Valery Ochkov · Konstantin Orlov
Volodymyr Voloshchuk

Thermal Engineering Studies with Excel, Mathcad and Internet

By general edition of Nikolay Rogalev

Springer

Thermal Engineering Studies with Excel, Mathcad and Internet

Valery Ochkov · Konstantin Orlov
Volodymyr Voloshchuk

Thermal Engineering Studies with Excel, Mathcad and Internet

By general edition of Nikolay Rogalev

Springer

Authors
Valery Ochkov
Moscow
Russia

Konstantin Orlov
Moscow
Russia

Volodymyr Voloshchuk
Rivne
Ukraine

Editor
Nikolay Rogalev
Moscow
Russia

ISBN 978-3-319-26673-2 ISBN 978-3-319-26674-9 (eBook)
DOI 10.1007/978-3-319-26674-9

Library of Congress Control Number: 2015958332

Printed on acid-free paper

This Springer imprint is published by SpringerNature
The registered company is Springer International Publishing AG Switzerland

Preface

What is the purpose of the publication of this book?

This book was actually written and published from the year 2000 [1] in the form of articles in various Russian and foreign journals or conference papers.

You can find all these “thermal engineering” publications on the two Web sites: http://twt.mpei.ac.ru/ochkov/work1.htm and http://twt.mpei.ac.ru/ochkov/work2.htm.

So why do you need an additional publication in book form? In particular nowadays, the traditional (“paper”) book publishing is going through hard times. Usually are only those books published, where the costs of publication are paid in advance, either by the author, state, university, research institute or a business organization. In this case, you have to save on everything: on the reviewers, scientific and literary editors, proofreaders, etc. Therefore, the quality of many books can be reduced.[1] Therefore, still numerous inaccuracies and/or errors remain in the text, and because of this, books often unfortunately are overlooked. Publishers often have no alternatives to this method. The competition between traditional publishing and electronic (by Internet) publishing is very intense. Often, someone just scans the book, posts it on the Internet, and consequently, blocks its sale fully or partially. Further details on this topic are provided in [2].

However, it was possible to publish this book, and the reader is able to hold it in his hands and read it. What is the price—this is the second question. The authors do not think about the commercial success of the book. The most important thing for the authors is that this book is useful for the readers. It does not matter how the book published: hard copy, electronic form including live Internet calculations, etc.

So, why this book was published?!

“There is nothing more pleasant than to smell a printed book”, especially, if you are one of the authors and you can see your own name on the book. The Internet has nowadays a great speed and accessibility, but traditional technical books usually are published as hard copies. If you have printed version of the book, you can present it

[1]It is often said: “reduced to the level of the Internet”. However, the Internet has also many decent titles with proper content and design.

to somebody, to a research institute, or even to a university. This cannot be done with an Internet site. Although an Internet site has also its own advantages, you can hotfix bugs and add new material or diagrams, illustrations, etc. We will not argue about what is better—the book or the Web site, and will actively take the advantages of both "media products." Another advantage of paper books is that you can read on the couch or even when you are on a trip. But nowadays, there are also e-books or tablet computers, with which can also read on the couch and additionally make electronic bookmarks and fast search in the text. The authors both fear and hope that after their book is published, someone will scan it and post it on the Internet (see above). They fear because the publishers and authors lose money, and they hope because it is an indirect indication that the book is interesting and the readers demand it. Bad books, which nobody reads, are not pirated and posted on the Internet.

Now, about the authors of this book.

There are Valery Ochkov, Konstantin Orlov (Moscow Power Engineering Institute—www.mpei.ru and Thermocenter, named V.P. Glushko, Joint Institute for High Temperatures, Russian Academy of Sciences—www.jiht.ru) and Volodymyr Voloshchuk (National University of Water Management and Natural Resources Use (Rivne, Ukraine—http://nuwm.edu.ua). Nikolay Rogalev (Moscow Power Engineering Institute) is an editor of this book. Usually, you note which of the authors of the book have written which chapter of the book. However, in this book, it is hard to do. Here, it is better to talk about the contribution by each of the authors. One of them created a function on the properties of substances (working fluids); the other one created a Web site and the third one wrote and debugged Mathcad programs. Most texts were written by Valery Ochkov and edited by Nikolay Rogalev and all of the other authors. If the reader encounters in the book the word "author," then all of the above persons will be meant. The words "I," "we," "me," "my," etc., in the texts of this book are all its authors and readers are meant, like "as we noted above," "we see on the chart," etc.

In this book, the reader will not find the traditional description of thermal engineering processes. The authors do not add anything new to these descriptions and only rewrite existing texts[2] available in "more respectable books" [3–7]. The purpose of the book is how to use modern software tools, in particular Mathcad and the Internet for the calculations of thermal engineering processes.

The book contains *Studies*. What is this? In the language of musicians and chess players, "Studies" ("chess problems" and "musical sketches") are a little exercise to practice some elements of the play/game. The studies of this book do not seek to teach those who wish to work professionally with Mathcad. The studies show the basics of working with Mathcad, "work out some elements of the play/game" in the environment of the program. The examples showed for simple and clear problems involving current issues of computational mathematics (solution of equations and systems—algebraic and differential, optimization, statistics, mathematical

[2]Therefore, by the way, many authors of study books do this.

modeling, the technique of symbolic transformations, graphs and animations, etc.). The word "study" in art means a sketch, which is a part of a future picture. Studies of this book are not only exercises, but also a kind of sketches, where the attentive reader will find many useful tips and interesting ideas that will be useful to him in the future when writing large Mathcad paintings, related not only to thermal engineering.

The Web site of the book is https://www.ptcusercommunity.com/groups/thermal, where readers will also find how to contact the author. A lot of effort and money are invested in the Web site. The development of the Web site was supported by grants of the Russian Foundation for Basic Research (www.rffi.ru) and the budgetary funds allocated for the development of the Moscow power Engineering Institute (www.mpei.ru) as a national research university.

Contents

Introduction

Nowadays, engineering, scientific, and technical calculations are carried out on computers. This naturally also applies to thermal engineering calculations.

There are established and successfully operated powerful programs for such tasks, in particular, for the calculation of the thermodynamic cycle. The programs are based on a “black box,” in which raw data arrays are put, “close the lid” (press “calculate”), open the “lid of the box, and take the response out of it”—these are the parameters of the design of the heating equipment.

But you always want to have at least a general idea of what is in the “black box,” to know how the calculations are carried out and to see the intermediate results or all the formulas by which they are conducted. In addition, in order to open the lid of the box, it is useful to study mathematical models embedded in the calculation. It is also necessary to remember that any kind of these powerful programs for thermal calculations cannot be used very often for nonstandard, but for current and operational tasks, which you can principally solve with “pen on paper,” but to speed up the calculations and avoid errors, the computer is used. In addition, we must remember that these “program-monsters” are very expensive and require complex and expensive servicing and that their study requires a lot of time and effort, which is never enough.

On the other hand, there are universal, cheap (and in some cases even free), and easy-to-learn programs for engineering, scientific, and technical computing like Excel, Mathcad, MATLAB, Maple, Mathematica, and others. The Mathcad[3] is one of the most suitable engineering purposes, because it has three advantages: universality, accessibility, and cheapness. You can download a free version of Mathcad Express (see http://www.ptc.com/product/mathcad/free-trial). You can use one month the full version of Mathcad Prime for free, and then, it is shortened to Mathcad Express.

Excel is the most suitable program for financial calculations. Moreover, you can also carry out complex calculations for scientific and engineering problems, but

[3]It is often also named engineering calculator (super calculator) or engineering office.

those calculations become confusing after a while for the author, not to mention those, who want to understand these calculations and to supplement them. This also applies to the above-mentioned “programs-monsters”, in which you can look at the source code and decompile it. However, it is almost impossible for an external to understand how the program works.

The language of MATLAB, which people often try to compare with the Mathcad, is not a mathematical one. It is a programming language, which is too difficult to master.

In addition, MATLAB is not good for itself, but its highly specialized applications, which are also some “monsters” with their inherent disadvantages noted above, are good. We should also not forget that Excel and MATLAB unlike Mathcad cannot work with physical quantities and units of measurement, which is very uncomfortable and prone to error [8].

We can also say something about the math programs Maple and Mathematica, which principally can also be used for engineering calculations. Primarily, Maple and Mathematica are programs for symbolic math and computer analytic transformations. In thermal engineering calculations, you can basically use numerical mathematics, with some elements of symbolic transformations.

So Mathcad! Why is it so good for the thermal calculations? First, of course the fact that one of the authors, who educates thermal power engineering and works in the field of power engineering, is well aware of this software and even wrote a few books about it [8–20].

But there are, of course, also objective conveniences for the work with Mathcad. Here, they are as follows:

1. Good documentation of calculations

You can print out the calculation, which is made in the environment of Mathcad, and give it to an examination or review to someone, who has never worked on a computer. Mathcad calculations are similar to calculations made on paper by the system WISIWIG (What You See Is What You Get). What you see (on screen), you will get (on paper). Printouts of calculations made in Mathcad can be left in the archive in order to read them in 50–100 years. The combination of documentation and calculations performed in natural math notation will allows you to understand what has been written in the past and reproduce it without much effort in a new software environment, which will be available by then. This is very important. Nowadays, we can see a crisis in information technology (IT), which people name “nightmare of old software.” Imagine a company, corporation, or university that developed and accumulated in 30–40 years of intensive use of computers a large number of programs of varying complexity to compute and simulate a variety of processes and also a lot of equipment and technology in various fields of science. New computers, which replace old outdated computers, come with new hardware and also new operating systems. Computers are combined in local area networks, which do not remain aloof from the process of “globalization” and integrate into the Internet—wired or wireless. The development of wireless computers to servers was the cause of the appearance of “clouds” in information technology, which will be discussed in the book. Modernization of the computer park often leads to the fact that older applications will refuse to run on new or upgraded computers, workstations, and servers. Sometimes, you just cannot read the program from the media (punched cards, punched tapes, floppy disks of different diameter, obsolete “flash cards,” etc.), because new computers do not have the appropriate reading devices (“UI”). Either you have just to abandon such programs and to create new ones from “zero”, or you have to spend time and money to create or acquire a certain utility (emulator) to run old programs on new or upgraded computers. But it is not so bad. The real trouble begins when the professionals, who establish and maintain these programs leave the firms, corporations or universities, and the younger people, who replace them, cannot upgrade the programs to new requirements. This stems from the lack of documentation of the code and the lack of appropriate software tools and a simple inability to correctly read a program written in the old style—(“dead”) programming languages. If you have managed to recreate or upgrade an old program using the old or new programming languages, the “nightmare of old software” will pop up again after a while.

2. Working with units

We have already mentioned that this is a very useful tool of Mathcad [8]. We add only that Excel, MATLAB, and other programming languages weaned us to work with measurement units or rather have taught us to work with dimensionless quantities and their units (SI units—pascal, kelvin, joule, watt, etc.). Keep in mind that, again, it is very uncomfortable and prone to error in the calculation. Even the basic SI units are inconvenient: The basic unit of pressure (Pa) is very small and always requires multipliers such as kilo or mega, and the temperature in kelvin is not easy to understand for normal people and requires translation into Celsius or Fahrenheit scales. The unit of measurement is completely dedicated to the second study of this book.

3. A flexible system of variable names

The variables and functions in Mathcad with a few exceptions have the same names that were defined for them in various scientific and technical disciplines long before the appearance of computers. For example, the Greek letter η with different indices represent the thermodynamics efficiency (thermal, internal, relative, etc.—see the pictures of the book). This along with the use of traditional writing of mathematical operators and functions makes the “language” of Mathcad to be available to all people (see item 1 above) without any further comment. The remaining four features are common to other so-called mathematical programs (MATLAB, Maple, Mathematica, SMath, Derive, etc.), but, nevertheless, we tell about these.

4. Numerical und symbolic mathematics

Mathcad allows us to use an extensive library of numerical methods in order to solve mathematical problems. There we can anticipate or complement the analytical solutions (successful or unsuccessful) of the problem. Mathcad originally designed as a package of numerical mathematics, which later was extended with symbolic kernel of the Maple, which then (in the 13th version) has been replaced by the kernel of the symbolic mathematics of the MuPAD software. The same "biography" can be told about the numerical part of the package MATLAB, which also had at the beginning the symbolic kernel of Maple and later was replaced by the same kernel from the MuPAD. Software programs such as Maple and Mathematica were originally symbolic mathematical programs with elements of numerical calculations.

The authors have received during the writing of this book an invitation from the company MapleSoft to test new version of Maple—Maple 2016. This version has a library of thermophysical properties of working fluids for thermal power systems. Authors solved some of problems this book in Maple 2016 and posted sheets on the website of this book and on the Maple Application Center http://www.maplesoft.com/applications.

5. In Mathcad are quite powerful and flexible tools built in for creating flat and volume graphics, and animation. This allows you to visualize the initial, intermediate, and final data, which contributes to better understanding of the calculation and the identification of possible errors and false solutions of the problem. Graphics help preparing calculations for publications, which we have also used in this book, for example, creating a chart of a thermodynamic cycle.

6. In Mathcad, mathematical operations were performed on a worksheet like on an ordinary sheet of paper from left to right and from top to bottom. But sometimes, it is necessary to change this order of the calculation, for example, not performing some part of the calculation and perform some other or perform a selected group of operators several times. This possibility is provided in Mathcad, and it is not only successfully used by advanced users, but also by those who believed that they would never be able to program. With these programming tools, Mathcad can solve quite complex problems that do not fit within the narrow confines of the sequential algorithm (from left to right and top to bottom). In Fig. 3.5 in Chap. 3, we can see the problem-solving process by successive approximation in "manual mode," when the user makes himself in Mathcad the last approximation of the previous calculations. Figure 4.1 shows how this routine work automated by programming ("while"-cycle). We noted only the first feature (tool) for programming. Two others—one is the use of *local variables* and the other is the possibility of unification of operators in *blocks* that are executed as a single operator.

7. The functions of the Mathcad are possible to expand in three different ways:

The first way. Attaching to Mathcad through the mechanism of DLL (Dynamic Link Library) for functions written, for example, in C programming language.

The second way is a referring to another Mathcad document. After such a reference, user-defined variables and functions will be available (visible, as programmers say) in the working document, which are stored in the document (file) to which the reference has been made. These and other Mathcad documents can be downloaded to your computer or local network and used as templates.

Third way. Mathcad tools allow the user to quickly write and debug functions that return, for example, the properties of working fluids, based on the formulas, tables, or graphics, taken from an external source—from paper or electronic books, as well as from the Internet.

These three features can and should be used in the calculation/document, for example, for functions with parameters as arguments to a specific point of a thermodynamic cycle and return the desired thermal properties of the working fluid at this point: specific enthalpy, specific entropy, density, specific heat (isobar or isochoric), thermal conductivity, viscosity, etc.

Without this, it is impossible to calculate thermal engineering processes. After this description of Mathcad, we start our book with Chap. 1.

Chapter 1
Properties of Working Fluids, Coolants and Structural Materials for Thermal Engineering Calculations

Valery Ochkov, Konstantin Orlov and Volodymyr Voloshchuk

Abstract You will learn how to create a function that returns thermophysical properties of working fluids, heat transfer and energy materials using interpolation and smoothing. You will also learn what a cloud function for the thermal calculations is.

As we noted in the introduction, calculations of thermal processes are impossible without a proper knowledge of the properties of working fluids. We begin with the Study by not focusing on the use of software tools such as WaterSteamPro (we will talk about this software in different parts of this book). Instead, we will talk about some unique techniques associated with new information technologies.

1.1 From Databases on the Properties of Substances to the Functions and Templates in Mathcad

Some time ago in an internet forum about Mathcad (https://www.ptcusercommunity.com/community/mathcad—see also Chaps. 6 and 12), the author saw a request by a stranger from a far-away India for some help (see Table 1.1)

The site of the chapter: https://www.ptcusercommunity.com/message/422996.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Voloshchuk
National University of Water Management and Natural Resources, Rivne, Ukraine
e-mail: Vl.volodya@gmail.com

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_1

Table 1.1 Pressure, density and temperature of any substance

	A	B	C	D	E	F	G	H	I
1	p, bar\ t, °C	-50	0	50	100	150	200	300	400
2	1	1,563	1,275	1,078	0,932	0,823	0,736	0,607	0,517
3	50	83,79	65,20	53,96	46,25	40,57	36,18	29,08	25,37
4	100	175,6	131,4	107,1	91,13	79,66	70,92	58,37	49,71
5	200	340,3	253,7	205,4	174,3	152,2	135,6	111,8	95,41
6	300	449,3	350,8	288,6	246,7	216,4	193,4	160,3	137,4

A stranger from a far-away India, which, after our conversation on the forum became the "a far-away Indian friend", asked visitors of the forum to create a function that would use as arguments the pressure ranging from 1 to 300 bar (see Table 1.1, first column) and temperature ranging from –50 to +400 °C (see the first row of the table) and then return the density of a substance in kg/m^3 (content of the table).

The author had a function for solving this problem at his fingertips, edited it a bit and then posted the solution on the forum—see Fig. 1.1. In Mathcad, the first

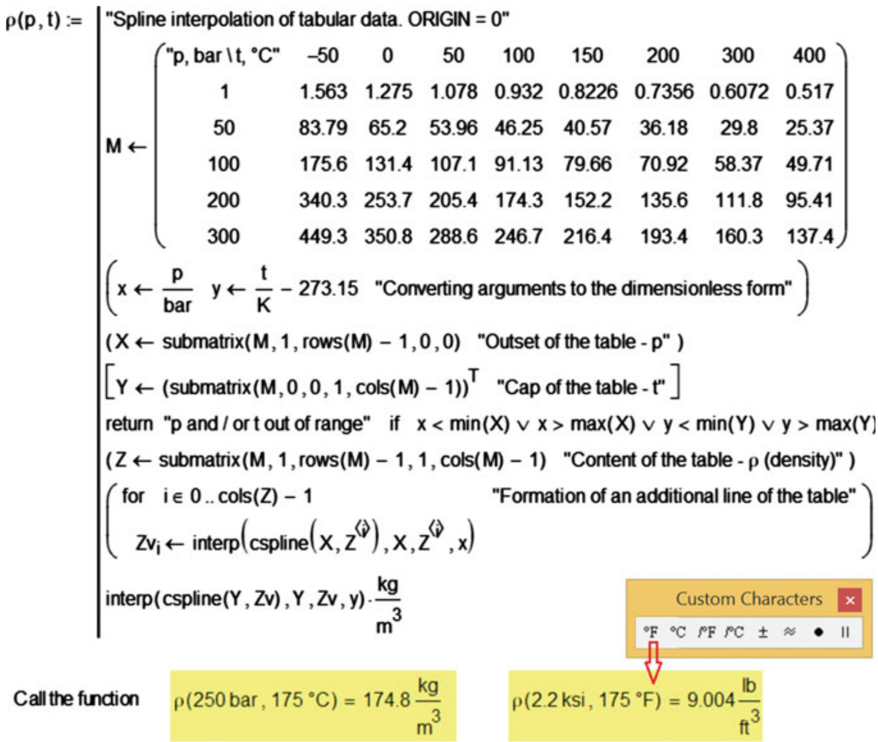


Fig. 1.1 Spline-interpolation of tabular data (Mathcad 15)

row/column designated with the index 0. By using the built-in variable ORIGIN, the user has the opportunity to change how to designate the first row/column. Therefore, for example, if the variable ORIGIN is set to 1, you can call the first row/column with the index 1.

In the program shown in Fig. 1.1, you can see how double-spline interpolation is used. What happens in the program? First, the arguments of the function ρ lose their units x ← p/bar and y ← t/K—273.15. Without this the built-in function **cspline** in Mathcad 15 would not work. In the newest version, Mathcad Prime, you can also use **cspline** with units—see Fig. 1.2. The last statement of the program (function shown in Fig. 1.1) is, because of the desired unit density. This is a common technique for empirical formulas—see Chap. 2.

The built in function **submatrix** creates a matrix out of the matrix M—the outset (vector X) and "cap" (vector Y). The line with the return statement controls the arguments of the function ρ, so they remain within the specified pressure- and temperature range (see also Sect. 1.4 "extrapolation" in this chapter below). Hereinafter the same function generates a submatrix Z—it filters Table 1.1, so there will be a submatrix created without the "cap" and the outset. Next, out of the columns of the density matrix Z an additional line (vector Zv) for a given pressure will be generated by spline interpolation. Our "intermediate" in this example will have for p = 250 bar, following elements 403.703, 306.196, 249.182, 211.977, 185.394, 165.393, 136.665 and 116.848 kg/m^3 for the temperatures −50, 0, 50, 100, 150, 200, 300 and 400 °C, respectively (see "cap" of Table 1.1). Further, in the program shown in Fig. 1.1 again spline interpolation is required for the density dependence on temperature, absent of the "cap" of the table. Spline interpolation of two vectors is shown in Fig. 3.3.

In Fig. 1.1 the last two operators show how to call the generated function ρ(p, t) for different values of pressure (p) and temperature (t) and with various European and American units. Function ρ(p, t) returns the default density (ρ) with the basic SI units

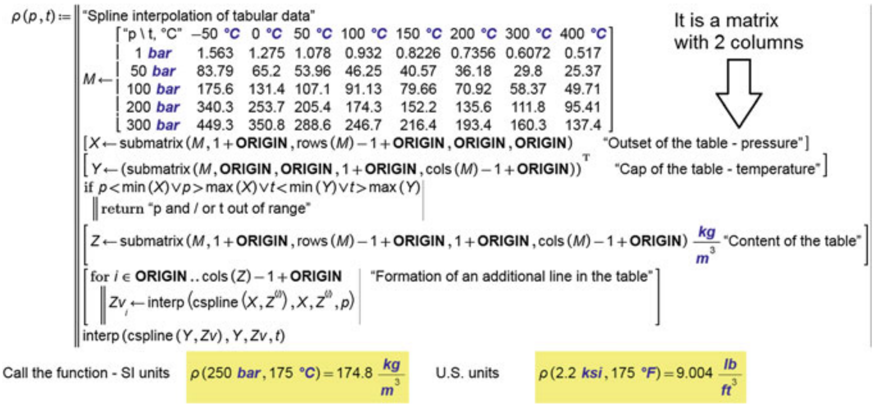


Fig. 1.2 Spline interpolation of tabular data (Mathcad Prime)

(kilograms and meters), but the user also may choose other units—pounds and feet, for example. Working with units (not just with the values and the physical quantities)—it is, we repeat, a unique feature of Mathcad, greatly facilitates calculations and prevents many errors [8], which we describe in more detail in Chap. 2.

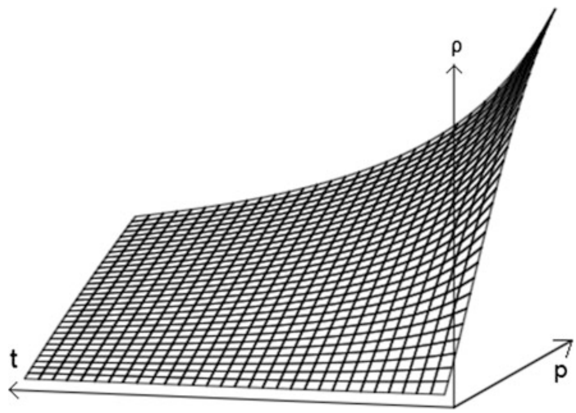
Figure 1.2 shows a program with double spline interpolation for Mathcad Prime, where it becomes possible to use a matrix with different dimension. The function shown in Fig. 1.2 does not depend on the value of the system variable ORIGIN. Again the "cap" and outset of the table in Fig. 1.2 have the units pressure and temperature. In order to save space the units of the table content (density) not shown.

For a function with two arguments, it is easy to construct the surface in Mathcad. Figure 1.3 shows that the function ρ(p, t) is smooth, increases with increasing pressure and decreases with increasing temperature.

