\Big) + [W_\delta]\Big(\{u\}_n - \{\tilde{u}\}_n\Big) \tag{15.29}$$

Effectively, these are Type 3 or mixed boundary conditions. As ($[W_\alpha]$, $[W_\delta]$) become large, we approach stronger Type 1 conditions on u_0 and u_n.

Direct solution of this linear system is possible. It requires constructing and factoring a large matrix and is not pursued further here. An iterative strategy is developed below.

[2]We have used equation (15.40) in developing the BC's.

Iterative Solution by Gradient Descent

We iterate using a Gradient Descent method. We enforce the gradient conditions (15.10 - 15.14), and use ρ and α as control parameters for the reduction of Z. The imbalance in the gradient conditions (15.8, 15.9) will drive the descent.

Initially, assume $\{\rho\} = \{\alpha\} = 0$. Then:

- Compute $\{u\}$ via forward model, forced by the best guess for $\{\rho\}$ and $\{\alpha\}$; evaluate $\{\delta\}$, the terminal error:

$$\begin{aligned} \{u\}_0 &= \{\tilde{u}\}_0 + \{\alpha\} && (15.30) \\ [M]\{u\}_{k+1} &= [A]\{u\}_k + \{b\}_k + \{\rho\} \qquad k = [0, n-1] && (15.31) \\ \{\delta\} &= \{u\}_n - \{\tilde{u}\}_n && (15.32) \end{aligned}$$

- Compute $\{\lambda\}$ by backwards-in-time integration of the adjoint model, forced by $\{\delta\}$:

$$\begin{aligned} \left\{\tilde{\lambda}\right\}_0 &= \left[A^T\right]\{\lambda\}_1 && (15.33) \\ \left[M^T\right]\{\lambda\}_k &= \left[A^T\right]\{\lambda\}_{k+1} \qquad k = [1, n-1] && (15.34) \\ \left[M^T\right]\{\lambda\}_n &= -[W_\delta]\{\delta\} && (15.35) \end{aligned}$$

- Compute $\{\nabla Z\} = (\left\{\frac{\partial Z}{\partial \rho}\right\}, \left\{\frac{\partial Z}{\partial \alpha}\right\})$

$$\begin{aligned} \left\{\frac{\partial Z}{\partial \rho}\right\} &= [W_\rho]\rho - \sum_{k=1}^{n}\{\lambda\}_k && (15.36) \\ \left\{\frac{\partial Z}{\partial \alpha}\right\} &= [W_\alpha]\alpha - \left\{\tilde{\lambda}\right\}_0 && (15.37) \end{aligned}$$

  If $\{\nabla Z\} \simeq 0$, **STOP**. All first-order conditions for a minimum are satisfied. Otherwise,

- Compute an appropriate descent direction $(\{\partial\rho\}, \{\partial\alpha\})$, *e.g.* via Steepest Descent or Conjugate Gradient Descent; and increment $(\{\rho\}, \{\alpha\})$ in that direction with stepsize σ:

$$\begin{aligned} \{\Delta\rho\} &= \sigma\{\partial\rho\} && (15.38) \\ \{\Delta\alpha\} &= \sigma\{\partial\alpha\} && (15.39) \end{aligned}$$

- Reiterate to convergence at $\nabla Z \approx 0$.

An interesting property of the adjoint is gotten by summing equations (15.15), (15.16), and (15.17):

$$\left([M]^T - [A]^T\right)\sum_k \{\lambda\}_k = -[W_\alpha]\{\alpha\} - [W_\delta]\{\delta\} \qquad (15.40)$$

Optimal Step Size

In each iteration we increment $(\{\rho\}, \{\alpha\})$ from their current values $(\{\rho_k\}, \{\alpha_k\})$, in a prescribed direction $(\{\partial\rho\}, \{\partial\alpha\})$, with stepsize σ:

$$\{\rho\}_{k+1} = \{\rho\}_k + \sigma\{\partial\rho\} \tag{15.41}$$
$$\{\alpha\}_{k+1} = \{\alpha\}_k + \sigma\{\partial\alpha\} \tag{15.42}$$

This results in increments in $\{\partial u\}$ and $\{\partial\delta\}$:

$$\{u\}_{k+1} = \{u\}_k + \sigma\{\partial u\} \tag{15.43}$$
$$\{\delta\}_{k+1} = \{\delta\}_k + \sigma\{\partial\delta\} \tag{15.44}$$

where $(\{\partial u\}, \{\partial\delta\})$ is the impact of a forward model run with $\sigma = 1$. [3] These in turn alter the objective as follows:

$$\begin{aligned} Z_{k+1} = {} & Z_k + \sigma\Big(\{\partial\alpha\}^T[W_\alpha]\{\alpha\}_k + \{\partial\rho\}^T[W_\rho]\{\rho\}_k + \{\partial\delta\}^T[W_\delta]\{\delta\}_k\Big) \\ & + \frac{\sigma^2}{2}\Big(\{\partial\alpha\}^T[W_\alpha]\{\partial\alpha\} + \{\partial\rho\}^T[W_\rho]\{\partial\rho\} + \{\partial\delta\}^T[W_\delta]\{\partial\delta\}\Big) \end{aligned} \tag{15.45}$$

Proper selection of the direction guarantees that the objective Z will decrease locally. The scalar σ is selected to have maximum impact on Z. The first-order condition $\frac{dZ}{d\sigma} = 0$ gives the optimum step size:

$$\sigma = -\frac{\{\partial\alpha\}^T[W_\alpha]\{\alpha\}_k + \{\partial\rho\}^T[W_\rho]\{\rho\}_k + \{\partial\delta\}^T[W_\delta]\{\delta\}_k}{\{\partial\alpha\}^T[W_\alpha]\{\partial\alpha\} + \{\partial\rho\}^T[W_\rho]\{\partial\rho\} + \{\partial\delta\}^T[W_\delta]\{\partial\delta\}} \tag{15.46}$$

Procedurally, we can get away with only a single adjoint solution per iteration as follows. First, compute $\{\partial u\}$ and $\{\partial\delta\}$ using $\sigma = 1$ as per the definition. This enables calculation of σ. Then by superposition (equations 15.43 and 15.44), recomputation of the adjoint model is unnecessary to complete the iteration.

Direction of Descent

Earlier in Section 14.2 we discussed the selection of the descent direction $(\{\partial\rho\}, \{\partial\alpha\})$. The most intuitive approach is Steepest Descent: proceed in the direction of the negative gradient. This method is workable and will ultimately succeed for a linear problem, since one is always going "downhill" and a unique minimum exists. It is well-established however that this can lead to slow convergence. A common alternative is the Conjugate Gradient Descent. Like Steepest Descent, this method begins in the direction of the negative gradient; however the second and subsequent directions are selected with memory of the previous directions in addition to the local gradient. The details are such that the minimum is reached theoretically in finite number of steps (n, the dimension of the search space), with the most rapid progress realized early. Since the method is unstable in the presence of roundoff error, it is important to stop the iteration well short of n steps.

[3] Notice that by definition (equation 15.32), $\{\partial\delta\} \equiv \{\partial u\}_n$ here.

Special Case #1: "Shooting"

Suppose that the IC's are known to be perfect, *a priori*. In that case, we need to remove $\{\alpha\}$ from the system of unknowns ($\{\alpha\} \equiv 0$ everywhere) and delete the gradient of Z with respect to $\{\alpha\}$, $\left\{\frac{\partial Z}{\partial \alpha}\right\}$. The forward model is:

Initial Condition

$$\{u\}_0 = \{\tilde{u}\}_0 \tag{15.47}$$

Dynamic

$$[M]\{u\}_{k+1} = [A]\{u\}_k + \{b\}_k + \{\rho\} \tag{15.48}$$

Terminal Condition

$$\{u\}_n = \{\tilde{u}\}_n + \{\delta\} \tag{15.49}$$

Objective: Minimize Z:

$$\begin{aligned}
Z &= \frac{1}{2}\left(\{\delta\}^T [W_\delta]\{\delta\} + \{\rho\}^T [W_\rho]\{\rho\}\right) \\
&+ \left\{\tilde{\lambda}\right\}_0^T (\{u\}_0 - \{\tilde{u}\}_0) \\
&+ \left\{\tilde{\lambda}\right\}_n^T (\{u\}_n - \{\tilde{u}\}_n - \{\delta\}) \\
&+ \{\lambda\}_1^T ([M]\{u\}_1 - [A]\{u\}_0 - \{b\}_0 - \{\rho\}) \\
&+ \{\lambda\}_2^T ([M]\{u\}_2 - [A]\{u\}_1 - \{b\}_1 - \{\rho\}) \\
&+ \vdots \\
&+ \{\lambda\}_k^T \left([M]\{u\}_k - [A]\{u\}_{k-1} - \{b\}_{k-1} - \{\rho\}\right) \\
&+ \vdots \\
&+ \{\lambda\}_n^T \left([M]\{u\}_n - [A]\{u\}_{n-1} - \{b\}_{n-1} - \{\rho\}\right)
\end{aligned}$$

Essentially, starting from known IC's, minimize the misfit with the TC's, $\{\delta\}$, in a GLS sense, by adjusting the unknown controls $\{\rho\}$. We are "shooting" at the TC's. If $\{\rho\}$ were known with certainty, the problem would be overdetermined.

The first-order necessary conditions are

$$\left\{\frac{\partial Z}{\partial \rho}\right\} = [W_\rho]\{\rho\} - \sum_{k=1}^{n} \{\lambda\}_k \tag{15.50}$$

$$\left\{\frac{\partial Z}{\partial \delta}\right\} = [W_\delta]\{\delta\} - \left\{\tilde{\lambda}\right\}_n \tag{15.51}$$

$$\left\{\frac{\partial Z}{\partial u}\right\}_0 = \left\{\tilde{\lambda}\right\}_0 - [A]^T \{\lambda\}_1 \tag{15.52}$$

$$\left\{\frac{\partial Z}{\partial u}\right\}_k = [M]^T \{\lambda\}_k - [A]^T \{\lambda\}_{k+1} \quad k = [1, n-1] \tag{15.53}$$

$$\left\{\frac{\partial Z}{\partial u}\right\}_n = [M]^T \{\lambda\}_n + \left\{\tilde{\lambda}\right\}_n \tag{15.54}$$

$$\left\{\frac{\partial Z}{\partial \lambda}\right\} = \textit{Forward Model} \tag{15.55}$$

The adjoint model is then

Initial Condition

$$-[A]^T\{\lambda\}_1 = -\left\{\tilde{\lambda}\right\}_0 \tag{15.56}$$

Dynamic

$$[M]^T\{\lambda\}_k - [A]^T\{\lambda\}_{k+1} = 0 \qquad k = [1, n-1] \tag{15.57}$$

Terminal Condition

$$[M]^T\{\lambda\}_n = -\left\{\tilde{\lambda}\right\}_n = -[W_\delta]\{\delta\} \tag{15.58}$$

and the relations between adjoint and forward model are

$$[W_\delta]\{\delta\} = \left\{\tilde{\lambda}\right\}_n \tag{15.59}$$

$$[W_\rho]\{\rho\} = \sum\left\{\tilde{\lambda}\right\}_k \tag{15.60}$$

Summing the adjoint equations we obtain a modification of (15.40)

$$\left([M]^T - [A]^T\right)\sum_k \{\lambda\}_k = -\left\{\tilde{\lambda}\right\}_0 - [W_\delta]\{\delta\} \tag{15.61}$$

Special Case #2: Agnostic ρ

Consider the special case where the variance of $\{\rho\}$ is infinite, *i.e.* there is no prior knowledge about it. In this case the above relations are valid with the simple proviso $[W_\rho] = 0$; we do not care about the size and shape of these variables. The most notable change in the result is the gradient expression

$$\left\{\frac{\partial Z}{\partial \rho}\right\} = -\sum_{k=1}^{n}\{\lambda\}_k \tag{15.62}$$

which would be used to direct a gradient descent iteration.

McGillicuddy *et al.* [85] used a shooting formulation with agnostic ρ to estimate population dynamics for a marine species based on observations of its spatial distribution. The forward model was a 2-D transient Advective-Diffusive FEM, with known parameters. An adjoint-based solution was used, with Conjugate Gradient-directed iterative descent in the solution space ρ.

Parameter Estimation

Suppose one or more of the parameters of the PDE are unknown precisely; then we have the problem of estimating them as well. As discussed above, this makes an otherwise linear problem *necessarily nonlinear*; there is the product of parameter and dependent variable to contend with. As a result, an iterative strategy is necessary. We assume familiarity with the previous discussion

of static problems in section 14.6; we will extend that discussion by example here in the dynamic case.

The basic extension needed is to introduce unknown parameters $\{y\}$ to the problem. The expression for the objective is now augmented by the extra regularization term

$$Z \to Z + \frac{1}{2} \{y\}^T [W_y] \{y\} \tag{15.63}$$

All of the previously defined gradient conditions remain unchanged, although we need to keep in mind that the matrices $[M]$ and $[A]$ are now dependent on $\{y\}$; therefore they change every time the parameter estimate changes. And, there is a new first-order condition to satisfy, the gradient with respect to the parameters. For the case studied here, we have

$$\left\{ \frac{\partial Z}{\partial y} \right\} = [W_y] \{y\} + \sum_{k=1}^{n} \left(\{\lambda\}_k^T \left[\frac{\partial M}{\partial y} \right] \{u\}_k - \{\lambda\}_k^T \left[\frac{\partial A}{\partial y} \right] \{u\}_{k-1} \right) \tag{15.64}$$

Now the gradient of a matrix is an unpleasant thing to need – a three-dimensional array. But we can take advantage of the FEM origin of $[M]$ and $[A]$ here, and avoid the explicit computation and storage of this gradient.

For simplicity, let's concentrate on the estimation of the first-order reaction parameter κ. The part of the matrices affected by κ are:

$$M_{ij} = \langle \kappa \phi_i \phi_j \rangle \theta \Delta t + \ldots \tag{15.65}$$

$$A_{ij} = -\langle \kappa \phi_i \phi_j \rangle (1-\theta) \Delta t + \ldots \tag{15.66}$$

Furthermore, assume that $\kappa(x,y)$ is expressed in a known scalar basis $\psi(x,y)$:

$$\kappa(x,y) = \sum_l \kappa^l \psi^l(x,y) \tag{15.67}$$

with parameters κ^l. (Here we use superscripts to isolate the parameter index.) The matrix gradients are

$$\frac{\partial M_{ij}}{\partial \kappa^l} = \langle \psi^l \phi_i \phi_j \rangle \theta \Delta t \tag{15.68}$$

$$\frac{\partial A_{ij}}{\partial \kappa^l} = -\langle \psi^l \phi_i \phi_j \rangle (1-\theta) \Delta t \tag{15.69}$$

If we express continuous fields $\overline{u}_k(x,y)$ and $\lambda_k(x,y)$ in the FEM basis $\phi(x,y)$:

$$\overline{u}_k(x,y) = \sum_j \overline{u}_{j,k} \phi_j(x,y) \tag{15.70}$$

$$= \sum_j \left(\theta u_{j,k} + (1-\theta) u_{j,k-1} \right) \phi_j(x,y) \tag{15.71}$$

$$\lambda_k(x,y) = \sum_i \lambda_{i,k} \phi_i(x,y) \tag{15.72}$$

Then the quadratic gradient forms may be assembled as follows:

$$\left\{ \frac{\partial Z}{\partial \kappa} \right\} = [W_\kappa] \{\kappa\} + \left\{ \sum_{k=1}^{n} \langle \psi(x,y) \lambda_k(x,y) \overline{u}_k(x,y) \rangle \Delta t \right\} \tag{15.73}$$

The vector nature of the second term is associated with the several bases ψ^l. The scalar components of the gradient may be easier to comprehend:

$$\frac{\partial Z}{\partial \kappa^l} = Regularization + \sum_{k=1}^{n} \langle \psi^l \lambda_k \overline{u}_k \rangle \Delta t \tag{15.74}$$

These are quantities which are readily assembled within FEM, assuming the current estimates of λ and u are available. Essentially, during the element loop to assemble $[M]$ and $[A]$, assemble the inner product of λ and $\overline{u}$, weighted with the functions ψ^l. The summation is over all time steps. The work involved is roughly the same as that for assembling the RHS for a single forward model run.

The reader is encouraged to revisit the discussion in section 14.6 where the static version of this problem is discussed in terms of a gradient descent iteration. That discussion is directly applicable to the dynamic case here. In particular, care must be exercised when there are Dirichlet or other conditions which distort the matrices $[M]$ and $[A]$ relative to the formulas presented here.

15.2 Hyperbolic Model: Telegraph Equation

Problem Statement

Here we consider a Hyperbolic PDE in one variable $u(x, y, t)$:

$$\frac{\partial^2 u}{\partial t^2} + \tau \frac{\partial u}{\partial t} - \nabla \cdot (gh \nabla u) = \beta(x, y, t) \tag{15.75}$$

We will consider the IC's, the forcing $\beta(x, y, t)$, and the parameters τ and gh to be known; and the BC's to be unknown. We assume data are available at unstructured locations in (x, y, t), and seek to minimize the misfit in a GLS sense. As usual we assume a prior estimate of the BC's and their covariance; and a prior estimate of the covariance of the model-data misfit.

The problem will be discretized on a conventional mesh of elements for (x, y), and with a conventional FD timestepping method. The result is a set of linear difference equations in the discrete variables $\{u\}_k$, with subscript k indicating a time level. Because of the second derivative in time, two initial conditions are required and assumed known with certainty.

Forward Model

Initial Condition

$$\{u\}_{-1} = \{\tilde{u}\}_{-1} \tag{15.76}$$

$$\{u\}_0 = \{\tilde{u}\}_0 \tag{15.77}$$

Dynamic

$$[M]\{u\}_{k+1} = [A]\{u\}_k + [B]\{u\}_{k-1} + [E_{k+1}]\{\rho\}_{k+1} + \{b\}_{k+1} \qquad k = [0, N-1] \tag{15.78}$$

Misfits

$$\{\delta\}_k = \left\{\tilde{d}\right\}_k - [T_k]\{u\}_k - [U_k]\{u\}_{k-1} \qquad k = [1, N] \tag{15.79}$$

For a conventional discretization using C^o elements, and a 3-level in time integration, implicit and centered with time parameter θ, we have:

$$
\begin{aligned}
M_{ij} &= \langle \phi_j \ \phi_i \rangle \left(1 + \frac{\tau \Delta t}{2}\right) + \frac{\theta \Delta t^2}{2} \langle gh \nabla \phi_j \cdot \nabla \phi_i \rangle \\
A_{ij} &= 2\langle \phi_j \ \phi_i \rangle - (1-\theta)\Delta t^2 \langle gh \nabla \phi_j \cdot \nabla \phi_i \rangle \\
B_{ij} &= \langle \phi_j \ \phi_i \rangle \left(-1 + \frac{\tau \Delta t}{2}\right) - \frac{\theta \Delta t^2}{2} \langle gh \nabla \phi_j \cdot \nabla \phi_i \rangle \\
b_i &= \langle \beta \phi_i \rangle + \oint gh \frac{\partial u}{\partial n} \phi_i \ ds
\end{aligned}
$$

where $\phi(x, y)$ is the basis for u; and $<>$ indicates integration over x, y; and $\oint \ ds$ is the boundary integral.

Table 15.2: FEM discretization of Wave equation (15.78).

Definitions

- $\{u\}_k$: Vector of unknowns at time k (State variables in forward model)
- $\{\tilde{u}\}_k$: IC's; *assumed known with certainty*
- $\left\{\tilde{d}\right\}_k$: Observations of u in time window $[k-1, \ k]$
- $\{\delta\}_k$: Model-data misfits in time window $[k-1, \ k]$
- $[T_k]$, $[U_k]$: sampling matrices; these interpolate the model output $\{u\}$ to the space-time observation points
- $\{\rho\}$: the vector of control variables (Dirichlet BC's).
- $[E_k]$: projects $\{\rho\}$ onto the simulation as Dirichlet BC's at time k
- $\{b\}_k$: Vector of known forcing at time k

The matrices for a typical FEM treatment of 15.78 are given in Table 15.2. Note that the BC control vector $\{\rho\}$ is not time-indexed here. In this formulation $\{\rho\}$ contains the temporal as well as spatial degrees of freedom in the BC's; and $[Cov(\rho)]$ is assumed to account for that. The selection of BC info in time as well as space is handled by $[E_k]$.

Optimal Fit: GLS Objective and First-Order Conditions

We seek the least squares fit:

$$
\text{Minimize } \frac{1}{2}\left\{ \{\rho\}^T [W_\rho] \{\rho\} + \sum_{k=1}^{N} \{\delta\}_k^T [W_\delta] \{\delta\}_k \right\}
$$

Introduce the Lagrange Multipliers $\left\{\tilde{\lambda}\right\}$, and $\{\lambda\}$, and minimize the functional Ω:

$$\begin{aligned}
\Omega &= \frac{1}{2}\left\{\{\rho\}^T [W_\rho] \{\rho\} + \sum_{k=1}^{N} \{\delta\}_k^T [W_\delta] \{\delta\}_k \right\} \\
&+ \sum_{k=1}^{N} \left\{\tilde{\lambda}\right\}_k^T \left([T_k] \{u\}_k + [U_k] \{u\}_{k-1} - \left\{\tilde{d}\right\}_k + \{\delta\}_k\right) \\
&+ \{\lambda\}_1^T \left([M] \{u\}_1 - [A] \{u\}_0 - [B] \{u\}_{-1} - [E_1] \{\rho\} - \{b\}_1\right) \\
&+ \{\lambda\}_2^T \left([M] \{u\}_2 - [A] \{u\}_1 - [B] \{u\}_0 - [E_2] \{\rho\} - \{b\}_2\right) \\
&+ \vdots \\
&+ \{\lambda\}_k^T \left([M] \{u\}_k - [A] \{u\}_{k-1} - [B] \{u\}_{k-2} - [E_k] \{\rho\} - \{b\}_k\right) \\
&+ \vdots \\
&+ \{\lambda\}_n^T \left([M] \{u\}_n - [A] \{u\}_{n-1} - [B] \{u\}_{n-2} - [E_n] \{\rho\} - \{b\}_n\right)
\end{aligned}$$

The gradient of Ω with respect to the Lagrange Multipliers recovers the forward model. The other gradient terms are:

$$\left\{\frac{\partial \Omega}{\partial u}\right\}_k = [M]^T \{\lambda\}_k - [A]^T \{\lambda\}_{k+1} - [B]^T \{\lambda\}_{k+2} + [T_k]^T \left\{\tilde{\lambda}\right\}_k + [U_{k+1}]^T \left\{\tilde{\lambda}\right\}_{k+1} \quad (15.80)$$

$$\left\{\frac{\partial \Omega}{\partial \delta}\right\}_k = [W_\delta] \{\delta\}_k + \left\{\tilde{\lambda}\right\}_k \quad (15.81)$$

$$\left\{\frac{\partial \Omega}{\partial \rho}\right\} = [W_\rho] \{\rho\} - \sum_{k=1}^{N} [E_k]^T \{\lambda\}_k \quad (15.82)$$

Setting the first two of these conditions to zero, and eliminating $\left\{\tilde{\lambda}\right\}$, gives the **Adjoint Model** for $k = [1, N]$:

$$\begin{aligned}
[M]^T \{\lambda\}_k - [A]^T \{\lambda\}_{k+1} - [B]^T \{\lambda\}_{k+2} &= [T_k]^T [W_\delta] \{\delta\}_k \\
&+ [U_{k+1}]^T [W_\delta] \{\delta\}_{k+1}
\end{aligned} \quad (15.83)$$

with **Terminal Conditions** for $k > N$

$$\{\delta\}_k, \{\lambda\}_k = 0 \quad (15.84)$$

It is interesting to note that the adjoint model is forced by the model-data misfit. No misfit, then the adjoint solution is the null one. Also, due to the homogeneous terminal conditions, the adjoint variables will be zero for all time following the last observation. For a Hyperbolic problem, forcing information propagates forward in time; in its adjoint, misfit information propagates backward in time.