It Fig. 1.3 shows typical "Thermophysical behavior" of many substances (gases) in the single-phase region. The graph of the function is not only useful for visual analysis of the behavior of matter, but also to fix possible errors and typos. Thus, if in the matrix shown in Fig. 1.1, intentionally or accidentally the density 107.1 kg/m^3 (Table 1.1 allocated in the frame) is overwritten by 170.1 kg/m^3 (a very common error when publishing numbers in magazines and books, and/or when manually entering numbers on a computer), the surface shown in Fig. 1.3, change its form—see Fig. 1.4. However, this is not so simple. A splash on the chart may be not just a mistake, but also maybe some anomalous behavior of substances. Therefore, when seeing such a "mistake", you need first to check before correcting it—it is either a typo, or a scientific discovery. The graph is not created by interpolation (conducting the surface through the points)—it is just created by approximation (smoothing—conducting surface near points) of the table.

Of course, you would have to immediately ask the "distant Indian friend" either what this substance (gas) is or if the formula is known. This rather ordinary episode of Mathcad problems raises a very important issue, which will be outlined in this chapter.

Fig. 1.3 Graphic display of a function (density) with two arguments—pressure and temperature



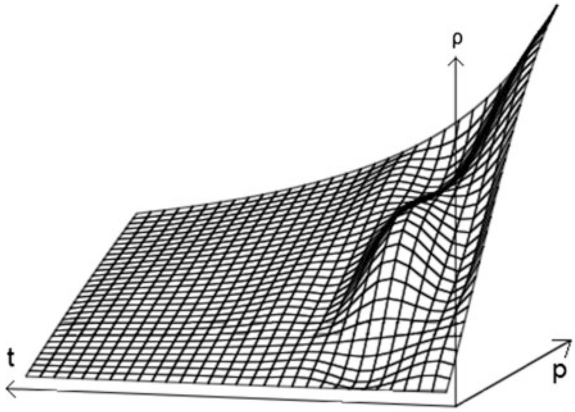


Fig. 1.4 Graphical display according to a possible error source data

Currently various scientific and educational (academic), as well as commercial organizations have accumulated a huge amount of databases on the properties of substances. These data partially published on paper or electronic media in the form of tables (a typical example—Table 1.1), graphs, formulations (sets of formulas and guidelines for their use) or computer programs. We just described such a program—see Fig. 1.1. But on the other hand, specialists, which are involved in research, design, development, operation and/or disposing of various "processes, devices and technologies" cannot always effective use of these databases. Why? First, they must find them somewhere. Second, we must ensure that they are of adequate quality—that they are certified. Thirdly, and most importantly, you need to spend a lot of time and effort to connect these databases to the working environment (Excel, Mathcad, MATLAB, other programming languages and specialized programs, etc.). Often you can see this picture: a specialist working on a computer with expensive and powerful specialized programs forced to use paper guides, the Internet or run separate programs to learn and tell the computer, for example, the density of a working fluid or any material of construction. Yes, some of these "special programs" have databases on the desired properties of substances, but usually they are quite primitive—contain only of constants and do not include, for example, the dependence of properties on some parameters (temperature, for example), do not work with new materials, or/and simply outdated. This explained by the desire of companies to reduce the cost of developing programs. On the other hand, in many engineering organizations it prohibited to manually enter external data on the properties of substances in specialized programs, despite the fact that they can be updated. This is done because of the fear, that manual inputs may include a trivial typo (compare Figs. 1.3 and 1.4) or wrong units—entering some values in °C (see Table 1.1), although the computer is set to the Kelvin scale, or (U.S. program) the Rankine scale or Fahrenheit.

Ideally your computer, if you need some properties of some substances, should automatically synchronize itself over a computer network to a ("cloud") specialized

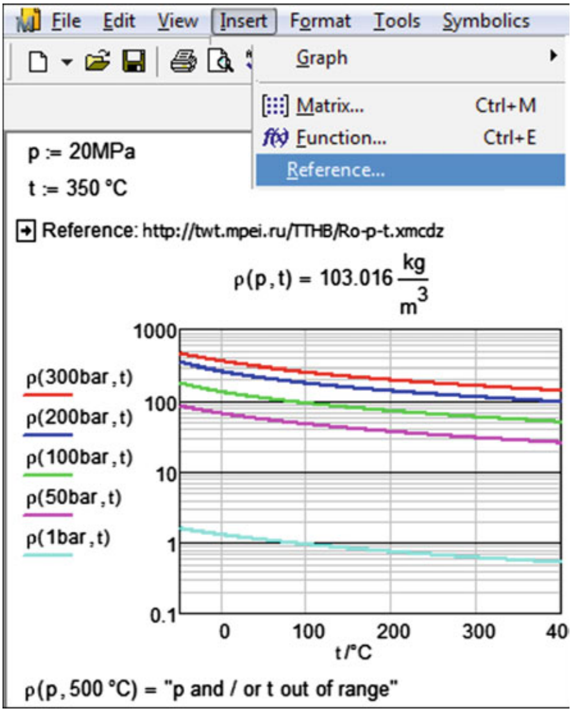


Fig. 1.5 Clickable link to external Mathcad document

and certified server with parameters of the needed substance and the server should return to the user's computer the needed properties.

How is it realized? A specific example: the Mathcad document shown in Fig. 1.1 placed on the server http://twt.mpei.ru/TTHB/Ro-pt.xmcdz. If the property you have to calculate is the density of the substance is a function of pressure and temperature, you only have to insert a reference (Fig. 1.5) to the desired Mathcad-document (shown in Fig. 1.1) and the function ρ(p, t) is visible in your calculation.

After the data entry of t and p and entering a reference to the file named Ro-pt, the "cloud" function ρ(p, t) becomes available. This function is here called with the parameters p = 20 MPa and t = 350 °C. This function further visualized by some isobars. When entering t = 500 °C (this temperature is outside of the range covered by Table 1.1), the function ρ(r, t) returns an error message.

On the current server, which is a collaboration of specialists from the National Research University "Moscow Power Engineering Institute" (www.mpei.ru), Joint Institute for High Temperatures (www.jiht.ru) RAS and Ltd "Trieru" (see www.trie.ru), collected a large number of such "cloud" Mathcad-functions. In addition, their number is growing.

These functions created for different reasons and in different ways.

First, if the author sees in a book or online a table similar to Table 1.1, he instructs his students to "revive" it by using, for example, by the program shown in Fig. 1.1. Sometimes it is enough to insert it into a new matrix and make some other minor changes. If in the book or on the Internet the formulas showed, which used to calculate the properties of substances, it is significantly easier. We have also developed a technology to transfer schedules to the "cloud" function. All of these techniques described in Ref. [21], which is released by Russian Foundation of Basic Research (www.rfbr.ru) grant fifth additional volume reference series [22].

Second, these "cloud" functions customized for Mathcad for working in the field of thermal and nuclear power industry [23]. Thus, for example, have been created and placed on the server functions on thermophysical properties of ethanol for calculation of steam turbine cycles with organic working fluids (ORC—organic Rankine cycle), for the calculation of refrigeration units [24] and other cycles. ORC posted at http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Rankin-Ethanole.xmcd.

The main part of the "cloud" functions hosted on MPEI-JIHT-Trieru, linked, of course, with properties of working fluids and coolants. These functions described in Ref. [23], and the technology they use in the calculations in the articles [24, 27]. Functions related to water and steam (which is the main working body of domestic and foreign thermal and nuclear power), based on the formulations approved by the International Association for Properties of Water and Steam (IAPWS—www.iapws.org; authors of this book are members of this Association). These formulations not based on table data, but formulas. For example, the pressure and the saturation temperature of water in the range from the triple point to a critical point are associated with quadratic equations. However, in practice very often calculation formulas replaced by interpolation from individual points. That is why we pay special attention to this issue. This does not only apply for calculating the properties of saturated (a function of one argument), but also for individual single-phase regions and areas of subcooled liquid or superheated steam (a function of two arguments). Waiver of calculation formulas and the transition to working with tables can be considered a step backwards in the historical process of creating databases on the properties of substances. At the dawn of this process, such databases published in the form of tables and/or graphs (visualized tables). Reading a table or graph without a computer or calculator is much faster than using the formulas. Loss of accuracy here was not so important. With the advent of electronic computing people slowly began to move away from tables, graphs and formulas and moved to computer programs, where these formulas programmed. Formulation process of not yet fully investigated or newly synthesized compounds carried out as follows: create a table with the so-called skeletal material properties through experimental measurements with various physical methods. Then out of these discrete tabular data one common function or a set of functions is generated by various mathematical methods for different areas or applications, and predicting possible errors for the primarily use on computer programs—this is called the formulation. However, often there is only a table published without the mathematical processing. Having at hand powerful and convenient means of interpolation—such, for example, which built in Mathcad (e.g., Fig. 1.1), you can refuse to create complex equations of state, covering a wide range of parameters of the material, and limit the amount of tables. Interpolation replaces work with equations of state for another reason. When you create, such programs for thermal and nuclear power plants and need to know the parameters of working fluids—water and steam, air, fuel, combustion products, etc., it is necessary for the calculations of properties of these substances, that they are carried out very quickly, even at the expense of a loss of accuracy. Often the interpolation requires polynoms with very high degrees, but for better understanding of the mathematical process and higher speed, polynomial with lower degrees are used. The problems with functions on the properties of substances, based on previously created databases can be described by an old "computer" situation. When there were only mechanical devices, before electronic devices, which facilitated calculations and reduced the costs, these devices could carry out only four basic arithmetic operations—addition, subtraction, multiplication and division. If, however, it was necessary to compute a more complex calculation, for example, the square root or sine of an angle, then you had to go away from computing and to use tables—search there for the desired rows and columns and if necessary to carry out interpolation. All this greatly hampers and slows down calculations and increasing the risk of errors in them.

Now a situation, which frequently occurs in calculations on properties of substances. Yes, these calculations usually carried on computers, but you often have to break away from the computer and look in the directory—paper or electronic, to clarify the nature of a substance appearing in the calculation. All this "greatly complicates and slows down the calculations and increases the risk of errors in them."

Sinus problem and other elementary and special functions (Bessel functions, for example) solved in electronic calculators by inserting these functions in these devices. Modern mathematical-, but also other programming languages can load extension packages, which include not only additional special functions (Bessel functions, for example), but also the means for specialized tasks—signal and image processing, solution of equations, statistical data processing, etc.

Embedding all functions on all properties of all substances in one modern calculating device is of course, unrealistic and not rational. You should be allowed to make visible only those functions that are needed at the moment. This information technology we did not only test on the Mathcad, but also in other environments and it described in this book.

Now we outline some of the nuances of this work.

1.2 Initial Data Nonstandard Configuration

The program shown in Fig. 1.1 is easy because the original rectangular matrix is full. If the matrix is square shaped, it will simplify the work. Spline interpolation on a square matrix in Mathcad shown in Fig. 1.1. If you want to cover a wide range of parameters of a substance by interpolation, including solid, liquid and gaseous phases, the corresponding table of material properties (density, for example) will be

"t, K \ p, MPa"	0.1	1	3	5
380	737.51	738.50	740.65	742.75
390	"-"	730.87	733.75	735.38
400	"-"	723.16	725.59	727.95
410	"-"	715.39	717.97	720.48
420	"-"	707.53	710.28	712.95
430	"-"	699.58	702.52	705.35
440	"-"	691.54	694.68	697.69
450	"-"	683.38	686.74	689.96
460	"-"	675.11	678.71	682.15
470	"-"	666.71	670.57	674.25
480	"-"	658.16	662.32	666.26
490	"-"	649.44	653.93	658.16
500	"-"	640.55	645.40	649.95
510	"-"	"-"	636.70	641.61
520	"-"	"-"	627.83	633.13
530	"-"	"-"	618.74	624.48
540	"-"	"-"	609.43	615.66
550	"-"	"-"	599.85	606.64

"t, K \ p, MPa"	0.1	1	3	5
380	737.51	738.50	740.65	742.75
390	(728.66)	730.87	733.75	735.38
400	(722.04)	723.16	725.59	727.95
410	(714.21)	715.39	717.97	720.48
420	(706.27)	707.53	710.28	712.95
430	(698.22)	699.58	702.52	705.35
440	(690.08)	691.54	694.68	697.69
450	(681.82)	683.38	686.74	689.96
460	(673.43)	675.11	678.71	682.15
470	(664.91)	666.71	670.57	674.25
480	(656.21)	658.16	662.32	666.26
490	(647.32)	649.44	653.93	658.16
500	(638.26)	640.55	645.40	649.95
510	(628.96)	(631.44)	636.70	641.61
520	(619.41)	(622.12)	627.83	633.13
530	(609.56)	(612.52)	618.74	624.48
540	(599.37)	(602.63)	609.43	615.66
550	(588.78)	(592.39)	599.85	606.64

Fig. 1.6 Not completely filled (sparse) matrix with fictitious numbers

"cracked" by lines of phase transitions. These tables usually published in various reference books on the properties of substances, for example [23]. In these tables phase transition marked by lines. Furthermore, such tables often only partially filled.

We have developed [27], an effective interpolation technology for this type of data through their breakdown into separate areas (partially filled matrix), the boundaries of these are represented by pairs of vectors (curves, fixing phase transitions), separate equations, etc. This work done in Germany [28].

Figure 1.6 shows an example of a partially filled matrix. Left—source matrix, right—matrix with added fictitious values for "smoothness" (enclosed in parentheses). These additional points generated by extrapolating the values of the matrix. Having completed the matrix, it is easy to carry out the interpolation, for example, with the algorithm shown in Fig. 1.1. When you call the function created for two input arguments, you have to make sure that the values of pressure and temperature are included in the scope of permissible values. Data shown in Fig. 1.6—is a piece of a density table of a gas condensate (see http://twt.mpei.ac.ru/TTHB/2/GasCondensat.html). The missing elements of the matrix (see table on the left in Fig. 1.6) or values framed by brackets—this is area represents the phase transition.

In the blank cells of the table you can put the system variable **NaN** (Not a Number)—see Fig. 1.7. In this case, spline interpolation cannot be carried out, but linear interpolation can be carried out, using the built-in Mathcad function **linterp**. Linear interpolation has the advantage over spline interpolation that in linear interpolation oscillations not observed—see Fig. 1.8.

$\rho_{NaCl}(t, \omega) :=$
"Density of NaCl aqueous solution as function of temperature and mass ratiol - see the table as a picture at right"
"Density of NaCl aqueous solution at t=20C from one separate tablel"

$\left(x \leftarrow \frac{t}{K} - 273.15 \quad y \leftarrow 100 \cdot \omega \right)$

return "t or/and mass ratio is out of the range" if $x > 20 \vee x < -15$
return "t or/and mass ratio is out of the range" if $y > 25 \vee y < 10$
return "t or/and mass ratio is out of the range" if $x < -5 \wedge y < 15$
return "t or/and mass ratio is out of the range" if $x < -10 \wedge y < 19$
return "t or/and mass ratio is out of the range" if $x < -10 \wedge y > 24$

M ←

"t, °C / ω, %"	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
20	1071	1078	1086	1093	1101	1109	1116	1124	1132	1140	1148	1156	1164	1172	1180	1189
15	1075	1082	1089	1098	1103	1111	1119	1127	1134	1141	1151	1160	1168	1174	1184	1193
0	1078	1086	1093	1101	1108	1116	1124	1133	1141	1147	1158	1165	1174	1181	1191	1199
−5	1079	1089	1095	1102	1110	1117	1125	1134	1142	1148	1160	1168	1176	1183	1194	1202
−10	NaN	NaN	NaN	NaN	NaN	1119	1125	1135	1144	1149	1162	1169	1178	1185	1196	1204
−15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1151	1163	1171	1180	1187	1198	NaN

X ← reverse(submatrix(M, 1 + ORIGIN, 6 + ORIGIN, ORIGIN, ORIGIN))

$Y \leftarrow \text{submatrix}(M, ORIGIN, ORIGIN, 1 + ORIGIN, 16 + ORIGIN)^T$

Z ← submatrix(M, ORIGIN + 1, ORIGIN + 6, 1 + ORIGIN, 16 + ORIGIN)

for i ∈ ORIGIN .. cols(Z) − 1 + ORIGIN

$Zv_i \leftarrow \text{linterp}\left(X, \text{reverse}\left(Z^{\langle i \rangle}\right), x\right)$

$\text{linterp}(Y, Zv, y) \cdot \frac{kg}{m^3}$

Fig. 1.7 Not completely filled matrix with NaN values and linear interpolation

Source data for tables cannot only taken from directories and the Internet, but also from other programs. Data on the properties of ethanol to calculate the above-mentioned steam turbine cycle with organic working fluid (see separate study) and some refrigerants (see separate study), were taken from the program RefProp [29]. This program can connected directly to Mathcad through the mechanism of DLL (Dynamic Link Library). However, this technology does not work with all versions of Mathcad, and only few can do it. Therefore, (alternatively) the technology described in Figs. 1.1 and 1.6 is used. RefProp generates the desired vector or matrix, which will be inserted into a Mathcad function, which is similar to that which is shown in Fig. 1.1. You can obtain a function, which returns the desired property of a substance with the right dimensions and has dimensional arguments, but does not work with equations of state, which hidden in the RefProp, and can be viewed and edited if necessary. Figure 1.9 displays the process of generating a matrix in the RefProp software with two columns that hold the temperature and pressure of ethanol on the saturation line. In the table (she left in Fig. 1.9) it is sufficient to select the desired track and copy and paste it in Mathcad-document (right part of Fig. 1.9), in which the number will be converted to a string (character string enclosed in quotation marks, where we need to remove the commas and replace them by points, and then Mathcad-function str2num to translate it into numbers). Another way to obtain direct numbers in Mathcad-document from a program like RefProp—is to transfer data to a spreadsheet in Excel, which will be transferred to a Mathcad-document. Data source tables similar to that, which shown in the left half of Fig. 1.9, can often be found on the

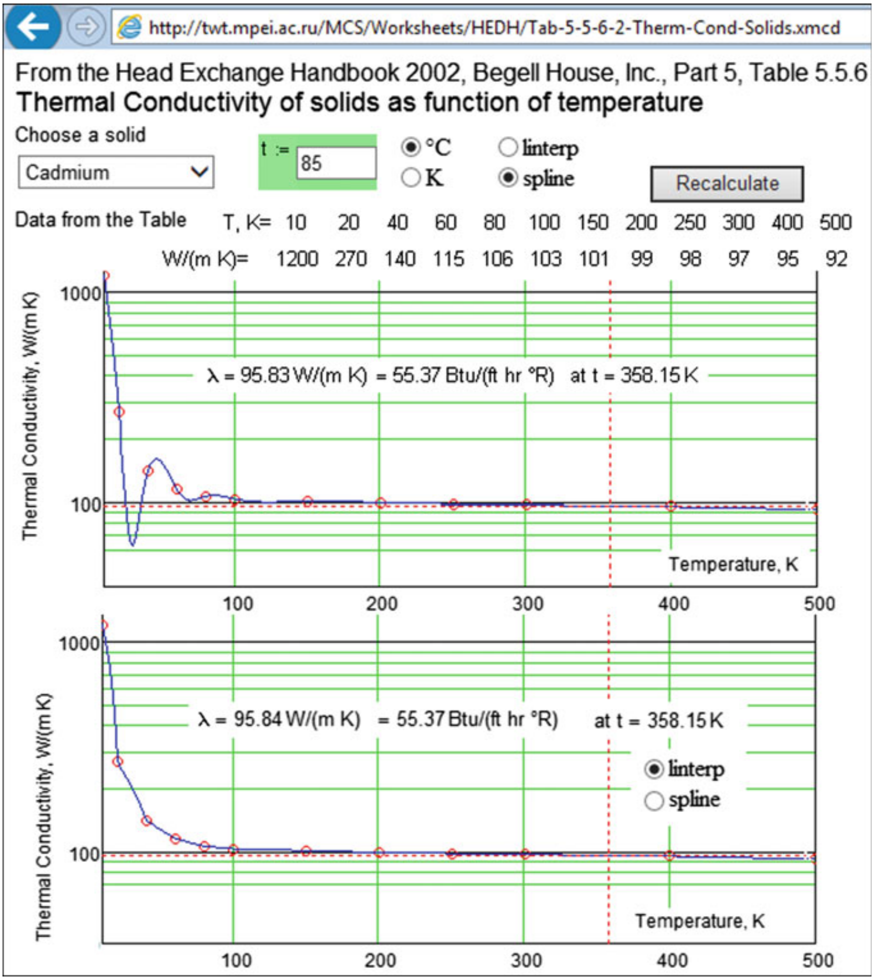


Fig. 1.8 Comparison of spline interpolation and linear interpolation: oscillation

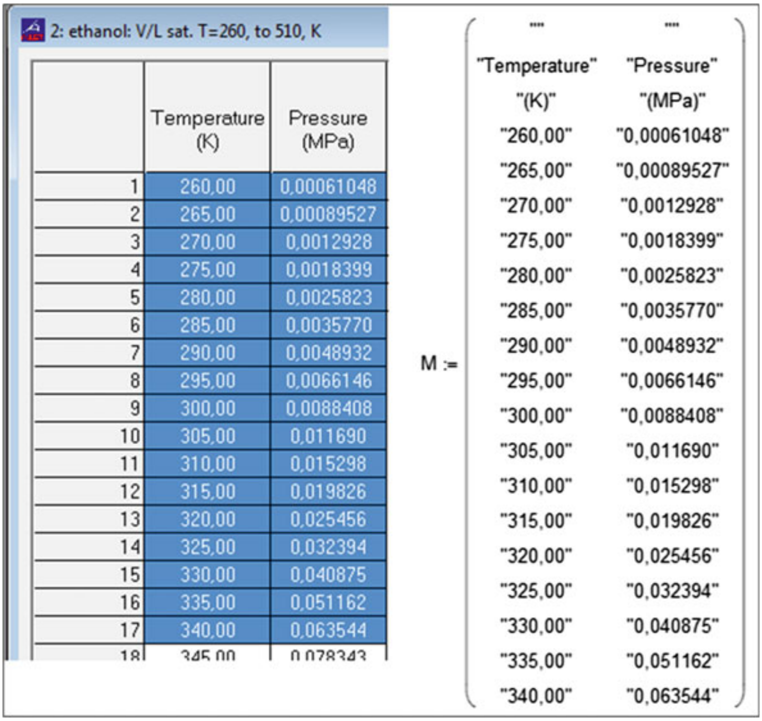
Internet. They are easy to move in Mathcad-document with the methods described above. Here the main "fuss" is about the problem of "point—comma as decimal separator". Create a function on the table with two columns is not difficult, as shown in Fig. 3.3.

The interval between two temperatures can be increased in order to save computer memory or decreased in order to increase the accuracy. Speaking of computer memory—previously, it was one of the main limiting factors of calculations. Therefore, interpolation was often refused, because of the required storage of large data sets on the computer memory, and the work on equations of state proceeded. Now this restriction eliminated.

1.3 Backward Functions Development

If there is a dependence of one parameter from another one, it is possible to create two functions—forward and backward. The establishment of a pair of functions shown in Fig. 3.3. Built in Mathcad means of solving equations and finding zeros of functions make it easy to quickly create backward functions on the properties of substances with more complex (not only unique) dependence, as well as dependencies of three or more parameters. Now we return to our problem of the density of matter, depending on pressure and temperature (Fig. 1.1). In specific calculations, we often have to solve the backward problem—finding the pressure dependent on density and temperature or the temperature dependent on density and pressure. If the equation of state known, you can try to solve analytically in this case and find its backward functions—modified equation of state. We can go another way—see Fig. 1.9.

Mathcad has a built-in function (root), which returns the zero value of analyzed functions by using bisection method (see 1 in Fig. 1.10) or secant method (see 2 in Fig. 1.10). In the first case you need to specify the range of possible values of the desired parameters for the numerical solution of the problem—temperature, keep in



2: ethanol: V/L sat. T=260, to 510, K

	Temperature (K)	Pressure (MPa)
1	260,00	0,00061048
2	265,00	0,00089527
3	270,00	0,0012928
4	275,00	0,0018399
5	280,00	0,0025823
6	285,00	0,0035770
7	290,00	0,0048932
8	295,00	0,0066146
9	300,00	0,0088408
10	305,00	0,011690
11	310,00	0,015298
12	315,00	0,019826
13	320,00	0,025456
14	325,00	0,032394
15	330,00	0,040875
16	335,00	0,051162
17	340,00	0,063544
18	345,00	0,078343

M :=

""	""
"Temperature"	"Pressure"
"(K)"	"(MPa)"
"260,00"	"0,00061048"
"265,00"	"0,00089527"
"270,00"	"0,0012928"
"275,00"	"0,0018399"
"280,00"	"0,0025823"
"285,00"	"0,0035770"
"290,00"	"0,0048932"
"295,00"	"0,0066146"
"300,00"	"0,0088408"
"305,00"	"0,011690"
"310,00"	"0,015298"
"315,00"	"0,019826"
"320,00"	"0,025456"
"325,00"	"0,032394"
"330,00"	"0,040875"
"335,00"	"0,051162"
"340,00"	"0,063544"

Fig. 1.9 Copying data from the program RefProp

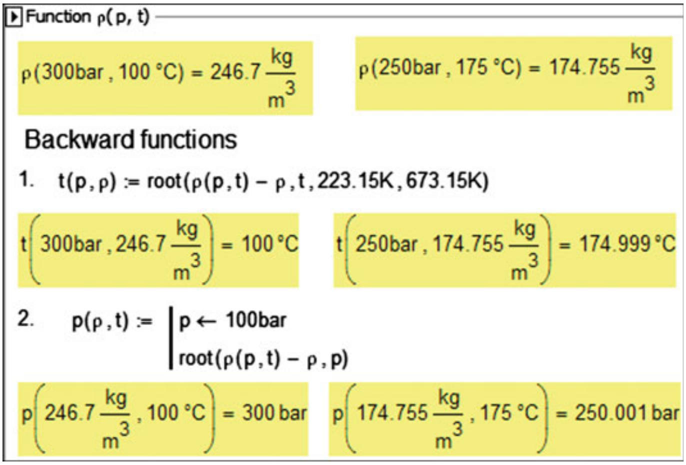


Fig. 1.10 Backward functions development

mind if you claim 1 in Fig. 1.9, while the second—the guess value to the desired parameter—the pressure (see Section 2 in Fig. 1.10).

Our function, the creation of this is shown in Fig. 1.1, is continuous and monotone dependent on each of the parameters in the selected range of pressure and temperature. When you work with more complex regions of state you should be noted that for such functions or there is no backward functions somewhat (which all have the same domain, but different value). In this case, the given parameters need to know exactly what the backward function we will seek and will need to be created in the backward function to provide an additional argument—the initial approximation to the desired solution. If it theoretically proved that in the defined area the backward function is uniquely defined, we will get the same result for any approximation. For the existence and uniqueness of constructing the backward function of one variable its strict monotony is sufficient. In general, the search for the backward function is a complex mathematical problem.

A specific example: the water pressure and/or steam temperature, and the specific enthalpy can then be either determined in the 1-phase region (pressurized water) or in the two-phase region (wet steam). This situation described in Chap. 5, wherein the compression curve is constructed in the T, h-diagram.

Sometimes it is useful as a first approximation to use a random numbers in a specified range as values of the backward function. Figure 1.11 illustrates the change of the pressure dependent on the temperature for different χ (χ—the heat capacity ratio), for χ 1.26, 1.27… 1.4. The **wspPTK** function based on function **wspKPT**, which is included in the WaterSteamPro software (see www.wsp.ru). The **wspPTK** function returns different values depending on the first approximation. Random number generator helped us to construct the required family of curves, although in fact they are not the curves, and the new family of points with

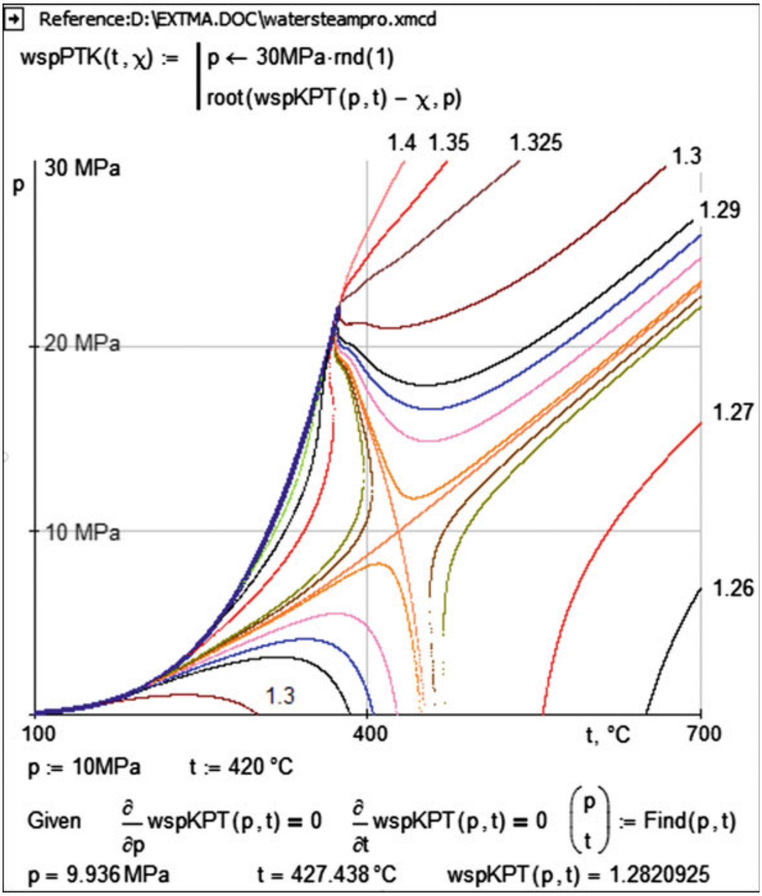


Fig. 1.11 Isentropic dependence

discontinuities in those places where the Mathcad could not find solutions function root.

In Fig. 1.11, as well as in Fig. 1.10, we use to create the backward function the built-in Mathcad function **root**. However, as a first approximation of the pressure we use a random value in the range of 0–30 MPa. This done with the built-in Mathcad function **rnd**, which returns a random number in the range from zero to the value of its argument. In Fig. 1.11 you can also see a **Give-Find** block. There pressure and temperature is calculated at which the partial derivatives of **wspKPT** are equal to zero. In mathematics, this is called a saddle point.

About Fig. 1.11 an interesting story can be told. One of MPEI's teachers promised his students, that he will give to the one, who can construct this family of curves using Mathcad, a six in the exam (not five—actually the best mark in Russia). The instructor was hundred percent sure, that this can be done only with the help of more powerful programs. Students were not able to do it, and asked the author of this book for some help. The problem solved, but none of students received a six in the exam, because they asked for help from outside.

In Mathcad, with “direct” functions on the properties of materials and tools for solving equations and systems, you can understand the essence of this problem; it is easy to create backward functions. We will show this in some studies of this book.

The issue of creating backward functions will continue in Chap. 11.

1.4 Extrapolation

The problem of creating backward functions closely related to the problem of extrapolation.

In the program shown in Fig. 1.1 the operator before the calculation checks either the inputs do fall within the specified range of pressure and/or temperature, or not. This is on the one hand good and on the other bad. Good because it cuts off any possible errors a user can do—either accidentally or deliberately entering invalid data and getting the wrong answer—the result of extrapolation, not interpolation. On the other hand, such a restriction may interfere with the procedure of calculating the backward functions (see above).

The implementation of numerical algorithms for searching zero function can go beyond the agreed limits of the function—intermediate iterations can reach higher/lower values than allowed in order to obtain a completely correct answer. In this regard, the “cloud” features on the properties of substances contain an optional argument, which allows or excludes extrapolation.

This technique illustrated by following four figures.

Figure 1.12 shows a table find in the Internet by keyword “density of the solution NaCl”. (Incidentally, in Fig. 1.12 in the table header is a link to the source, but actually, you can find this table on the following website: http://thermalinfo.ru/publ/zhidkosti/voda_i_rastvory/teplofizicheskie_svojstva_vodnykh_rastvorov_khloristogo_natrija_i_kalcija/32-1-0-225 not established.)

Figure 1.13 shows the program Mathcad returning interpolated densities of an aqueous NaCl solution, depending on its concentration (rather, mass fraction (percent)) and temperature. This table scanned from paper books in order to translate into “digital” form—an Excel spreadsheet or a matrix in Mathcad (see Fig. 1.13). This can be done by using the special software OCR. But if the quality of the original “pattern” is low or the table itself is not very voluminous, then such work can be also done “by hand”. (The National Institute of Standards and Technology, which already mentioned in this chapter, engaged this summer a group of students for this handiwork. One student dictates the table, and the second enters the numbers into a computer. This first student alternates his eyes from the paper to the display in order to check the correctness of the input.)

The matrix M shown in Fig. 1.13 not completely filled, as it repeats the data recorded in the table shown in Fig. 1.12. However, it not so—to “empty” areas of

%	ρ (kg / m³) at temperature, °C				
	15	0	−5	−10	−15
10	1075	1078	1079	—	—
11	1082	1086	1087	—	—
12	1089	1093	1095	—	—
13	1098	1101	1102	—	—
14	1103	1108	1110	—	—
15	1111	1116	1117	1119	—
16	1119	1124	1125	1125	—
17	1127	1133	1134	1135	—
18	1134	1141	1142	1144	—
19	1141	1147	1148	1149	1151
20	1151	1158	1160	1162	1163
21	1160	1165	1168	1169	1171
22	1168	1174	1176	1178	1180
23	1174	1181	1183	1185	1187
24	1184	1191	1194	1196	1198
25	1193	1199	1202	1204	—

Fig. 1.12 Table from the internet—density of an aqueous NaCl solution

the matrix M are "fictitious" number prescribed white on white. This technique works with non-standard data and described in the following [21] (in Fig. 1.6 such data registered "black and white", but in brackets.)

The function **ρ(NaCl)** checks for wrong input data and returns an error message, if the third argument of the function (Check) has a value of 1 (not zero). This shown in Fig. 1.14.

Figure 1.14 shows three calls of the function **ρ(NaCl)**: "normal" call, when the arguments of the function are within the acceptable ranges of temperature and concentration of the solution NaCl, "false", when the third argument (Check) is 0, and the function output of **ρ(NaCl)** is wrong, extrapolation value and clipping the wrong case, false answer.

Figure 1.15 also shows how to create an backward function on the basis of the function **ρ(NaCl)**.

The new function **ω(NaCl)** returns the NaCl solution concentration depending on the temperature and the density (for example: in a NaCl solution a thermometer and hydrometer was inserted and now we want to know its concentration). If the function is enabled **ρ(NaCl)** check the source data, but the backward function does not work.

The created backward function can be supplemented by operators in order to avoid wrong answers—see Fig. 1.16.

The function shown in Fig. 1.16, determines the concentration of NaCl in the solution depending on its temperature and density, without any limitation, since

```
ρNaCl(t, ω, Check) := | x ← t/K − 273.15   y ← ω/%
  if Check
    Err ← "t or / and p out fron range"
    return Err if x > 15 ∨ x < −15 ∨ y < 10 ∨ y > 25
    return Err if x < −5 ∧ y < 15
    return Err if x < −10 ∧ y < 19
    return Err if x < −10 ∧ y > 24
       ( "ω, %t \ t, °C"  15    0     −5    −10   −15  )
       (  10             1075  1078  1079              )
       (  11             1082  1086  1089              )
       (  12             1089  1093  1095              )
       (  13             1098  1101  1102              )
       (  14             1103  1108  1110              )
       (  15             1111  1116  1117  1119        )
       (  16             1119  1124  1125  1125        )
  M ←  (  17             1127  1133  1134  1135        )
       (  18             1134  1141  1142  1144        )
       (  19             1141  1147  1148  1149  1151  )
       (  20             1151  1158  1160  1162  1163  )
       (  21             1160  1165  1168  1169  1171  )
       (  22             1168  1174  1176  1178  1180  )
       (  23             1174  1181  1183  1185  1187  )
       (  24             1184  1191  1194  1196  1198  )
       (  25             1193  1199  1202  1204        )
  X ← reverse(submatrix(M, 0, 0, 1, 5)^T)
  Y ← submatrix(M, 1, 16, 0, 0)
  Z ← submatrix(M, 1, 16, 1, 5)
  for i ∈ 0.. cols(Z) − 1
    Zv_i ← interp(cspline(Y, Z^⟨i⟩), Y, Z^⟨i⟩, y)
  interp(cspline(X, reverse(Zv)), X, reverse(Zv), x)·kg/m^3
```

Fig. 1.13 Program with spline interpolation

```
ρNaCl(10 °C, 10.5%, 1) = 1080.78 kg/m^3
ρNaCl(−15 °C, 7%, 0) = 507.47 kg/m^3
ρNaCl(−15 °C, 7%, 1) = "t or / and p out fron range"
```

Fig. 1.14 Calling a function with and without scrutiny

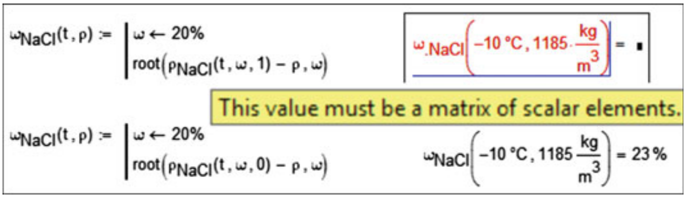


Fig. 1.15 Creating an backward function

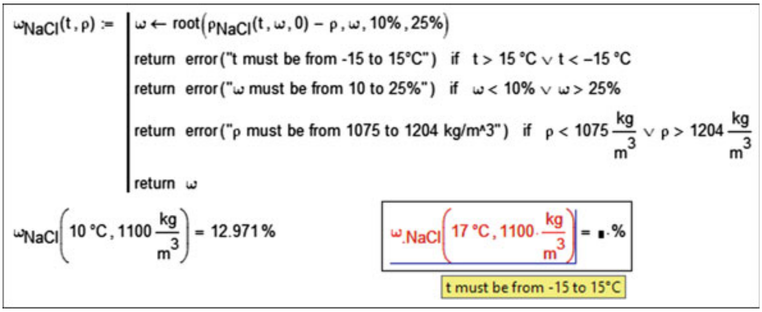


Fig. 1.16 Backward function to the control response

third argument of ρ(NaCl) is zero. But before the output response it is monitored, if the values of the function arguments of ω(NaCl) and the values of the function are in a reasonable range. The density value controlled by checking whether the value does fall within the range of the minimum density (1075 kg/m^3) and maximum (1204 kg/m^3) density—see the table in Fig. 1.12 and the matrix in Fig. 1.13.

Remark. Two pitfalls can be noted in the function shown in Fig. 1.16. First, the root feature seeking zero value of functions not near the point 20 % but within the range 10–25 %—see Fig. 1.16. Secondly, the function, the creation of which shown in Fig. 1.16, when erroneous data returns, not text message detailing the user response.

Speaking of extrapolation among Mathcad—functions with the root name spline (see Fig. 1.1) can have three preposition (prefix)—l (el), p and c: lspline, psline and cspline. These prefixes represent linear (l), square (parabolic—p) and cubic (c) extrapolation. Inside the area of tabular data interpolation is conducted by cubic extrapolation.

1.5 People of Experiment and People near Experiment

One of the authors of this book is often criticized that he "in his life has never spent a single measurement of properties of substances, and is only concerned with the fact that someone else takes the published experimental data, processes them, and puts it on his server in the form of online calculations or "cloud" functions." Sometimes this reproach enveloped in a cruder form. They say: "hit the keys of the computer—every <one rather rude word> can, and you try to run an experiment here!" These criticisms can be continued. The persons conducting experiments on the stand and measure some properties of substances can blame the author for something like this: "to conduct experiments on the final stand—every <…> can. Did you try it yourself here to collect and establish a similar stand!" In addition, more. "Collect and organize stand on the properties of substances, based on the known techniques—every <…> can, and you try to come up here and implement a new method for the study of these properties," etc.

Since quite a long time and not only in manufacturing, but also in science occurs the process of division of labour. There are scientific organizations (like the already mentioned NIST, or for example, the National Institute of Standards and Technology in the U.S.), which along with the study of the properties of individual substances are gathering previously obtained data around the world from others, and create this way databases on properties of the substances (and also of course programs). A concrete example—the program RefProp (see above). Here, the main thing is not to give out wrong information and explain where you got this information. However, in reality it is not so easy to do. Why!? You can find in many books and directories tables of data on the properties of substances, integrate them in newly created programs, and declare where they were taken from. Everything seems to be correctly. Then it turns out that in the handbook, to which you referred to, said that this table taken from a different book, and in this "other book" is a new link to another source, etc. Sometimes it seems that certain widespread table properties of the substance filled the Creator himself when creating this substance.

The author of this book, by the way, did the following experiment at the beginning of his career—he collected a stand and held it experiences. Then the fate forced him to give serious attention to computer technology. This work carried away and let him do something. When the writer hears the words that "every <…> can hit the keys of a computer", he answers (usually mentally to himself) something like that and at the ready, each laboratory bench can <…> work, and here you try to create a program that will be a success—and scientific and commercial—for example, the program RefProp [29], ThermoData Engine [30] or WaterSteamPro [31]. The scientific process of creating databases on the properties of substances and provide them to users in digital form involvs many different specialists. Everybody contributes an important part to this work! The main thing is that this work carried out professionally and meets the principles of scientific ethics. You can not only learn about nature, but also about the "second" nature, in particular, the programs created by people.

A scientist, who wants to learn about the world, cannot speak directly to the Creator—God or Nature, and should ask the questions to himself, i.e., he must carry out an experiment.

A program has authors. Their names not always listed on the boxes and documentations, but they exist. Consequently, any experiments on programs are unnecessary. Emerging questions should be addressed to either the documentation or the authors themselves. However, If a user, for example, need to clarify, either the argument of sine is measured in degrees or radians, he will not check the documentation, and just simply write x := sin(30) and see what result he gets for the variable x. Such experiments give users an option, i.e., checking the documentation only in severe cases (or not) and usually not finding the answer there. This appeals to the creator as problematic. On the hotlines do not sit the creators of the program —usually just some mildly skilled employees, and as the reader knows, it is not the same thing. Consultant firms are likely to ask you to call back in a couple of days during which they will perform some experiments with the program and try to find the answer. Yes, and according to the author, you often do nothing in the meantime, forget about the problem and will be fully absorbed to a new project. Even if it is not this case, then the author cannot remember all the details of his program after he calls back.

Because of this when working with programs the user often forgets that it is a product of the mind and human hands, i.e., which is the fruit of the work of anonymous persons, that cannot be reached via a hotline. Here, apparently, lies the philosophical explanation (but not in any way an excuse) for the widespread use of illegal copies of programs. Here we are talking about relatively honest people that put the program on your computer in order to get to know more about nature and pass on their knowledge, such as students.

We can assume that the Creator, throwing an apple on top and please them on the head of Newton, we opened one of the secrets of his Divine Plan. The merit of genius (Newton)—the ability to be in the right place at the right time. Program (and not only brilliant) would also like to fall on top of us, and there is no obligation for them to pay crazy money by Russian standards. Hence the idea of vitality and shareware, which oppose primarily sellers and not the authors of programs.

We can assume that God created from his own image and likeness, not only humans, but also the computer. Writing for a program—the inhalation of soul into a lifeless pile of iron, plastic and sand. Trading bodies (organ transplants, blood transfusions or, finally, prostitution)—is a reality of our days. Buying and selling of souls is found only in fairy tales and fiction (the story of Faust, for example). By purchasing the software, we buy only the "body"—disks, documentations, information and discounts on new versions, and most importantly—after-sales service.

Hence our conclusion: experiments with programs like RefProp [29], ThermoData Engine [30], WaterSteamPro [31] or Thermoflow (see Introduction) have the same right to life as classic experiments.

1.6 Once Again Why Mathcad?

Often you can hear the statement (another rebuke to the author) that, Mathcad is a "frivolous" program, designed only for pupils or students. Engineers and scientists need to work with more complex programs, like MATLAB for example.

What can I say?

First, MATLAB—like it is mentioned in the Introduction, is a programming language, or rather, a programming language for technical computing. The developer of MatLab MathWorks company also says about it (www.Mathworks.com). Mathcad is primarily a scientific calculator with an interface as close as possible to the "interface" of calculations written down on paper, and then a programming tool.

In order to master the program MATLAB, you actually need to master a second specialty—the specialty of a programmer. The situation in Mathcad: a scientific/technical worker or engineer, sitting at the computer for the first time with the program Mathcad, begins after a few hours to solve quite complex problems—calculating formulas (involving units; which MATLAB, we repeat, not can), solving equations and systems of equations, building complex graphs and charts, etc. In addition, all this without any loss of their qualification in their main specialty.

The program MATLAB is without special courses or "thick" books very difficult to master. Because of this, often more than one person works with MATLAB, usually two. The first level (an experimenter) forms the task, and the second (an expert in the MATLAB) translates this problem into the language of the computer. In such a chain are often glitches and inconsistencies. Of course, there are exceptions of this rule, but they are isolated.

We do not need to compare the MATLAB and Mathcad. Nobody bothers specialist of applied work with Mathcad and MATLAB. Moreover, when working with Mathcad, you can automatically send data to MATLAB, where they processed by using specialized tools in MATLAB, and return the new data to Mathcad. Such technology, incidentally, works also with a spreadsheet program, like Excel—or other software environments, which often used for scientific and technical calculations, although they intended primarily for accounting calculations. Table 1.1, by the way, is a table from Excel—it is convenient to store and edit tabular data sets there.

The Mathcad is very good for this kind of work, e.g., when you need to make certain estimates or rough calculation of a process. Here in interactive continuous interaction work knowledge and human intelligence and computer speed and accuracy. And if this process to mount the database on the properties of substances, such creative work is very productive in what we hope reader will see very soon.

1.7 Temperature Scales

In the table, placed at the beginning of the Chapter (Table 1.1 and Fig. 1.3) the Celsius temperature scale used. This is not a typical case. Usually these tables based on the absolute temperature—on the Kelvin scale. Working with different temperature scales (Kelvin, Rankine, Celsius, Fahrenheit, and sometimes Reaumur) often leads to confusion and errors, when we work with tables. Mathcad greatly facilitates the work with its ability to work with units. In addition, we must remember that old tables, graphs and formulas, which display the properties of substances, can rely on the temperature scale of 1968 (ITS-68). Currently a new scale since 1990—ITS-90, is used. To set the "cloud" features on the properties of substances applied a pair of functions, providing for a new recount temperature (ITS-90) and old (ITS-68) scales, which eliminates possible errors in the use of old, but, nonetheless, demand data. The site http://twt.mpei.ac.ru/PVHB/pvhbeng.html—a web version of the book "Physical quantity" (http://twt.mpei.ac.ru/PVHB/pvhb.html) in the "thermometer" the reader will find on-line recalculations on the temperature scale and corresponding functions for embedding them in Mathcad.

Figure 1.17 shows the difference of the both temperature scales (ITS-90 and ITS-68). On the site you can also find links to functions for recalculations of metrological standards and description of these.