The last gradient condition (equation 15.82) is enforced in different ways, depending on the method of solution. For iterative gradient descent algorithms, it gives the **direction of steepest descent**, and is set to zero by iteration. In that case it directs the recipe for improvement in the estimate for $\{\rho\}$. For direct solution by the method of representers, equation 15.82 is set to zero directly in order to compute $\{\rho\}$. Both methods are described below.

Gradient Descent Algorithms

We search for the minimum Ω in the parameter space ρ. From any given point $\{\rho\}_k$, there are 2 decisions: which direction to move in (the vector $\{\partial\rho\}$) and how far to go in that direction (σ, the scalar step size) before changing direction. With superscript l indicating iteration number, we have

$$\{\rho\}^{l+1} = \{\rho\}^l + \sigma\{\partial\rho\} \tag{15.85}$$

The **direction of most rapid decrease in** Ω is given by its negative gradient $-\{g\} \equiv -\left\{\frac{\partial\Omega}{\partial\rho}\right\}$. The steepest descent algorithm selects this direction:

$$\{\partial\rho\} = -\{g\} \tag{15.86}$$

and therefore

$$\{\rho\}^{l+1} = \{\rho\}^l - \sigma\{g\} \tag{15.87}$$

with still-arbitrary $\sigma > 0$. This or any other selection of $\{\partial\rho\}$ will produce new misfits[4]

$$\{\delta\}_k^{l+1} = \{\delta\}_k^l + \sigma\{\partial\delta\}_k \tag{15.88}$$

with $\{\delta\}_k^l$ the current misfit, at time level k, following iteration l; and $\{\partial\delta\}_k$ the change in the misfit achieved with $\sigma = 1$. In turn the objective function will become

$$\begin{aligned}\Omega^{l+1} = \Omega^l &+\sigma\left[\{\partial\rho\}^T[W_\rho]\{\rho\}^l + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\delta\}_k^l\right] \\ &+\frac{\sigma^2}{2}\left[\{\partial\rho\}^T[W_\rho]\{\partial\rho\} + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\partial\delta\}_k\right]\end{aligned} \tag{15.89}$$

The **optimal** σ is obtained by simple minimization:

$$\sigma = -\frac{\{\partial\rho\}^T[W_\rho]\{\rho\}^l + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\delta\}_k^l}{\{\partial\rho\}^T[W_\rho]\{\partial\rho\} + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\partial\delta\}_k} \tag{15.90}$$

Substitution of this value of σ gives the change in the objective function:

$$\Omega = \Omega^0 - \frac{\left(\{\partial\rho\}^T[W_\rho]\{\rho\}^l + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\delta\}_k^l\right)^2}{\{\partial\rho\}^T[W_\rho]\{\partial\rho\} + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\partial\delta\}_k} \tag{15.91}$$

which is obviously an improvement in Ω. Equivalently,

$$\Omega = \Omega^0 + \sigma\left(\{\partial\rho\}^T[W_\rho]\{\rho\}^l + \sum_k \{\partial\delta\}_k^T[W_\delta]\{\delta\}_k^l\right) \tag{15.92}$$

The steepest descent algorithm is summarized as follows:

- Given $\{\rho\}^l$, $\{\delta\}_k^l$, and the gradient $\left\{\frac{\partial\Omega}{\partial\rho}\right\}$, compute the direction of descent $\{\partial\rho\}$

[4]The nomenclature here can confuse; $\{\delta\}_k^l$ indicates a misfit at time k, as estimated during iteration l. During each iteration, all time levels k are recomputed.

- A single forward-model calculation with $\sigma = 1$ is sufficient to calculate $\{\partial\delta\}_k$
- Compute the optimal value of σ
- By superposition, update the new estimates $\{\rho\}^{l+1}$ and $\{\delta\}_k^{l+1}$

This completes the new, improved forward solution. It is followed by an adjoint solution forced by the new values of $\{\delta\}_k$, resulting in a new gradient $\left\{\frac{\partial\Omega}{\partial\rho}\right\}$, and the cycle continued to convergence. A suitable convergence rule is the achievement of vanishingly small values of the gradient, thereby satisfying the final first-order condition for an optimal solution. Note that only a single forward and adjoint solution is required per iteration.

Conjugate Gradient Descent

An alternative iterative strategy employs the same general machinery as the Steepest Descent method, except that the direction in each iteration is selected differently. We identify a sequence of gradients $\{g\}^l \equiv \left\{\frac{\partial\Omega}{\partial\rho}\right\}^l$ and directions $\{h\}^l$, with l indicating iteration number.[5] The gradients are computed from the first-order condition as above. The direction is computed as a blend of the current gradient and the previous direction:

$$\{h\}^{l+1} = -\{g\}^{l+1} + \gamma\{h\}^l \tag{15.93}$$

$$\gamma = \frac{\left(\{g\}^{l+1} - \{g\}^l\right)\cdot\{g\}^{l+1}}{\{g\}^l\cdot\{g\}^l} \tag{15.94}$$

$\{\partial\rho\}$ is then computed as an increment of arbitrary length in this direction,

$$\{\partial\rho\}^{l+1} = \{h\}^{l+1} \tag{15.95}$$

The iteration is otherwise the same as steepest descent, in particular the procedure for computing the optimal step size σ once the direction is known. The method reduces to the steepest descent method if γ is arbitrarily set to zero. Initially, $\{h\}^0 = -\{g\}^0$ is sufficient to get things started.

Lynch and Hannah [68] solved a generalized hyperbolic system of this form, using the adjoint approach with Conjugate Gradient descent, in order to fit an ocean circulation model to velocity data.

Solution by Representers

This is a direct (non-iterative) solution strategy. The idea is to compute impulse responses of the adjoint/forward system, each forced by a *unit misfit* at a single measurement point in space-time. With m observations, there will be m impulse responses. The solution is then synthesized as a linear combination of these "representer" solutions. Each representer is computed non-iteratively as follows:

- Beginning with a single unit misfit, compute the adjoint solution $\{\lambda\}$

[5]The notation here is taken from Press *et al.* [99], Chapter 10.6, with the sign of g reversed.

- Compute the boundary conditions $\{\rho\}$ from the gradient condition (equation 15.82) $\left\{\frac{\partial\Omega}{\partial\rho}\right\} = 0$:[6]

$$[W_\rho]\{\rho\} = \sum_{k=1}^{N} [E_k]^T \{\lambda\}_k \tag{15.96}$$

Since $[W_\rho] = [Cov(\rho)]^{-1}$, then this calculation is conveniently restated:

$$\{\rho\} = [Cov(\rho)] \sum_{k=1}^{N} [E_k]^T \{\lambda\}_k \tag{15.97}$$

and the inversion of the Covariance matrix is not needed.

- Compute a forward solution forced by $\{\rho\}$ and sample and record the model response $\{u\}$ at all the space-time observation points (equation 15.79). Call this vector of sampled $\{u\}$, the *representer* $\{r\}_j$; it is the forward model response, sampled at all the data points, to a unit misfit at data point j.

- Assemble the misfit relation by superposition of the representers:

$$\{\delta\} = \{\tilde{d}\} - [R]\{\delta\} \tag{15.98}$$

and finally,

$$[I + R]\{\delta\} = \{\tilde{d}\} \tag{15.99}$$

with $[R]$ the representer matrix, with columns $\{r\}_j$

This equation may be solved directly and finally for the misfits $\{\delta\}$. Once these are computed, a single adjoint/forward run completes the calculation. With m observations, there are m adjoint/forward solutions required to form the representer matrix $[R]$; an inversion of the $m \times m$ system for $\{\delta\}$; and a final adjoint/forward solution. The total number of equivalent forward model solutions is therefore $2m + 2$. If m is small compared to n (the state space of the system) then this is very attractive. (Recall that Conjugate Gradient Descent scales with n.)

Notes:

- Here the vectors $\{\delta\}$ and $\left\{\tilde{d}\right\}$ are the concatenation of their counterparts in specific time windows $\{\delta\}_k$ and $\left\{\tilde{d}\right\}_k$.

- Each column of $[R]$ is a single representer, projecting the influence of a single unit misfit onto the forward model solution at the observation points.

- The representers are independent of the data themselves; they depend only on their space-time locations. In cases where the observational program is known ahead of time, they may be pre-computed. In addition they contain information essential to the evaluation of an experimental design.

- There is an obvious computational advantage if m is small; and conversely, a computational penalty for redundant sampling.

[6]Note that this condition is enforced directly here; in the gradient descent algorithms it is satisfied only after the iterations have converged.

15.3 Regularization

Almost any inverse PDE problem rapidly grows in the number of unknowns, if for no other reason than the process of increasing resolution. One result is that the number of unknowns being estimated is likely always to overwhelm the number of data available. So, relative to model data misfit, we can expect to have many nearly equivalent inverse solutions. *Regularization* is the general term for injecting additional considerations to discriminate among them. In effect, we either penalize unlikely candidates in the GLS sense, based on their statistical improbability; or we eliminate them entirely, *a priori*, by restricting the form of solutions considered. Generally, a blend of these strategies is used, serially.

Reduction of the DoF's

The first step is to avoid creating a flood of inverse Degrees of Freedom (DoF's) in the first place. Remember that it is always necessary to obtain prior estimates of the mean and covariance among all the DoF's; and that *more* does not always lead to a *better* solution, but it does necessarily lead to more work and slower estimation. It is far less work to solve a problem with a few DoF's, than to solve it with many more which are strongly correlated. The solutions should be materially the same, but the work of getting there is not.

Here one needs to be creative, using physical insight into the system being modeled. Some strategies for reducing the number of DoF's:

- Reduce the spatial basis of the forcing. For FEM fields, this amounts to using simpler bases for the forcing than for the dependent variable; generally, using appropriate resolution for each field, by tayloring the FEM basis to it.
- Express the forcing in non-FEM bases: *e.g.* , the first few terms of a spatial Fourier Series for boundary conditions [75]; or the first few EOF's of observed variability.
- Use coarse temporal resolution (large Δt) for timeseries of forcing which are known to vary more slowly than the system response might allow [68]. Effectively this creates temporal smoothing *a priori.*
- Use a time-harmonic representation for variables which are known to be periodic, or when the periodic steady-state is of interest. This greatly reduces the number of degrees of freedom from of order 20 (enough to discretize a sine wave) to 2 (amplitude and phase) assuming the frequency is known [75].
- Use the convolution of a measured surrogate timeseries, with lags of undetermined size, to represent an unknown timeseries which is known to be correlated. The sizes of the lag coefficients become the unknowns. This was used in [74, 60]; observed wind timeseries at lags of 0, 3, and 6 hours were a useful surrogate for oceanic pressure boundary conditions. The result was the reduction of temporal degrees of freedom to about 6% of baseline; more rapid convergence of a Conjugate Gradient iteration; and run-time reduction by a factor of 5-10.
- The inversion DoF's need not be limited to single fields, nor is space-time separation necessary. There is general idea of specifying *Features* – dynamical space-time, multi-field entities which approximate modes of the solution. EOF's (Empirical Orthogonal Functions) are Features

obtained via statistical analysis. Features may also be identified and specified by less formal means, and may in fact represent solutions to simplified forms of the governing equations [34, 56].

Naturally, all of these strategies require considerable physical insight about the solution and dynamics under investigation. There is no substitute for this; the alternative is brute force which will almost always produce slow and costly results of limited value.

The Weight Matrix

Once the inversion DoF's are specified, the mathematics takes over through the GLS regularization (weight) matrices $[W]$. Since $[W_x] \equiv [Cov(x)]^{-1}$, we need to either express the covariance among the DoF's, or construct $[W]$ directly on heuristic or "acceptability" arguments. The specification of these matrices is substantive and challenging.

The derivation of covariance among variables is beyond our scope. If the data are available to support that, it is surely desirable. If data are available from a suitable simulation, that may suffice as well. In practice it is common to assume (or derive) a closed-form model of variability based on pseudo-distance among points in the same scalar or vector field, as for example described previously in the section on "noise models" (section 13.1). Normally, analytic distance-based covariance models lead to completely full covariance matrices; these may be truncated by setting tiny covariances among distant points to zero; however the matrix $[W] = [Cov(x)]^{-1}$ will still be completely full.

Absent these approaches, one normally resorts to heuristic arguments about expected size, shape, and/or smoothness as qualities of acceptable fields. These arguments are quite common, hence the meaningful labels "weight matrices", "weighted least squares", etc. For example, a baseline practice is to simply weight the size of the variance of a variable $[x]$ by its expected variance σ^2:

$$[W_x] = \frac{1}{\sigma^2}[\ I\ ] \tag{15.100}$$

This is equivalent to the assumption of uniform, independent (uncorrelated) variation among the x_i, each with variance σ^2:

$$[Cov(x)] = [W_x]^{-1} = \sigma^2[\ I\ ] \tag{15.101}$$

This discriminates against large x but expresses no concern for smoothness, perhaps in the case where the selection of the DoF's has taken care of that. At least, this specification takes care of units! Further decoration of this idea is common and becomes increasingly problem-specific.

It is easy to automate this practice with standard FEM apparatus, as described below.

Heuristic Specification of $[W]$ using FEM

Suppose a field f to be estimated is expressed in a basis $\phi_i(x, y)$:

$$f(x, y) = \sum_{j=1}^{n} f_i \phi_i(x, y) \tag{15.102}$$

We might be concerned with its mean size and/or slope:

$$\frac{1}{A}\int\int f^2 dx\ dy = \frac{1}{A}\sum_{i=1}^{n}\sum_{j=1}^{n} u_i \langle \nabla\phi_i \nabla\phi_j \rangle u_j \tag{15.103}$$

$$= \frac{1}{A}\{f\}^T [\langle \phi_i \phi_j \rangle]\{f\} \tag{15.104}$$

$$\frac{1}{A}\int\int (\nabla f)^2 dx\ dy = \frac{1}{A}\sum_{i=1}^{n}\sum_{j=1}^{n} f_i \langle \nabla\phi_i \nabla\phi_j \rangle f_j \tag{15.105}$$

$$= \frac{1}{A}\{f\}^T [\langle \nabla\phi_i \nabla\phi_j \rangle]\{f\} \tag{15.106}$$

where A is the domain area, $A = \int\int dx\ dy = \langle 1 \rangle$. So we could construct a heuristic weighting matrix comprising a weighted blend of mean size and slope:

$$w_0 \int\int f^2 dx\ dy + w_1 \int\int (\nabla f)^2 dx\ dy = w_0 \{f\}^T [\langle \phi_i \phi_j \rangle]\{f\} \tag{15.107}$$

$$+ \ w_1 \{f\}^T [\langle \nabla\phi_i \nabla\phi_j \rangle]\{f\} \tag{15.108}$$

and the quadratic form would have weight matrix

$$[W_f] = [w_0 \langle \phi_i \phi_j \rangle + w_1 \langle \nabla\phi_i \nabla\phi_j \rangle] \tag{15.109}$$

Similarly for a boundary field $g(s) = \sum g_i \phi_i(s)$, the weight matrix

$$[W_g] = \left[w_0 \oint \phi_i \phi_j ds + w_1 \oint \frac{\partial\phi_i}{\partial s}\frac{\partial\phi_j}{\partial s} ds \right] \tag{15.110}$$

would penalize mean size and smoothness of $g(s)$ on the boundary. These matrices are easily assembled using standard FEM technology. In [68] this type of regularization is used to estimate oceanic boundary conditions.

15.4 Example: Nonlinear Inversion

Here we illustrate by example the construction of a nonlinear inverse from linearized inverses, iteratively. In the literature this is referred to variously as the "Incremental" or the "Tangent Linear" approach, the latter usually referring specifically to the use of linearized adjoint models. We proceed intuitively. The works involved include [70, 75, 68, 74, 89, 61, 59].

The physical context is hindcasting oceanic motion, specifically the motion occurring over the continental shelf. There exists a canonical set of 3-D nonlinear dynamical equations (mixed, hyperbolic and parabolic; a specialization of the general Navier-Stokes and advective-diffusion equations) and several well-known approaches to their discrete forward solution. That used here is the "Quoddy" model, [70]. It requires IC's and BC's which are imperfectly known. Therefore the problem posed is to deduce these, and the resultant motion, from measurements. In our case, the measurements include fluid velocity and density inside an artificially bounded domain; the boundaries are mainly ones of convenience but not natural in the mathematical sense. So estimation of BC's and IC's is a prime issue. By comparison with these, local meteorological forcing (local wind and heat flux) are reasonably well-known from direct observation.

The principal motions can be separated scales of time variability and motion, corresponding to primary physical forcing:

- tide: $V \sim 1$ m/s, $\tau \sim 3$ hours
- wind: $V \sim .10 - .50$ m/s, $\tau \sim 1 - 5$ days
- baroclinicity: $V \sim .10 - .20$ m/s, $\tau \sim 10 - 30$ days

Isolation of the tidal motion is further simplified by the fact that the motions are periodic with known frequencies, with persistent amplitudes and phases which vary spatially and possibly on long (baroclinic) timescales.

A principal nonlinearity in the forward (Quoddy) model is the turbulence and stratification. Both affect the effective eddy viscosity in the vertical, which affects all motions; and in addition the stratification creates internally-driven baroclinic motions in itself. Therefore, the linearized inverses are based on mean values of these features, estimated from the latest run of Quoddy.

Figure 15.1 is a schematic of such an inverse, from [68]. It is designed to estimate BC's in the wind band; Saco is a linearized forward model (linearized Quoddy); Moody is an exact adjoint of Saco. Both require the linearized representation of stratification and turbulence as indicated. Gradient Descent iteration is used to solve these models. Basic input is interior velocity data which are otherwise unexplained; output is a set of BC's which minimize (in GLS sense) the Saco misfit with same. The union of Saco and Moody constitutes the linearized inverse Casco for the wind-band motions. Essentially, Casco operates in the time domain; it seeks wind-band motions by restricting the BC variability to a) appropriately long time steps; b) a convolution of the observed local wind timeseries with proper smoothing; and/or c) by heuristic regularization, penalizing the square of $\frac{\partial BC}{\partial t}$.

The Casco model may be nested in a larger iteration in order to deal with the actual nonlinearities in the forward model. This is illustrated in Figure 15.2. The premise of this arrangement is that the linearization of the turbulence-dependent terms is valid and captures the dominant nonlinearities. There are other nonlinearities, of course; the hypothesis here is based on physical intuition, that the dominant nonlinearity affecting the wind-band motions can be captured in this iteration. The question of valid linearization remains theoretically open in this and any other application.

Figure 15.3 is a condensation of Figure 15.2. Added is a second linearized inverse, Truxton [75], arranged in series with Casco in each overall iteration. Truxton inverts the Quoddy misfits first, seeking fast tidé-band signals at pre-determined temporal frequencies. It is a direct GLS inversion, with degrees of freedom corresponding to harmonic amplitudes and phases for the various tidal frequencies. Because of this, the linearized forward model involved is an elliptic (Helmholtz) equation which permits direct solution easily. Regularization is achieved by restricting the spatial bases *a priori*, or by GLS weighting which discriminates against size and slope of the BC's. This arrangement was used in the studies [89, 74].

The serial arrangement of Casco and Truxton can be described as "detiding" followed by "dewinding" of the same misfit signal. The two models are sensitive to separate time-variabilities, and experience proves them to be effectively orthogonal to each other. Specifically, the Casco inversion produces essentially the same result whether or not it is preceded by Truxton. By design, the two inverse solution spaces are practically independent.

In the present hierarchy, Baroclinic variability is the slowest; it is expressed most clearly in the density data. The studies cited here are focused on a temporal window of 5-10 days. Accordingly, these data are "assimilated" as Initial Conditions only. Figure 15.4 illustrates this, using an Objective Analysis or Statistical Interpolation procedure (see Chapter 18). Longer Quoddy simulations would need to compute misfits with these data at later times, and properly invert them.

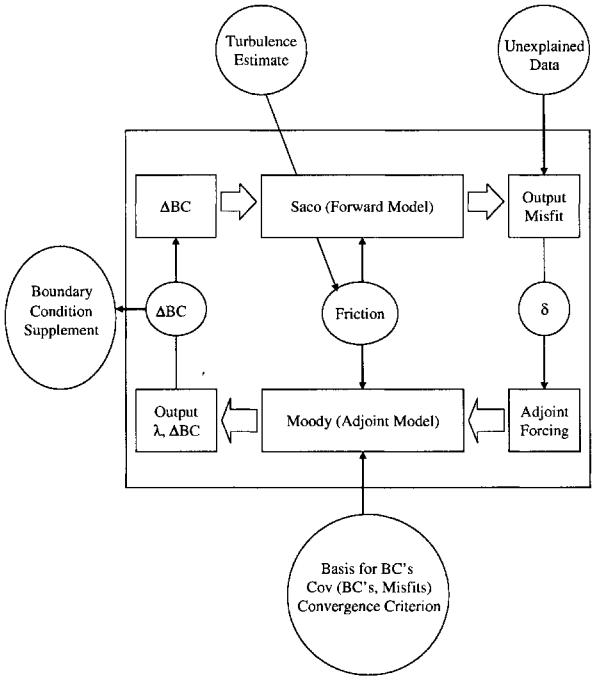


Figure 15.1: Iteration to solve Forward (Saco) and Adjoint (Moody) models. via gradient descent. Control variables are limited to the barotropic boundary conditions. The large square box encloses the linear inverse model Casco. As in [68]

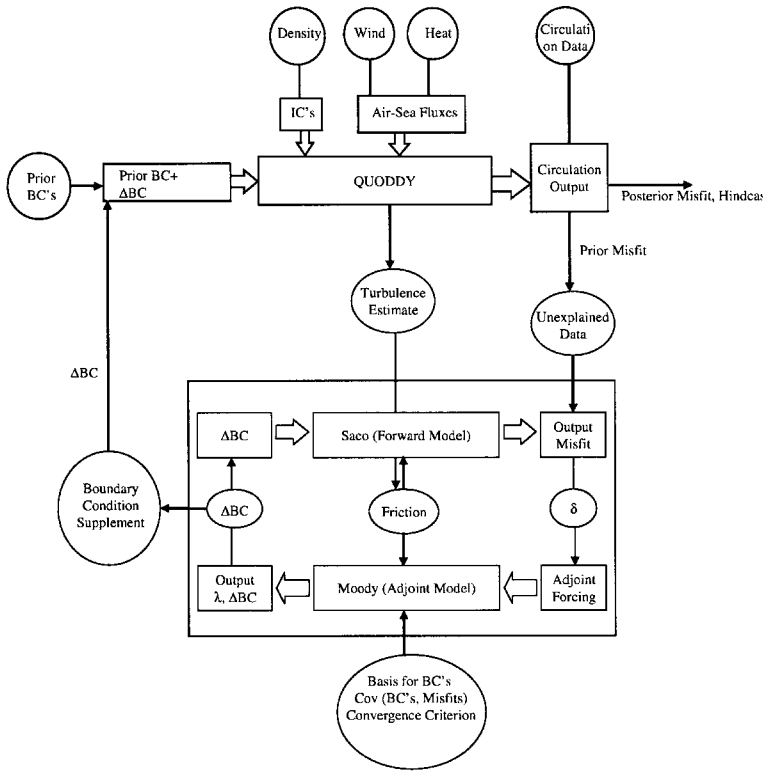


Figure 15.2: Grand iteration to solve nonlinear Forward (Quoddy) and linearized Inverse (Casco) models. Saco is a linearized version of Quoddy; Moody is its exact adjoint. Casco is the gradient descent solution of Saco/Moody, enclosed in the rectangular box as in 15.1. The turbulence estimate is updated in each iteration from the latest Quoddy run. Control variables are limited to the barotropic boundary conditions, reflecting prior physical reasoning about the principal unknowns affecting wind-band motions.

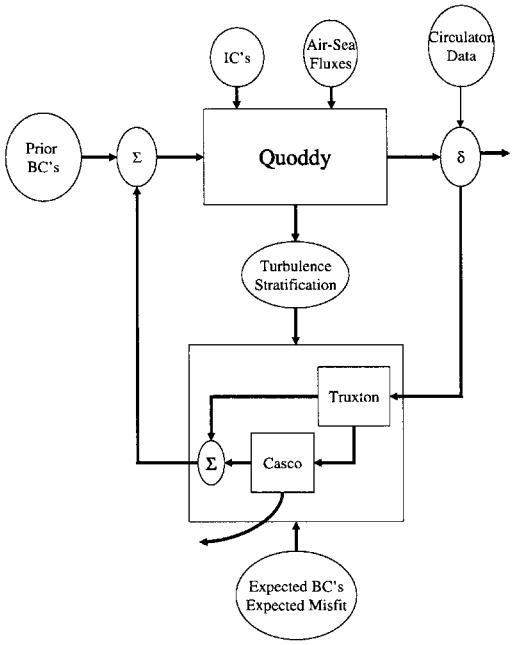


Figure 15.3: Flowchart of iterative nonlinear inversion and sequential arrangement of the nonlinear forward model (Quoddy) and the two linearized inverses, Truxton and Casco. As in [74]

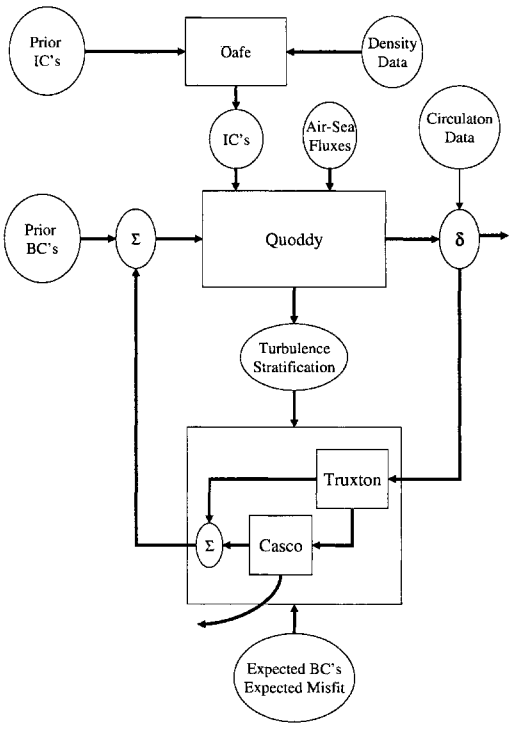


Figure 15.4: Assimilation of density data by OA (the Oafe module) to IC's. Casco, Truxton, Quoddy, Oafe in a big network.

Chapter 16

Time Conventions for Real-Time Assimilation

16.1 Time

To begin with, we distinguish two relevant "times". The first is the *Time of Occurrence* of a natural event. The second is the *Time of Availability i.e.* the time at which an observation of that event is recorded, processed, and resident in a secure data server with instant access. Availability here means available to a data assimilative modeling system, on demand. Figure 16.1 shows these graphically. All data and data products are located on this plot by their times of occurrence and availability.

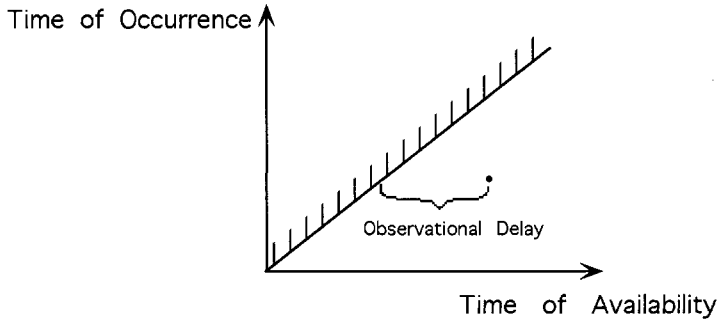


Figure 16.1: Timing diagram relating the time at which an event occurs in nature, versus the time at which information about that occurrence is available.

16.2 Observational Data

By necessity, all observational data lie on or below the 45° line. Instantaneously available observations would lie on the line; practical observations of necessity involve a nonzero *Observational Delay.* Contributors to this delay include instrument response time and, more importantly, averaging intervals.

Figure 16.2 plots a simple timeseries where each individual datum is available serially, as soon

as it is observed. Figure 16.3 shows a batch process wherein data is published less frequently than it is observed. This introduces an additional *Publication Delay* related to the batch size.

The process of becoming available thus involves time delays for observation and publication. The distinction is only meaningful when significant publication delays are necessitated by a slow network connecting measurement sources with the assimilative modeling system. For example, a mooring which computes an hourly average and transmits data once a day would have a 0.5 hour observational delay and a 24 hour publication delay.

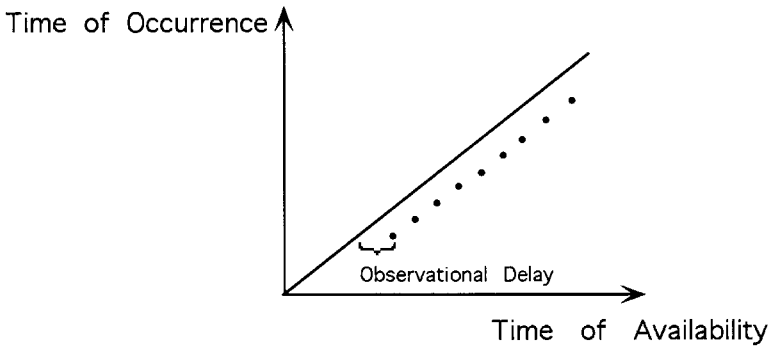


Figure 16.2: Timeseries published one point at a time.

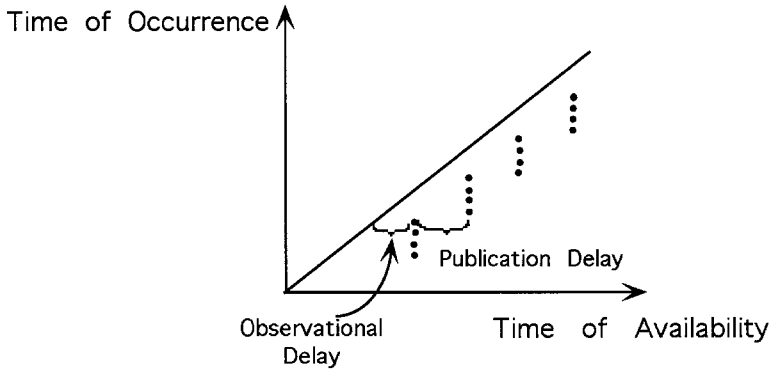


Figure 16.3: Timeseries published in batch mode; a publication delay is introduced.

16.3 Simulation Data Products

Outputs of simulation models constitute a second type of data product. These data are synthetic, constituting an alternative estimate of the state of the ocean. The same time conventions are relevant. In this case the 45° line divides *Forecast* from *Hindcast* as illustrated in Figure 16.4.

Figure 16.5 shows a single assimilative simulation. This simulation is launched at the *Bell Time* t_b, and is complete and published after an *Assimilation Delay*. The hindcast period terminates at $t = t_b$. The forecast period is from there onward. The assimilation delay has two consequences:

a) Data which occur during the assimilation delay are not assimilated. This includes the very latest observations as well as older observations which were delayed in publication.

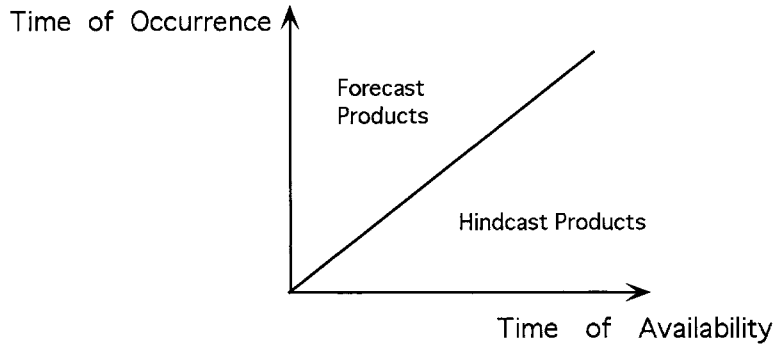


Figure 16.4: Simulation data products address both historical (hindcast) and future (forecast) events.

b) At the time of its publication, the forecast is already old, and 'forecast' extends backwards on the occurrence (vertical) axis to t_b.

A variant is presented in Figure 16.6 wherein simulation results are published incrementally during a simulation, rather than in a single batch at the end of a simulation. This decreases the assimilation delay for the early results. Also, data availability for the later results is increased, since in this "just-in-time" model we use data which occurs after the bell.

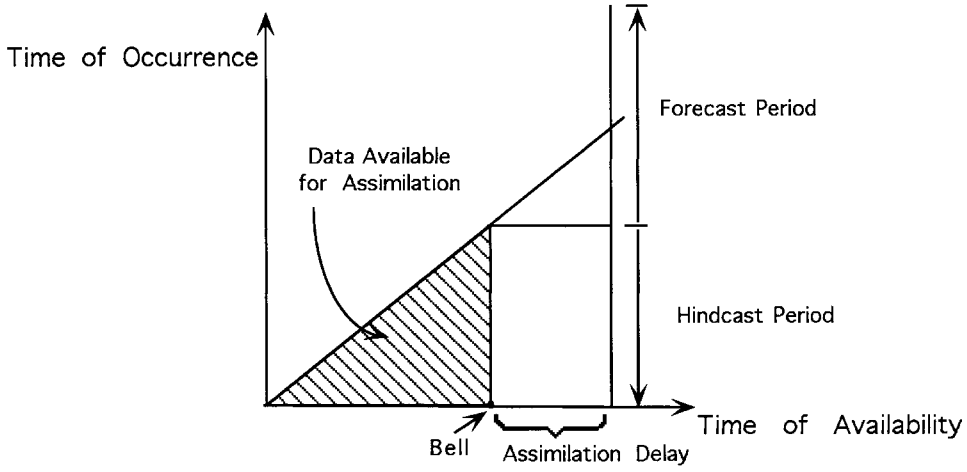


Figure 16.5: Operational timing. A hindcast/forecast product is initiated at the bell time and published following an assimilation delay.

16.4 Sequential Simulation

Figure 16.7 illustrates a sequence of two data-assimilative simulations, separated by a *Bell Interval.* The first simulation constitutes the *Best Prior Estimate*, or BPE, for initializing and forcing the second. The second simulation is fitted to the most recent data. The memory of the earlier data is entrusted to the BPE. Note that some late-availability data goes unused; and depending on the details of the data assimilation method, a nonzero *Initialization Period* may be required in the second simulation. This Figure applies directly to the more general case of a sequence of

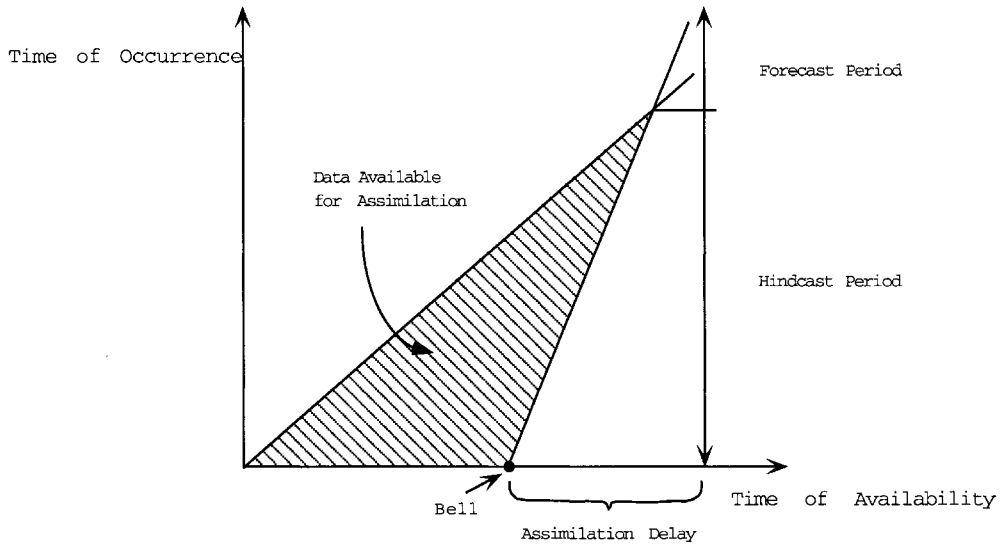


Figure 16.6: Just-in-time version of Figure 16.5. Calculations are published sequentially during the simulation.

data-assimilative simulations, at regular bell intervals.

16.5 What Time Is It?

Geophysical-scale phenomena place serious demands on the space-time location of things. Both observations and model calculations have to be put into a consistent coordinate system for any comparison to be meaningful. Fortunately we are in the realm of classical phenomena, where time and space are well-defined on planetary scales. Appeal to those standards is very important.

In particular, the issue of precise time-registration of both observational and computational data products is critical. As soon as more than one source of information is presented, the issue of time-stamping becomes critical. Appeal to Greenwich time, with the Gregorian standard for $t = 0$, is routine and recommended. Care has to be taken to avoid the buildup of round-off by keeping integer and real parts of "time", and there are a series of bookkeeping considerations stemming from that. Related is the variable length of individual months and years. In the Appendix we present one example of a little library and convention set which deals with this.

Another issue in timing relates to periodic phenomena – tides – which are routinely reported in terms of amplitude and phase. Because accumulation of time errors due to finite frequency can be serious, it is crucial to invoke standards for tidal timing – notably the conventions for phase, frequency, and start time. The Appendix contains one instance of standards and software for this second critical aspect of timing, affecting both observational and simulation data products.

16.6 Example: R-T Operations, Cruise EL 9904

In April 1999 the first operational forecast system for Georges Bank was deployed at sea, aboard R/V Edwin Link. Georges Bank is a topographic feature with length scale 200 km, on the US-Canadian border at the edge of the continental shelf. The system involved an atmospheric forecast

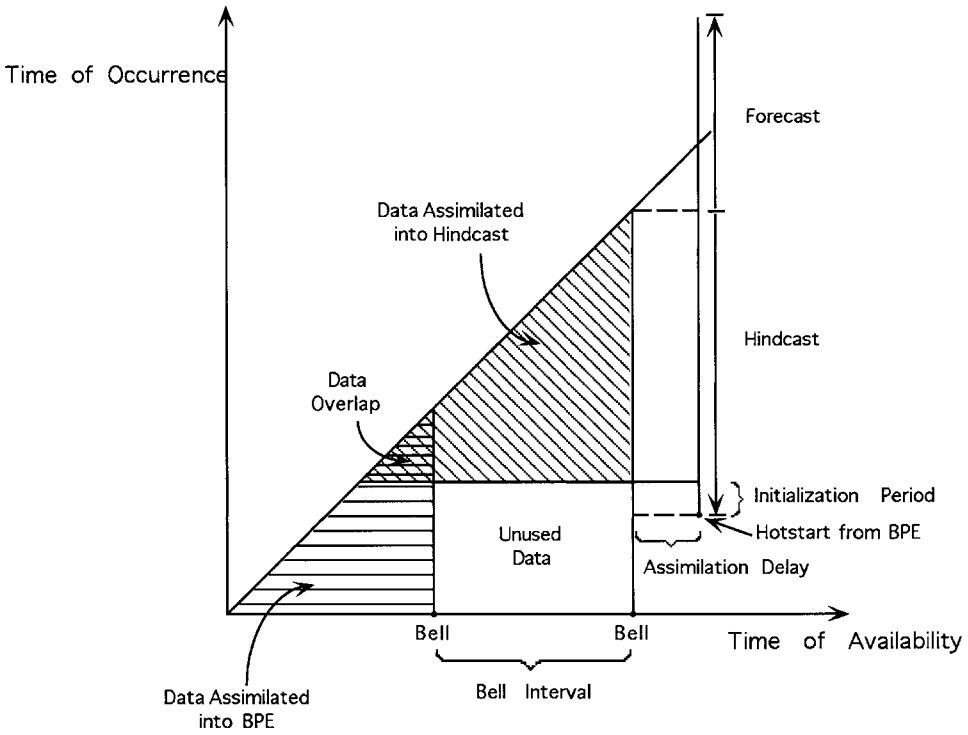


Figure 16.7: A sequence of hindcast/forecast simulations.

and two marine forecasts, and was operated once a day. The *atmospheric* forecast was taken from modeled winds, as web-published by the National Center for Environmental Prediction. This product included a three-day forecast period subject to a 4-hour assimilation delay, preceded by a data-assimilative hindcast. This was used to drive a far-field *oceanic* hindcast/forecast, consisting of a barotropic wind+tide calculation on a wide-area finite element mesh covering roughly half of the Atlantic basin. The purpose of this was to provide the wind-band pressure forcing in the vicinity of Georges Bank (GB). This product was then used to force the boundary conditions on the *Georges Bank* near-field calculation, using a limited-area, high-resolution mesh. Additional forcing to this final calculation included observed/forecast local wind and heat flux plus tides. The near field calculation was initialized with observed hydrography and assimilated velocity observations from drifters on the bank.

The timeline achieved in this initial operation is illustrated in figure 16.8. Delays associated with assimilation and publication of the various products are indicated. Overall the delay from atmospheric bell to GB forecast was 25 hours, partitioned as follows:

- Assimilation: 9.5 hours
- Elective Assimilative Delay: 6 hours
- Publication: 9.5 hours

The assimilation delay is largely the sum of the three computational times. In the short run this may be presumed to be at an irreducible minimum. The publication delay represents several complications related to the network and the distribution of effort. The far-field oceanic calculation was performed at a shore station; the GB calculation was performed at sea. Shore-to-ship communications and the initial procedure for procuring the atmospheric data products imposed important limitations and contributed most of the publication delay. The Elective Assimilative delay occurred

between receipt of the oceanic forecast on ship, and the GB bell. It represents burn-in of the assimilation procedure, tuning of archival strategies, manual examination of incoming data streams, and various scientific investigations. Much of this elective delay can be eliminated with further automation of the data feeds and experience. This results in a minimum overall delay of 9.5 hours from atmospheric bell to GB forecast publication. It is interesting that if the oceanic forecast were not needed, the delay between shipboard observations and forecast could be reduced to 3.5 hours.

More details of this and subsequent real-time exercises are described in [59].

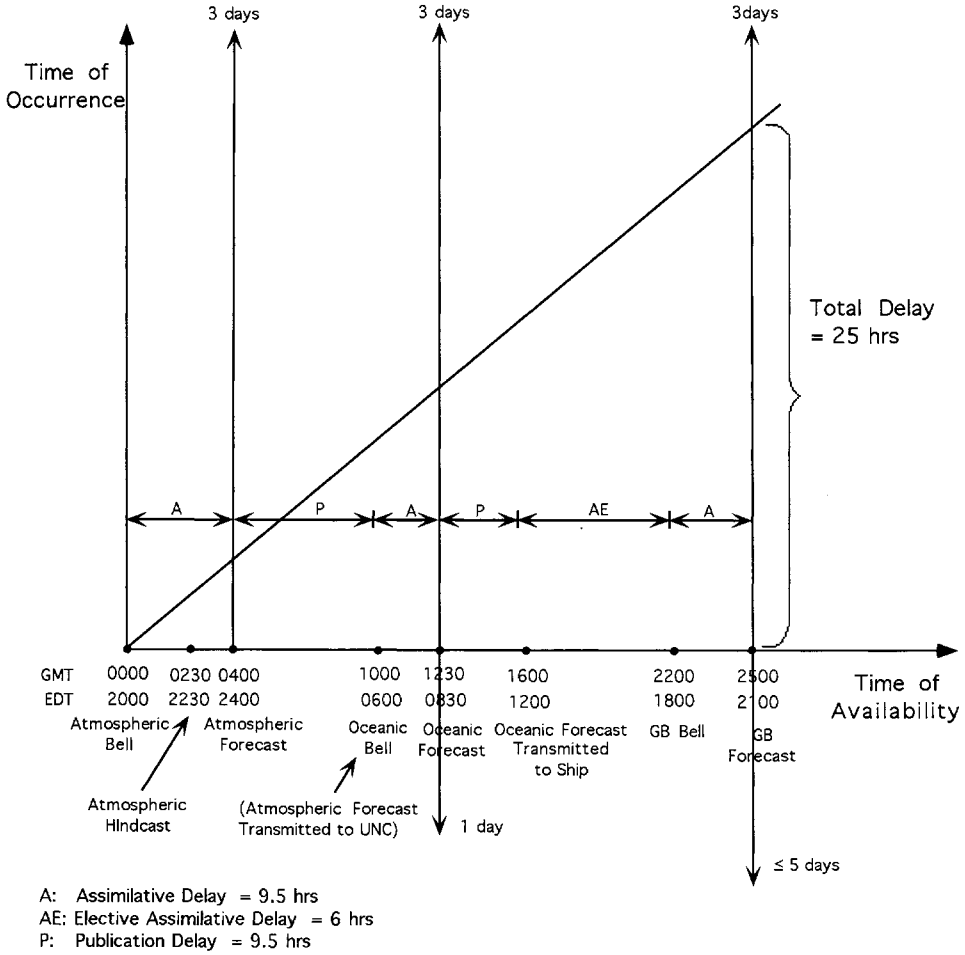


Figure 16.8: Timeline for cruise EL9904, the first Georges Bank at-sea assimilation exercise in real time. From [59]

Chapter 17

Skill Assessment for Data Assimilative Models

In this chapter we describe vocabulary for assessment of skill, as used generally in [68, 74, 89]. Then we present an example drawn from [68].

17.1 Vocabulary

Forward and Inverse Models

Forward Model: simulates nature with a fixed set of discretized differential equations. Any model has a set of necessary and sufficient inputs which are required to produce a simulation: parameters, initial conditions, boundary conditions, and forcing functions e.g. atmospheric and river forcing. These are referred to as *Essential Data.*

Inverse Model: a procedure for estimating missing or imprecise essential data by fitting a forward model to observations of other, non-essential data. There are multiple levels:

- Parameter Estimation
- Initial Condition Estimation
- Boundary Condition Estimation
- Forcing Function Estimation

The quantities being estimated are called the *Control Variables.* These comprise a subset of the Essential Data.