In Old Russian (Soviet) references, along with the old temperature scale can be (see above) to meet and outdated symbol of the unit of temperature, the temperature

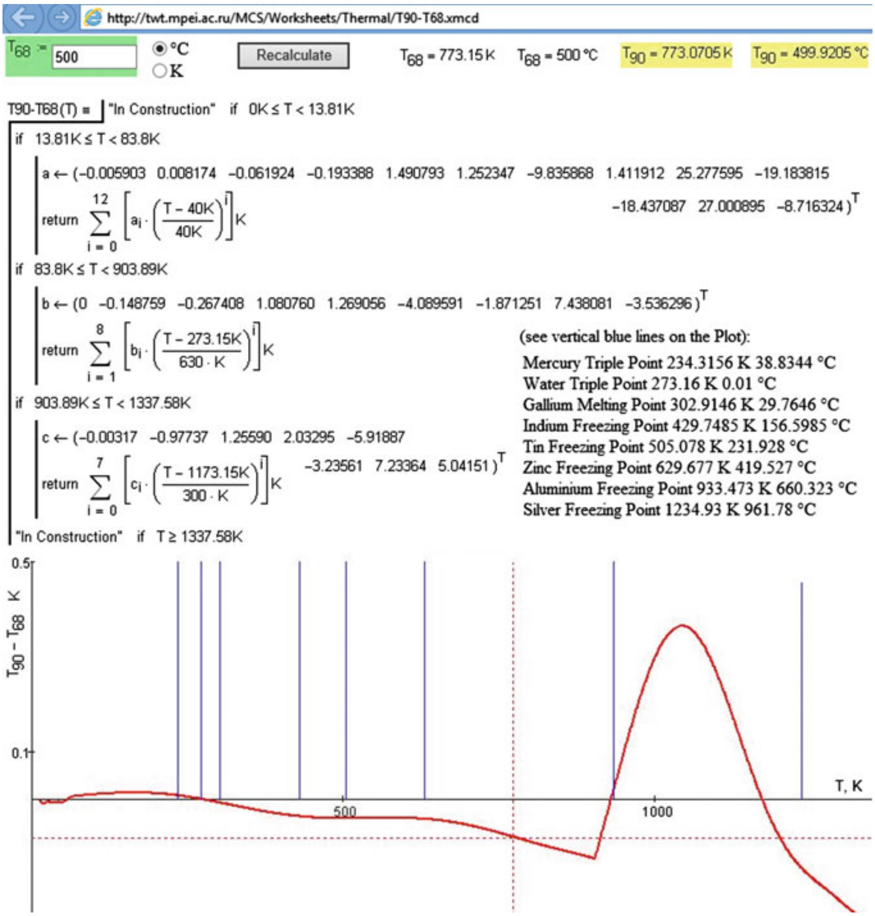


Fig. 1.17 Difference between old (ITS-68) and new (ITS-90) temperature scales

difference rather. The unit of thermal conductivity is watts divided by meter and degree Celsius—W/(m °C) and the unit of specific heat—calories per kilogram and degree Celsius—cal/(kg °C). In modern references replace Celsius by Kelvin, although not always. In U.S. references you can find as the unit of thermal conductivity—Btu/(hr ft °F), while the specific heat has the unit (or unit specific entropy) Btu/(hr lb °R). The author asked one of the authors of a handbook published by the National Institute of Standards and Technology (NIST), why they used degrees Rankine in the unit of heat capacity, but degree Fahrenheit for heat conduction and entropy. The answer was—it is an American tradition of engineering and we cannot change it. We can change the U.S. Constitution rather than the tradition.

In Mathcad, by the way, in support of this tradition two temperature units introduced Δ°C and Δ°F numerically equal to Kelvin and degrees Fahrenheit, respectively.

1.8 Uncertainly

Many properties of substances given in databases come with a certain error (uncertainly). This error fixed in various formulations on the properties of substances, but not always shown in tables and graphs.

"Cloud" functions, for online calculations on the properties of substances, described in this chapter. In these calculations, is not only the calculated value of the property given, but also the calculation error—the interval of admissible values (see Fig. 1.18).

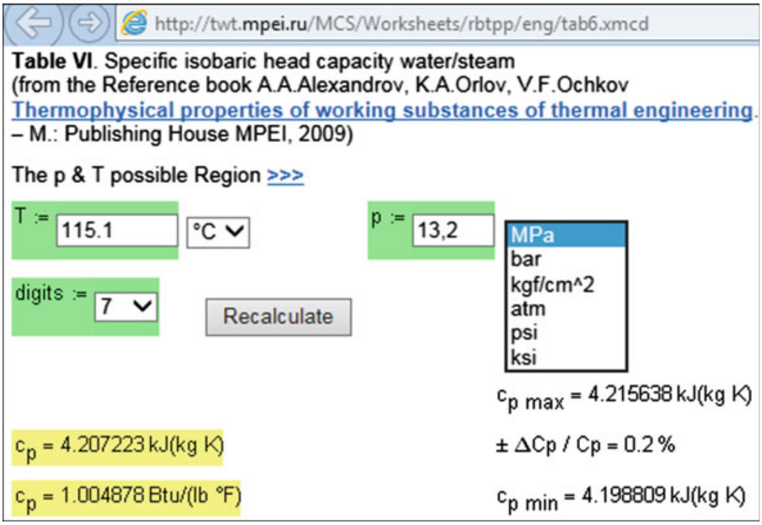


Fig. 1.18 Calculation of the specific isobaric heat capacity of water, indicating the error

If it is necessary, you can modify the “cloud”-function, which is discussed in the article, that it returns more than one value (scalar or vector)—the value of the substance itself, the error (absolute and/or relative), and the maximum and minimum value of the determined error.

1.9 IT-Security

Many “serious” organizations (research institutes, design bureaus, commissioning organizations) deny or restrict Internet access for end users’ computers. Internet access is often only available for the system administrator. This done in order to guarantee 100 % safety. These organizations can use cloud functions in this way: download them from the “cloud” and place them on the organization’s local server or end user’s workstation. In Chap. 17, we show how the same problem can be solved by referring to cloud features (Fig. 17.8) and how to work with templates (Fig. 17.11).

However, the benefits of working with cloud features are undeniable. First, these functions are always “fresh”—they are continually optimized, redefined, and extended. Secondly, it eliminates the problems associated with the loss of installed programs—by changing databases or the computer (operating system).

1.10 Templates

The problem of security (see above) is such an important aspect of information technology as the work with templates.

The fact that the technology options “cloud” function in Mathcad 15—is, frankly, an undocumented trick that, like any other undocumented, developers locked in a new version of Mathcad—Mathcad Prime. In addition, it done for security reasons. An alternative to the same references to the “cloud” function in an environment Mathcad Prime 3 are templates, including “cloud”.

What is this template?

For us, the more familiar term is the term form, which goes back to typewriters and writing with a pen on paper. If we need to write a formal or informal letter, invitation or something else, we do this by using a form or template, as it is now more common to call. Such templates used in almost all organizations. They have been inserted earlier in typewriters, and then in printers. Now we use electronic forms. For paper forms for printers now usually resort to if the form color and the text on it is printed on black-and-white printer.

Blanks integrated in many text editors. Thus, Fig. 1.19 shows a part of the screen after starting Word 2013, with which, by the way, this book was written.

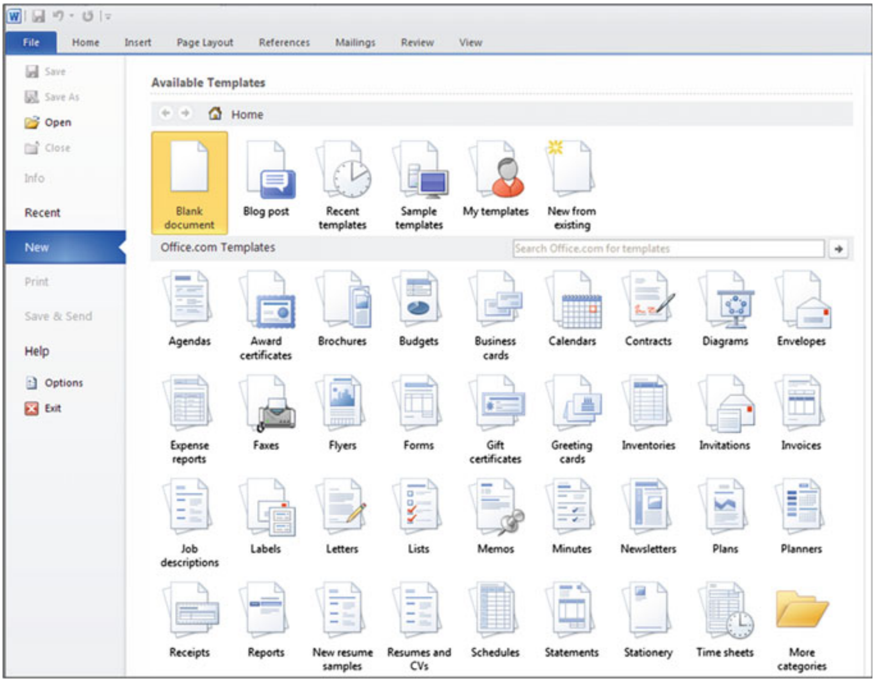


Fig. 1.19 Word templates

Figure 1.19 shows how Word offers its users not to start writing on a blank sheet of paper (which, incidentally, is also a template, however, a primitive one), and to select the appropriate template.

The ready-made Word document (official letter, for example), we repeat, consist of two parts—a constant (name of organization, its details and so on.), which was in the template and a variable part—what we enter in the template (main part of the letter). All this significantly accelerates and simplifies the work, and reduces the risk of errors therein.

Mathcad also comes with a few templates. Usually they are rarely used when working with Mathcad due to the fact that after the launch of this program not a list of patterns (as in Word—see Fig. 1.19), and a blank document (the most primitive pattern) appear. If we press the small button next to the icon with the image of a new blank document in Mathcad and give the command "New" from the File menu (Mathcad 15), then the screen will display a list of templates, which are built into Mathcad—see Fig. 1.18.

Not only built in templates can be used (see Figs. 1.20 and 1.21), but also user created ones. Creating a template is easy—you need to open a built-in template, change something in it and save it in the folder "template" with a new name and the ending *.hmct. The same operation can be also done with a finished document.

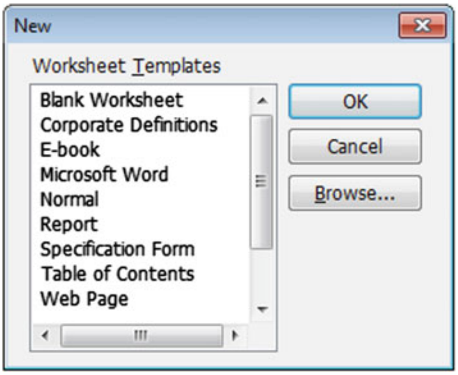


Fig. 1.20 Built-in templates in Mathcad 15

Figure 1.19 after adding a new template the user can use his own created template just like the built-in templates.

If you open a custom template, as shown in Fig. 1.21, Russian units can be used in the Mathcad document (if Russian units have been introduced in this template), which are not built into Mathcad—see Fig. 1.22. (You can also refer to the “cloud” Mathcad document with full name http://twt.mpei.ac.ru/ochkov/units.xmcdz, and some additional Russian and English units of measure become available. They also used in the “cloud” document named http://twt.mpei.ru/tthb/H2O.xmcdz, to which we often refer in this book.)

Custom pattern shown in Fig. 1.22, a built-in Mathcad standard template (Normal), which inserted in an area named Additional units. In this area, the operators stored kJ: = 1000 * J, g: = gm, etc., i.e.

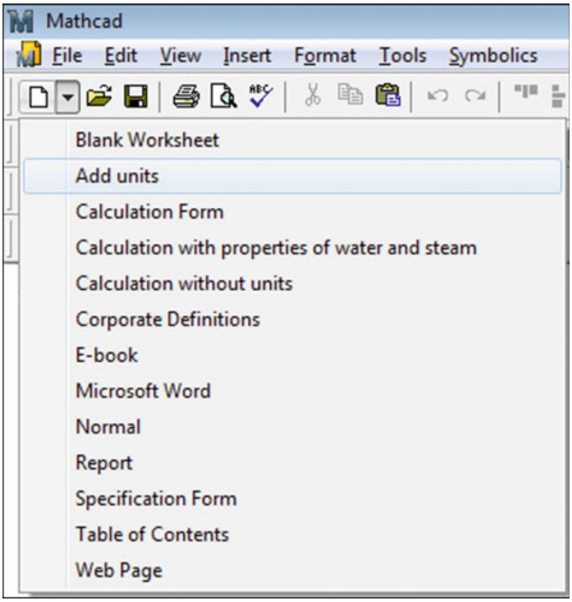


Fig. 1.21 List with custom template

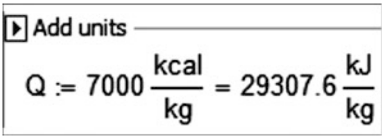


Fig. 1.22 Working with custom template

Among Mathcad Prime 3.0 the opportunities to work with templates are expanded. In particular, you cannot only use your own templates (My Templates—see Fig. 1.23), but also templates from all users of the local computer network (Shared Templates) and standard templates from the firm PTC—Developer Mathcad.

With the ability to share templates among all users in the network, it is possible to replace Mathcad servers in those organizations, which for various technical reasons and/or for security reasons do not have access to the Internet.

Mathcad templates primarily contain no text (except heading, etc.) and calculation elements. Thus, Fig. 1.24 shows a template for the use with aqueous NaCl

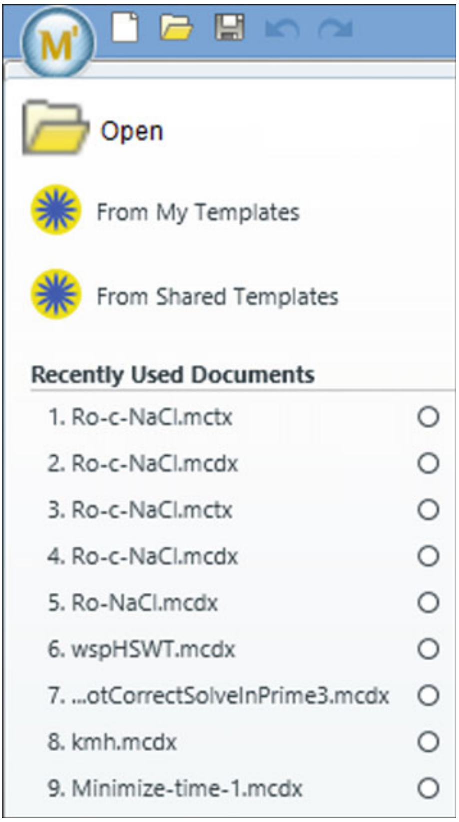


Fig. 1.23 Opening Templates in Mathcad Prime 3.0

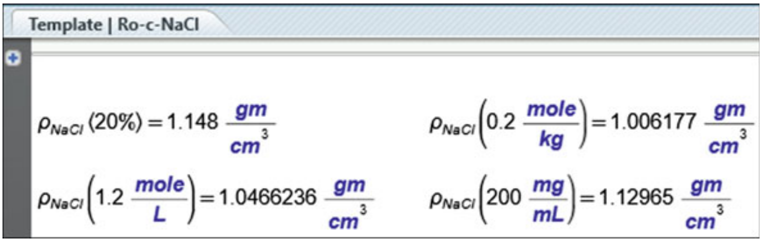


Fig. 1.24 Template with the properties of aqueous NaCl solution

(this will be described in detail in Chap. 3). In this pattern are following functions integrated: a function that returns the density of NaCl in aqueous solution depending on the concentration, expressed in different ways—the weight percent (%), molality (mole/kg), molarity (mole/L) and the titre (g/mL), the backward function (concentration of density), a scaling function of concentration depending on the function of the freezing point of the solution, and other useful functions. All this considerably simplifies and accelerates the calculation where this solution appears—for example, the calculation of air conditioning systems with this solution as a coolant or refrigerant.

Looking ahead, we can, say that a template for Mathcad 15 may contain links to cloud features on the properties of water and steam—the main working body in engineering sciences.

Figure 1.25 shows the calculation of the density of water for a given temperature and pressure with a function named **wspDPT**, which is stored in the "cloud" file named **H2O.xmcdz**. More details about this calculation described in Chap. 6 "Maximum density of water".

Working with templates can greatly simplify the calculations. Starting a new project does not require opening a new blank document and not be lazy and search the World Wide Web right template with the right functions. The used template can be edited and expanded—add to it, for example, new functions and return it back to the Internet, providing new functions for other users. It is one possible way of

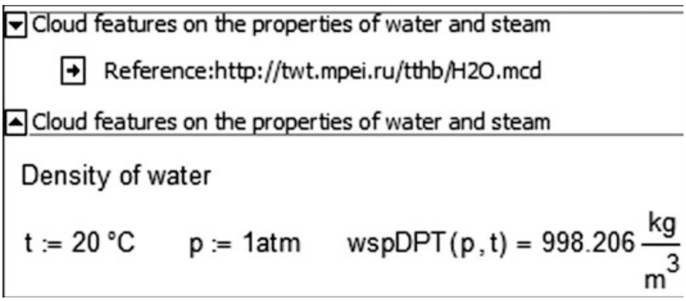


Fig. 1.25 Template with the properties of water and steam

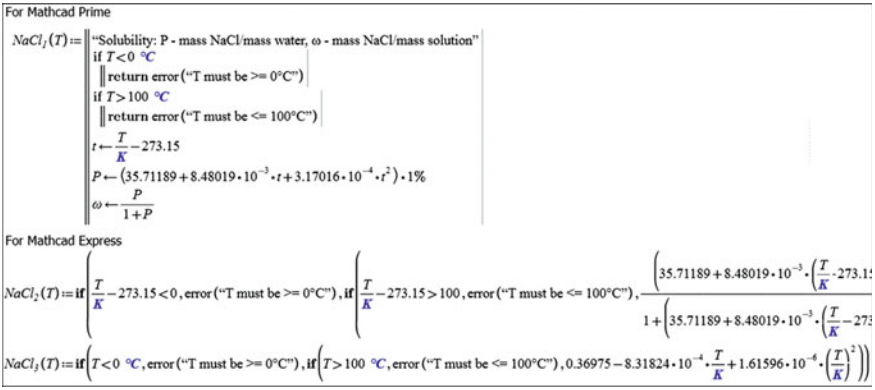


Fig. 1.26 Function solubility NaCl: two options

communication about databases and calculations on properties of substances and CAD programs.

The free and shortened version of Mathcad Prime—Mathcad Express has not programming tools. In this connection, will be very useful to create functions that return the properties of substances, without programming. Figure 1.26 shows three functions with different appearances but the same content. They return the solubility (wt%) of sodium chloride in water depending on the temperature.

The dependence of the solubility of sodium chloride on the temperature well described by a second-degree polynomial. Many calculations about solubility provided on the site: http://twtmas.mpei.ac.ru/mas/Worksheets/Chem/solutions.html. The calculations based on the tables from the Web reference book "Analytical Chemistry" (http://twt.mpei.ac.ru/MAS/Worksheets/Chem/solutions.html). Figure 1.27 shows such a calculation for an aqueous solution of NaCl.

In Fig. 1.26 we can see three functions named NaCl with the subscripts 1, 2 and 3, which return the solubility of sodium chloride depending on the temperature. The first of them created using programming tools. The second function (only a part of it) uses only the built-in Mathcad function that allows this function with the name of NaCl work as the full version of Mathcad Prime, and in its shortened version of Mathcad Express. Nevertheless, the first function (with programming elements) is more transparent in terms of its understanding and provides more possibilities for editing. The third function is a simplification of the second function. It different polynomial coefficients because once the temperature considered Kelvin, Celsius, and not that it outputs a value polynomial ω (mass ratio of solute to the mass of the solution), not **P** (ratio of the mass of solute to the mass of water).

Solutions devoted to concentrations will be discussed additionally in Chap. 3. Now it may be mentioned that in different books can be found widely differing data on thermophysical properties of fluids. This occurs in part because that some references realize the percentage ratio of the weight concentration of the solute in the

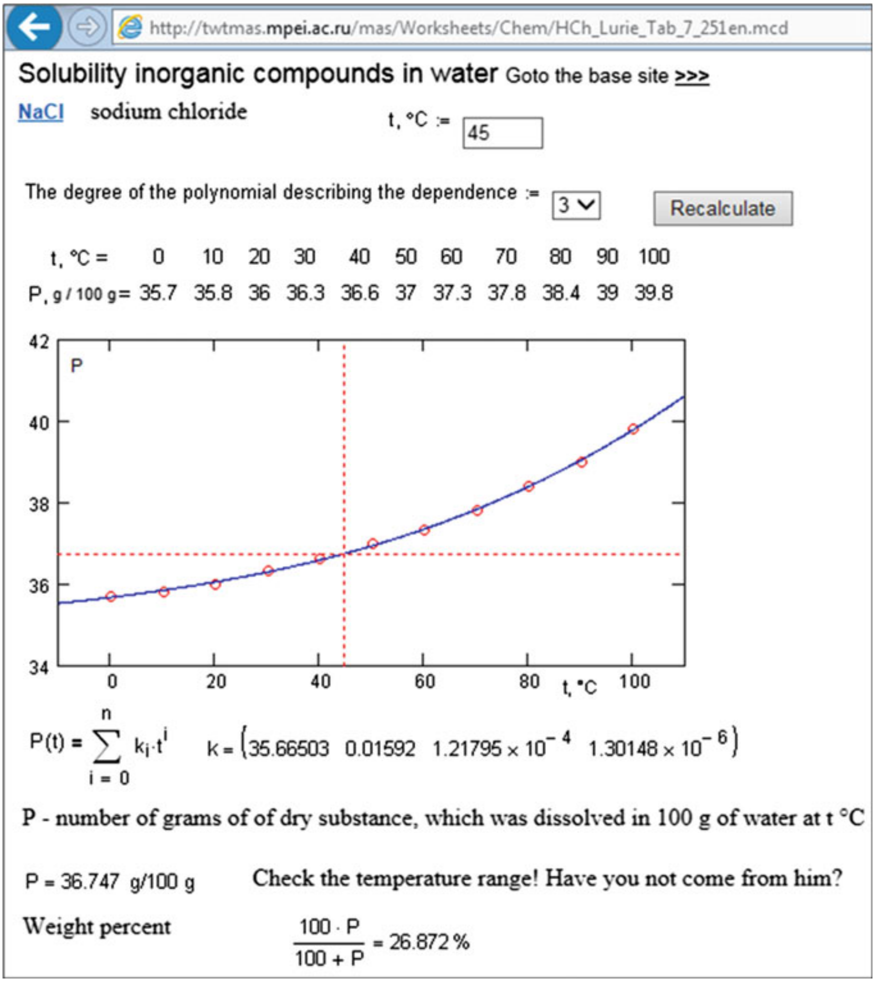


Fig. 1.27 Online calculation of the solubility of NaCl in water

bulk solution, to the other—the ratio of the mass of solute to the mass of solvent, and the third—the ratio of the amount of solute to solvent, etc. So use this data with great care.

Another modern way to publish data about properties of substances is to create a website, which contains a code for specific programs. Thus in Fig. 1.28 you can see a website, which generates a code for engineering calculations with MATLAB.