Model Identification deals with selecting among various candidate forward models which have different mathematical structure, typically by comparing model performance with observations. The formal version of this utilizes inversion.

Truth, Data, Prediction

Truth is that which we measure and simulate. In nature, Truth can only be estimated. But its existence and uniqueness is not questioned here.

Estimate: an approximation to Truth. Estimates can be based on observations, simulations, or simply prior opinion; or a composite of these.

Data = Truth + Observational Error, at given space-time observation points. Only a statistical description of Observational Error is meaningful. Data are either active or passive in terms of their role in inversion.

Active Data are used in inversion, *i.e.* the model is required to fit these data.

Passive Data are not used in inversion; they are available only post-inversion. Since the model is not required to fit these data, they provide an approximate measure of prediction error.

Data Product: published information formed from observation, simulation, or other method.

Prediction is a Model-based estimate of truth. It is defined and known perfectly everywhere.

Prior: without the benefit of data
Posterior: with the benefit of data
Note that "prediction" does not imply "future" here. It is meaningful to predict past events.

Notice that in the special case of an Observational System Simulation Experiment (OSSE, see example below), the only Truth is a virtual truth, *i.e.* a model-generated product. This is known without error or uncertainty, everywhere. Data in this case is synthetic; a Truth sample plus a fabricated measurement error. Prediction Error for OSSE's is known perfectly because Truth is known.

Skill

Prediction Error = Truth - Prediction, defined everywhere. In nature it is unknowable; it must be estimated as it refers to Truth.

Skill is good if Prediction = Truth i.e. Prediction Error = 0, statistically. The scale of this zero is set by a) the observational error, and b) the model's fidelity to Truth processes. Four Skill distinctions are useful:

Interpolation: Prediction among the data, spatially
Extrapolation: Prediction beyond the data, spatially
Hindcast: Prediction within the time window of the data
Forecast: Prediction beyond the time window of the data

Misfit = Data - Prediction. Misfit is defined only at the observation points; it is the sum of Observational and Prediction Errors.

Fit is good if Misfit = 0 statistically; if models were perfect, Misfit = Observational Error.

Overfitting drives the Misfit below the Observational Error, fitting noise at the expense of Skill.

Accuracy/Bias, Precision/Noise

Skill can be further decomposed. Let the observations d be expressed as the sum of Truth $\overline{d}$ plus a random component $\widehat{d}$ representing observational error.

$$d = \overline{d} + \widehat{d} \tag{17.1}$$

$\overline{d}$ is the ensemble mean over many measurements; $\hat{d}$ has zero mean. Linear inverse models produce a prediction P which is linear in the data:

$$P = [L]\, d = [L]\, \overline{d} + [L]\, \hat{d} \tag{17.2}$$

Any prediction will be the superposition of the inverse truth $\overline{P} = [L]\, \overline{d}$ and one realization of the inverse noise $\hat{P} = [L]\, \hat{d}$. The Prediction Error E_P would then be

$$E_P = T - P = (T - \overline{P}) - \hat{P} \tag{17.3}$$

and in the ensemble mean,

$$\overline{E}_P = (T - \overline{P}) \tag{17.4}$$

Agreement between $\overline{P}$ and Truth is a measure of *accuracy*. Inaccuracy amounts to a *bias* in the inversion – a mean preference for something other than Truth. The inverse noise $\hat{P}$ is a measure of the *precision* of the prediction in the face of observational error. An ensemble of possible $\hat{P}$ exists; only the statistics of this ensemble are meaningful. Although these definitions are strictly valid for linear Predictions, we incorporate them into our vocabulary:

Accuracy concerns the agreement between the ensemble mean Prediction and Truth. (The ensemble here is all the possible realizations of the Observational Error, for a given Prediction procedure.) Inaccuracy or *Bias* is quantified by the ensemble mean Prediction Error $\overline{E}_P$.

Precision concerns the variance of the Prediction as caused by the ensemble of Observational Errors. This is the *inverse noise* $\hat{P}$.

17.2 Observational System Simulation Experiments: Example

Lynch and Hannah [68] studied inversion of 2-D hydrodynamical data to obtain circulation estimates in an idealized, rectangular segment of the coastal ocean (figure 17.1). Truth (figure 17.2) was a simulation; this allowed perfect knowledge of the truth everywhere. Data were obtained by sampling the truth at six locations, without error. Subsequently the data were contaminated by adding different types of error and the effect of the error on the inversion studied. This constitutes an Observational System Simulation Experiment (OSSE) – truth and data are simulated with complete experimental control. Further, the inverse model was identical to that which generated the truth. So, a perfect inverse is possible, *i.e.* one which exactly reproduces the truth everywhere, not just the data. OSSE's with this characteristic are referred to as "Twin Experiments". Control variables were pressure BC's; data were velocity observations. Model physics represented linear, hyperbolic oceanic fluid mechanics. The experiment focused on the steady-state model response following a brief startup transient. The inverse method was adjoint-based with gradient descent. Regularization terms involved the size and smoothness of the boundary conditions. A Monte Carlo approach was used, with data errors simulated via an ensemble of data noise vectors.

The "Inverse Truth" is illustrated below (figures 17.3). This is the inverse solution forced by perfect data. It demonstrates interpolative skill, but limited extrapolation skill. Figure 17.4 is a map of error or bias – the discrepancy between Inverse Truth and the real Truth. The bias is negligible in the interpolation zone among the data locations; it grows as one extrapolates away from the data, as the regularization terms become dominant. Especially important in this case is the fact that boundary controls along the bottom boundary cause only very small motions at the observation

points; hence the regularization controls their estimation. In this OSSE the regularization favors no pressure variation there, hence flow parallel to that boundary. This phenomena is a manifestation of the physics of the system.

The inverse noise map for this inversion is illustrated in Figure 17.5. This was obtained by inverting an ensemble of data vectors containing statistical noise only (zero mean); and then compiling a map of RMS speed across the ensemble. As can be seen, an ensemble size of about 50 was sufficient. Any real inversion of this linear system will produce the superposition of the inverse truth (with its bias) and one realization of the inverse noise.

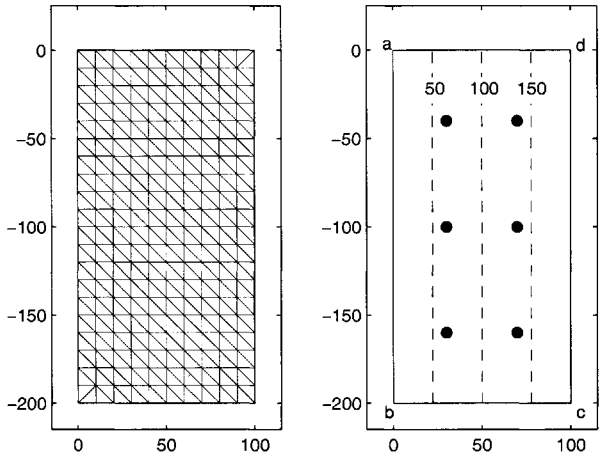


Figure 17.1: Test case geometry. Left: FE mesh. Right: bathymetry (dash contours) and data locations (dots). From [68].

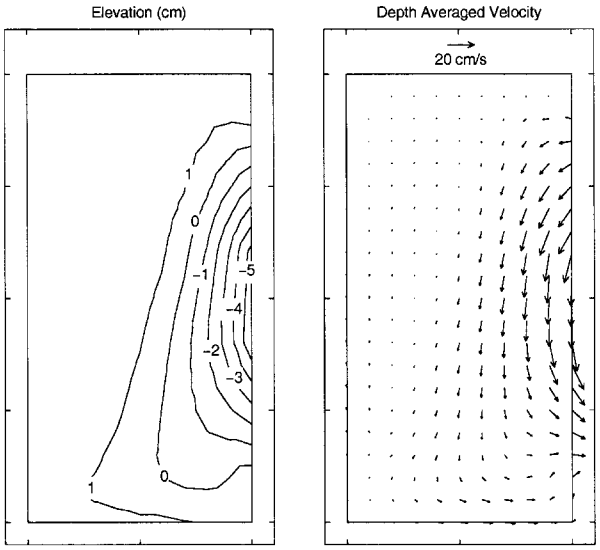


Figure 17.2: Steady-state Truth. From [68].

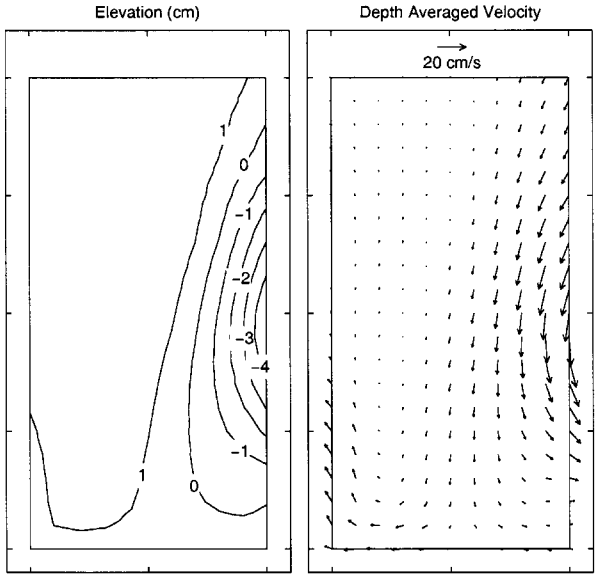


Figure 17.3: Inverse Truth. This results from inverting *perfect data*, *i.e.* truth sampled with no observational error. From [68].

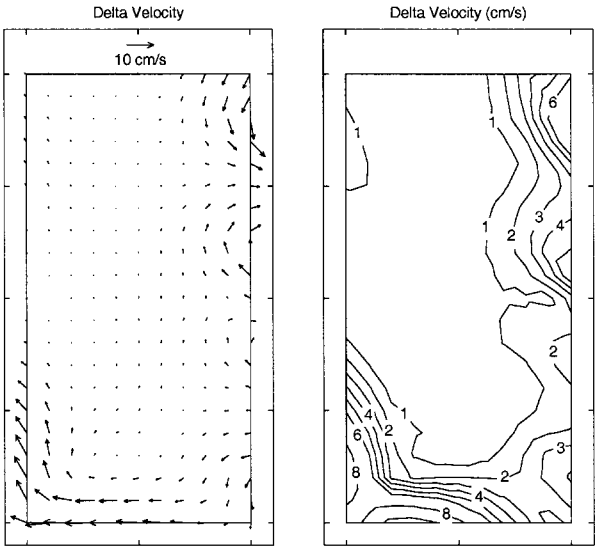


Figure 17.4: Inverse Error: the difference between the inverse solution (figure 17.3) and truth (figure 17.2). This is a map of inaccuracy or bias. The bias is introduced by the regularization which prefers small, smooth solutions; it controls the inversion where the influence of the data is weak. From [68].

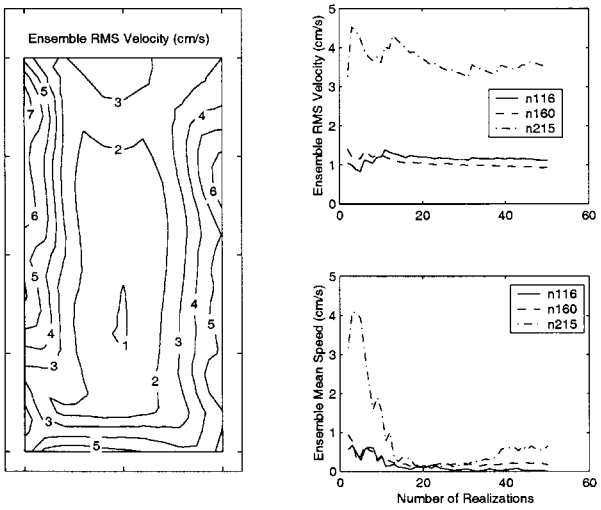


Figure 17.5: Inverse noise, *i.e.* inverse model response to data noise only. Left: map of ensemble RMS speed. Right: RMS (top) and mean (bottom) speeds at selected nodes, versus ensemble size. From [68].

Chapter 18

Statistical Interpolation

In this chapter we address the interpolation of data. It is assumed that the data are imperfect and that therefore a good interpolant should be immune to the peculiarities of the data error. Also, it is assumed that the data are sparse and leave a lot unmeasured. So we have to assume the availability of a statistical model of the data errors; and combine that with a statistical model of the variability of the field being estimated; and make a good balance of these in an average sense. The field of Statistical Interpolation addresses this topic in general; it is highly developed. (See *e.g.* Liebelt [57], Cressie [27], Journel and Huijbregts [50], Daley, [28], Bras and Rodriguez-Iturbe [18], Christakos [24].) Our objective here is to introduce basic notions, and then to develop a credible argument for the classic Gauss-Markov theorem. This theorem is the cornerstone of the ideas presented here. Approaches based on this theorem are variously known as Objective Analysis (OA) and Optimal Interpolation (OI) in some communities; and as Kriging and its variants in others. All rely on a statistical description of the quantities being interpolated, estimated, predicted, or mapped. For present purposes these words all mean the same practical thing.

We illustrate the use of stochastic differential equation (SDE) solutions in developing the necessary knowledge of the field covariance. The relation to GLS minimization of model-data misfit is developed.

Overall, an effective FEM strategy for Statistical Interpolation is demonstrated.

18.1 Introduction: Point Estimation

To fix ideas, imagine that we have a scalar physical variable being measured simultaneously by two instruments. Both instruments introduce measurement error. The two measurement errors have distinct statistical descriptions. The relations among data d, truth μ, and measurement error ϵ are

$$d_1 = \mu + \epsilon_1 \tag{18.1}$$

$$d_2 = \mu + \epsilon_2 \tag{18.2}$$

and we want an estimate of μ. Call this estimate u, and assume that u is to be a linear combination of the data:

$$u = b_1 d_1 + b_2 d_2 \tag{18.3}$$

This is a *linear estimator*. The *estimation error* ϵ_e is the difference between u and truth μ:

$$\epsilon_e \equiv u - \mu = (b_1 + b_2 - 1)\mu + b_1\epsilon_1 + b_2\epsilon_2 \tag{18.4}$$

Now we invent an ensemble of such data, all statistically the same. (The measurement errors are all from the same distribution). The mean of all such estimates is

$$\overline{u} - \overline{\mu} = (b_1 + b_2 - 1)\overline{\mu} + b_1\overline{\epsilon}_1 + b_2\overline{\epsilon}_2 \tag{18.5}$$

with the overbar indicating the ensemble average. Some elementary points:

- We are allowing natural variability, hence $\overline{\mu}$ is the mean state of nature.
- We will assume that the measurement errors ϵ_i have zero mean, $\overline{\epsilon}_i = 0$. If not, we are assuming that the data and errors are mathematically demeaned prior to this analysis.
- The *unbiased* case requires the constraint $b_1 + b_2 = 1$. Otherwise the average estimate is not equal to the average truth.

The variance of the estimation error, in the unbiased case, is the expected or mean squared departure from truth:

$$\overline{(u-\mu)^2} = b_1^2\overline{\epsilon_1^2} + b_2^2\overline{\epsilon_2^2} + 2b_1b_2\overline{\epsilon_1\epsilon_2} \tag{18.6}$$

Suppose we minimize the variance subject to the constraint $b_1 + b_2 = 1$

$$Min\left\{Z = b_1^2\overline{\epsilon_1^2} + b_2^2\overline{\epsilon_2^2} + 2b_1b_2\overline{\epsilon_1\epsilon_2} + \lambda\left(b_1 + b_2 - 1\right)\right\} \tag{18.7}$$

The first-order conditions for a minimum are:

$$\frac{\partial Z}{\partial b_1} = 2b_1\overline{\epsilon_1^2} + 2b_2\overline{\epsilon_1\epsilon_2} + \lambda = 0 \tag{18.8}$$

$$\frac{\partial Z}{\partial b_2} = 2b_1\overline{\epsilon_1\epsilon_2} + 2b_2\overline{\epsilon_2^2} + \lambda = 0 \tag{18.9}$$

$$\frac{\partial Z}{\partial \lambda} = b_1 + b_2 - 1 = 0 \tag{18.10}$$

The solution is

$$u = \frac{\left(\overline{\epsilon_2^2} - \overline{\epsilon_1\epsilon_2}\right)d_1 + \left(\overline{\epsilon_1^2} - \overline{\epsilon_1\epsilon_2}\right)d_2}{\overline{\epsilon_1^2} + \overline{\epsilon_2^2} - 2\overline{\epsilon_1\epsilon_2}} \tag{18.11}$$

This is the *Best Linear Unbiased Estimator* (BLUE) in this context. The simplest case is that of uncorrelated measurement errors, $\overline{\epsilon_1\epsilon_2} = 0$. The BLUE is intuitively sensible here, with higher weight going to the datum with the smaller error.

If we abandon the requirement of no bias, we have

$$\begin{aligned}\overline{(u-\mu)^2} &= \beta^2\overline{\mu^2} + b_1^2\overline{\epsilon_1^2} + b_2^2\overline{\epsilon_2^2} \\ &+ 2b_1b_2\overline{\epsilon_1\epsilon_2} + 2\beta b_1\overline{\mu\epsilon_1} + 2\beta b_2\overline{\mu\epsilon_2}\end{aligned} \tag{18.12}$$

where $\beta \equiv b_1 + b_2 - 1$. Notice the cross-correlation terms among errors, $\overline{\epsilon_1\epsilon_2}$ and between error and truth, $\overline{\mu\epsilon_i}$. We will ignore the latter assuming $\overline{\mu\epsilon_i} = 0$. This is common practice. However, this asserts, among other things, that large signals μ are not accompanied by large errors, etc. An error model of the form $\epsilon = \alpha\mu$, *i.e.* the error is proportional to the truth, *i.e.* truth = data $\pm$ a fixed percentage, is ruled out. Beware of glossing over these little details at the outset.

The first-order conditions in this case are

$$\left(\overline{\epsilon_1^2} + \overline{\mu^2}\right)b_1 + \left(\overline{\epsilon_1\epsilon_2} + \overline{\mu^2}\right)b_2 = \overline{\mu^2} \tag{18.13}$$

$$\left(\overline{\epsilon_1\epsilon_2} + \overline{\mu^2}\right)b_1 + \left(\overline{\epsilon_2^2} + \overline{\mu^2}\right)b_2 = \overline{\mu^2} \tag{18.14}$$

and the solution is

$$u = \frac{\left(\overline{\epsilon_2^2} - \overline{\epsilon_1\epsilon_2}\right) d_1 + \left(\overline{\epsilon_1^2} - \overline{\epsilon_1\epsilon_2}\right) d_2}{\left[\overline{\epsilon_1^2} + \overline{\epsilon_2^2} - 2\overline{\epsilon_1\epsilon_2}\right] + \frac{1}{\mu^2}\left[\overline{\epsilon_1^2}\,\overline{\epsilon_2^2} - (\overline{\epsilon_1\epsilon_2})^2\right]} \tag{18.15}$$

Note the simplest case $\overline{\epsilon_1\epsilon_2} = 0$. By comparison with equation 18.11 it is clear that this estimator has a bias toward underprediction. So here, if we allow a biased estimator, we can expect a smaller mean squared error than the BLUE.

18.2 Interpolation and the Gauss-Markov Theorem

Next, add spatial variations to the truth, $\mu = \mu(x, y)$; and assume that the two measurements d_1 and d_2 are at different points $(x, y)_1$ and $(x, y)_2$:

$$d_1 = \mu_1 + \epsilon_1 \tag{18.16}$$
$$d_2 = \mu_2 + \epsilon_2 \tag{18.17}$$

The field estimate wanted is in general at a third spatial point $(x, y)_k$:

$$u_k = b_{k,1}d_1 + b_{k,2}d_2 \tag{18.18}$$

The variance of the Estimation Error $\epsilon_e = u - \mu$ is

$$\overline{(u_k - \mu_k)^2} = \overline{(b_{k,1}d_1 + b_{k,2}d_2 - \mu_k)^2} \tag{18.19}$$
$$= b_{k,1}^2\overline{d_1^2} + b_{k,2}^2\overline{d_2^2} + \mu_k^2 + 2b_{k,1}b_{k,2}\overline{d_1 d_2} - 2b_{k,1}\overline{d_1\mu_k} - 2b_{k,2}\overline{d_2\mu_k} \tag{18.20}$$

The First Order Conditions, with no constraints on $b_{k,i}$ concerning the estimation bias, are:

$$\overline{d_1 d_1}\; b_{k,1} + \overline{d_1 d_2}\; b_{k,2} = \overline{d_1\mu_k} \tag{18.21}$$
$$\overline{d_2 d_1}\; b_{k,1} + \overline{d_2 d_2}\; b_{k,2} = \overline{d_2\mu_k} \tag{18.22}$$

and the solution is

$$\left\{ \begin{array}{c} b_{k,1} \\ b_{k,2} \end{array} \right\} = \left[\begin{array}{cc} \overline{d_1 d_1} & \overline{d_1 d_2} \\ \overline{d_2 d_1} & \overline{d_2 d_2} \end{array} \right]^{-1} \left\{ \begin{array}{c} \overline{d_1\mu_k} \\ \overline{d_2\mu_k} \end{array} \right\} \tag{18.23}$$

This gives the vector of estimator coefficients for a single estimation point k, based on observations at the two data points 1 and 2. If we add a third datum, the system of first-order conditions extends to

$$\overline{d_1 d_1}\; b_{k,1} + \overline{d_1 d_2}\; b_{k,2} + \overline{d_1 d_3}\; b_{k,3} = \overline{d_1\mu_k} \tag{18.24}$$
$$\overline{d_2 d_1}\; b_{k,1} + \overline{d_2 d_2}\; b_{k,2} + \overline{d_2 d_3}\; b_{k,3} = \overline{d_2\mu_k} \tag{18.25}$$
$$\overline{d_3 d_1}\; b_{k,1} + \overline{d_3 d_2}\; b_{k,2} + \overline{d_3 d_3}\; b_{k,3} = \overline{d_3\mu_k} \tag{18.26}$$

$$\left\{ \begin{array}{c} b_{k,1} \\ b_{k,2} \\ b_{k,3} \end{array} \right\} = \left[\begin{array}{ccc} \overline{d_1 d_1} & \overline{d_1 d_2} & \overline{d_1 d_3} \\ \overline{d_2 d_1} & \overline{d_2 d_2} & \overline{d_2 d_3} \\ \overline{d_3 d_1} & \overline{d_3 d_2} & \overline{d_3 d_3} \end{array} \right]^{-1} \left\{ \begin{array}{c} \overline{d_1\mu_k} \\ \overline{d_2\mu_k} \\ \overline{d_3\mu_k} \end{array} \right\} \tag{18.27}$$

and so on for an arbitrary collection of data:

$$\{b\}_k = \left[\overline{\{d\}\,\{d\}^T}\right]^{-1} \overline{\{d\}\,\mu_k} \tag{18.28}$$

The vector $\{b\}_k$ contains the interpolation weights to be assigned to the various data, for estimating u_k:

$$u_k = \{b\}_k^T \{d\} \tag{18.29}$$

This easily generalizes to several estimation points, $k = 1, K$; in that case we have a matrix of coefficients $[B]$, whose *rows* are the vectors $\{b\}_k$:

$$[B] = \left(\left[\overline{\{d\}\{d\}^T} \right]^{-1} \left[\overline{\{d\}\{\mu\}^T} \right] \right)^T = \left[\overline{\{\mu\}\{d\}^T} \right] \left[\overline{\{d\}\{d\}^T} \right]^{-1} \tag{18.30}$$

such that the estimator is

$$\{u\} = [B]\{d\} \tag{18.31}$$

Here we need to introduce a more compact notation for covariance matrices:

$$[C_{ab}] \equiv \left[\overline{\{a\}\{b\}^T} \right] \tag{18.32}$$

Using this convention, the estimation matrix $[B]$ is

$$[B] = [C_{\mu d}][C_{dd}]^{-1} \tag{18.33}$$

and the estimate is

$$\{u\} = [B]\{d\} = [C_{\mu d}][C_{dd}]^{-1}\{d\} \tag{18.34}$$

It is useful to pause here and examine these matrices and vectors. The underlying field $\mu(x, y)$ is continuous in space. $\{d\}$ is a vector of M different data; $\{u\}$ is a vector of K estimates of μ, each at a different location. The covariances involved are among these $M + K$ items and are finite matrices. In particular the covariances involving μ address the truth values at the K estimation points; those involving data address the M specific data at hand. $[C_{\mu d}]$ has dimension $K \times M$; $[C_{\mu\mu}]$ is $K \times K$; $[C_{dd}]$, $M \times M$; etc. Everything is finite, determined by the observation points and the estimation points.