In conclusion we can say that the International Association for the Properties of Water and Steam (www.iapws.org) uses for interactive website design calculations basic properties of this working fluid and material engineering (e.g., Fig. 1.28). There's a link to a white paper with instructions how to calculation the thermal conductivity of water and steam (PDF of document), then a short description of this

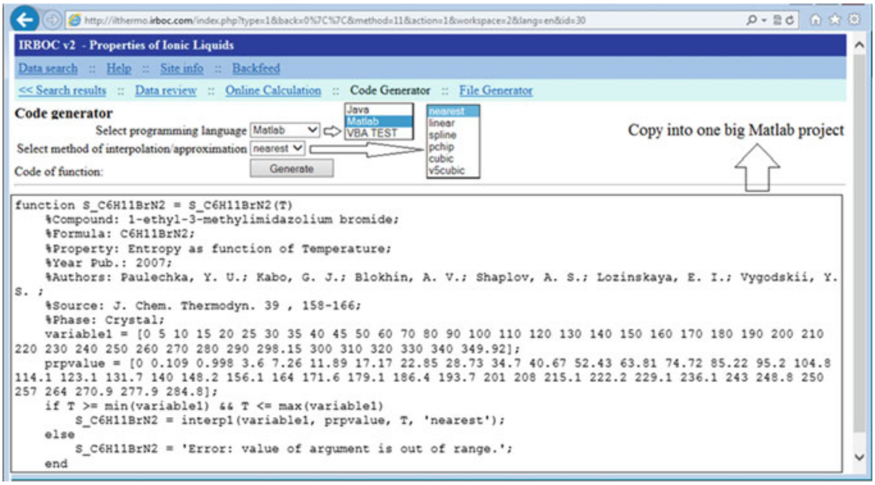


Fig. 1.28 Internet generation user functions on the properties of substances for MATLAB

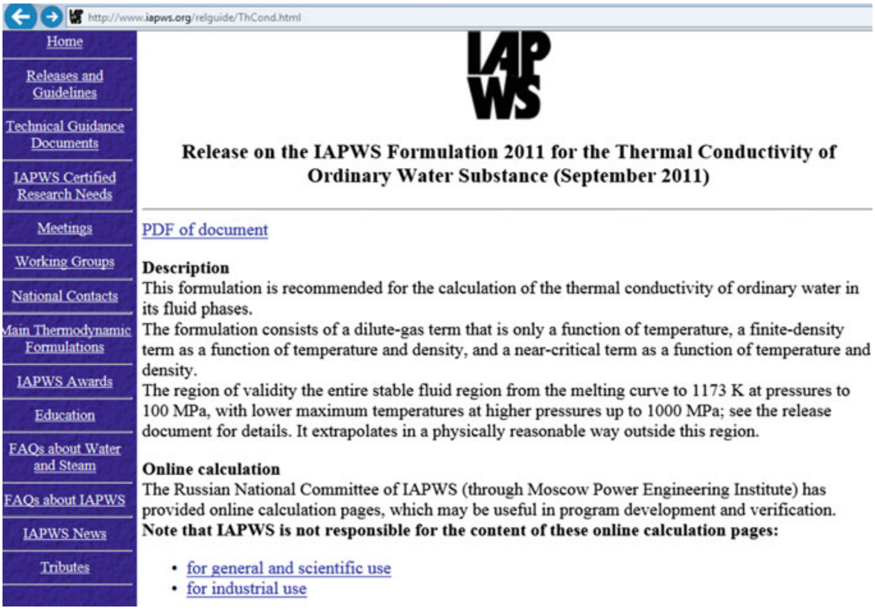


Fig. 1.29 Link to the on line calculation website from the site IAPWS

document (Description) and references to "live" calculations of thermal conductivity (Online Calculation) for general and scientific use (a function of density and temperature) with and sharing of, for industrial use (a function of pressure and temperature). After the data input is given not only the final result of the calculation,

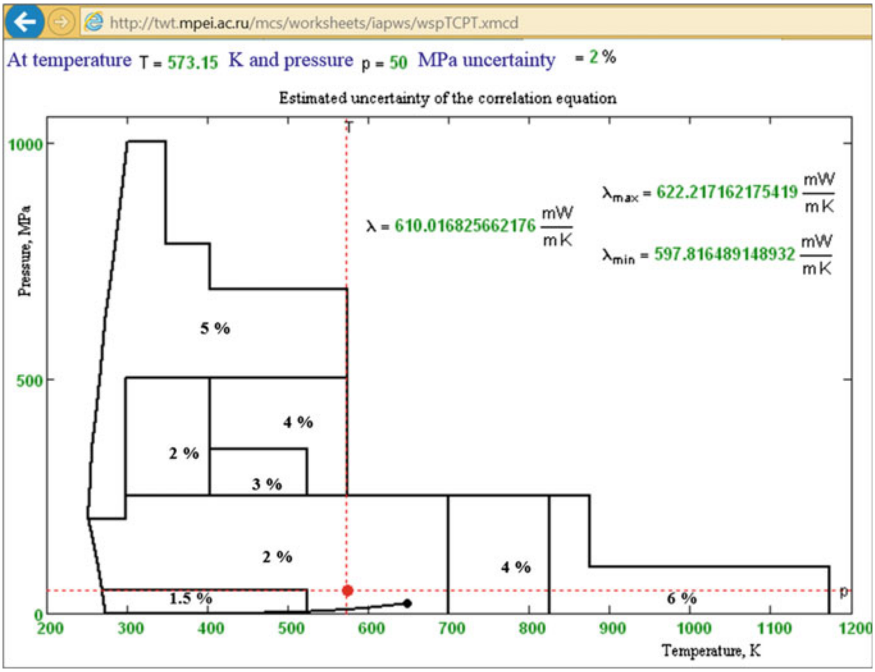


Fig. 1.30 Page of the site with an interactive calculation the thermal conductivity of water and steam

but also all intermediate data. When debugging a program comparing intermediate calculations will quickly find and resolve the error, if it took place. In addition, the "live" calculations, links to which are shown in Fig. 1.29, contain diagrams with area where operating point is marked, and the value of the calculation error is shown (Fig. 1.30).

Chapter 2
Working with Physical Quantities: Problems and Solutions

Valery Ochkov and Konstantin Orlov

Abstract This chapter deals with the use of dimensional quantities in thermal calculations. The main aspects of work with dimensional variables in thermal calculations are given. The concepts of physical and empirical formulas are extended by pseudo-empirical formula concept. Few samples are provided.

Programming languages and spreadsheet programs usually calculate with simple quantities (numbers) rather than with physical quantities, like mass, velocity, energy, etc. That reduces the "readability" of the calculations, slows them down, and is additionally a source of errors [8].

The consequences of misusing units can be extremely significant and even cause real disaster. Here are three examples, of which the first two directly related to heat-engineering.

A Boeing 767 belonging to Canadian airlines was flying on 23 July 1983 from Montreal to Edmonton. Halfway (at an altitude of 12 500 m) the plane was forced to make an emergency landing at an abandoned military airfield in the town of Gimli, because of empty fuel tanks. The reason for the emergency was a simple calculation error—the mass of fuel needed for the trip was incorrectly calculated. Usually Air Canada used the British metric system (pounds) for the calculation of the needed fuel. However, this airplane was a new one, where the SI-units had to be used (kg). In order to calculate the needed mass of fuel the volume had to be multiplied with

The site of the chapter: https://www.ptcusercommunity.com/message/423012.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_2

the density. However, the density was used with wrong units and therefore the tanks have been just half-full.

A similar mistake has led to the loss of the American satellite Mars Climate Orbiter (MCO). MCO launched on 11 December 1998 with a PH-2 Delta rocket. The unit arrived at Mars after 9 months, at the 23 September 1999. The engine should break the unit and go into the orbit of Mars. After 5 min, when the satellite went behind Mars, no more signals have been detected. From the analysis of data the scientists found out, that the device has passed the surface at a height of 57 km instead of 140 km and simply burned in the atmosphere (unit cost $125 million). This large deviation was caused by the fault of experts, who prepared the mission: when calculating the braking pulse one of the groups used Imperial-units (pound-force) and the other SI-units (newton).

You can mention another engineering error associated not with units themselves, but with the scale of measurement. In the past was a bridge across the Rhine built. On one side the Germans started and on the other side the Swiss started to build it. When both side met, they recognized that the height difference is about a half meter. The reason for the error was that the German standard of the construction zero height is the average level of the North Sea, and in Switzerland the average level of the Mediterranean Sea.

Any expert, who works in the field of science and technology, can tell numerous examples of such errors and incidents associated with the incorrect handling of units. The transition from calculations by hand to "dimensionless" programming languages, could not either solve this problem.

Mathcad—it's not just physical/mathematical program. Often people do not use units and just mention the unit of the used constant/variable just in the comments.

So you can, for example, often see that the pressure is just written down in Mathcad as: **p := 120**, instead of using the more clear and correct way: **p := 120 atm**.

What are the reasons for the under-utilization of Mathcad? Firstly, some users are just unaware of this useful tool. It is possible to use units in Mathcad and then nevertheless transfer the calculation to another programming language or spreadsheet program, in which, I repeat, the variables are stored as numerical values and their units noted in the comments.

There is a second group of Mathcad-users, which do not use physical quantities in their calculations, explaining or justifying it by the fact that all quantities prescribed in only one fundamental measurement system (e.g., international SI-Units) and they do not have any problems with the translation of the units. This motivation often buttressed by the fact that Mathcad-document without physical quantities is much easier to prepare for transferring to other programming languages. In addition, a Mathcad-document with disabled mechanism to work with units of measurements works faster.

The third and the main cause of the failure of many users to work physical quantities in the calculations lies deeper. It is associated with some features and shortcomings associated with the work with physical quantities, which even the most experienced users sometimes force to translate them into the category of comments.

However, using comments represents extra work. Often the user says to himself that he will add the necessary comments later. Often this “later” doesn’t come and the comments are never added. Nevertheless, when someone wants to use this in the past-created document he cannot understand which values he has to use for the variables, because he does not know which units are used there. Even for the creator of the document it can be difficult to remember which units are need to be used, because the document was created too long ago in the past.

2.1 Tools for Using Physical Quantities in Mathcad

Using units in Mathcad is quite simple. When you enter a numeric value you can connect it with units through typing in a built-in/user-unit or also a functional dependence. Additionally you can also choose a unit through the dialog box (Fig. 2.1). Usually the multiplication sign between the numerical value and unit is invisible. When you enter a dimensional value the option can be given to the user to select the unit of measurement, for example, through the use of switches: the temperature/pressure can be shown at different temperature or pressure scales with different units of temperature/pressure.

Among Mathcad (Ver. 13, 14, 15) it became possible not only to work with the built-in units (SI, MKS (meter, kilogram, second), CGS (centimeter, gram, second) and the U.S. Units), but also with custom units. This in particular means that the user can modify one of the predefined units or create even new units. Further calculations with custom units are in the same way monitored like calculations with old units: meters, for example, cannot be added with kilograms; which helps to avoid errors and typos.

When you use the “ = ” operator Mathcad will display the following: ∎ = ∎ ∎. The first place holder contains variable name, the second one the numerical value, and the third one the unit. By clicking on the third placeholder you can change the used unit and this convert to another system of measurement (e.g., bar → Psi). In some cases it is useful to duplicate the result value and show this way the result with different units; so the reader can choose, which one is the most convenient one for him.

What forces even experienced users to avoid units in purely physical calculation or to withdraw them from an almost finished document?

These “pitfalls”:

Some Mathcad tools are not designed to work with dimensional values. They interrupt the work or return an error message such as “There should be no dimension value” or, much even worse, give the wrong answer. In such cases it is necessary to temporarily withdraw the units by dividing the variables by the unit, and then after the calculation add the unit by multiplying the variable with the base

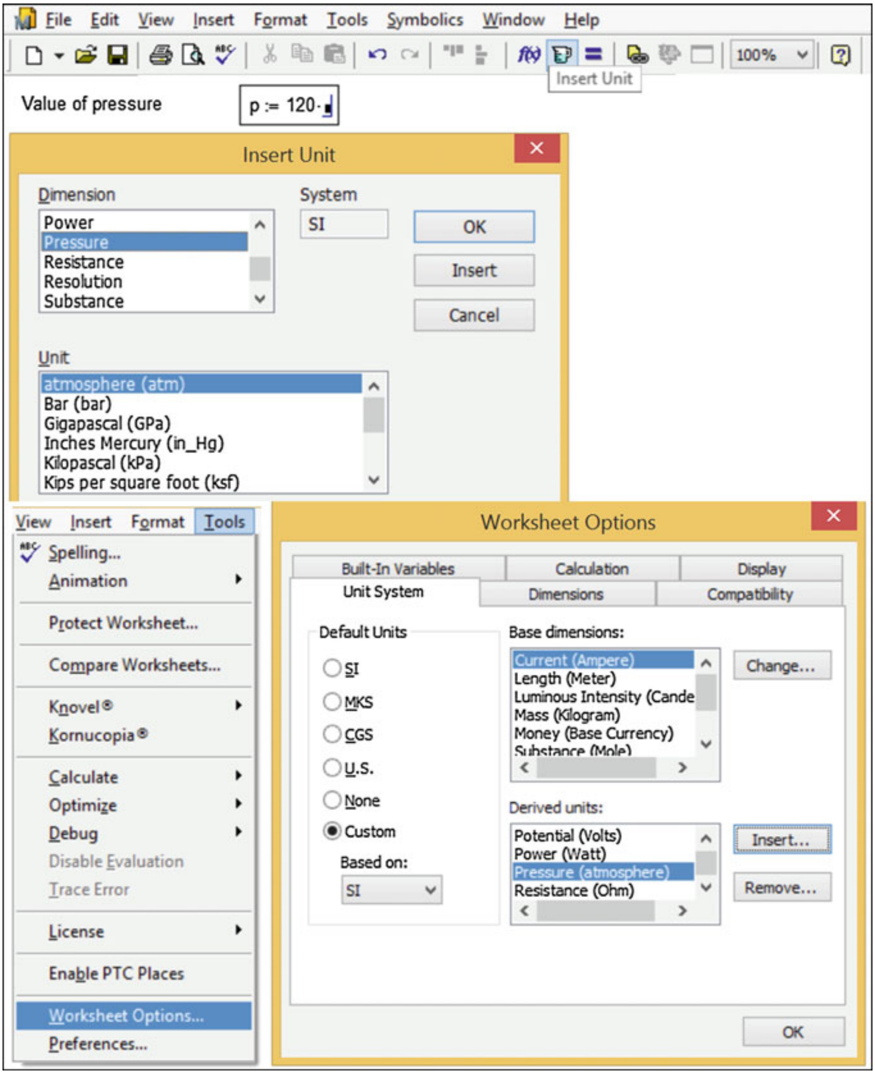


Fig. 2.1 Using units in Mathcad 15

unit. This method will be described in more detail when we talk in more detail about working with empirical formulas.

If the user does not plan to use physical quantities, then the use of units should be disabled. Therefor you have to make the following commands: press Tools, Options, and then a new window will pop up. There you press on the button Unit System and set the switch to No (see Fig. 2.1). This way it is possible to avoid errors. For example, when the user wants to create a function like $f(x) = x^2$, and accidently uses instead of x an unit like: **A**, **S**, **V**, etc. Mathcad will not return an error as expected. When using Quickplot we will see just one point instead of the functional correlation. When turning of units we can use these former unit-names also as variables (like x). This way mistakes can be avoided.

Symbolic math units operate in Mathcad as simple variables, they do not take into account that 1 m is equal to 100 cm or an hour has 60 min, etc. The symbolic math tool is a foreign element in Mathcad. It is taken out of the program Maple, in which the units have appeared only in the eighth version (Mathcad 5–13), or out of the program MuPAD, in which are either no units of measurement used (Mathcad 14/15 and Mathcad Prime). Symbolic Mathematics is an auxiliary tool in Mathcad, which is rarely used in calculations, and only plays a secondary role. If you want to use units in symbolic mathematics, than more should be done: it is necessary to use the substitute command in Mathcad, and tell Mathcad that 1 m = 100 cm, 1 h = 60 min, etc.

In Mathcad 15 you can store only dimensionless quantities or quantities of the same dimension in vectors/matrices, i.e., the same physical quantity like time, strength, weight, etc. There is known only one exception to this rule. The function Find, for example, can return a vector with mixed elements. If you need the results of several different function united in a vector or matrix, it is possible to deprive the units first and then bring them back. This restriction has been removed in Mathcad Prime.

Often so-called empirical formulas are used—formulas that relate not just on physical interrelations. Example of an empirical formula showed on Fig. 2.2 provides the information how to calculate with Mathcad the (estimated) used car value depended on age and mileage.

The empirical formula, with which the price of an old car can be calculated, was created with the help of statistical processing. This example shows how to supplement these formulas without interference with physical quantities. When empirical formulas are published it is always clearly marked, what units should be used as the initial values and in what units the answer will be. In this case, the vehicle age must be entered in years (yr) and mileage in Miles (mi). The formula returns the car price in U.S. dollars ($). These variables must lose their units in the formula. Figure 2.2 shows the formula of the **CarPrice** where variables are divided by their assigned units, the formula itself is multiplied by the currency. (Figure 2.2 was created at a time when the U.S. dollar was worth 50 rubles.)

The list of usable units in Mathcad does not contain all units. There are missing, for example, often used units in information technology, like bits, bytes, etc. In Mathcad 14 are also missing currency-units (dollar, ruble, euro, etc.). Mathcad 14 deals only with "seven" SI-Units (time, length, mass, current, temperature, light intensity and amount of substance) and their derivatives—units, which composed of these basic units (force, energy, power, etc.). And how to deal with older Mathcad versions, if you want to perform the so-called economical calculations, in which are dollars, rubles, euros, etc. are present? One solution is shown in the upper part of Fig. 2.2. The Dollar ($US) is assigned with a random unit of measurement, a physical quantity which does not appear in this calculation—candela (unit of luminous intensivity): **$:= cd** (see the first operator in Fig. 2.2). Next, you can

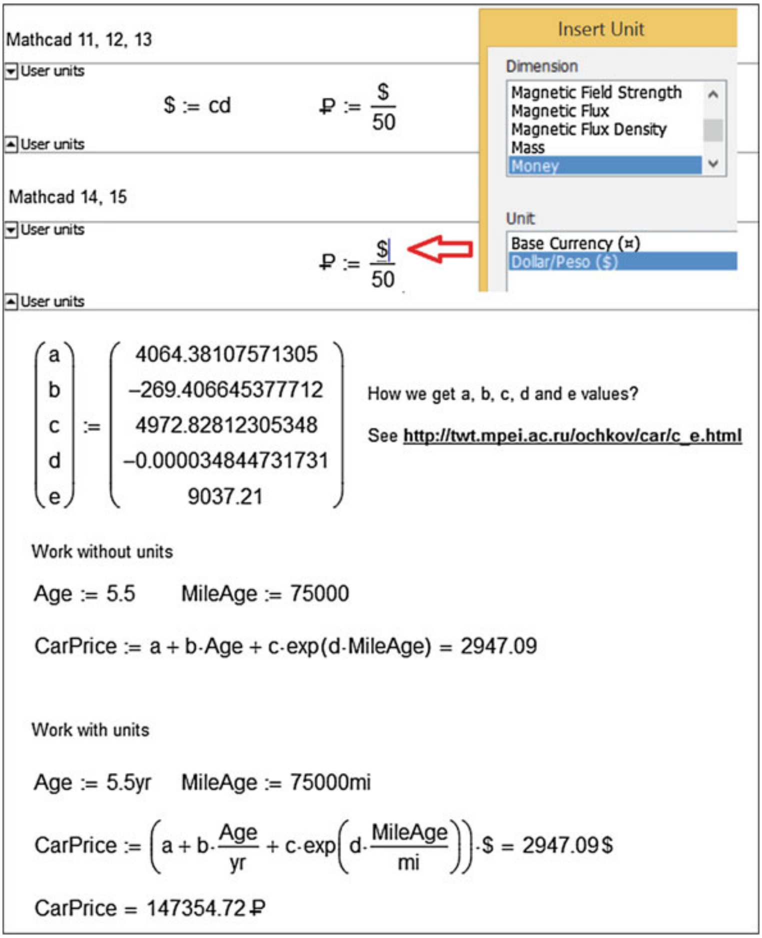


Fig. 2.2 Working with empirical formulas

connect the dollar with other currencies used in the calculation, for example, **Rub := $/50**. After that the price will be displayed by default, candelas (Mathcad 14 and 15), or in base currency units, which has to be replaced by the appropriate unit—dollars or rubles (see the last row in Fig. 2.2).

Another example shows Fig. 2.3, where we have to solve the following problem of thermal engineering: there is a power plant efficiency η given and you need to calculate the specific fuel consumption for electricity generation. All information for this calculation is given. The formula for the needed fuel is: **b := 12300/η**. It is said that the efficiency (**η**) must be expressed as a percentage (or in relative units) and the result of the needed fuel will tell us how many gram (**gm**) of fuel we need for the production of 1 kWh of electricity. For example: 12300/32 = 384.4 or

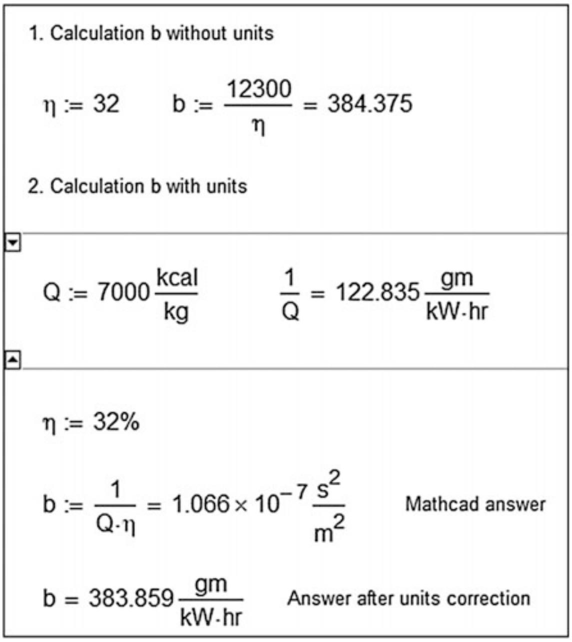


Fig. 2.3 Working with pseudo empirical formulas

123/0.32 = 384.4—thermal power plant with an efficiency of 32 % burns 384.4 g of conventional fuel for the generation of 1 kWh of electricity.

Like already often mentioned, it is possible to withdraw the units from all variables and then to obtain the answer dimensionless. The answer can either be multiplied afterwards with the demanded units or it is also possible to write the units just in the comment. Do not forget that these pseudo empirical formulas have been created with good intentions. This way we are liberated from additional unit conversions and these formulas are also very ease to remember/use. In this case (because of the easy task), it is also possible to obtain to answer by modifying the formula: **b := 1/(Q · η)**. Q is in this formula equal to the heating value, which has an assumed value of 7000 kcal/kg (the heat of combustion of coal with good quality). When you enter this modified formula you do not have to withdraw the units of the input and additionally you will get a more precise answer.

There formulas (a lot of them in the technical literature), which you cannot immediately understand and this can lead to errors in the calculation. For example, you need to calculate the molality of a solution out of its molarity. You go to the Internet, do some search and find the site with the desired formula (see Fig. 2.4).

This formula is easy to implement in Mathcad (see Fig. 2.5).

Figure 2.5 shows the encircled formula from table (see Fig. 2.4) implemented in Mathcad. The result of this formula looks very reasonable and this is also the crux, since nobody will check this formula for possible mistakes. All formulas in Fig. 2.4 are pseudo-empirical. Therefore you should check in which units all variable have to be entered. Otherwise Mathcad will convert the units how it is needed. And here http://erichware.info/sposob/ximia/ximikonc.htm

	0 < K < 100 % mass ratio	T titer	L molality	M molarity
M =	$\frac{10*q*K}{wp}$	$\frac{1000*T}{wp}$	$\frac{1000*q*L}{1000+wp*L}$	$\frac{1000*yp}{v}$
L =	$\frac{1000*K}{wp*(100-K)}$	$\frac{1000*T}{wp*(q-T)}$	$\frac{1000*yp}{mb}$	$\frac{1000*M}{1000*q-wp*M}$
T =	$\frac{q*K}{100}$	$\frac{mp}{v}$	$\frac{q*wp*L}{1000+wp*L}$	$\frac{wp*M}{1000}$
K =	$\frac{mp*100\,\%}{mb+mp}$	$\frac{100*T}{q}$	$\frac{100*wp*L}{1000+wp*L}$	$\frac{wp*M}{10*q}$

wp – molar mass of solute (g/mole)
mb – mass of solvent (g)
mp – mass of solute (g)
yp – moles of solute (mole)
v – volume of solution (cm³)
q – density of solution (g/cm³)

Fig. 2.4 Formula for the conversion of concentrations from the Internet

I want to calculate the molality (L) NaCl aqueous solution with a molarity M = 2 mole/L and the density of q = 1.076 gm/mL

$$M := 2\,\frac{mol}{L} \qquad q := 1.076\,\frac{gm}{cm^3} \qquad wp := 58.44\,\frac{gm}{mol}$$

$$L := \frac{1000\,M}{1000\,q - wp \cdot M} = 1.859\,\frac{mol}{kg}$$

Fig. 2.5 Conversion of Molarity → Molality

is the mistake. Although we use in this case even the right units we still get the wrong result. The Multiplier 1000 is integrated in the formula because of the relation between the units g/kg and cm^3/L. If we use the pseudo-empirical formula as a real empirical formula in Mathcad, we will get the correct result (see Fig. 2.6).

If we want to use the formula like a real physical formula we need to think about the multiplier 1000. In this case we need to delete it. After correcting the formula we will get the correct result (see Fig. 2.7).

$$L := \frac{1000\,\frac{M}{mol \cdot L^{-1}}}{1000\,\frac{q}{gm \cdot cm^{-3}} - \frac{wp}{gm \cdot mol^{-1}} \cdot \frac{M}{mol \cdot L^{-1}}} \cdot \frac{mol}{kg} = 2.085\,\frac{mol}{kg}$$

Fig. 2.6 Pseudo-empirical Formula as a purely empirical formula

$$L := \frac{M}{q - wp \cdot M} = 2.085\ \frac{mol}{kg}$$

Fig. 2.7 Using Formula like a physical one

Units support for physical quantities was till the 13th Mathcad version only possible on an absolute scale. This means in particular that if the value is zero itself, then it is not necessary to attribute any unit, although nevertheless it should be done: for example, **l := 0 m**, in order not to disrupt the possibilities to control the units. But often we also need a relative measurement scale for some tasks. For example, we usually measure the temperature in degrees Celsius (relative scale), and not in Kelvin (thermodynamic temperature—absolute scale). In Mathcad (also in new versions) we do not have a multiplication sign between the numeric value (**25**) and unit (**°C**), when we use a relative scale (**t := 25 °C**). We have more functional correlation between the unit and the numeric value, connected by a postfix-operator.