For present purposes, equations 18.33 and 18.34 are the bottom line of our simple demonstration. They prescribe a statistically optimal (minimum variance) estimation for the true field $\mu(x, y)$ based on linear combinations of the data. The coefficients of $[B]$ invoke prior knowledge of the covariance of the data with itself and with the truth. Rigorous development of this result is found elsewhere under several guises; we refer to its proof as the *Gauss-Markov Theorem*, and to the use of equation 18.34 for estimating fields from data as Gauss-Markov Estimation. The estimation problem may be viewed as the statistical interpolation of data, and this procedure is frequently referred to as Optimal Interpolation (OI). The phrase Objective Analysis (OA) is frequently associated with the process of Gauss-Markov field estimation. An additional key result is the variability of the estimation error, $\epsilon_e \equiv u - \mu$:

$$[C_{\epsilon_e \epsilon_e}] = [C_{\mu\mu}] - [B][C_{d\mu}] \tag{18.35}$$

The diagonals of this matrix are the expected squared Estimation Error[1]; their sum is minimized compared with any other selection of $[B]$ – *i.e.* the Gauss-Markov estimator is a minimum variance estimator. Finally, the *bias* of the estimate,

$$\{\overline{u - \mu}\} = [B]\left\{\overline{d}\right\} - \{\overline{\mu}\} \tag{18.36}$$

[1]Notice that here, ϵ indicates a measurement error, defined at the measurement points; while ϵ_e is the estimation error, defined at the estimation points. As always, *error* indicates discrepancy with truth.

is generally nonzero. If μ and d have zero means, then the estimate is also unbiased and therefore it is the BLUE. It is desirable to deal with de-meaned data and fields to achieve this.

It is common to go one practical step further. Since $\{d\} = \{\mu\} + \{\epsilon\}$, we have

$$[C_{\mu d}] = [C_{\mu\mu}] + [C_{\mu\epsilon}] \tag{18.37}$$
$$[C_{dd}] = [C_{\mu\mu}] + [C_{\epsilon\epsilon}] + [C_{\mu\epsilon}] + [C_{\epsilon\mu}] \tag{18.38}$$

If we assume that measurement error is not correlated with truth:

$$[C_{\mu\epsilon}] = 0 \tag{18.39}$$

then we have a final form of 18.33 and 18.34:

$$[C_{dd}] = [C_{\mu\mu}] + [C_{\epsilon\epsilon}] \qquad [C_{\mu d}] = [C_{\mu\mu}] \tag{18.40}$$

$$[B] = [C_{\mu\mu}] \left[[C_{\mu\mu}] + [C_{\epsilon\epsilon}] \right]^{-1} \tag{18.41}$$

and finally, the optimal estimate is

$$\{u\} = [B]\{d\} = [C_{\mu\mu}] \left[[C_{\mu\mu}] + [C_{\epsilon\epsilon}] \right]^{-1} \{d\} \tag{18.42}$$

18.3 Interpolating and Sampling Finite Fields

In this section we revisit some of the terminology and ideas examined in section 14.1 concerning model error, measurement error, sampling, and the model-data misfit. It is a good idea to first review what we covered there.

Suppose we introduce interpolating bases $\phi_k(x, y)$ among the estimation points, such that the estimated field is continuous:

$$u(x, y) = \sum_{k=1}^{K} u_k \phi_k(x, y) \tag{18.43}$$

Naturally, we are anticipating here the use of the FEM bases ϕ_k and nodal values u_k; with the estimation points the nodes of a FE grid. We will refer to the field $u(x, y)$ as the *model field.* Now use the same bases for the true field $\mu(x, y)$:

$$\mu(x, y) = \sum_{k=1}^{K} \mu_k \phi_k(x, y) + \epsilon_s(x, y) \tag{18.44}$$

where $\epsilon_s(x, y)$ is the part of the truth which is outside the basis. It is variously the *subgrid-scale variability* (hence the subscript); the truncation error; or the unresolvable truth. Assuming $\epsilon_s(x, y)$ is unknowable, we have a finite truth vector $\{\mu\}$ and its estimate $\{u\}$.

Data are imperfect linear samples of truth, with observational error $\{\epsilon_d\}$ added. For an individual datum d_k we have a linear sampling operator L_k:

$$\begin{aligned} d_k &= L_k(\mu) + \epsilon_{d,k} && (18.45)\\ &= L_k\Big(\sum_j \mu_j \phi_j(x,y) + \epsilon_s\Big) + \epsilon_{d,k} && (18.46)\\ &= \sum_j \mu_j L_k(\phi_j(x,y)) + L_k(\epsilon_s) + \epsilon_{d,k} && (18.47) \end{aligned}$$

Therefore the data vector is expressed as a matrix equation

$$\{d\} = [S]\{\mu\} + \{\epsilon\} \tag{18.48}$$

with sampling matrix $[S]$ comprising the scalar entries

$$S_{k,j} = L_k(\phi_j(x,y)) \tag{18.49}$$

i.e. $S_{k,j}$ is the k^{th} sample of the j^{th} basis function. Notice that $\{\epsilon\}$ is a combination of observational error and a sampling of the unresolvable, subgrid truth:

$$\{\epsilon\} = \{L(\epsilon_s)\} + \{\epsilon_d\} \tag{18.50}$$

Sampled, unresolved truth is indistinguishable from measurement error in this context. Their sum $\{\epsilon\}$ is referred to here as *Sampling Error*, to distinguish it from measurement error *per se*.

We may proceed from here as above. Assuming we are minimizing the variance of the estimation error at the node points, $\{\epsilon_e\} \equiv \{u\} - \{\mu\}$, then the previous development is unchanged, leading to the same result for the estimate

$$\{u\} = [B]\{d\} = [C_{\mu d}][C_{dd}]^{-1}\{d\} \tag{18.51}$$

and its precision

$$[C_{\epsilon_e \epsilon_e}] = [C_{\mu\mu}] - [B][C_{d\mu}] \tag{18.52}$$

Now the final step needs to recognize the effect of the sampling:

$$\{d\}\{d\}^T = \Big[[S]\{\mu\} + \{\epsilon\}\Big]\Big[[S]\{\mu\} + \{\epsilon\}\Big]^T \tag{18.53}$$

$$[C_{dd}] = [S][C_{\mu\mu}][S]^T + [C_{\epsilon\epsilon}] + \Big([C_{\epsilon\mu}][S]^T + [S][C_{\mu\epsilon}]\Big) \tag{18.54}$$

Similarly,

$$\{\mu\}\{d\}^T = \{\mu\}\Big[[S]\{\mu\} + \{\epsilon\}\Big]^T \tag{18.55}$$

$$[C_{\mu d}] = [C_{\mu\mu}][S]^T + [C_{\mu\epsilon}] \tag{18.56}$$

With the usual assumption that $[\mu]$ and $[\epsilon]$ are uncorrelated, we have

$$\begin{aligned} [C_{dd}] &= [S][C_{\mu\mu}][S]^T + [C_{\epsilon\epsilon}] && (18.57)\\ [C_{\mu d}] &= [C_{\mu\mu}][S]^T && (18.58)\\ [C_{d\mu}] &= [S][C_{\mu\mu}] && (18.59) \end{aligned}$$

Finally the Gauss-Markov estimator is:

$$[B] = \left[[C_{\mu\mu}][S]^T\right]\left[[S][C_{\mu\mu}][S]^T + [C_{\epsilon\epsilon}]\right]^{-1} \tag{18.60}$$

with the estimate

$$\{u\} = [B]\{d\} = \left[[C_{\mu\mu}][S]^T\right]\left[[S][C_{\mu\mu}][S]^T + [C_{\epsilon\epsilon}]\right]^{-1}\{d\} \tag{18.61}$$

and precision

$$[C_{\epsilon_e\epsilon_e}] = [C_{\mu\mu}] - [B][S][C_{\mu\mu}] \tag{18.62}$$

The key from this point on is the estimation of the Covariance among the truth $\{\mu\}$ and among the "noise" $\{\epsilon\}$.

An alternative expression for the Gauss-Markov estimator $[B]$ is obtained by application of the matrix identity 18.103. The result:

$$[B] = \left[[S]^T[C_{\epsilon\epsilon}]^{-1}[S] + [C_{\mu\mu}]^{-1}\right]^{-1}[S]^T[C_{\epsilon\epsilon}]^{-1} \tag{18.63}$$

In the present context this is a less appealing form due to the multiple inverses; later it will be used and we note it here.

Aside, notice that the model-data misfit now is a sampled version of equation 18.48, based on the estimate $\{u\}$:

$$\begin{aligned}\{\delta\} \equiv \{d\} - [S]\{u\} &= [S]\{\mu\} + \{\epsilon\} - [S]\{u\} &(18.64)\\ &= [S]\{\mu - u\} + \{\epsilon\} &(18.65)\\ &= [S]\{\mu - u\} + \{L(\epsilon_s)\} + \{\epsilon_d\} &(18.66)\end{aligned}$$

$\{\delta\}$ is an estimate of $\{\epsilon\}$, which is unknowable except statistically. Included in $\{\delta\}$ are a sample of the estimation error $\{\mu - u\}$, a sample of the unresolved field, plus measurement error. These are three different things; computation will inevitably focus on the first, since the latter two are known only in a statistical sense. This illustrates the importance of not reducing the misfit below the expected level of the latter two terms combined; that would be overfitting, chasing noise. Essentially, statistical zero is set by this noise level – the combination of sampled subgrid-scale variability and measurement error. Correct specification of $[C_{\epsilon\epsilon}]$ is essential here. In the absence of formal statistical descriptions of noise covariance among instruments and locations, one relies on the intuition expressed in the experimental plan or design.

Relative to the field variability, appreciate the great strength of the FEM here. In setting out the estimation points one presumes prior knowledge of $[C_{\mu\mu}]$ among them. In a heterogeneous field with spatially-variable length scales, a uniform lattice of estimation points would seem to be the worst choice. Instead, one should strive for closely-spaced points where spatial gradients are high, and sparse spacing where gradients are low. This is of course exactly what the FEM grid generator is expected to achieve! And although grid generation is an heuristic and iterative process, relying on the intuition and experience of the generator, nevertheless an acceptable FEM grid embodies all the best wisdom about resolving expected variations in the field being approximated. So a prior estimate of $[C_{\mu\mu}]$ has already been stated, implicit in the grid.

So to proceed we need to be able to specify the covariances of $\{\mu\}$ and $\{\epsilon\}$. We will concentrate on the specification of $[C_{\mu\mu}]$ via the FEM, in the subsequent sections. But first, here is a collection of definitions which accompany the basic G-M estimator, equations 18.60 - 18.62:

$\{\mu\}$ is the vector of truth at the sampling points (nodes of a computational mesh)

$\{u\}$ is the estimate of $\{\mu\}$

$\{\epsilon_e\} \equiv \{u - \mu\}$ is the estimation error.

$\{\epsilon\} \equiv \{d\} - [S]\{\mu\}$ is the sampling error, defined at the data points; it is the sum of measurement error *per se* plus a sample of the unresolved or subgrid-scale truth.

$\{\delta\} \equiv \{d\} - [S]\{u\}$ is the model-data misfit, an estimate of $\{\epsilon\}$

$\{\mu\}$ and therefore $\{\epsilon\}$ are unknowable; $\{u\}$ and $\{\delta\}$ are estimates of them. $\{\epsilon_e\}$ is the estimation error; its covariance needs to be estimated posterior to the data, to accompany the estimate $\{u\}$.

18.4 Analytic Covariance Functions

Practical procedures try to fit simple analytic covariance functions to data. In section 13.1 we introduced an example of a distance-based covariance function. Many have been proposed and used. Almost universally, these are spatially homogeneous, isotropic functions with a small number of degrees of freedom (1 or 2), the amount of data limiting the introduction of further sophistication. This is a potentially serious limitation, especially when the assumptions of homogeneity and isotropy are basically inappropriate. But practical estimation procedures have been realized. Bretherton *et al.* ([19]) provides a general discussion. Typical are covariances of the form $C(r)$, where r is scaled distance or pseudo-distance separating any two points. These are readily made and easily interpreted conceptually.

Vector as well as scalar fields have been described this way. In particular, this is important when there are constraints among vector components and/or a scalar – e.g. an incompressibility constraint; or a directional constraint; or a relation of a vector to a potential function. Effectively, this amounts to embedding *analytic* PDE constraints into the estimation. Some examples follow.

Freeland and Gould [32] estimated streamfunction from oceanic velocity measurements. Distance-based covariance functions $C_{\psi\psi}(r)$, $C_{uu}(r)$, $C_{vv}(r)$ describe the streamfunction and the longitudinal and transverse[2] velocity covariances, as functions of the spatial separation distance r. Classical fluid mechanical constraints were imposed (non-divergent, geostrophic flow). The forms used were

$$C_{\psi\psi}(r) = (1 + br + b^2r^2/3)e^{-br} \tag{18.67}$$

$$C_{uu}(r) = (1 + br)e^{-br} \tag{18.68}$$

$$C_{vv}(r) = (1 + br - b^2r^2)e^{-br} \tag{18.69}$$

[2]These directions are relative to the separation vector **r**, which was isotropic and homogeneous.

This is one-parameter, homogeneous isotropic covariance within 3 scalar fields which recognizes simple PDE constraints. The data were used to estimate b. McWilliams [86] made similar estimates, fitting the 3-parameter functions

$$C_{\psi\psi}(r) = (1-\gamma^2 r^2)e^{-\delta^2 r^2/2} \tag{18.70}$$

$$C_{uu}(r) = (1-b^2 r^2)e^{-\delta^2 r^2/2} \tag{18.71}$$

$$C_{vv}(r) = (1-[5+\delta^2/\gamma^2]b^2 r^2 + \delta^2 b^2 r^4)e^{-\delta^2 r^2/2} \tag{18.72}$$

The constraint $b^2 = \gamma^2\delta^2/(\gamma^2+\delta^2)$ reduced the three parameters (b,γ,δ) to two.

Denman and Freeland [30] estimated 2-parameter covariance functions from oceanic observations over the continental shelf. Fitted hydrodynamic fields are reported under similar assumptions as Freeland and Gould [32]. Related scalar fields are also estimated. Functional forms included

$$C(r) = (1-r^2/b^2)e^{-r^2/a^2} \tag{18.73}$$

$$C(r) = \cos(r/b)e^{-r^2/a^2} \tag{18.74}$$

$$C(r) = J_0(r/b)e^{-r^2/a^2} \tag{18.75}$$

(J_0 is the Bessel function of the first kind.)

Hendry and He [41] implemented OA for shelf estimations using the covariance function as in Freeland and Gould [32]

$$C_{uu}(\rho) = \left(1+\rho+\rho^2/3\right)e^{-\rho} \tag{18.76}$$

In that implementation, ρ is an *anisotropic, space-time pseudo-distance*, with principal axes and scaling defined locally, prior to the data. Several studies including [90], [70], [40], [58], and [85] have used this procedure to estimate scalar oceanic fields, with the local axes defined by local topography.

Zhou [119] studied space-time interpolation of oceanic plankton data, using the covariance function

$$C(\rho) = (1-\rho)e^{-\rho} \tag{18.77}$$

where again ρ is a scaled pseudo-distance, in this case involving an isotropic 2-D spatial scale and a temporal scale. The observations were advected either forward or backward to a common time, to offset the non-synoptic sampling.

These types of covariance functions have served well in the studies cited. But in practical estimation problems, there are many potentially serious complications:

- boundary constraints: analytic covariance functions ignore them
- anisotropy: advection creates locally-anisotropic covariance, oriented to streamlines; other dynamical processes may be oriented by local parameters and their gradients
- inhomogeneity: correlation scales and principal axes may vary locally, with covariance based on distance along curved (not straight) lines.

Practical incorporation of these effects into $[C_{uu}]$ is our goal in the next section. An underlying PDE in invoked, with boundary conditions and inhomogeneous, anisotropic coefficients, and stochastic forcing.

18.5 Stochastically-Forced Differential Equation (SDE)

In this section (adapted from Lynch and McGillicuddy, [72]), the prior covariance of the field to be estimated is posed as the outcome of stochastically-forced differential equation, subject to boundary conditions with inhomogeneous, anisotropic parameters. Numerical solution is readily implemented in standard finite element methodology. Far from boundaries and inhomogeneities, the procedure defaults to standard OA methods using distance-based covariance functions.

Our basic idea is to represent the underlying field variability as the outcome of a structured stochastic process; specifically, the result of a stochastically-forced differential equation (SDE). For example, the simple equation

$$\frac{\partial^2 \mu}{\partial x^2} - k^2 \mu = \eta(x) \tag{18.78}$$

with the *process noise* η a $(0,1)$ random disturbance with no spatial correlation, has covariance

$$C_{\mu\mu}(r) = (1+kr)\, e^{-kr} \tag{18.79}$$

when posed in an unbounded domain. Thus (18.78) and (18.79) are equivalent statements of the same problem. The 2-D pair

$$\nabla^2 \mu - k^2 \mu = \eta(x,y) \tag{18.80}$$

$$C_{\mu\mu}(r) = krK_1(kr) \simeq \left(\frac{\pi}{2} kr\right)^{\frac{1}{2}} \left(1 + \frac{3}{8kr}\right) e^{-kr} \qquad kr \to \infty \tag{18.81}$$

are likewise equivalent statements ([7]). (K_1 is the Bessel function of the second kind.)

The importance of the SDE approach is that $C_{\mu\mu}$ can be computed numerically for realistic conditions – realistic anisotropy, variable coefficients, resolution, boundary conditions – which make analytic solutions impossible. The parameters and process noise model η can be chosen to represent real processes affecting the field of interest. The limiting case, far from boundaries and inhomogeneities, will default to an equivalent $C(r)$ structure. Balgovind *et al.* [7] used the SDE approach (specifically, the Helmholtz equation (18.80) with k^2 varying with latitude) to compute a realistic spatially-varying covariance for meteorological forecast error.

Discretization of (18.78), (18.80) or any other differential operator on a finite element grid leads to the matrix form

$$[K]\,\{\mu\} = \{\eta\} \tag{18.82}$$

in which all boundary constraints are incorporated automatically. The discrete process noise $\{\eta\}$ drives the variability in $\{\mu\}$. The resulting covariance of $\{\mu\}$ is obtained directly:

$$[C_{\mu\mu}] = [K]^{-1}\,[C_{\eta\eta}]\,[K]^{-T} \tag{18.83}$$

$$[C_{\mu\mu}]^{-1} = [K]^{T}\,[C_{\eta\eta}]^{-T}\,[K] \tag{18.84}$$

Immediately we have accounted, formally, for a) realistic oceanic transport processes and BCs in $[K]$, and b) realistic process errors in $C_{\eta\eta}$, representing processes not modeled in $[K]$ or in the prior estimate or forcing. Knowledge of these two effects allows us to compute $[C_{\mu\mu}]$ using standard finite element solution techniques. Coupling (18.83) with (18.60 - 18.62), we have Objective Analysis.

Two examples follow. Notice here we are using $[K]$ to indicate a well-posed FEM system matrix for a conventional (forward) problem.

Example 1

Consider the SDE

$$\frac{d}{dx}D\frac{d\mu}{dx} + V\frac{d\mu}{dx} - k^2\mu = \eta \tag{18.85}$$

Assuming constant D, the discrete forward model is

$$[1 - Pe - a\mathcal{K}^2]\ \mu_{i-1} - [2 + \mathcal{K}^2(1-2a)]\ \mu_i + [1 + Pe - a\mathcal{K}^2]\ \mu_{i+1} = N_i \tag{18.86}$$

with three dimensionless quantities

$$Pe = \frac{Vh}{2D} \quad \mathcal{K}^2 = \frac{k^2h^2}{D} \quad N = \frac{\eta h^2}{D} \tag{18.87}$$

The parameter a accounts for either Galerkin FEM on linear elements ($a = 1/6$); the lumped-mass version of same ($a = 0$); or standard FD ($a = 0$). In matrix form we have

$$[K]\{\mu\} = \{N\} \tag{18.88}$$

$[K]$ is tridiagonal; $[K]^T[K]$ is pentadiagonal. In the case $a = 0$, an interior row of $[K]^T[K]$ will be

$$\left[\ (1 - Pe^2) \qquad -2(2 + \mathcal{K}^2) \qquad 2(1 + Pe^2) + (2 + \mathcal{K}^2)^2 \qquad -2(2 + \mathcal{K}^2) \qquad (1 - Pe^2)\ \right]$$

Using the centered difference operators δ^n, these rows represent the discrete operator

$$\left[h^4\delta^4 - h^2(4Pe^2 + 2\mathcal{K}^2)\delta^2 + \mathcal{K}^4\right] \tag{18.89}$$

The original PDE has been elevated,term by term, and become symmetric:

- the Diffusion term, formerly representing Laplacian smoothing, has become biharmonic
- the Advective term has lost its directional (forward-backward) bias and plays a diffusive role
- the Helmholtz term now provides a balance between diffusive smoothing and local decay, setting the decorrelation length scale at $\sqrt{D/k^2}$ or, normalizing by mesh size, $1/\mathcal{K}$.

The inverse of $[K]^T[K]$, assuming $[C_{NN}] = [I]$, gives the covariance $C_{\mu\mu}$. This will be a full matrix, with the covariance structure centered on the diagonal. It is plotted in Figures 18.1 and 18.2, assuming periodic boundary conditions. Note that $\mathcal{K} = \frac{1}{M}$ where M is the number of grid cells per e-folding length for the analytic solution as well as for the discrete solution (above). Thus a reasonable discretization has $\mathcal{K}$ =O(1) or less. Similarly, $Pe = 1$ is a threshold for poor resolution of advection and accompanies loss of diagonal dominance in the case $\mathcal{K}^2 = 0$.

In Figure 18.1 we show $C_{\mu\mu}$ with periodic BCs, for $Pe = 0$. From (18.79) and the above discussion we expect the decorrelation scale to be $1/\mathcal{K}$ grid cells. Figure 18.1 confirms that for large $\mathcal{K}$ we have essentially an unbounded domain. Decreasing $\mathcal{K}$ broadens the covariance. Figure 18.1 also confirms that for large $\mathcal{K}$ we recover the analytical result; while at small $\mathcal{K}$ (=.01 in this example) the BCs have effect and the free-space analytic solution becomes invalid.