Figure 2.8 shows one possible solution of the problem of using relative measurement scales: given is the temperature t_1 (heater entrance) and the temperature difference **Δt** (difference between the entrance and exit). It is necessary to determine the temperature $\mathbf{t_2}$ at the outlet of the heater. There you can see three objects called **°C** (each time another User has to be selected in Mathcad): two constants named **°C**

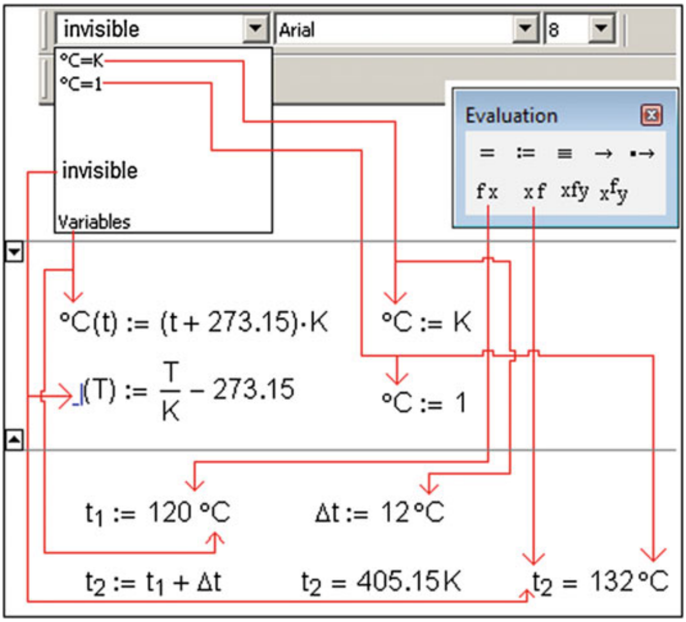


Fig. 2.8 Working with temperature scales

(the first is called **°C := K** and the second **°C := 1**) and a function called **°C**. Additionally you can see another function with an invisible name—**(T) := T/K-273.15**. Invisible function name can be entered by space sign (using Ctrl-Shift-K) or using text style with white color of characters.

The names of all three objects coincide (**°C**). However they are different objects, because they are used by different users in Mathcad (selectable via the button left next to the font-button). When working with relative temperature relative scales are three situations, in which the above introduced objects called **°C** are useful.

Situation 1
In the calculation it is necessary to enter the temperature in Celsius. To do this, we use the function called **°C** with the postfix operator. The input of the function will be temperature in **°C**. Internally Mathcad will convert the Celsius temperature into kelvin. Therefore all calculations can be carried in Mathcad, since Mathcad works with the absolute temperature scale.

Situation 2
You must enter the temperature difference **Δt**. It is very important that differences entered with another functions. Usuall (and very bad) mistake is to use °C. By using °C you will increase temperature difference value for 273.15 degrees. Since **Δ°C** is equal to **K** you can use the object **°C = K**.

Situation 3
You want to convert the temperature from Kelvin to Celsius scale. To do this you use the prefix operator and the invisible operator. The result will be first displayed without any units. So u have to use additionally the **°C = 1** in order to add some "virtual" units. Otherwise you can also just write the units in a comment.

With the described situations and the three similar objects named **°C**, it is fully possible to realize the work with each temperature scale: entering the temperature in one scale and converting it, or calculating the input and output values of a

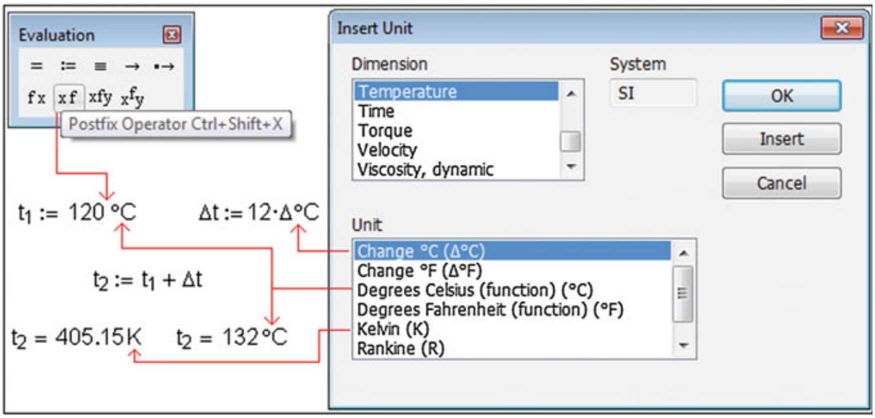


Fig. 2.9 Working with temperature scales in Mathcad 13/14/15

temperature difference. Nevertheless it is necessary to follow a simple but important rule: no matter with which scale we start, the temperature in the formulas should appear only in absolute values (kelvin).

Among Mathcad 13/14/15 and Mathcad Prime is the possibility to work relative temperature scales (Fig. 2.9), but Fig. 2.8 is necessary, since with this method also other relative measurement scales can be introduced to Mathcad. For example, the height of a building measured from the ground level or another point. In addition, it is possible to disable the units of measurement, and leaving only a imitation of units.

Mathcad versions 13/14/15 and also Mathcad Prime have not only standard units of temperature. There is also (temperature difference) **Δ°C** and **Δ°F**. This is done in order to provide the option to use **Δ°C/ΔK** or **Δ°F/Δ°R** equally. It is a tribute to outdated units of measurement. Previously, for example, the thermal conductivity was measured in **W/(m · °C)**, and now in **W/(m · K)**. In Mathcad the unit of thermal conductivity can to be written as: **W/(m · Δ°C)**, but it is better like this: **W/(m · K)**, as **Δ°C = K**.

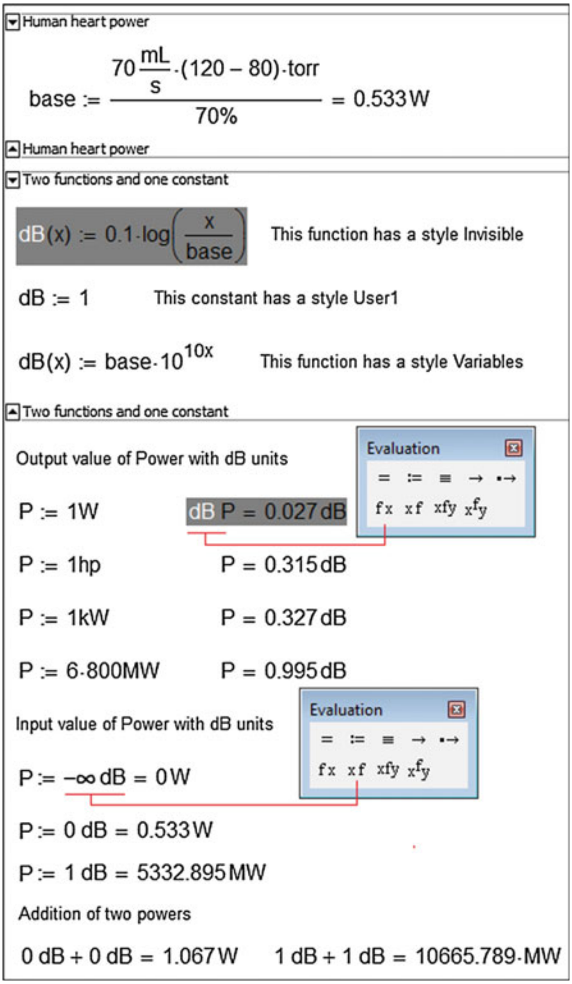
There is another unit, which could cause some difficulties in Mathcad—it is decibel (dB). Bel is the logarithmic ratio of two one-dimensional physical quantities, and deci means one-tenth. In order to measure a value in decibel, we thus have to build some kind of logarithmic scale of. Usually we measure the strength of sound in decibel. The intensity of sound waves is compared with the lowest sound intensity $\mathbf{I_0}$, which is audible with the human ear (usually $\mathbf{I_0 = 0.01\ W/m^2}$). We can directly calculate the ratio of the measured sound intensity I and the smallest intensity, but the range of values of this ratio so broad that it is necessary to introduce a logarithmic scale with a decimal multiplier—$0.1 \times \log(I/I_0)$. Very loud sounds, e.g., a jack-hammer reaches a level about 100 dB, a conversation in a room is about 60 dB. When using decibels we must select a base from which we use as the base value. In Fig. 2.10 is a scale introduced in relation to the power of the human heart.

In order to work with decibels in Mathcad we introduce two functions named **dB** (the name of one of them is invisible—write with white on white background) and one constant, which is also named **dB**. But these are different objects because they are used by different users (see example above). The operator, which defines the invisible function, “lies” on a colored background in order to make the function name visible. When calling this function with the prefix operator the user cannot see it; instead of seeing **dB p = 2 dB**, he will see **p = 2 dB** (the visible unit **dB** has to be entered manually by the user. Therefore he has to use **dB := 1**. It simulates a unit measurement).

Sometimes when working with decibels is necessary also indicate base (initial value). Thus: **p := 100 dB (re 0.533 W)**, where is just a reference and 0.533 W represents the base. This function requires an infix operator (Fig. 2.11).

Some features can be noted when using dimensionless physical quantities—angles, mass-/ volume-/ and mole-fractions, etc. So, from the 12th version of Mathcad onwards the Steradian become dimensionless (**sr = 1**); before that it was a dimensional quantity (**sr = 1 sr**). The developers on the one hand, restored certainly some logic—radians (the ratio of two lengths) was always dimensionless and Steradian (the ratio of two areas) was dimensional before Mathcad 12 (see above),

Fig. 2.10 Working with dB



and on the other hand, they opened the possibility of making mistakes such as addition of radian. This problem of lack of control of the dimensions you may encounter when working with other "dimensionless" quantities.

Units referred to the number of pieces (e.g., price per piece) can cause some additional problems. Thus, in the calculation, which is shown in Fig. 2.12, you have to calculate the total surface of 2200 tubes with a diameter of 15 mm. First we have to calculate the surface of one tube. The result we obtain is the surface per piece. This value has afterwards to be multiplied with the total number of pieces to get the final result. The surface per piece is different to the surface of one piece (different units). For example the surface per piece cannot be added to the surface of one piece. To create this new unit of surface per piece we introduce the new unit "piece" (see Fig. 2.12).

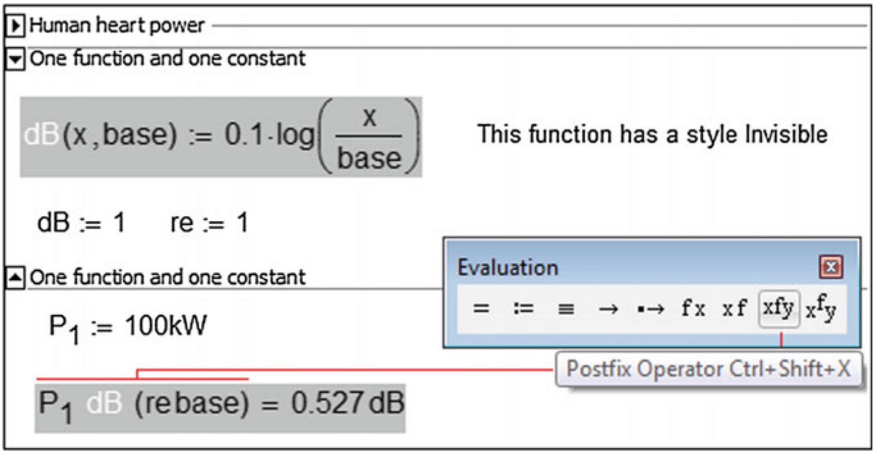


Fig. 2.11 Working with decibels and a reference to the base

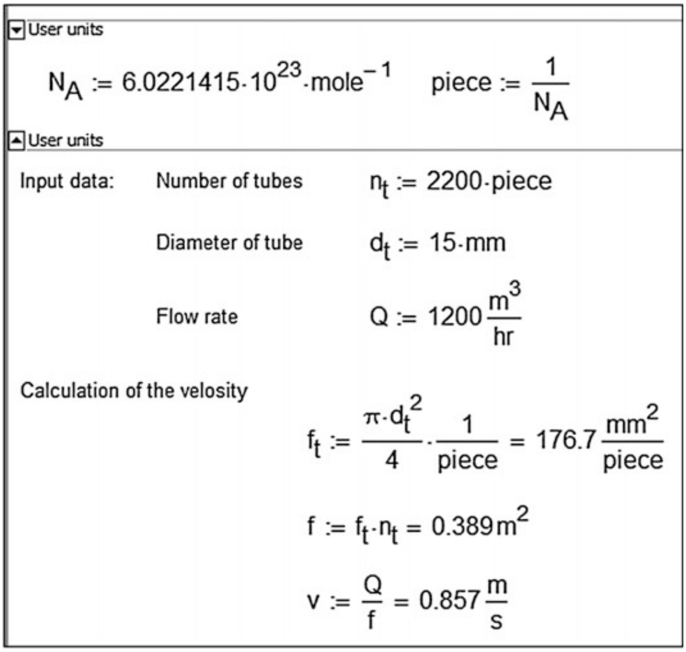


Fig. 2.12 Tube calculation

In some calculations you need two variables with two different units for storing one physical value, for example, the temperature in thermodynamic calculations, where the variable **t** (or **θ**) stores the numerical value in degrees Celsius and **T** in Kelvin. There are tools built into Mathcad in order to eliminate this undue dichotomy.

Without this integrated tool, the calculation of the human heart (Fig. 2.10) would be more complicated and it would be necessary to introduce operators, which translate the units (**mL** and **torr**) in a more popular unit system, for example, in the international SI-system: milliliter (**mL**) → cubic meter (**m**^{3}), millimeter of mercury (**torr**) → Pascal (**Pa**), etc.

There is another source of possible difficulties when using units. There are, if I may say so, psychological and linguistic reasons related to the unclear definition of physical quantities and units of measurement in everyday speech and even in scientific papers. If two people may have a different understanding of one expression, similar discrepancies can also occur in human-computer "conversations". We say "light year", where the name intends the unit of length—not time, etc.

2.2 Completely Dimensional Functions

We continue with a problem about a pump. Thus, Fig. 2.13 shows an emergency stop mechanism, if the source data contains an error: here was to variable $\mathbf{p}_1$ mistakenly assigned the unit of mass, not pressure.

But in order to avoid the emergency stop of the calculation, it is not only possible to fix the error ($\mathbf{p_1 := 1\ kg}$), but also to introduce a new error; ($\mathbf{p_2 := 20\ kg}$; see Fig. 2.14).

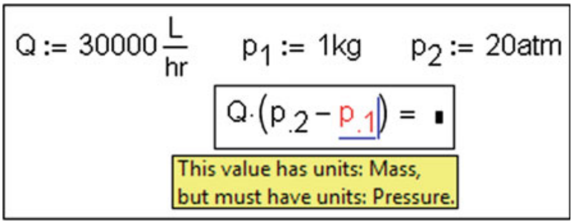


Fig. 2.13 Emergency stop during pump-calculation

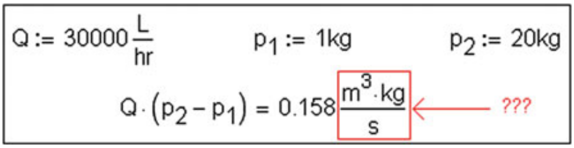


Fig. 2.14 "Fixing" the emergency stop with an error

$$Q := 30000 \qquad p_1 := 1 \qquad p_2 := 20$$
$$Q \cdot (p_2 - p_1) = 5.7 \times 10^5$$

Fig. 2.15 Pump-calculation without units

Of course there will also no emergency stop occur, when using no units at all (see Fig. 2.15).

In principle, all dimensions of the control mechanism should be understood, so that in the formulas the units occur only with certain dimensions—the examples shown in Figs. 2.14 and 2.15 must be completed, so that they stop the calculation, as shown in Fig. 2.13. Another possible problem is the possibility of mixing up the input parameters $\mathbf{p}_1$ and $\mathbf{p}_2$.

The essence of this problem and its possible partial and complete solution, we illustrate in a more common and understandable task—the creation of a function with the name of Vc, which calculates the volume of a cylinder. In Fig. 2.16 shows the function in Mathcad.

The figure shows that if the variables **d** and **h** have the dimensions of length, the function will return the value of volume of the cylinder in m^3. But the same trouble-free function takes also the value of its arguments with any units or even without any units (Fig. 2.17).

$$V_c(d,h) := \frac{\pi \cdot d^2}{4} \cdot h$$
$$V_c(20mm, 15mm) = 4.712 cm^3$$

Fig. 2.16 Calculating the volume of a cylinder

Fig. 2.17 Function call with different dimensions of the arguments

$$V_c(d,h) := \frac{\pi \cdot d^2}{4} \cdot h$$
$$d := 20 \qquad h := 15kg$$
$$V_c(d,h) = 4712.4 kg$$

To find out how to get out of this situation, remember the wonderful principle formulated, which Fourier presented in his classic work "The Analytical Theory of Heat", which was published in 1822. This principle is now called the "principle of dimensional homogeneity" and it states that if the dimensions of each term on both the sides of equation are same, then the physical quantity will be correct. To illustrate the difference between the usual algebraic and "physical" equations, we show an example given in the seminal work of Bridgman ("Regular and Chaotic Dynamics", 2001). There we analyze the problem of a body falling, Bridgman notes that there are at least two equations linking the distance (s), speed (v), time (t) and the acceleration of gravity (g):

$$v = gt$$
$$s = gt^2/2.$$

If we now add both as purely algebraic equation, we obtain the following equation:

$$v + s = gt + gt^2/2$$

That firstly looks plausible, but it is wrong in terms of dimensions—the speed cannot be added to the distance! Having an instrument of controlling the dimensions embedded in Mathcad, we can control the dimension of the entered value by adding a "size zero", namely by adding and subtracting a certain value of a given dimension. Figure 2.18 shows how to do that.

The figure shows that the newly created function, which returns the volume of a cylinder, takes the arguments only in meter (**m**) and returns only values with $\mathbf{m}^3$.

The following example (Fig. 2.19) shows that Mathcad monitors only the dimensions of the variables, but does not check the position of the variables. Therefore it is possible to accidently mix up the variables **d** and **h**.

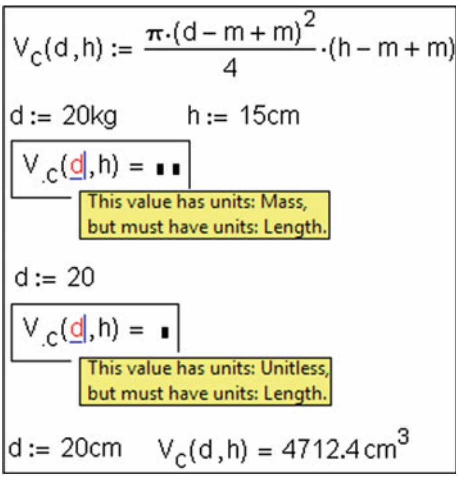


Fig. 2.18 Creating and calling "dimensional" user function

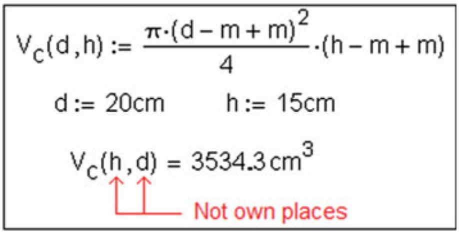


Fig. 2.19 Error while using a user function

The figure shows an error when calling a function that returns the volume of a cylinder, which may occur if the function arguments are reversed: instead of the diameter, we enter the height of the cylinder. This error can be avoided by using further monitoring tools. Here the diameter of the cylinder and its height are two different physical quantities with the same length dimension. First, in the most complete form this idea, how to control the inputs, has been effectively developed in the book Huntley (Huntley G. Dimensional Analysis. Mir, Moscow, 1970), which, in particular, suggested the use of a "vector of dimension".

Thus, the variables have to be connected to the dimensions (length–meters, feet, miles, etc., weight–pounds, pounds, etc.), and to the physical quantities. It should be assumed that the two variables (diameter of the cylinder and its height) have two different units of measurement, and their "difference", in particular, should be reflected in the fact that these parameters cannot, for example, be added.

In Mathcad are eight dimensions integrated: length, time, mass, current, temperature, light intensity, the amount of substance, and currencies (only in the 14th and 15th versions and Mathcad Prime). In order to avoid the possibility of mixing up the input parameters we introduce two different lengths. The diameter is connected to the unit **m-d** (m = meter, d = diameter) and the height is connected to the unit **m-h** (m = meter, h = height). Basically the new unit **m-h** has not the unit meter. Therefore we divide in the function $\mathbf{V}_c$ the input parameter by his unit **m-h**. Afterwards the function is multiplied with $\mathbf{m}^3$ in order to obtain the correct result with correct units (Fig. 2.20).

When you now swap wittingly or unwittingly the arguments of the function the calculation is interrupted by an error message, which can be described as follows: "The inputs are swapped".

There are also real thermodynamic problems, which can be solved with Mathcad, whereby we need or, at least, it would be desirable to handle different physical variables, which have the same dimension.

You need to calculate the thermal efficiency of a combined-cycle (see Chap. 14). To solve this problem, you need functions, which return the thermodynamic properties of the working fluids. These functions can be integrated to Mathcad by using the already mentioned WaterSteamPro software (www.wsp.ru)—see Fig. 2.21.

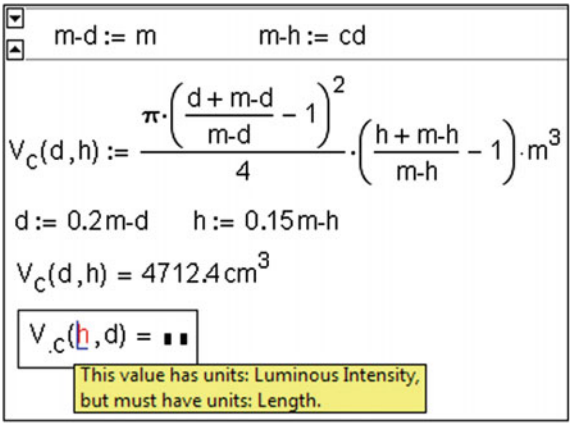


Fig. 2.20 Calculation of the volume of a cylinder with full control

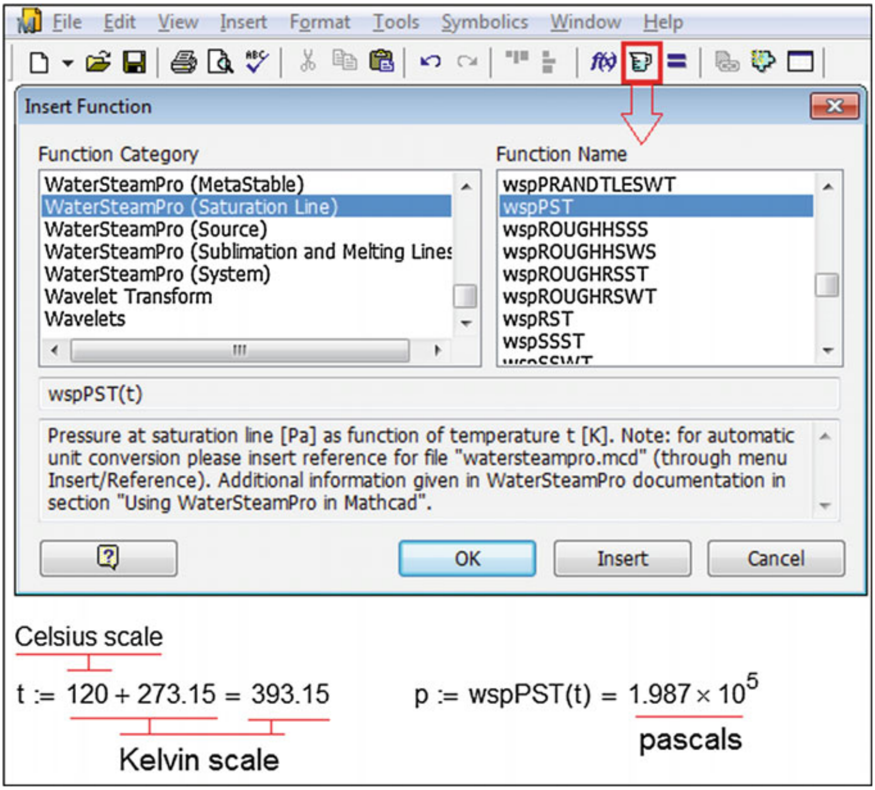


Fig. 2.21 Using WaterSteamPro in Mathcad

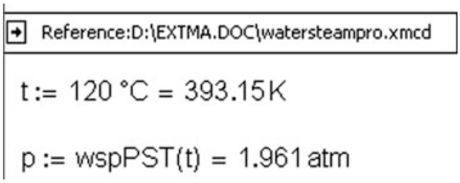


Fig. 2.22 Using WaterSteamPro with units

Figure 2.21 shows how you can calculate the boiling point of water for a given pressure by using one of the functions from WaterSteamPro—function wspTSP. Moreover, all values are non-dimensional, but it is meant that the pressure is expressed in pascal, and the temperature is measured in kelvin. All functions from WaterSteamPro by default (without referencing to "watersteampro.xmcd") use dimensionless arguments but expressed in SI units or their mixture.