Figure 18.2 shows $C_{\mu\mu}$, again with periodic BCs, for various values of Pe. The effect of advection is to lengthen the correlation scale, increasing covariance in the along-stream direction. In the steady state, the upstream and downstream effect is symmetric. Creative use of these two parameters sets a baseline correlation scale via $\mathcal{K}$ and directional anisotropy via Pe, both of which can be locally variable. Modulation near boundaries is naturally incorporated from the outset.

In Figure 18.3 we show the effect of a spatial variation in diffusion coefficient. In this case there is a step change by a factor of 10 in D at node 19. The result is asymmetry (right curve) and inhomogeneity (left curve versus right) in the covariance. Figure 18.4 shows the result of an impermeable (Neumann) boundary between nodes 31 and 32. Disturbances introduced near this boundary must diffuse around it by a much longer path; thus the covariance of *geographic* near-neighbors is greatly reduced in the presence of isolating boundaries.

These results illustrate local control of the correlation length scales using physical parameters and boundary conditions of the underlying differential equation.

Objective analyses of 5 data sampled at intervals of $10\Delta x$ are plotted in Figure 18.5. The three cases illustrate the significant variations among the interpolants, depending on the details of boundary conditions. Also shown are the estimated standard deviations of the interpolations and of the data. At Neumann boundaries, the interpolation is discontinuous and the interpolation uncertainty grows where data needs to be extrapolated. Another effect of the internal Neumann boundary is the reduction of the interpolation maximum at node 35, due to its isolation from the data to the left (bottom panel of Figure 18.5).

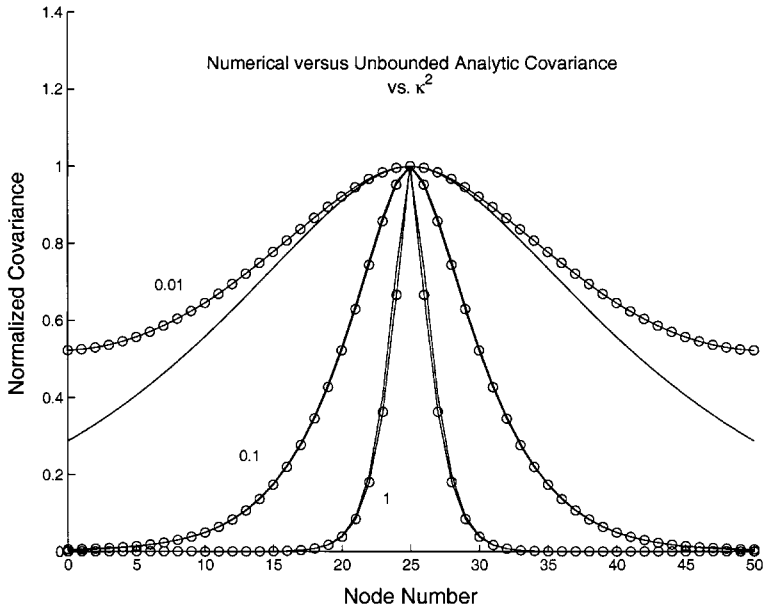


Figure 18.1: Covariance for the advective-diffusive-reactive equation (18.86) with periodic boundary conditions (dots), compared with the analytic free-space result (solid). Three different values of $\mathcal{K}^2$ are shown; $Pe = 0$. The plots have been self-normalized. (From [72].)

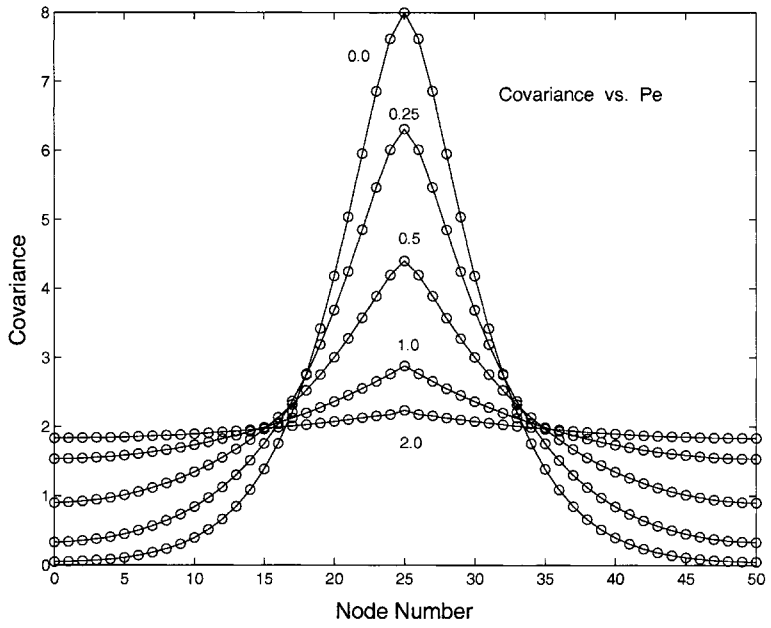


Figure 18.2: Covariance for the advective-diffusive-reactive equation, for 5 values of Pe; $\mathcal{K}^2 = 0.1$. Periodic BCs. (From [72].)

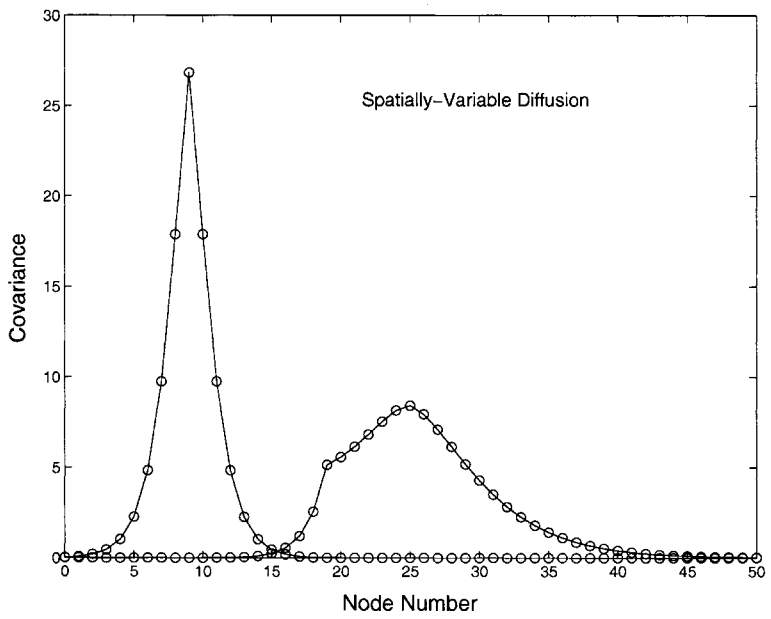


Figure 18.3: Effect of variable diffusion coefficient on covariance. To the left of node 19, $D = D_o/10$; to the right, $D = D_o$. The two curves are associated with nodes 10 (left curve) and 25 (right curve). Periodic BCs; $Pe = 0$; $\mathcal{K}_o^2 = 0.1$ (From [72].)

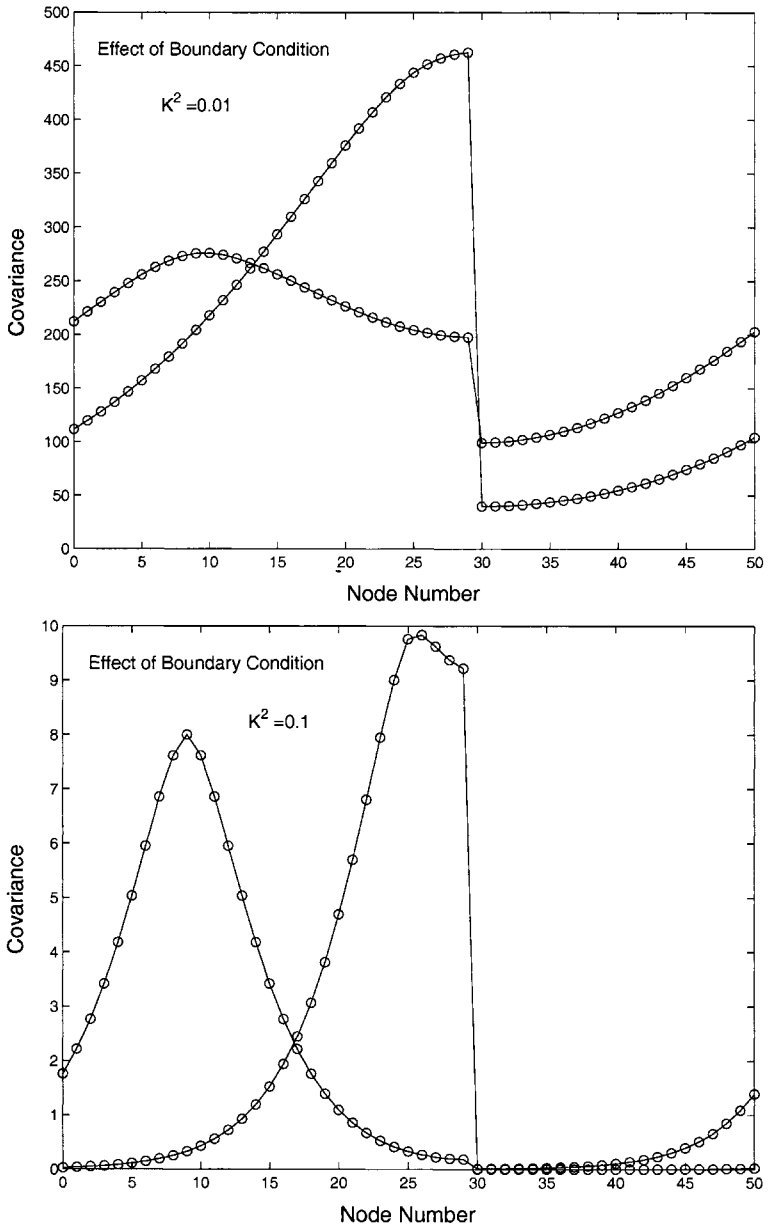


Figure 18.4: Effect of no-flux boundary condition between nodes 30 and 31 on covariance with nodes 10 (left curves) and 25 (right curves). Periodic BCs; $P_e = 0$; $\mathcal{K}^2 = .01$ (top) and 0.1 (bottom). (From [72].)

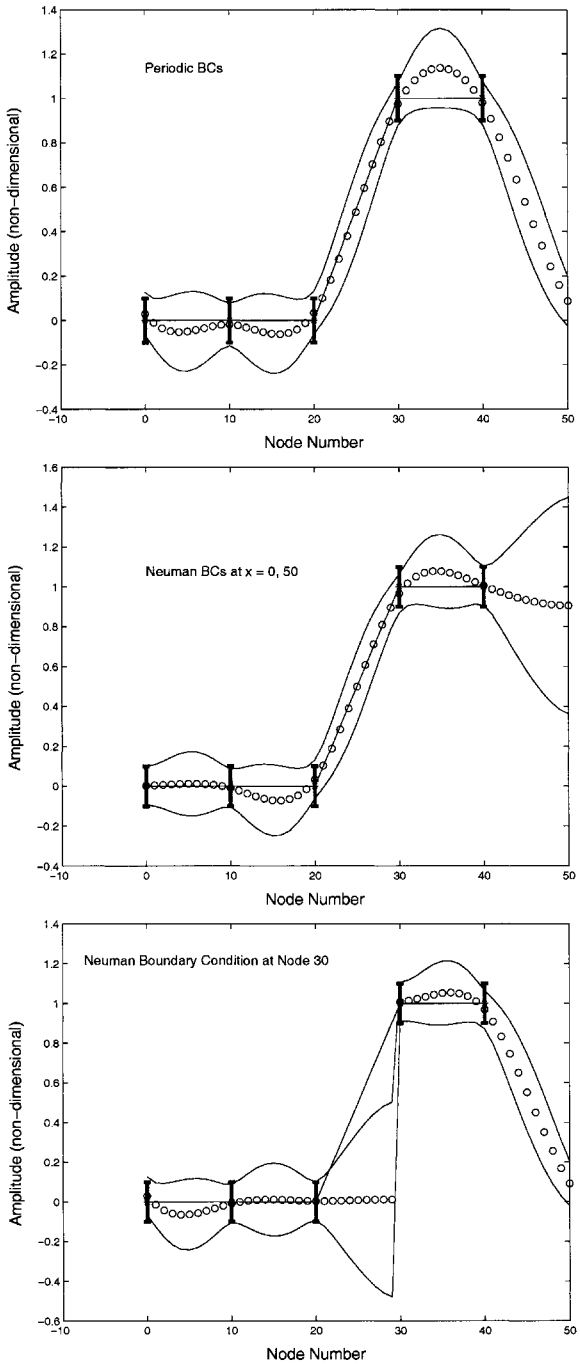


Figure 18.5: Objective Analysis of 5 data with various boundary conditions; $Pe = 0$, $\mathcal{K}^2 = .01$. The data [0 0 0 1 1] are connected by a solid line representing simple linear interpolation. The OA estimates are indicated by the circles. The standard deviations of the data (error bars) and of the estimated interpolant (solid lines) are also shown. **Top**: periodic BCs at left and right. **Middle**: Neumann BCs at left and right. **Bottom**: Neumann BCs in center of domain (just left of Node 30) and periodic BC's at left and right. Standard deviations of observational noise $S\epsilon$ (equation 18.50) and process noise $S\eta$ (equation 18.82) are (0.1, 0.05) respectively. The middle and bottom panels illustrate the successful blocking of interpolation across a no-flux (homogeneous Neumann) boundary. (From [72].)

Example 2

Next consider the 2-D transport equation

$$\frac{1}{H}\nabla \cdot HD\nabla\mu - \mathbf{V}\cdot\nabla\mu - k^2\mu = \eta_s + \eta_p \tag{18.90}$$

with

u=field anomaly
H=bathymetric depth
D=dispersion coefficient
$\mathbf{V}$=fluid velocity
k^2=first-order decay rate
η_s= surface forcing
η_p= isolated inputs from point sources (e.g. river discharges)

We imagine for example that estimates of all transport parameters are available, and that a prior estimate of the transported field has been subtracted from the data. Standard Galerkin FEM discretization leads to

$$[K]\{\mu\} = \{\eta_s\} + \{\eta_p\} + \{r\} \tag{18.91}$$

$$K_{ij} =< -HD\nabla\phi_j\cdot\nabla\phi_i - H\mathbf{V}\cdot\nabla\phi_j\ \phi_i - Hk^2\phi_j\phi_i > \tag{18.92}$$

$$r_i = -\oint HD\frac{\partial u}{\partial n}\phi_i \tag{18.93}$$

$$\eta_{si} =< H\eta_s\phi_i > \tag{18.94}$$

$$\eta_{pi} =< H\eta_p\phi_i > \tag{18.95}$$

where the usual FEM conventions are used: $<\ >$, integration over the spatial domain; $\oint$, integration over its boundary. The $\{\eta_s\}$ vector might be constituted as random noise plus a highly structured response correlated with weather; $\{\eta_p\}$ might be correlated with hydrological conditions or industrial activity. They are separated because they are assumed uncorrelated.

In [72] this formulation was used to interpolate oceanic plankton data. The context included locally dominant advection and resulting anisotropy along/across streamlines; high spatial variability in the diffusion coefficient; and important no-flux boundaries.

18.6 OA-GLS Equivalence

Statistical Estimation, equations 18.60 - 18.62, with $[C_{\mu\mu}]$ from a discretized SDE as in equation 18.83, is equivalent to solving the GLS problem

$$Minimize \quad \left\{\{\delta\}^T[C_{\delta\delta}]^{-1}\{\delta\} + \{\eta\}^T[C_{\eta\eta}]^{-1}\{\eta\}\right\} \tag{18.96}$$

subject to the constraints which define the model-data mismatch δ and the process noise η in terms of the model 18.82 and the sampling 18.64:

$$[K]\{u\} = \{\eta\} \tag{18.97}$$

$$\{\delta\} = \{d\} - [S]\{u\} \tag{18.98}$$

This is the standard problem of fitting the model 18.97 to the data, by adjusting the controls $\{\eta\}$. To see this, we will examine and manipulate the two estimates.

GLS:

Earlier we developed the GLS estimator for this problem, in several forms. The direct expression based on the normal equations is given in equation 14.21. Adjusting to the terminology here (equations 18.96-18.98), we have

$$\{u\} = [K]^{-1} \left[\left[SK^{-1} \right]^T [C_{\delta\delta}]^{-1} \left[SK^{-1} \right] + [C_{\eta\eta}]^{-1} \right]^{-1} \left[SK^{-1} \right]^T [C_{\delta\delta}]^{-1} \{d\} \tag{18.99}$$

OA/SDE:

Equation 18.61 gives the general OA estimator

$$\{u\} = \left[[C_{\mu\mu}] [S]^T \right] \left[[S] [C_{\mu\mu}] [S]^T + [C_{\epsilon\epsilon}] \right]^{-1} \{d\} \tag{18.100}$$

Using the SDE-based approach to $[C_{\mu\mu}]$, equation 18.83

$$[C_{\mu\mu}] = [K]^{-1} [C_{\eta\eta}] [K]^{-T} \tag{18.101}$$

we get the OA/SDE estimator:

$$\{u\} = [K]^{-1} \left[[C_{\eta\eta}] [K]^{-T} [S]^T \right] \left[[S] [K]^{-1} [C_{\eta\eta}] [K]^{-T} [S]^T + [C_{\epsilon\epsilon}] \right]^{-1} \{d\} \tag{18.102}$$

This appears to be a different estimate from that just given for GLS. Now we need to invoke an obscure but powerful matrix identity ([57], equation 1-51):

$$[X] [Y]^T \left[[Y] [X] [Y]^T + [Z] \right]^{-1} = \left[[Y]^T [Z]^{-1} [Y] + [X]^{-1} \right]^{-1} [Y]^T [Z]^{-1} \tag{18.103}$$

Use of this with $[X] = [C_{\eta\eta}]$, $[Y] = [S] [K]^{-1}$, $[Z] = [C_{\epsilon\epsilon}]$ gives us the equivalent expression

$$\{u\} = [K]^{-1} \left[[K]^{-T} [S]^T [C_{\epsilon\epsilon}]^{-1} [S] [K]^{-1} + [C_{\eta\eta}]^{-1} \right]^{-1} \left[[K]^{-T} [S]^T [C_{\epsilon\epsilon}]^{-1} \right] \{d\} \tag{18.104}$$

This is identical to the GLS estimator, with the only difference being the use of $[C_{\epsilon\epsilon}]$ rather than $[C_{\delta\delta}]$. The practical distinction is null. Recall that ϵ is the gap between truth and model estimate, *i.e.* the model *error* at the data points. It is unknowable except in the sense of the model-data misfit δ; and a computable distinction between the two prior estimates of covariance is unlikely. So we have practical identity between OA/SDE and GLS estimators. (The reader is referred back to equation 18.64 and the adjacent material.)

An alternative path to 18.104 is to use the alternative OA form 18.63 identified earlier, which was also based on the identity 18.103. Substitution of 18.101 for $[C_{\mu\mu}]$ and a little bit of rearrangement leads directly to 18.104, as one should expect.

This OA - GLS equivalence is a remarkable result. It is at once intuitive, unlikely, and pleasantly surprising; and it suggests a hunt for some estimation algorithms which exploit it. It was pointed out by Wunsch [115] in the oceanographic context; presumably it has been brought down from the

ancient sources in various forms. It is one of the pathways to the standard conclusion that a GLS weight matrix should be the inverse of the covariance matrix for any control variable. (See also the Appendix.)

The identity 18.103 is from Liebelt [57], who provides a collection of related identities plus a derivation. These are also reported selectively in Westlake [113] and Wunsch [115]. Here they are used only to demonstrate the present equivalence. However they also suggest alternative computational paths to estimation and potentially new algorithms which take advantage of specific properties of selected matrices. We leave this idea for the reader to explore.

The bottom line here is: OA with SDE-based covariance is the same GLS estimation – within the linear world that we pose.

18.7 Kriging

An independent development of the same ideas presented here as OA or OI has been pursued in the statistical description of mineral deposits. The methods are referred to as Kriging and its variants. Cressie [26] gives an historical account of this development. There is no substantive difference in intent or outcome, although the terminology is different and that leads to some different issues.

Probably the most central distinction is the use of the *Variogram* or SemiVariogram as the basic descriptor of field variability. This occupies the place of the covariance $[C_{\mu\mu}]$ in the OA development. Briefly, if we have two locations X_i and X_j with field values U_i and U_j, the SemiVariogram γ is defined in terms of the *difference* among them:

$$\gamma_{ij} = \frac{1}{2}\overline{(U_i - U_j)^2} \tag{18.105}$$

while the covariance is defined simply in terms of their product:

$$C_{ij} = \overline{(U_i U_j)} \tag{18.106}$$

It is easy to discover that

$$\gamma_{ij} = \frac{1}{2}\left(\overline{U_i^2} + \overline{U_j^2}\right) - \overline{(U_i U_j)} \tag{18.107}$$

$$= \frac{Var(U_i) + Var(U_j)}{2} - C_{ij} \tag{18.108}$$

and if the statistics are homogeneous,

$$\gamma_{ij} = \sigma^2 - C_{ij} \tag{18.109}$$

As in OA/OI, the common practice is to make prior estimates of the SemiVariogram assuming homogeneous functional forms of the distance between i and j.

A subtle issue is that the Variogram describes measured quantities, and as such includes measurement error as well as natural variation. As a result, $\gamma_{ii} \neq 0$ – an apparent contradiction which must be assigned to either measurement error or to true small-scale variability. This is referred to as the *nugget effect* in the Kriging literature. Cressie [27] states that ambiguity on the distinction between these two contributors to the nugget "is the source of the 'kriging is/is not an exact interpolator' controversy".

For a fuller exposition of Kriging and its variants, the reader is referred to [50], [27], or [23]. Kitanidis [51] gives an excellent practical exposition. Cressie [26] provides a detailed map of the relationship between Kriging and OA/OI – essentially the Rosetta Stone of the business. It establishes the equivalence between the mining (Kriging) and meteorological (OA/OI) literatures which grew up independently with different terminology during the 1960's. (Gandin [33] produced the definitive modern work in the latter area.)

18.8 Concluding Remarks

The observational opportunities for environmental fields are enormous today, and full of promise. At the same time, the computational power available is unprecedented. The merger of the formerly separate fields of PDE solution and Field Estimation represents a major opportunity with broad implications for scientific advance and engineering practice.

In classical Gauss-Markov estimation, prior knowledge of the covariance of the field to be estimated is critical. Stochastically-forced differential equations representing physical processes are a natural way forward, accommodating the complexity represented in modern PDE solution. The combination of G-M estimation with SDE-based field covariance is potentially very powerful, and equivalent to a certain GLS minimization of model-data misfit. This OA-SDE approach focuses special attention on a) what *processes* are relevant to the observed field variability; b) the nature of the stochastic forcing (the *process noise*); and c) the nature of the *observational noise.*

It is important to pay attention to the spatial correlation in the *process noise.* There is much substance in the specification of this Covariance model. Distance-based Covariance models for this forcing may suffice in many practical cases, especially since the intervening PDE insulates the estimation by an extra layer of structured space-time covariance. But essentially, these are physical questions and must ultimately be resolved in those terms.