But Mathcad software supports units, and this mechanism is necessary to solve our problem and indeed in many other heat and power calculations.

In order to be also able to use units for WaterSteamPro it is necessary to make a reference link to **watersteampro.xmcd**, included in the WaterSteamPro (Fig. 2.22).

What contains this file? There are firstly some additional units definitions, as well as various constants needed for calculations in heat and power engineering (the universal gas constant, critical points of water and steam, etc.). Secondly, and most importantly, it makes units available for WaterSteamPro.

In the programs Mathcad Prime 1 and 2, WaterSteamPro cannot be integrated through the mechanism of DLL. Such an opportunity came only in the third version of Mathcad Prime. Figure 2.23 shows how to call a function named **wspPST**, which returns the saturation pressure of water and steam as a function of temperature. Calling this function in Mathcad 15 is shown in Fig. 2.21 ("dimensionless" call) and Fig. 2.22 ("dimensional" call through the supporting document watersteamepro. mcd). The mechanism of function redefining shown on Fig. 2.24 on the example made im Mathcad 15 won't work in Mathcad Prime 3 environment. WaterSteamPro functions, that ypu can see on Fig. 2.23 must be "dimensionless" or supplemented by the corresponding basic units (such as **Pa** for pressure). We will examine this methods more closely in the Chap. 9 on the example of nuclear power plant calculation.

However in Mathcad 15 it isn't necessary to insert reference on watersteampro. xmcd (which redefines all the functions of WaterSteamPro program), instead, it is possible to manually redefine only those functions that are used in the users' project taking into account, that different physical values might have same units. For example in Chap. 14 (calculation of the thermal efficiency of the CCGT unit) reader will encounter with different physical values—mass of the gas and mass of the water/steam, which measures in one and the same units of mass.

This is one of the key aspects of working with units.

In thermophysical calculations reader may face values that have identical unit, but different physical sense. If this fact doesn't taken into account, it may lead to the errors during calculations in the Mathcad environment, which we mentioned before.

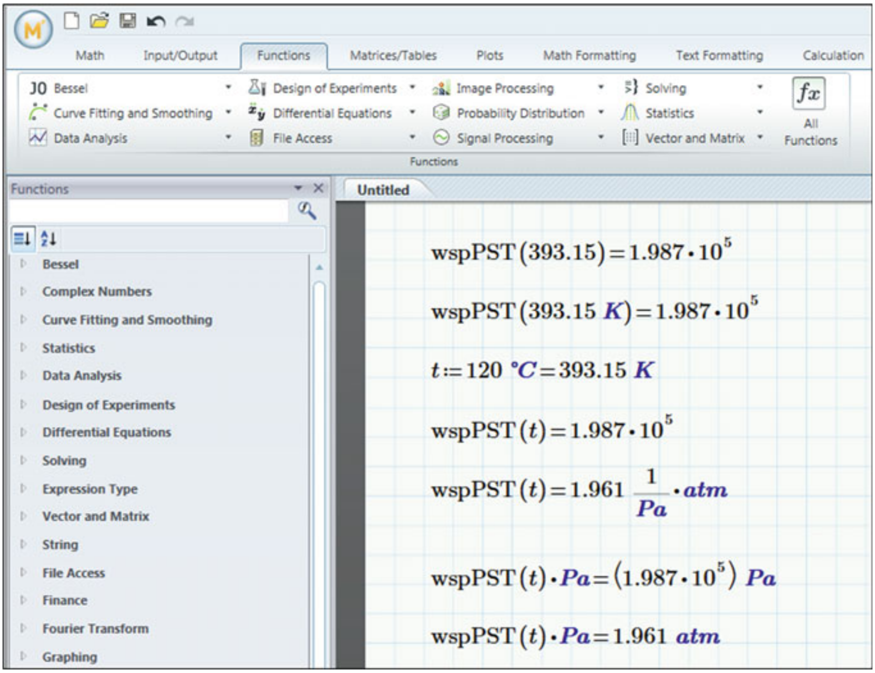


Fig. 2.23 WaterSteamPro functions with units in Mathcad Prime 3

Example In the binary thermodynamic cycle (CCGT unit for example) among the other values appear values such as specific enthalpy of the first working medium and specific enthalpy of the second working medium. Both of these values are typically measured in kJ/kg, but they have different kilograms. To avoid mistakes of mixing different physical values in such calculations, it is recommended to use built-in kilograms for one of the values and for the second assign basic SI unit, that isn't used in users' calculation (for example candela—**cd**). In that case value **m**, that usually records ratio of the first working body flow rate to the flow rate of the second in binary thermodynamic cycles, will be dimensional value, and it will help to avoid some typical mistakes.

The technology of entering different physical values with same units into calculation shown on Fig. 2.24.

As you can see at the top of Fig. 2.24 introduced two units of mass: for the measurement of water vapor and gas—the two working fluids of the combined-cycle plant. Kilograms of water/steam equal to the built-in Mathcad unit of mass (**kg**). Kilograms of gas equal a specific unit, which in this calculation is not used—the unit of luminous intensity **cd** (candela).

The next stage in the calculation in Fig. 2.24 is the input of initial data. Notice that in values of mass flow rate are used kilograms of water vapour and kilograms of gas for different physical quantities that have the same mass dimension.

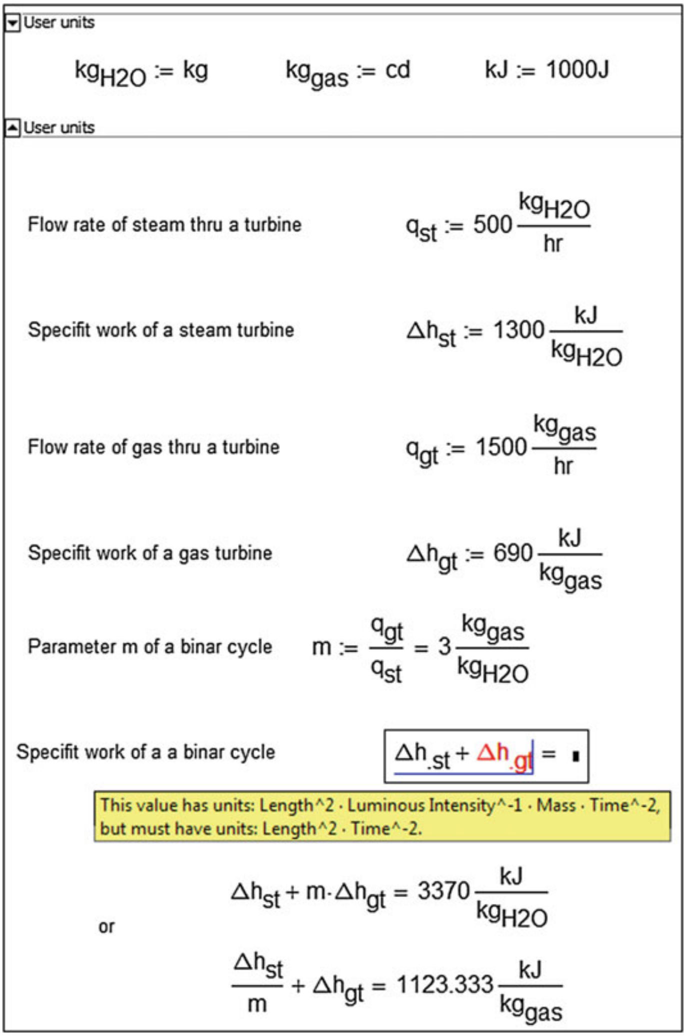


Fig. 2.24 Performance assessment of a combined cycle plant

This allows user to have in his calculation the value of **m** (the ratio of the flow rate of one of the working medium to the flow rate of another working body) not dimensionless, but dimensional. This helps to avoid some errors in the calculation of binary power cycles, one of which is shown in Fig. 2.24: if we had not introduced two units of mass for the two working fluids, the operator $\mathbf{\Delta h_{st} + \Delta h_{gt}}$ would not have been interrupted by the error message, and would have given an incorrect result. The correct result (specific work of the combined cycle) is calculated from the last two operators shown in Fig. 2.24.

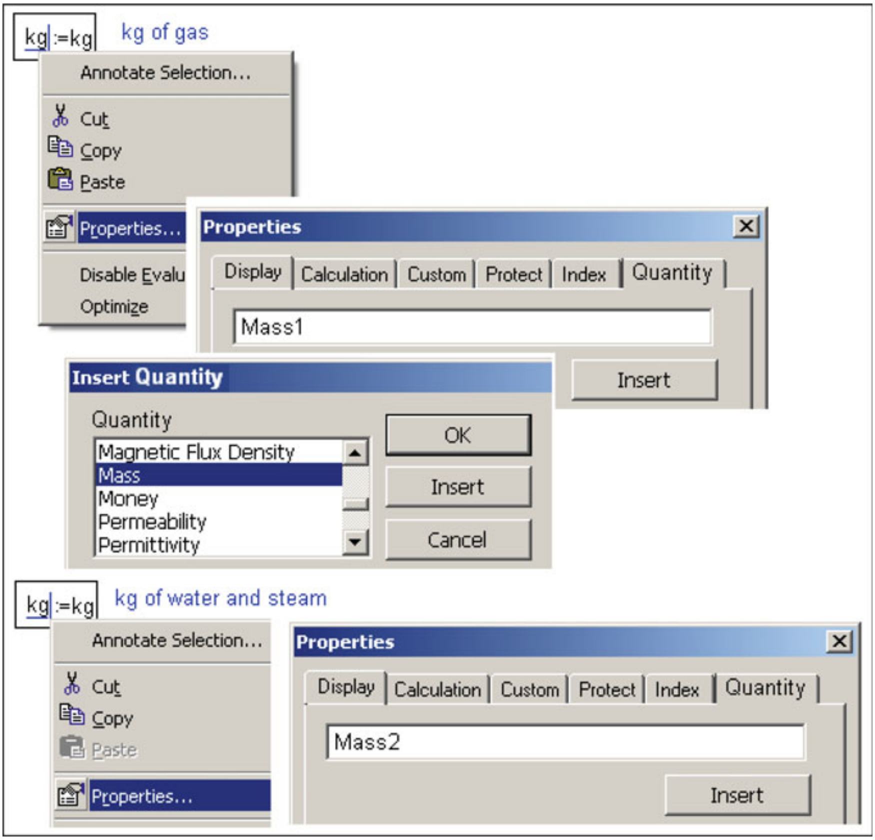


Fig. 2.25 Demonstration of the toolkit for creating different quantities with the same unit

One solution of Mathcad problems with different physical quantities of the same dimension can be offered through an additional tab "Quantity" in the dialog box "Properties", which occurs if we press the right mouse button on the name of the newly created function or variable. In the text box of this tab can be written the desired dimension/index—the base (weight, volume, etc.) or component (air, gas, etc.). Thus the name can be supplemented by the numbers 1, 2, 3, etc., dividing one from the other physical value. This technique is displayed in Fig. 2.25.

In the calculation shown in Fig. 2.24 dimensionless variable m (the mass flow in a gas turbine divided by the mass flow of steam in a steam turbine) is transformed into a ("pseudo"-) dimensional variable. And this is, as we have seen, good: the less dimensionless quantities, the more monitoring tools can be used. But in terms of dimensionless quantities we also have other ones—the degree of dryness of steam (dry weight of steam referred wet steam) and the three values of thermal efficiency ($\boldsymbol{\eta}$).

If the numerators and denominators in these fractions have the same dimension, these ratios must be dimensionless. If the numerators and denominators of these fractions are regarded to different physical quantities, these values (**m**, **η**, etc.) should be "pseudo"-dimensional. It is possible to use, for example, the technique shown in Fig. 2.25.

Chapter 3
Concentration of Solutions

Valery Ochkov

Abstract In the chapter you will learn how to use the value of the concentration of a solute in aqueous solutions in computer calculations, how to convert the species concentration using modern computer tools.

When modelling thermophysical properties of working fluids and heat transfer mediums presented as multicomponent mixtures it is necessary to define the term *concentration*. The thermal conductivity of aqueous solution of NaCl[1] from article [32] is described in this way as a function of pressure, temperature, and solution molality (see "live calculation" by this function here—http://twt.mpei.ac.ru/MCS/Worksheets/rbtpp/tcon.sol.sod.hl.lf.xmcd). For this reason and bearing in mind, that the term "concentration" also figures in chemical thermodynamics, it seems to be useful to describe the particularities of working with concentrations in Mathcad. As an example, we consider the following problem: The aqueous solution of NaCl was mixed with the same solution of another concentration. The volumes of both solutions, molality of the first and molarity of the second are given. Define the parameters of the mixture. Figure 3.1 shows solution of the problem in Mathcad 15 [33].

The site of the chapter: https://www.ptcusercommunity.com/message/423015.

---

[1]Aqueous solution of this salt is often used as a heat transfer, or rather, a cooling medium in conditioning systems (see Chap. 17). Adding a salt into the water results in decrease of the freezing temperature—see Fig. 3.6.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_3

Molarity is the amount[2] of dissolved substance (mole) divided by the volume of solution, and molality is that divided by the mass of solvent medium. Molarity of an aqueous solution can vary with change of the solution temperature while molality does not. To solve the problem we need first, molar mass of the dissolved substance (NaCl) and second, dependence of density of NaCl aqueous solution on its concentration.

The first is easy to do: just have a look in Mendeleev's periodic table and summarize the atomic weights of sodium and chlorine. Also, one can get in the site shown on Fig. 3.2, enter name of an element or molecular species, obtain the result, and transfer it into the calculation.

Usually reference materials represent tabular density and concentration dependences of aqueous solutions of various salts, acids, and alkalies at specified temperature. As a rule, it is room temperature, the one in the laboratory where chemical analysis are carried on. As noted in Chap. 1, it is inconvenient to work in Mathcad with tables but one can easily develop functions connecting density and solution concentration—see Fig. 3.3.

Figure 3.3 shows, how spline interpolation (built-in function cspline[3]) is used to create two functions $\boldsymbol{\rho}_{\mathbf{NaCl}}(\boldsymbol{\omega})$ and $\boldsymbol{\omega}_{\mathbf{NaCl}}(\boldsymbol{\rho})$, where $\boldsymbol{\rho}$—density of NaCl solution, $\boldsymbol{\omega}$—mass percentage, which numerically equals to NaCl mass measured in grams divided by 100 g of solution (not solvent).

The problem of mixture of two solutions reduces to two simultaneous linear equations, the first is equality of sum of masses of initial solutions and mass of the mixture, and the second is equality of mass sum of dissolved solids (NaCl) in initial solutions and in the mixture (law of perdurability). The main thing here is to perform all concentrations in one form—as a mass percentage and define masses of the solutions.

Mass percentage of the first solution is easy to define. It is required to find out or to develop a formula for this value depending on molality of the solution. Figure 3.4 shows the Internet site for such calculations. Here it is sufficient to choose the type of concentration (**T**—titer, $\boldsymbol{\omega}$—mass percentage, **M**—molarity, and **m**—molality), enter numerical value of the concentration, and type of required concentration.

The site shown on Fig. 3.4 both makes calculations and represents the formulas used. One of them was transferred from the site shown on Fig. 3.4 into the calculation shown on Fig. 3.1 (finding mass fraction[4] of the first solution from its molality). The formulas for calculations of concentration values used in the

---

[2]Usually "measured in moles" is added here. Why?! An expression like this: "Amount of substance taken—200 mg" is often found in the literature. Certainly, here should be written "mass" rather than "amount". These terms are often misused in everyday life and in science—hence it is necessary to clarify.

[3]It was used in the first Mathcad document of the book, see Fig. 1.1.

[4]Mass fraction is mass of dissolved solids divided by mass of the total mixture ($\boldsymbol{\omega}$), which equals to number of dissolved solids measured in grams in 100 g of solvent (**P**). In this case it is important to distinguish between this type of concentration and another one—mass of dissolved solids divided by solvent mass.

We have mixed 100 ml of a 0.2 molal and 1 liter of 2 molar aqueous sodium chloride sodium (NaCl). How much and what concentration yielding a mixture (t=20°C).

Input data:

Volume first solution $V_1 := 100\,\mathrm{mL}$ Molality first solution $m_1 := 0.2\,\frac{\mathrm{mole}}{\mathrm{kg}}$

Volume second solution $V_2 := 1\,\mathrm{L}$ Molarity second solutiona $M_2 := 2\,\frac{\mathrm{mole}}{\mathrm{L}}$

Molar mass NaCl $MM := 58.44\,\frac{\mathrm{gm}}{\mathrm{mole}}$

Solution of the problem

The weight percent of the first solution. See (Fig. 3.4): **http://twt.mpei.ac.ru/MCS/Worksheets/Chem/CRC.xmcd** $\omega_1 := \frac{m_1 \cdot MM}{1 + m_1 \cdot MM} = 1.1553\,\%$

Link on cloud function Reference:http://twt.mpei.ru/tthb/H2O.xmcdz

Density first solution $\rho_1 := \rho_{NaCl}(\omega_1) = 1.006\,\frac{\mathrm{gm}}{\mathrm{mL}}$ Mass first solution $mass_1 := V_1 \cdot \rho_1 = 100.618\,\mathrm{gm}$

Molarity first solution $M_1 := \frac{\rho_1 \cdot \omega_1}{MM} = 0.199\,\frac{\mathrm{mole}}{\mathrm{L}}$

The percentage by weight of the second solution (calculated by successive approximations)

$$\omega_2 := \left| \begin{array}{l} (\omega_2 \leftarrow 10\% \quad \omega'_2 \leftarrow 20\%) \\ \text{while } |\omega_2 - \omega'_2| > 0.0001\% \\ \quad \left| \begin{array}{l} \omega'_2 \leftarrow \omega_2 \\ \omega_2 \leftarrow \frac{MM \cdot M_2}{\rho_{NaCl}(\omega_2)} \end{array} \right. \end{array} \right. = 10.853\,\%$$

Density second solution $\rho_2 := \rho_{NaCl}(\omega_2) = 1.077\,\frac{\mathrm{gm}}{\mathrm{mL}}$ Проверка $\frac{\rho_2 \cdot \omega_2}{MM} = 2\,\frac{\mathrm{mole}}{\mathrm{L}}$

Molality second solution $m_2 := \frac{\omega_2}{MM \cdot (1 - \omega_2)} = 2.083\,\frac{\mathrm{mole}}{\mathrm{kg}}$ Mass second solution $mass_2 := V_2 \cdot \rho_2 = 1076.926\,\mathrm{gm}$

$$(mass \quad \omega) := \begin{pmatrix} mass_1 + mass_2 = mass \\ mass_1 \cdot \omega_1 + mass_2 \cdot \omega_2 = mass \cdot \omega \end{pmatrix} \left| \begin{array}{l} \text{solve}, \begin{pmatrix} mass \\ \omega \end{pmatrix} \\ \text{float}, 7 \end{array} \right. \rightarrow (1.177545\mathrm{kg} \quad 0.1002446)$$

Answer $mass = 1177.55\,\mathrm{gm}$ $\omega = 10.024\,\%$

Mix density $\rho := \rho_{NaCl}(\omega) = 1.071\,\frac{\mathrm{gm}}{\mathrm{mL}}$ Mix volume $V := \frac{mass}{\rho} = 1099.296\,\mathrm{mL}$

Mix molarity $M := \frac{\rho\,\omega}{MM} = 1.837\,\frac{\mathrm{mole}}{\mathrm{L}}$ Mix molality $m := \frac{\omega}{MM \cdot (1 - \omega)} = 1.906\,\frac{\mathrm{mole}}{\mathrm{kg}}$

Final table:

$M_1 = 0.199\,\frac{\mathrm{mole}}{\mathrm{L}}$	$m_1 = 0.2\,\frac{\mathrm{mole}}{\mathrm{kg}}$	$\rho_1 = 1.006\,\frac{\mathrm{gm}}{\mathrm{mL}}$	$\omega_1 = 1.155\,\%$	$mass_1 = 100.618\,\mathrm{gm}$	$V_1 = 100\,\mathrm{mL}$
$M_2 = 2\,\frac{\mathrm{mole}}{\mathrm{L}}$	$m_2 = 2.0832\,\frac{\mathrm{mole}}{\mathrm{kg}}$	$\rho_2 = 1.077\,\frac{\mathrm{gm}}{\mathrm{mL}}$	$\omega_2 = 10.853\,\%$	$mass_2 = 1076.926\,\mathrm{gm}$	$V_2 = 1000\,\mathrm{mL}$
$M = 1.837\,\frac{\mathrm{mole}}{\mathrm{L}}$	$m = 1.906\,\frac{\mathrm{mole}}{\mathrm{kg}}$	$\rho = 1.071\,\frac{\mathrm{gm}}{\mathrm{mL}}$	$\omega = 10.024\,\%$	$mass = 1177.545\,\mathrm{gm}$	$V = 1099.296\,\mathrm{mL}$

Fig. 3.1 Problem of solution parameters

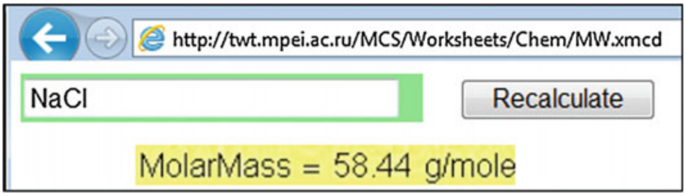


Fig. 3.2 Online calculation of molar mass

```
ωNaCl(ρ) := | "Concentration of water solution NaCl at t = 20°C according to it's Density"
            | ρ ← ρ·gm^-1·cm^3
            | M ← (1  1.005  1.013  1.020  1.027  1.034  1.041  1.049 ... 1.189  1.197  1.2 )
            |     (0  1      2      3      4      5      6      7     ... 25     26     26.4)
            | return error("Error: ρ must be >= 1 gm/cm^3") if ρ < 1
            | return error("Error: ρ must be <= 1.2 gm/cm^3") if ρ > 1.2
            | Ro ← (M^T)^<ORIGIN>
            | C ← (M^T)^<ORIGIN+1>
            | return interp(cspline(Ro, C), Ro, C, ρ)·%
ρNaCl(ω) := | "Density of water solution NaCl at t = 20°C according to it's concentration"
            | M ← (1  1.005  1.013  1.020  1.027  1.034  1.041  1.049 ... 1.189  1.197  1.2 )
            |     (0  1      2      3      4      5      6      7     ... 25     26     26.4)
            | return error("Error: ω must be >= 0") if ω < 0
            | return error("Error: ω must be <= 26.4%") if ω > 26.4%
            | C ← (M^T)^<ORIGIN+1>·%
            | Ro ← (M^T)^<ORIGIN>
            | return interp(cspline(C, Ro), C, Ro, ω)·gm·cm^-3
```

Fig. 3.3 Direct and inverse functions of density and concentration (mass percentage) of NaCl aqueous solution

calculation shown on Fig. 3.4 are differ from those containing in various reference books, online and paper. The fact is that formulas in the reference books sustain some coefficients (1000, 0.01, 10, etc.) connecting calculations with suitable units. This peculiarity of Mathcad calculations with units is described in Chap. 2.