Related is the need to describe the *observational noise* via a relevant statistical model. Every data product needs to have an associated error model! And, any data being analyzed should generally be reduced by removal of a structured prior estimate (spatially-variable mean), in the hope of achieving the zero-mean condition generally assumed and/or required. We can anticipate considerable scientific advances in areas fueled by this (formerly) interdisciplinary consideration.

From a computational point of view, the FEM-based Covariance matrices and their inverses are, *a priori*, full. As computational meshes grow in scope and refinement, dense matrix manipulations certainly scale badly. It will be practical to reduce matrix density by truncating covariance matrices beyond some practical limit, and exploiting the resulting sparse matrix structure.

Appendices

A1. Vector Identities

Triple Products

$$\mathbf{A} \cdot \mathbf{B} \times \mathbf{C} = \mathbf{B} \cdot \mathbf{C} \times \mathbf{A} = \mathbf{C} \cdot \mathbf{A} \times \mathbf{B} \tag{1}$$

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = \mathbf{B}\,(\mathbf{A} \cdot \mathbf{C}) - \mathbf{C}\,(\mathbf{A} \cdot \mathbf{B}) \tag{2}$$

Differentiation

$$\nabla\,(\phi\psi) = \phi\nabla\psi + \psi\nabla\phi \tag{3}$$

$$\nabla \cdot (\phi\mathbf{A}) = \phi\nabla \cdot \mathbf{A} + \mathbf{A} \cdot \nabla\phi \tag{4}$$

$$\nabla \times (\phi\mathbf{A}) = \phi\nabla \times \mathbf{A} + \nabla\phi \times \mathbf{A} \tag{5}$$

$$\nabla \cdot (\mathbf{A} \times \mathbf{B}) = \mathbf{B} \cdot (\nabla \times \mathbf{A}) - \mathbf{A} \cdot (\nabla \times \mathbf{B}) \tag{6}$$

$$\nabla \times (\mathbf{A} \times \mathbf{B}) = \mathbf{B} \cdot \nabla\mathbf{A} - \mathbf{A} \cdot \nabla\mathbf{B} + \mathbf{A}\nabla \cdot \mathbf{B} - \mathbf{B}\nabla \cdot \mathbf{A} \tag{7}$$

$$\nabla\,(\mathbf{A} \cdot \mathbf{B}) = \mathbf{A} \cdot \nabla\mathbf{B} + \mathbf{B} \cdot \nabla\mathbf{A} + \mathbf{A} \times (\nabla \times \mathbf{B}) + \mathbf{B} \times (\nabla \times \mathbf{A}) \tag{8}$$

$$\nabla \cdot (\nabla\phi) = \nabla^2\phi \tag{9}$$

$$\nabla \times (\nabla \times \mathbf{A}) = \nabla\,(\nabla \cdot \mathbf{A}) - \nabla^2\mathbf{A} \tag{10}$$

$$\nabla \times (\nabla\phi) = 0 \tag{11}$$

$$\nabla \cdot (\nabla \times \mathbf{A}) = 0 \tag{12}$$

$$\nabla \cdot (\nabla\phi \times \nabla\psi) = 0 \tag{13}$$

Integration

A right-handed $(n, \mathbf{s})$ boundary coordinate system is conventional, with $\hat{\mathbf{n}}$ directed outward and $\mathbf{s}$ a two-dimensional surface space normal to it. The scalar ds is an increment of surface area and has area units.

$$\int\int\int \nabla \cdot \mathbf{A}\; dv = \oint \hat{\mathbf{n}} \cdot \mathbf{A}\; ds \tag{14}$$

$$\int\int\int \nabla\phi\; dv = \oint \phi\; \hat{\mathbf{n}}\; ds \tag{15}$$

$$\int\int\int \nabla \times \mathbf{A}\; dv = \oint \hat{\mathbf{n}} \times \mathbf{A}\; ds \tag{16}$$

$$\int\int\int \nabla \cdot (\phi\nabla\psi)\; dv = \oint \hat{\mathbf{n}} \cdot \phi\nabla\psi\; ds \tag{17}$$

$$\int\int\int \left(\phi\nabla^2\psi + (\nabla\phi) \cdot (\nabla\psi)\right)\; dv = \oint \hat{\mathbf{n}} \cdot \phi\nabla\psi\; ds \tag{18}$$

$$\int\int\int \left(\phi\nabla^2\psi - \psi\nabla^2\phi\right)\; dv = \oint \hat{\mathbf{n}} \cdot (\phi\nabla\psi - \psi\nabla\phi)\; ds \tag{19}$$

In 2-D, a common scalar coordinate convention is (n, s, z) with $\hat{\mathbf{z}}$ normal to the plane of analysis and $\hat{\mathbf{n}}$ directed outward like before, but in the plane of analysis. Similarly, ds is a 1-D increment of line length in the plane, and dv is the increment of area in the plane. The 2-D circulation theorem is conveniently expressed as

$$\int\int (\nabla \times \mathbf{A}) \cdot \hat{\mathbf{z}} \, dv = \oint \hat{\mathbf{s}} \cdot \mathbf{A} \, ds \tag{20}$$

Integration by Parts

Combining differentiation and integration as recorded above we get these identities:

$$\oint \phi\psi \, \hat{\mathbf{n}} \, ds = \int\int\int (\phi\nabla\psi + \psi\nabla\phi) \, dv \tag{21}$$

$$\oint \hat{\mathbf{n}} \cdot \phi\mathbf{A} \, ds = \int\int\int (\phi\nabla \cdot \mathbf{A} + \mathbf{A} \cdot \nabla\phi) \, dv \tag{22}$$

$$\oint \hat{\mathbf{n}} \times \phi\mathbf{A} \, ds = \int\int\int (\phi\nabla \times \mathbf{A} + \nabla\phi \times \mathbf{A}) \, dv \tag{23}$$

In PDE usage, the following forms of these are common:

$$\oint (\hat{\mathbf{n}} \cdot \mathbf{A})\phi \, ds = \int\int\int (\nabla \cdot \mathbf{A}) \, \phi \, dv + \int\int\int (\mathbf{A} \cdot \nabla\phi) \, dv \tag{24}$$

$$\oint (\hat{\mathbf{n}} \times \mathbf{A})\phi \, ds = \int\int\int (\nabla \times \mathbf{A}) \, \phi \, dv - \int\int\int (\mathbf{A} \times \nabla\phi) \, dv \tag{25}$$

$$\int\int\int (\nabla \cdot \mathbf{A}) \, \phi \, dv = \oint (\hat{\mathbf{n}} \cdot \mathbf{A})\phi \, ds - \int\int\int (\mathbf{A} \cdot \nabla\phi) \, dv \tag{26}$$

$$\int\int\int (\nabla \times \mathbf{A}) \, \phi \, dv = \oint (\hat{\mathbf{n}} \times \mathbf{A})\phi \, ds + \int\int\int (\mathbf{A} \times \nabla\phi) \, dv \tag{27}$$

A2. Coordinate Systems

Cartesian $(x,\ y,\ z)$:

$$\mathbf{dl} = \hat{\mathbf{x}}dx + \hat{\mathbf{y}}dy + \hat{\mathbf{z}}dz \tag{1}$$

$$\nabla\phi = \hat{\mathbf{x}}\frac{\partial\phi}{\partial x} + \hat{\mathbf{y}}\frac{\partial\phi}{\partial y} + \hat{\mathbf{z}}\frac{\partial\phi}{\partial z} \tag{2}$$

$$\nabla\cdot\mathbf{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z} \tag{3}$$

$$\nabla\times\mathbf{A} = \hat{\mathbf{x}}\left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right) + \hat{\mathbf{y}}\left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right) + \hat{\mathbf{z}}\left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right) \tag{4}$$

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} \tag{5}$$

Cylindrical $(r,\ \theta,\ z)$:

$$\mathbf{dl} = \hat{\mathbf{r}}dr + \hat{\theta}rd\theta + \hat{\mathbf{z}}dz \tag{6}$$

$$\nabla\phi = \hat{\mathbf{r}}\frac{\partial\phi}{\partial r} + \hat{\theta}\frac{1}{r}\frac{\partial\phi}{\partial\theta} + \hat{\mathbf{z}}\frac{\partial\phi}{\partial z} \tag{7}$$

$$\nabla\cdot\mathbf{A} = \frac{1}{r}\frac{\partial\left(rA_r\right)}{\partial r} + \frac{1}{r}\frac{\partial A_\theta}{\partial\theta} + \frac{\partial A_z}{\partial z} \tag{8}$$

$$\nabla\times\mathbf{A} = \hat{\mathbf{r}}\left(\frac{1}{r}\frac{\partial A_z}{\partial\theta} - \frac{\partial A_\theta}{\partial z}\right) + \hat{\theta}\left(\frac{\partial A_r}{\partial z} - \frac{\partial A_z}{\partial r}\right) + \hat{\mathbf{z}}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(rA_\theta\right) - \frac{1}{r}\frac{\partial A_r}{\partial\theta}\right) \tag{9}$$

$$\nabla^2\phi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} + \frac{\partial^2\phi}{\partial z^2} \tag{10}$$

A good source for practical identities is Hildebrand [43].

A3. Stability of the Roots of the Quadratic Equation

Consider the general quadratic equation

$$a\lambda^2 + b\lambda + c = 0 \tag{1}$$

where it is assumed that a, b, and c are real and that a is positive. The standard solution is

$$\lambda = -\frac{b}{2a} \pm \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} \tag{2}$$

For $\frac{c}{a} > \left(\frac{b}{2a}\right)^2$, λ will be complex and its magnitude is given by

$$|\lambda|^2 = \frac{c}{a} \tag{3}$$

The necessary and sufficient condition for the stability of the complex roots of (1) is thus

$$\frac{c}{a} < 1 \tag{4}$$

If $\frac{c}{a} \leq \left(\frac{b}{2a}\right)^2$, λ will be real and for one of the roots (2),

$$|\lambda| \geq |\frac{b}{2a}| \geq \sqrt{\frac{c}{a}} \tag{5}$$

The condition (4) is thus necessary for the stability of the real roots as well. From (5), a second necessary condition is

$$|\frac{b}{2a}| \leq 1 \tag{6}$$

Assuming that (6) is satisfied, the stability constraint for the real roots is

$$|\frac{b}{2a}| + \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} \leq 1 \tag{7}$$

or,

$$\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} \leq 1 - \frac{b}{2a} \tag{8}$$

Since both sides of (8) are nonnegative, they may be squared to produce

$$\left(\frac{b}{2a}\right)^2 - \frac{c}{a} \leq 1 + \left(\frac{b}{2a}\right)^2 - |\frac{b}{a}| \tag{9}$$

Rearrangement of (9) gives the stability constraint for the real roots:

$$|\frac{b}{a}| \leq 1 + \frac{c}{a} \tag{10}$$

or,

$$|b| \leq a + c \tag{11}$$

As long as (4) is satisfied, (11) implies (6). It can be shown that when (4) is satisfied but (11) is not, the roots of (1) are necessarily real. From (11),

$$b^2 > a^2 + c^2 + 2ac = 4ac + (a - c)^2 \tag{12}$$

and thus,

$$b^2 > 4ac \tag{13}$$

Thus under these conditions the roots are real and unstable. The necessary and sufficient conditions, then, for the stability of the roots of (1) are:

$$\frac{c}{a} \leq 1 \tag{14}$$

and

$$|b| \leq a + c \tag{15}$$

A4. Inversion Notes

Some facts:

- $[A]^T$ indicates the transpose of $[A]$
- $\left[A^T\right]^{-1} = \left[A^{-1}\right]^T \equiv \left[A^{-T}\right]$
- $\mathcal{E}([A]\,x) = [A]\,\mathcal{E}(x)$
- $Cov([A]\,x) = [A]\,Cov(x)\,[A]^T$
- $Cov(x)$ is symmetric, positive definite
- $[A]$ Symmetric Positive Definite implies that $[A] = [R]\,[R]^T$ exists, with $[R]$ the "square root" matrix.
- $[X]\,[Y]^T\left[[Y]\,[X]\,[Y]^T + [Z]\right]^{-1} = \left[[Y]^T\,[Z]^{-1}\,[Y] + [X]^{-1}\right]^{-1}\,[Y]^T\,[Z]^{-1}$

Item 1: *The WLS weight matrix should be the inverse of the covariance matrix.*

We routinely minimize sums of mutually independent, squared errors. If we have *covariance* among the errors:

$$\epsilon = (0, [V]) \tag{1}$$

then $[V]$ has square-root factors $[K]$:

$$[V] = [K]^T\,[K] \tag{2}$$

Introduce the variable μ:

$$\mu = [K]^{-T}\,\epsilon \tag{3}$$

and we have

$$Cov(\mu) = Cov([K]^{-T}\,\epsilon) = [K]^{-T}\,Cov(\epsilon)\,[K]^{-1} \tag{4}$$

$$Cov(\mu) = [K]^{-T}\,[K]^T\,[K]\,[K]^{-1} = [I] \tag{5}$$

that is,

$$\mu = (0, [I]) \tag{6}$$

So μ are independent with zero mean, unit variance. If we minimize the quadratic form

$$\mu^T\mu = ([K]^{-1}\,\epsilon)^T([K]^{-1}\,\epsilon) = \epsilon^T\,[K]^{-T}\,[K]^{-1}\,\epsilon \tag{7}$$

$$\mu^T\mu = \epsilon^T([K]\,[K]^T)^{-1}\epsilon \tag{8}$$

$$\mu^T\mu = \epsilon^T\,[V]^{-1}\,\epsilon \tag{9}$$

In other words, minimizing the quadratic form

$$\epsilon^T\,[W]\,\epsilon \tag{10}$$

is equivalent to minimizing the norm of independent errors $\mu = (0, [V])$

$$\mu^T \mu \tag{11}$$

provided the weight matrix is the inverse of the covariance matrix:

$$[W] = [V]^{-1} \tag{12}$$

Item 2: *Regularization of a FEM result.*

Given a discretized PDE

$$[K]\, x = b + \epsilon \tag{13}$$

with known $[K]$ and b, and stochastic perturbations ϵ

$$\epsilon = (0, \sigma^2) \tag{14}$$

First, demean the relationship with $\hat{x} = x - \overline{x}$:

$$[K]\, \overline{x} = b \qquad and \qquad [K]\, \hat{x} = \epsilon \tag{15}$$

Now we have

$$\hat{x} = [K]^{-1}\, \epsilon \tag{16}$$

and the proper regularization weight $[W]$ for the quadratic form

$$\hat{x}^T\, [W]\, \hat{x} \tag{17}$$

is

$$[W] = [Cov(\hat{x})]^{-1} \tag{18}$$

From above,

$$[Cov(\hat{x})] = [K]^{-1}\, \sigma^2\, [K]^{-T} \tag{19}$$

and therefore

$$[W] = \frac{1}{\sigma^2}\, [K]^T\, [K] \tag{20}$$

Thus, we minimize the quadratic form

$$\frac{1}{\sigma^2} \hat{x}^T\, [K]^T\, [K]\, \hat{x} \tag{21}$$

For example, if $[K]$ is a (symmetric) Laplacian, then the proper quadratic form is $[W] = \frac{1}{\sigma^2}\, [K]^2$ *i.e.* a biharmonic smoother.

Item 3: *Generating random numbers.*

Given a random number generator which produces $\epsilon = (0, \sigma^2\, [I])$. We want to generate a distribution x with known covariance. Postulate a linear generator $[A]$:

$$x = [A]\, \epsilon \tag{22}$$

Since we know the covariance of x, then we have

$$Cov(x) = Cov([A]\, \epsilon) = [A]\, \sigma^2\, [A]^T \tag{23}$$

So the constraint on $[A]$ is

$$[A]\, [A]^T = \frac{1}{\sigma^2}\, [Cov(x)] \tag{24}$$

If we want to generate x from ϵ, $[A]$ must be the square root of the known $[Cov(x)]$. The Cholesky Square-Root algorithm is commonly invoked for this factorization; it is a special case of LU factorization for symmetric matrices, $[U]$=$[L]^T$

A5. Time Conventions for Geophysical Fields

Here we describe one example of a set of a library of software and standards that addresses this critical problem. **NML** refers the the Numerical Methods Laboratory at Dartmouth College; **IOS**, to the Institute of Ocean Sciences at Victoria, BC.

All subroutines and programs listed here are available in sourcecode at
http://www-nml.dartmouth.edu/Software/.

Time Registration

There are three time standards in current use: Gregorian, DMY, and UTC0. In all cases, timing is assumed to be referenced to the Greenwich Meridian and the contemporary Gregorian calendar.

Gregorian Time. Precision timing needs to be maintained in Geophysical models over long simulation times. Hence, time needs to be kept with an integer part and a real (precision) part. The most natural and robust integer part is the Gregorian Day, which establishes an absolute timeline[3] and properly accounts for variable numbers of days in months, leap years, etc. The real part S is assumed to be kept in seconds to accomodate the MKS system.

- The natural time convention for these models is the Gregorian time (K,S):
  - K = Gregorian Day #
  - S = elapsed time (seconds) since day K began at the Greenwich Meridian.

  The Gregorian time at 1:00 AM (Greenwich), January 1, 1999 is (730121, 3600.0).

The accumulation of large S threatens precision. The NML subroutine **Up_Date2**(K, S) increments the Gregorian K, if possible, and decrements S accordingly. This means the largest necessary value of S is 86,400. seconds (one day) and one-second precision should be possible in timekeeping on ordinary machines.

DMY Time. An equivalent integer part is the 3-integer DMY (day, month, year) convention. This is intuitively appealing but raises problems with respect to time intervals since months and years are of different duration. The Gregorian Calendar provides the unique, sequential day numbering system and the IOS routines **GDAY**(ID, IM, IY, K) and **DMY**(ID, IM, IY, K) provide the translations. GDAY converts to Gregorian Day; DMY is its inverse.

- An equivalent, intuitively appealing time stamp then is DMY time (ID, IM, IY, S)

[3]Gregorian day 1 is January 1, Year 0000 AD.

- ID = Day number of month
- IM = Month number (1 = January etc.)
- IY = Year number (*e.g.* 1999)
- S = elapsed time (seconds) since the day began (time 0000 on that day, at Greenwich).

S is identical in both Gregorian and DMY systems. The DMY time at 1:00 AM GMT, January 1, 1999 is (1, 1, 1999, 3600.0).

UTC0 (or GMT0) Time. An alternate time convention is the UTC0 time. In this case the Gregorian year (*e.g.* 1999) is recorded as the integer part, and the precision part is the elapsed time since the beginning of the year, in **decimal days**. UTC0 is identical to GMT0, an older nomenclature.

- The UTC0 time convention is (IY,D):
  - IY = Gregorian Year #
  - D = elapsed time (decimal days) since Year IY began at the Greenwich Meridian.

The UTC0 time at 1:00 AM (Greenwich), January 1, 1999 is (1999, .0416666666667).

Calculation of time intervals involving two different years requires access to the Gregorian standard; the NML routine **UTC0_GDAY**(IY, D, K, S) converts UTC0 (IY, D) to Gregorian (K, S). Its inverse **GDAY_UTC0**(K, S, IY, D) converts Gregorian to UTC0. To service the older GMT0 standard, routines GMT0_GDAY and GDAY_GMT0 are maintained. These simply call the appropriate UTC0 conversion.

Figure 1 depicts these standards and their conversions.

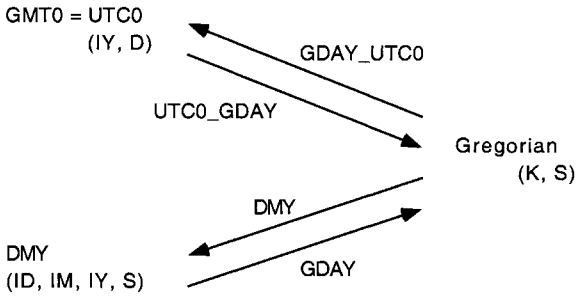


Figure 1: Time standards and their conversions.

Tidal Synthesis

Tidal timeseries are intrinsically periodic, with known harmonic frequencies established by astrophysical motions. The most commonly used conventions for analysis and synthesis of tidal timeseries are those of Godin, as implemented by Foreman [31]. Each constituent (individual harmonic component) is characterized by three real numbers:

- amplitude A, [meters]
- frequency ω [radians/sec]
- Greenwich phaselag g, [degrees]

These units are consistent with NML software standards. Synthesis of tidal timeseries is then a summation over all frequencies:

$$z(t) = Z_0 + \sum_k \left[f(t) \cdot A \cdot \cos\left(\omega[t - t_0] + V(t_0) + u(t) - \frac{\pi}{180} g \right) \right]_k \quad (1)$$

An absolute time standard t is implied; it is the Gregorian time. The reference time t_0 is arbitrary; $V(t_0)$ adjusts for it. f and u are slowly-varying functions of time and account for the nodal modulation (the result of bundling several closely-spaced constituents into a single standard constituent *e.g.* M2.) In principle f and u are needed at every synthesis time t. Because they vary slowly, in practical simulations it is often reasonable to evaluate them once, at $t = t_0$, and treat them as constants.

Subroutine **VUF_NML** provides values of ω for all standard tidal constituents; plus $V(t_0)$, $f(t_0)$, and $u(t_0)$ for any given time t_0 coinciding with the start of a Gregorian day. Thus the complete and unambiguous synthesis of tidal signals is achieved by the three data (A, ω, g) plus access to VUF_NML and IOS_tidetbl.

- Usage:
  - Call VUF_NML(Kd,KONk,xlat,fk,vuk,freqk)
- Inputs:
  - Kd: Gregorian day t_0. (The start of this day is t_0);
  - KONk [character*5]: Constituent name as a 5-character string, left-justified (`\eg M2bbb; or 2MS2b`).
  - xlat: latitude; decimal degrees North.
- Outputs:
  - fk: amplitude modulation factor f; dimensionless
  - vuk: $V(t_0) + u(t_0)$, radians.
  - freqk: constituent frequency ω, radians/sec.

VUF_NML is bundled from three standard IOS routines OPNVUF, SETVUF, and VUF, to provide I/O compatible with NML standards (see Table 1). It reads the data file IOS_tidetbl when first invoked. Notice that t_0 is restricted to be the start of a day when using this subroutine.

There is a simpler description of tidal timeseries, suitable for short-term analysis and synthesis:

$$z(t) = Z_0 + \sum_k \left[Z \cdot \cos\left(\omega[t - t_0] - \frac{\pi}{180} \phi \right) \right]_k \quad (2)$$

with t_0 defined by convenience, and phaselag ϕ linked to t_0 appropriately. With this standard, four data are needed for tidal synthesis: (Z, ϕ, ω, t_0). Note that ω is the same quantity as in the Foreman/Godin form; except here it is radians/sec, not cycles/hour. The amplitude and phase parameters differ due to the timing conventions and to the accounting for nodal modulation.

The NML subroutine RADIANS is available to evaluate $\omega[t - t_0]$ for a particular constituent. Its use is:

- Usage:
  - Call RADIANS(omegat, kd, kd0, seckd, seckd0, omega, k)
- Output:
  - omegat : result $\omega[t - t_0]$; radians
- Inputs:
  - k : constituent index k
  - (kd, seckd) : time of evaluation t; Gregorian standard (day, seconds)
  - (kd0(k), seckd0(k)) : array of t_0 for all constituents k; Gregorian standard
  - omega(k) : array of frequencies ω for constituent k; radians/sec

Equivalent to (2) is a description in terms of complex amplitude $\mathcal{Z}$:

$$z(t) = Z_0 + Re\left\{\sum_k \left[\mathcal{Z}\exp\left(\omega[t - t_0]\right)\right]_k\right\} \tag{3}$$

Here the amplitude and phase are embedded in the single complex number $\mathcal{Z}$:

$$Z = |\mathcal{Z}| \tag{4}$$

$$-\frac{\pi}{180}\phi = \arg(\mathcal{Z}) = \arctan\left(\frac{Im(\mathcal{Z})}{Re(\mathcal{Z})}\right) \tag{5}$$

This description is the most appealing for analytical work, as it utilizes the elegant apparatus of complex analysis.

Table 1: The basic IOS tidal routines which are bundled in VUF_NML. These are available at http://www-nml.dartmouth.edu/Software/iospak

The IOS subroutines may be used separately and provide values of V_0, f, and u at hourly increments. They are invoked in three steps:

- Call OPNVUF (Kh,KONk,xlat,fk,vuk,freqk) – this reads and saves the file IOS_tidetbl and calls SETVUF. It uses none of its arguments but passes them to SETVUF.
- Call SETVUF (Kh,KONk,xlat,fk,vuk,freqk) – this evaluates and saves f, $(V+u)$, and ω for all constituents listed in IOS_tidetbl. It uses only the time and latitude arguments Kh and xlat.
- Call VUF (Kh,KONk,xlat,fk,vuk,freqk) – this reports back (fk,vuk,freqk) for the single constituent KONk, based on the calculations done in SETVUF for all constituents. VUF ignores Kh and xlat.

The first two routines need to be called only once. Then VUF is invoked for each individual tidal constituent separately to get ω, f, and (V_0+u). The arguments are:

- Inputs:
  - Kh [integer]: Gregorian hour t_0 ; (GregorianDay*24)
  - KONk [character*5]: Constituent name as a 5-character string, left-justified (*e.g.* M2bbb; or 2MS2b).
  - xlat [real]: latitude; decimal degrees North.
- Outputs:
  - fk [real]: amplitude modulation factor f; dimensionless
  - vuk [real]: $V(t_0)+u(t_0)$, cycles.
  - freqk [undeclared] constituent frequency ω, cycles/hour

Beware of the units! IOS time is hours; and frequencies and angles are in cycles, not radians. Subroutine VUF_NML calls these in an efficient order, and translates the units.

Bibliography

[1] M. Ainsworth and J.T. Oden. *A Posteriori Error Estimation in Finite Element Analysis.* Wiley Interscience, 2000. 240 pp.

[2] W.F. Ames. *Numerical Methods for Partial Differential Equations.* Academic Press, 3rd edition, 1992. 451pp.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, 1995. 325pp; http:// www.netlib.org/lapack/lug.

[4] A. Arakawa. Computational design for long-term numerical integration of the equations of fluid motion. Two-dimensional compressible flow. Part I. *J. Comput. Phys.*, 1:119–143, 1966.

[5] A. Arakawa. *Design of the UCLA general circulation model.* Numerical Simulation of Weather and Climate,Tech. Report 7. UCLA Dept. of Meteorology, 1972. 116pp.

[6] I. Babuska, O.C. Zienkiewicz, J. Gago, and E.R.deA. Oliveira. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations.* Wiley Interscience, 1986. 393 pp.

[7] R. Balgovind, A. Dalcher, M. Ghil, and E. Kalnay. A stochastic-dynamic model for the spatial structure of forecast error statistics. *Mon. Wea. Rev.*, 111:701–722, 1983.

[8] M.L. Barton and Z.J. Cendes. New vector finite elements for three-dimensional magnetic field computation. *J. Appl. Phys.*, 61(8):3919–3921, 1987.

[9] A.F. Bennett. *Inverse Methods in Physical Oceanography.* Cambridge University Press, 1992.

[10] J. Berntsen, R. Cools, and T.O. Espelid. Algorithm 720: An algorithm for adaptive cubature over a collection of 3-dimensional simplices. *ACM Trans. on Mathematical Software*, 19(3):320–332, 1993.

[11] J. Berntsen and T.O. Espelid. On the construction of higher degree three-dimensional embedded integration rules. *SIAM J. on Numer. Anal.*, 25(1):222–234, 1988.

[12] J. Berntsen and T.O. Espelid. Algorithm 706 DCUTRI: An algorithm for adaptive cubature over a collection of triangles. *ACM Trans. on Mathematical Software*, 18(3):329–342, 1992.

[13] J. Berntsen, T.O. Espelid, and A. Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Trans. on Mathematical Software*, 17(4):437–451, 1991.

[14] W.E. Boyse, D.R. Lynch, K.D. Paulsen, and G.N. Minerbo. Nodal based finite element modeling of Maxwell's equations in three dimensions. *IEEE Trans Antennas and Propagat*, 40:642–651, 1992.

[15] W.E. Boyse and K.D. Paulsen. Accurate Solutions of Maxwell's equations around PEC corners and highly curved surfaces using nodal finite elements. *IEEE Trans Antennas Propagat*, 45:1758–1767, 1997.

[16] W.E. Boyse and A.A. Seidl. A hybrid finite element method for near bodies of revolution. *IEEE Trans. Magnetics*, 27:3833–3836, 1991.

[17] W.E. Boyse and A.A. Seidl. A hybrid finite element method for 3-D scattering using nodal and edge elements. *IEEE Trans Antennas Propagat*, 42:1436–1442, 1994.

[18] R.L. Bras and I. Rodriguez-Iturbe. *Random Functions in Hydrology.* Addison Wesley, 1985. 559pp.

[19] F.P. Bretherton, R.E. Davis, and C.B. Fandry. A technique for objective analysis and design of oceanographic experiments applied to MODE-73. *Deep Sea Res.*, 23:559–582, 1976.

[20] R.L. Burden and J.D. Faires. *Numerical Analysis.* PWS-Kent, 1989. 729pp.

[21] G.F. Carey. Derivative calculation from finite element solutions. *Comput. Meth. Appl. Mech. Eng.*, 35:1–14, 1982.

[22] G.F. Carey, S.F. Chow, and M.K. Seager. Approximate boundary-flux calculations. *Comput. Meth. Appl. Mech. Eng.*, 50:107–120, 1985.

[23] J.-P. Chiles and P. Delfiner. *Geostatistics: Modeling Spatial Uncertainty.* John Wiley, 1999. 695pp.

[24] G. Christakos. *Modern Spatiotemporal Geostatistics.* Oxford University Press, 2000. 288pp.

[25] G.R. Cowper. Gaussian quadrature formulas for triangles. *Int. J. Num. Meths. Engineering*, 7:405–408, 1973.

[26] N. Cressie. The origins of Kriging. *Mathematical Geology*, 22:239–252, 1990.

[27] N.A.C. Cressie. *Statistics for Spatial Data.* New York: J. Wiley, 1991. 900pp.

[28] R. Daley. *Atmospheric Data Analysis.* Cambridge University Press, Cambridge, 1991.

[29] A.J. Davies. *The Finite Element Method: A First Approach.* OxfordUniversity Press, 1980.

[30] K.L. Denman and H.J. Freeland. Correlation scales, objective mapping and a statistical test of geostrophy over the continental shelf. *J. Marine Res.*, 43:517–539, 1985.

[31] M.G.G. Foreman. Manual for Tidal Heights Analysis and Prediction. Pacific Marine Science Report 77-10, Institute of Ocean Sciences, Patricia Bay, Sidney, BC, 58pp., 1977.

[32] H.J. Freeland and W.J.Gould. Objective analysis of meso-scale ocean circulation features. *Deep Sea Res.*, 23:915–923, 1976.

[33] L. Gandin. Objective Analysis of Meteorological Fields. Leningrad:Gridromet, 1963. English translation 1965, Jerusalem: Israel Program for Scientific Translation.

[34] A. Gangopadhyay, A.R. Robinson, and H.G. Arango. Circulation and Dymamics of the Western North Atlantic: Part I. Multiscale Feature Models. *J. Atmos. Ocean. Tech.*, 14:1314–1351, 1997.

[35] G. H. Golub and C.F. van Loan. *Matrix Computations.* The Johns Hopkins University Press, 3rd edition, 1996. 664pp.

[36] P. Goovaerts. *Geostatistics for Natural Resources Evaluation.* Oxford University Press, 1997.

[37] W.G. Gray and M.Th. van Genuchten. Economical alternatives to Gaussian Quadrature over Isoparametric Quadrilaterals. *Int. J. Num. Meths. Engineering*, 1978.

[38] D.A. Greenberg, F.E. Werner, and D.R. Lynch. A diagnostic finite-element ocean circulation model in spherical-polar coordinates. *J. Atmospheric and Oceanic Technology*, 15(C9):942–958, 1998.

[39] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method.* Wiley, 1998. 1021pp.

[40] C. Hannah, J. Loder, and D. Wright. Seasonal variation of the baroclinic circulation in the Scotia Maine region. In D. Aubrey, editor, *Buoyancy Effects on Coastal Dynamics*, volume 53 of *Coastal and Estuarine Studies*, pages 7–29. American Geophysical Union, 1996.

[41] R. Hendry and I. He. Technical Report on Objective Analysis (OA) Project. Can. Tech. Rep. Hydrog. Ocean Sci. (in prep.)
http://www.mar.dfo-mpo.gc.ca/science/ocean/coastal_hydrodynamics.oax.html , 2000.

[42] M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

[43] F.B. Hildebrand. *Advanced Calculus for Applications.* Prentice-Hall, 1962. 646pp.

[44] F.B. Hildebrand. *Methods of Applied Mathematics.* Prentice-Hall, 1965. 362pp.

[45] R.W. Hornbeck. *Numerical Methods.* Quantum, 1975. 310pp.

[46] J.T.C. Ip and D.R. Lynch. Finite element solution of the two-dimensional incompressible Navier-Stokes equations with mixed interpolation. Numerical Methods Laboratory Report NML-91-1, Dartmouth College, Hanover NH USA. http:// www-nml.dartmouth.edu/ Publications/internal_reports/NML-91-1 , 1991.

[47] B.M. Jamart and D.F. Winter. Finite element computation of the barotropic tides in Knight Inlet, British Columbia. In H.J. Freeland, D.M. Farmer, and C.D. Levings, editors, *Fjord Oceanography*, pages 283–289. Plenum Press, 1980.

[48] B-n. Jiang, J. Wu, and L.A. Povinelli. The origin of spurious solutions in computational electromagnetics. *J. Comput. Phys.*, 125:104–123, 1996.

[49] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, 1994. 278pp.

[50] A.G. Journel and Ch.J. Huijbregts. *Mining Geostatistics.* Academic Press, 1978. 600pp.

[51] P.K. Kitanidis. *Introduction to Geostatistics.* Cambridge Universtiy Press, 1997. 249pp.

[52] L. Lapidus and G.F. Pinder. *Numerical Solution of Partial Differential Equations in Science and Engineering.* John Wiley, 1982.

[53] J.J. Leendertse. *Aspects of a computational model for long-period water-wave propagation.* Memorandum RM-5294-PR. The Rand Corporation, 1967. 164pp.

[54] C. LeProvost and A. Poncet. Finite element method for spectral modelling of tides. *Int. J. Num. Meths. Engineering*, 12:853–871, 1978.

[55] C. LeProvost, G. Rougier, and A. Poncet. Numerical modeling of the harmonic constituents of the tides, with application to the English Channel. *J. Phys. Oceanogr.*, 11:1123–1138, 1981.

[56] P Lermusiaux. Error Subspace Data Assimilation Methods for Ocean Field Estimation: Theory, Validation and Applications. PhD Thesis, Harvard University, Cambridge, MA., 1997.

[57] P.B. Liebelt. *An Introduction to Optimal Estimation.* Addison-Wesley, 1967.

[58] J. Loder, G. Han, C. Hannah, D. Greenberg, and P. Smith. Hydrography and baroclinic circulation in the Scotian Shelf region: winter versus summer. *Can. J. Fish. Aquat. Sci.*, 54(Suppl. 1):40–56, 1997.

[59] D. Lynch et al. Real-Time Data Assimilative Modeling on Georges Bank. *Oceanography*, 14:65–77, 2001.

[60] D. Lynch and K. Smith. Wind-based convolution in limited-area coastal ocean forecasting. In *CMWR04*. Elsevier, 2004.

[61] D. R. Lynch and D.J. McGilllcuddy. Objective Analysis for Coastal Regimes. *Contin. Shelf Res.*, 21:1299–1315, 2001.

[62] D. R. Lynch and F. E. Werner. Three-dimensional hydrodynamics on finite elements. part I: Linearized harmonic model. *Int. J. Num. Methods Fluids*, 7:871–909, 1987.

[63] D.R. Lynch. Comparison of spectral and time-stepping approaches for finite element modeling of tidal circulation. In *Oceans 81*, pages 810–814. IEEE Pub No. 81CH 1685-7, 1981.

[64] D.R. Lynch. Mass conservation in finite element groundwater models. *Adv. Water Res.*, 7:67–75, 1984.

[65] D.R. Lynch. Mass balance in shallow water simulations. *Comm. Appl. Numl. Meths.*, 1:153–159, 1985.

[66] D.R. Lynch. Mass balance in shallow water simulations. *Comm. in Applied Numerical Methods*, 1:153–158, 1985.

[67] D.R. Lynch and W.G. Gray. A wave equation model for finite element tidal computations. *Computers and Fluids*, 7(3):207–228, 1979.

[68] D.R. Lynch and C.G. Hannah. Inverse model for limited-area hindcasts on the continental shelf. *J. Atmos. Ocean. Tech.*, 18:962–981, 2001.

[69] D.R. Lynch and M.J. Holboke. Normal flow boundary conditions in 3-D circulation models. *Int. J. Num. Methods Fluids*, 25:1185–1205, 1997.

[70] D.R. Lynch, J.T.C. Ip, C.E. Naimie, and F.E. Werner. Comprehensive coastal circulation model with application to the Gulf of Maine. *Contin. Shelf Res.*, 16(7):875–906, 1996.

[71] D.R. Lynch and D. Paulsen K. Time-domain integration of the Maxwell equations on finite elements. *IEEE Trans Antennas Propagat*, 38:1933–1942, 1990.

[72] D.R. Lynch and D.J. McGillicuddy. Objective Analysis for coastal regimes. *Contin. Shelf Res.*, 21:1299–1315, 2001.

[73] D.R. Lynch and C. E. Naimie. The M_2 tide and its residual on the outer banks of the Gulf of Maine. *J. Phys. Oceanogr.*, 23:2222–2253, 1993.

[74] D.R. Lynch and C.E. Naimie. Hindcasting the Georges Bank circulation, part II: wind-band inversion. *Contin. Shelf Res.*, 22:2191–2224, 2002.

[75] D.R. Lynch, C.E. Naimie, and C.G. Hannah. Hindcasting the Georges Bank Circulation, Part I: Detiding. *Contin. Shelf Res.*, 18:607–639, 1998.

[76] D.R. Lynch and K.D. Paulsen. Origin of vector parasites in numerical Maxwell solutions. *IEEE Trans Microwave Theory and Technique*, 39:383–394, 1991.

[77] D.R. Lynch, K.D. Paulsen, and W.E. Boyse. Synthesis of vector parasites in finite element Maxwell solutions. *IEEE Trans Microwave Theory and Technique*, 41:1439–144, 1993.

[78] D.R. Lynch, K.D. Paulsen, and J.W. Strohbehn. Finite element solution of Maxwell's equations for hyperthermia treatment planning. *J. Comput. Phys.*, 58:246–269, 1985.

[79] D.R. Lynch, K.D. Paulsen, and J.W. Strohbehn. Hybrid element method for unbounded problems in hyperthermia. *Int. J. Num. Meths. Engineering*, 23:1915–1937, 1986.

[80] D.R. Lynch and J.M. Sullivan. Heat conservation in deforming element phase change simulation. *J. Comput. Phys.*, 57:303–317, 1985.

[81] D.R. Lynch, J.M. Sullivan, and K. ONeill. Finite element simulation of planar instabilities during solidification of an undercooled melt. *J. Comput. Phys.*, 69:81–111, 1987.

[82] D.R. Lynch, F.E. Werner, D.A. Greenberg, , and J.W. Loder. Diagnostic Model for Baroclinic and Wind-Driven Circulation in Shallow Seas. *Contin. Shelf Res.*, 12:37–64, 1992.

[83] J.N. Lyness and D. Jespersen. Moderate degree symmetric quadrature rules for the triangle. *J.Inst.Maths Applics*, 15:19–32, 1975.

[84] R.J. MacKinnon and G.F. Carey. Nodal superconvergence and solution enhancement for a class of finite-element and finite-difference methods. *SIAM J. Sci. Stat. Comput.*, 11(2):343–353, 1990.

[85] D.J. McGillicuddy Jr., D.R. Lynch., A.M. Moore, W.C. Gentleman, C.S. Davis, and C.J. Meise. An adjoint data assimilation approach to diagnosis of physical and biological controls on *Pseudocalanu* spp. in the Gulf of Maine-Georges Bank region. *Fish. Oceanogr.*, 7(3/4):205–218, 1998.

[86] J.C. McWilliams. Maps from the Mid-Ocean Dynamics Experiment: Part I, Geostrophic streamfunction. *J. Phys. Oceanogr.*, 6:810–827, 1976.

[87] K.W. Morton and D.F.Mayers. *Numerical Solution of Partial Differential Equations.* Cambridge University Press, 1994. 227pp.

[88] C.E. Naimie. Georges Bank residual circulation during weak and strong stratification periods - Prognostic numerical model results. *J. Geophys. Res.*, 101(C3):6469–6486, 1996.

[89] C.E. Naimie and D.R.Lynch. Inversion Skill for Limited-Area Shelf Modeling - an OSSE case study. *Contin. Shelf Res.*, 21:1121–1137, 2001.

[90] C.E. Naimie, J.W. Loder, and D.R. Lynch. Seasonal variation of the 3-D residual circulation on Georges Bank. *J. Geophys. Res.*, 99(C8):15,967–15,989, 1994.

[91] K.D. Paulsen, W.E. Boyse, and D.R. Lynch. Continuous potential Maxwell solutions on nodal-based finite elements. *IEEE Trans Antennas and Propagat*, 40:1192–1200, 1992.

[92] K.D. Paulsen and D.R. Lynch. Elimination of vector parasites in finite element Maxwell solutions. *IEEE Trans Microwave Theory and Technique*, 39:395–404, 1991.

[93] K.D. Paulsen, D.R. Lynch, and W. Liu. Conjugate direction methods for Helmholtz problems with complex-valued wavenumbers. *Int. J. Num. Meths. Engineering*, 35:601–622, 1992.

[94] K.D. Paulsen, D.R. Lynch, and J.W. Strohbehn. Numerical treatment of boundary conditions at points connecting more than two electrically distinct regions. *Communications in Applied Numerical Methods*, 3:53–62, 1987.

[95] K.D. Paulsen, D.R. Lynch, and J.W. Strohbehn. Three-dimensional finite, boundary, and hybrid element solutions of the Maxwell equations for lossy dielectric media. *IEEE Trans Microwave Theory and Technique*, 36:682–693, 1988.

[96] C.E. Pearson and D.F. Winter. On the calculation of tidal currents in homogeneous estuaries. *J. Phys. Oceanogr.*, 7:520–531, 1977.

[97] G.W. Platzman. A numerical computation of the surge of 26 June 1954 on Lake Michigan. *Geophysics*, 6(3-4):407–438, 1959.

[98] G.W. Platzman. Normal modes of the world ocean. Part I. Design of a finite element barotropic model. *J. Phys. Oceanogr.*, 8:323–343, 1978.

[99] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 1986. 818pp.

[100] G.A.F. Seber. *Linear Regression Analysis.* John Wiley and Son, 1977. 465pp.

[101] L.J Segerlind. *Applied Finite Element Analysis.* Wiley, 1984.

[102] G.D Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* Oxford University Press, third edition, 1985.

[103] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[104] C. Taylor and T. G. Hughes. *Finite Element Programming of the Navier-Stokes Equations.* Pineridge Press, 1981.

[105] L.N. Trefethen and D. Bau. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, 1997. 361pp.

[106] P. Vincent and C.LeProvost. Semidiurnal tides in the northeast Atlantic from a finite element numerical model. *J. Geophys. Res.*, 93(C1):543–555, 1988.

[107] M.A. Walkley, P.H. Gaskell, P.K. Jimack, M.A. Kelmanson, J.L. Summers, and M.C.T. Wilson. On the calculation of normals in free-surface flow problems. *Commun. Numer. Meth. Engng*, 20:343–351, 2004.

[108] R.A. Walters. A model for tides and currents in the English Channel and southern North Sea. *Advances in Water Resources*, 10:138–148, 1987.

[109] R.A. Walters and F.E. Werner. A comparison of two finite element models using the North Sea data set. *Advances in Water Resources*, 12:184–193, 1989.

[110] R. Weiss. *Parameter-Free Iterative Linear Solvers.* Akademie Verlag, Berlin, 1996. 217pp.

[111] J.J. Westerink, K.D.Stolzenbach, and J.J.Connor. General spectral computations of the nonlinear shallow water tidal interactions within the Bight of Abaco. *J. Phys. Oceanogr.*, 19:1348–1371, 1989.

[112] J. R. Westlake. *A Handbook of Numerical Matrix Inversion And Solution of Linear Equations.* Robert Krieger Publishing Company, 1975. pp.

[113] J.R. Westlake. *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations.* Robt. E. Krieger Publ. Co., Huntington, NY, 1975. Originally published by Control Data Corporation 1968. 457pp .

[114] W.L. Winston. *Operations Research; Applications and Algorithms.* Duxbury Press, Wadsworth, 1994. 1318 pp.

[115] C. Wunsch. *The Ocean Circulation Inverse Problem.* Cambridge University Press, 1996.

[116] K.S. Yee. Numerical solution of initial boundary value problems in isotropic media. *IEEE Trans. Antennas Propagt.*, AP-14:302–307, 1966.

[117] X. Yuan, D.R. Lynch, and K.D. Paulsen. Importance of normal field continuity in inhomogeneous scattering calculations. *IEEE Trans Microwave Theory and Technique*, 39:638–642, 1991.

[118] X. Yuan, D.R. Lynch, and J.W. Strohbehn. Coupling of finite element and moment methods for electromagnetic scattering from inhomogeneous objects. *IEEE Trans Antennas Propagat*, 38:386–393, 1990.

[119] M. Zhou. An objective interpolation method for spatiotemporal distribution of marine plankton. *Mar. Ecol. Prog. Ser.*, 174:197–206, 1998.

[120] O.C. Zienkiewicz. *The Finite Element Method in Engineering Science.* McGraw Hill, third edition, 1986. 787 pp.

[121] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method.* McGraw Hill, fourth edition, 1987. 648 pp.

[122] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: Volume 1, The Basis.* Butterworth-Heinemann, fifth edition, 2000. 712 pp.

[123] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: Volume 2, Solid Mechanics.* Butterworth-Heinemann, fifth edition, 2000. 480 pp.

[124] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: Volume 3, Fluid Dynamics.* Butterworth-Heinemann, fifth edition, 2000. 352 pp.

Index