Calculation of mass fraction of the second solution is more difficult. It is required first, access to the function returning density of solution by its mass fraction. This was done in the calculation shown on Fig. 3.1 by reference to the cloud file containing such a function. Second, one can calculate mass fraction of the second

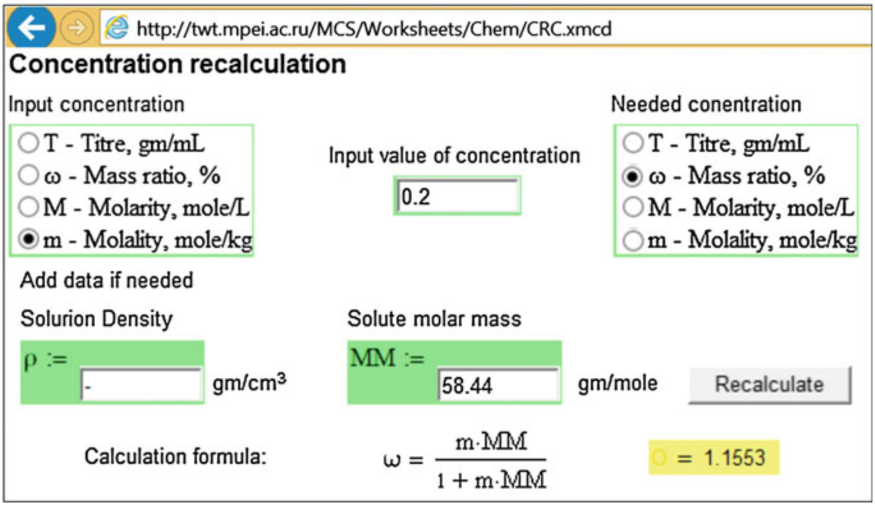


Fig. 3.4 Online calculation of concentration

solution only step-by-step because this value (ω_2) contains in both, left and right, parts of the equation connecting mass fraction and molarity. It was done in a program using **while** cycle, see Fig. 3.4. Although, such the method of sequential approximations can be performed without programming, in a manual mode, as shown on Fig. 3.5.

Figure 3.5 fixes such sequence of steps: first approximation to ω_2 (10 %) is assigned and then it is refined and transferred into the first (previous) approximation. The procedure is performed in such a way until the latest couple of values becomes closely equal (also see Sect. 6 of Introduction).

Having defined masses and mass percents of initial solutions it is easy to calculate required mixture parameters. In the calculation shown on Fig. 3.1 it was comprised and solved system of two equations. The most interesting is that the mixture volume is not equal to sum of volumes of initial solutions. We know it from "vodka phenomenon": 50 ml of ethanol plus 60 ml of water give volume little less than 100 ml. However, this phenomenon is explained by dependence of solution density from its concentration and by other reasons.

As was noted above aqueous solution of NaCl is used as a heat transfer (cooling medium) because temperature of such solution can be less than zero degrees centigrade. Figure 3.6 shows the Internet site for calculation of freezing temperature for some aqueous solutions (including NaCl).

A cold radiator in winter is awful. Although, it can be very useful in summer heat because it substitutes an air conditioner in a room. Such conditioning system have already been implemented. Certainly, there have been used special heat exchangers with blowers rather than radiators. In winter it is fed by warm NaCl solution, in summer—by NaCl solution at sub-zero temperature.

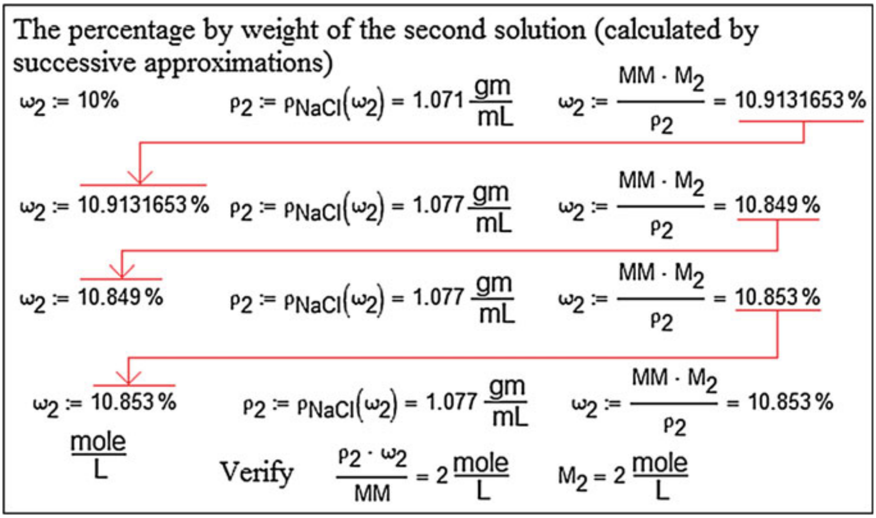


Fig. 3.5 Method of sequential approximations in a manual mode

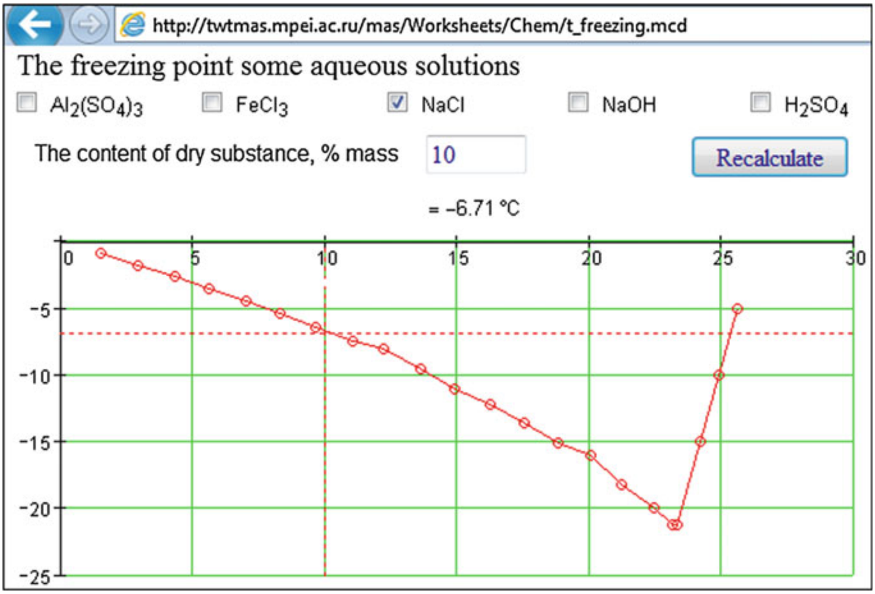


Fig. 3.6 Freezing temperature of some aqueous solutions depending on temperature

The calculation shown on Fig. 3.1 can be drastically simplified using function shown on Fig. 3.7. As the function shown on Fig. 3.3 it also returns density of NaCl solution depending on its concentration. However, argument of the new function can be not only mass percent of sodium chloride but also other types of concentration, namely, molality, molarity (see Fig. 3.1), and titre of the solution. Titre is a mass concentration expressed in g/mL or g/L. The key element of the function shown on Fig. 3.7 is Mathcad built-in function **SIUnitsOf** returning dimension of its argument. It allows us to define type of the concentration inserted by a user in program as an argument of function $\boldsymbol{\omega_{NaCl}(c)}$ and make corresponding recalculations.

Function shown on Fig. 3.7 can be simplified—it is possible to omit cycle defining of mass percent by molarity or titer adding three strings to matrix **M** that will hold discrete values of NaCl solution concentration expressed by molality, molarity, and titer.

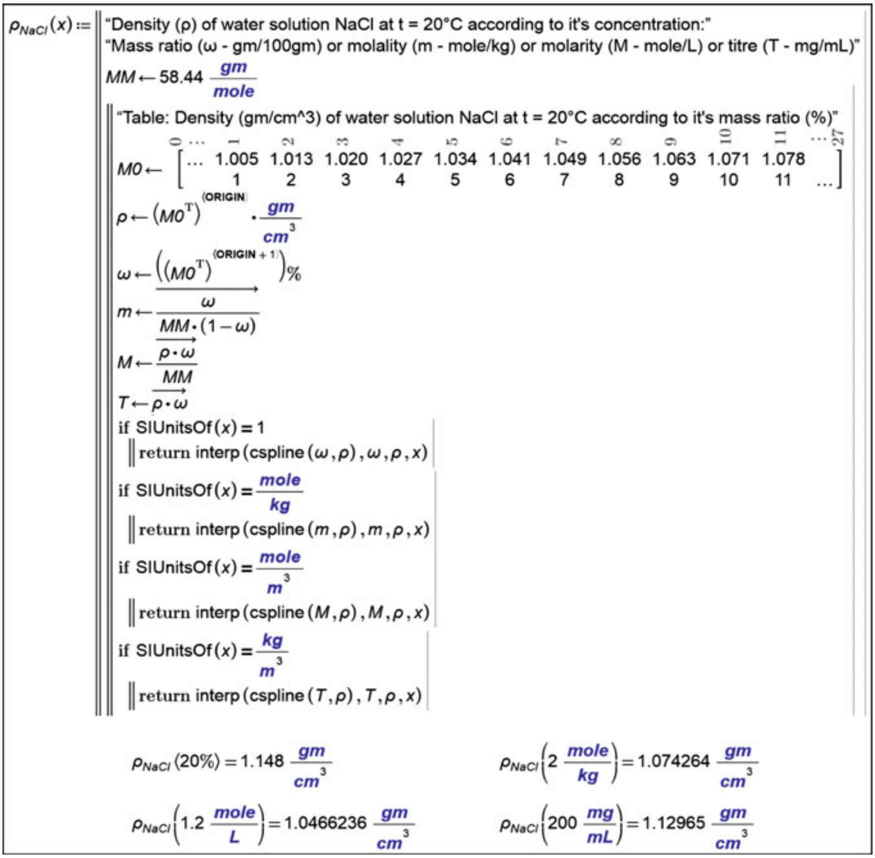


Fig. 3.7 Calculation of density of NaCl solution for any type of concentration

3.1 More About Concentration of Solutions

Chemists in Russia and in Europe have left off measuring hardness of water in degrees[5] and substituted meq/l. Using SI system is not just recommended but legitimized almost all over the world.

For example, according to SI system demands we cannot say “calcium concentration in solution equals to 1 eq/L”. There is no more such a unit—eq. Also, we cannot say “calcium concentration in solution equals to 1 mol/L”. We should specify type of concentration—molar or normal (to be more exact, molar concentration of equivalents), that is, note what is operational unit in a given solution, calcium ions or charges of calcium ions (cations).[6] Omission may result here in significant mistake: charges of calcium ions in volume unit twice as much as calcium ions[7]: for example, if molar concentration of calcium in solution equals to 1 mol/L, normal (equivalent)—2 mol/L. Old but holding ground method of disclosure of this omission is to use two groups of amount of a substance: moles (to be more exact, g-mole, mg-mole, gram-molecular, etc.) and equivalents (g-eq, mg-eq). They are similar by physical and chemical essence but, as a rule, differ in values. Although, sometimes is specified what type of concentration is meant. But it does not belong to do this if it is clear from concentration unit, g/L or mole/L. Sometimes disagreements transfers into denominator, too. It is considered that we cannot put liters (capacity unit) into denominator, we should change them for dm^3 (volume unit, length unit in a cube). Indeed, concentration is something divided into volume, rather than capacity. In thermal engineering, where water (plus steam) is the main working medium, water hardness is often related to solution mass (meq/kg) rather than volume (meq/L). Water volume varies greatly with heating (particularly with evaporation), but mass doesn’t.

SI system eliminates units ended with equivalents and reserves only mole and another way to note operational unit considering in concentration definition. Today, as was mentioned above, we cannot just say “calcium concentration equals to …” specifying operational unit of concentration—mol/L or eq/L (meq/L). It is required to note additionally type of concentration: molar (ionic) or equivalent (normal, “charge”). In other words, SI demands, allowing only mole and excluding equivalents transfer specification of operational unit from dimension to name of physical (more accurately chemical, physical and chemical) value.

[5]Many of properties were measured in degrees earlier: hardness of water, strength of spirits, hardness of metals, liquid viscosity, etc. SI system almost completely excludes degrees from metrology, which were clear only to focused specialists, and change them to more “physical” units. Now speaking of temperature we even should say just kelvin rather than degree Kelvin.

[6]This thesis can be illustrated for the readers far from chemistry with a geometric analogy. For example, we cannot say “circle size equals to 20 cm”. It is required to specify what we mean, diameter of radius of a circle.

[7]Diameter of a circle twice as much as its radius (see previous footnote).

Chapter 4
My First Power Engineering Mathcad-Calculation

Valery Ochkov and Konstantin Orlov

Abstract This chapter introduces how to use cloud functions and the WaterSteamPro software for calculation of thermophysical properties of water and steam. The thermal efficiency of steam-turbine Rankine cycle calculation is taken as example. A comparison between spreadsheet software and mathematical package is provided.

If we have the functional properties of the basic working fluids at hand and functions for coolant thermal engineering problems—(see Chap. 1) and are able to work with some dimensional quantities (see Chap. 2), we will be able to solve easily some tasks—to calculate thermal efficiency (η_t) of a simple steam turbine power generation cycle.

There is a very useful tool in Mathcad environment—to make links to the other documents, we already discussed it in Chap. 1 (see Fig. 1.5, for example). It is possible to make such a link to a document, saved on your own computer, local network or the Internet in Mathcad 15. The command is shown in Fig. 4.1 for Mathcad 15; by doing that you can make a link to a very interesting and useful Mathcad-sheet named **H2O**, with extensions **xmcdz**, **xmcd** or **mcd** saved on the Internet at http://twt.mpei.ru/tthb.

After such a reference in the Mathcad-worksheet functions will become available (visible, as programmers say), which calculate the properties of water/steam, depending on the parameters of the working fluid (see upper part of Fig. 4.2).

The site of the chapter: https://www.ptcusercommunity.com/message/423016.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_4

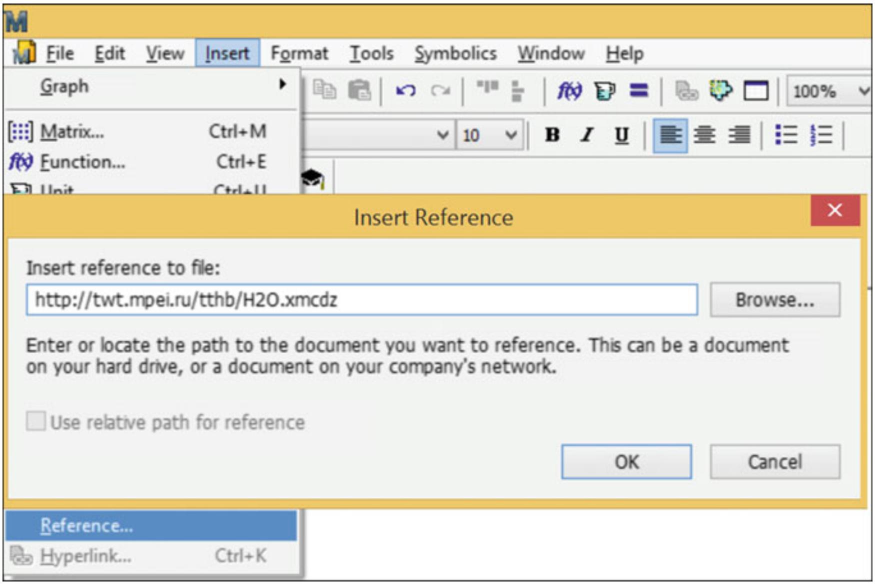


Fig. 4.1 Link to a cloud Mathcad-worksheet

When calculating the thermal efficiency of a steam turbine cycle we need the following functions with the prefix **wsp** (see in Fig. 2.21):

- **wspHPT**—specific enthalpy (**H**) of water/steam as a function of pressure (**P**) and temperature (**T**);
- **wspSPT**—specific entropy (**S**) of water/steam as a function of pressure (**P**) and temperature (**T**);
- **wspTSP** (or **wspTSatP**)—temperature (**T**) of water/steam at saturation line (**S/Sat**—here saturation) as a function of pressure (**P**);
- **wspXTS**—dryness factor (**X**) of wet steam as a function of the temperature (**T**) and the specific entropy (**S**);
- **wspHSWT** (or **wspHSatWT**)—specific enthalpy (**H**) at the water/steam saturation line (**S/Sat**) of water (**W**) as a function of temperature (**T**);
- **wspSSWT** (or **wspSSatWT**)—specific entropy (**S**) at the water/steam saturation line (**S/Sat**) of water (**W**) as a function of temperature (**T**);
- **wspTPS**—temperature (**T**) water/steam as a function of pressure (**P**) and specific entropy (**S**).

Having at hand the above functions, it is easy to calculate the specific enthalpy of water/steam at various points in the steam turbine (Rankine) cycle, the specific work of the turbine ($\mathbf{l_t}$ − t, turbine), the specific work to drive the feed pump ($\mathbf{l_p}$ − p, pump), the specific heat needed in the steam boiler ($\mathbf{q_b}$ − b, boiler), and, finally, the desired value of the thermal efficiency (see end of Fig. 4.2).

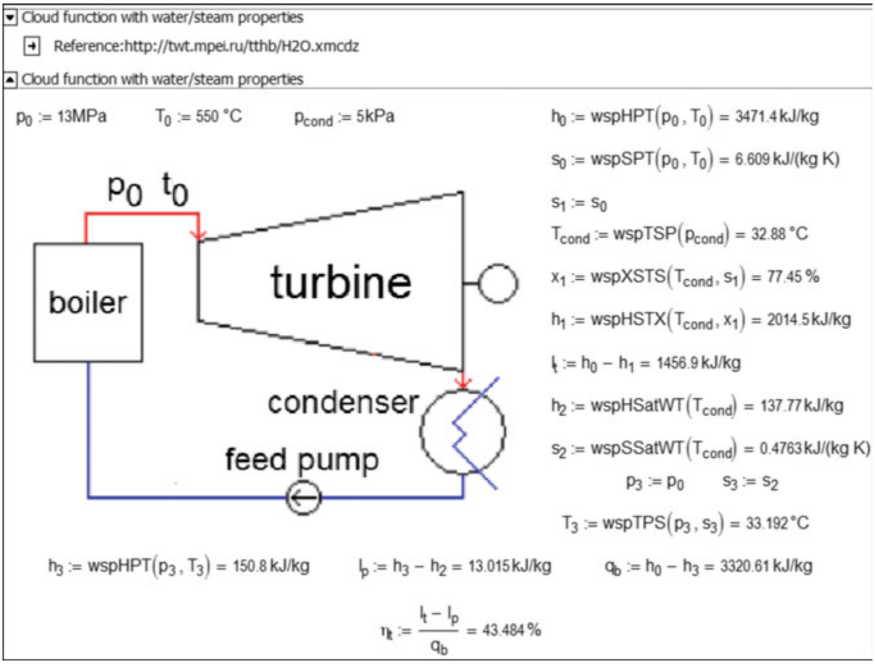


Fig. 4.2 The calculation of the thermal efficiency of a steam turbine cycle in Mathcad 15 by using the "cloud" functions

The calculation, shown in Fig. 4.2, can only work if the user's computer is connected to the Internet. If this is not the case, the link shown in Fig. 4.2 will turn red and we get an error message (see Fig. 4.3).

There are two ways for correcting this emergency situation, which was recorded in Fig. 4.3.

Firstly, you can upload the computer program WaterSteamPro, which we have already mentioned in the Chap. 2 (see Figs. 2.21 and 2.22). File "H2O.xmcdz" was designed to provide the subset of WaterSteamPro functions. And functions from "H2O.xmcdz" can be used without installation of the WaterSteamPro. However functions from WaterSteamPro can be used not only in Mathcad. For using WaterSteamPro functions with units in Mathcad you only need to insert a calculation link to the file watersteampro.xmcd, supplied by the program WaterSteamPro (In Fig. 4.3 you can find a pair of operators: assignment of units of specific enthalpy (**kJ/kg**) and entropy [**kJ/(kg K)**]. Using such a complex variable name is done through <Shift> + <Ctrl> + <k>).

If you click twice on a link to the "cloud" of the function (in Fig. 4.3 it does not work, because there is no connection to the Internet), it will be downloaded and opened the Mathcad-sheet, the beginning and the middle part of which is shown in Fig. 4.4.

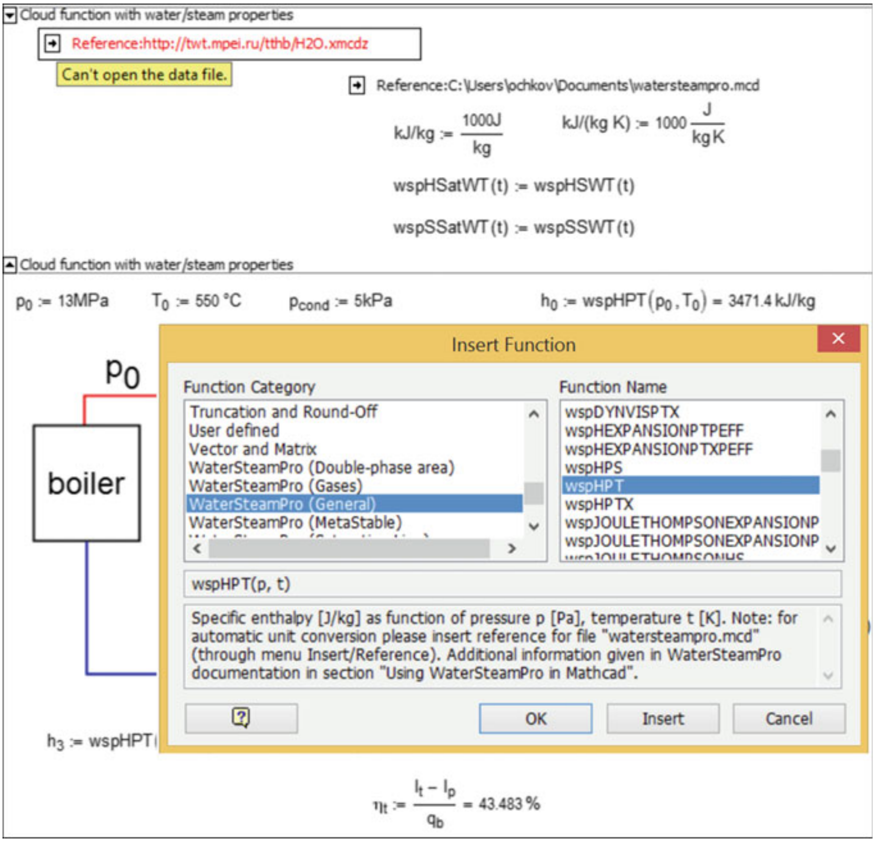


Fig. 4.3 The calculation of the thermal efficiency of steam turbine cycle with using the functions of the WaterSteamPro

The following figure shows the call of the function **wspHPT**, calculating the specific enthalpy of water and steam as a function of pressure and temperature. The function is hidden and password protected in a separate area. This and other functions are created on the recommendations of the International Association for the properties of water and steam (www.iapws.org), in which the author of this book works.

A downloaded file (named H2O.xmcdz) can be saved on your computer or on your local network, and you can make a link to it, as shown in Fig. 4.5.

With the functions, shown in Fig. 4.2, it is easy to create a function, that returns the thermal efficiency $\boldsymbol{\eta_t}$ of a simple steam turbine power generation cycle as function of the pressure $\mathbf{p_0}$, temperature $\mathbf{T_0}$, and condenser pressure $\mathbf{p_{cond}}$ (see Fig. 4.6).

The intermediate results are presented in Fig. 4.2, which can be used for debugging the function in Fig. 4.6.

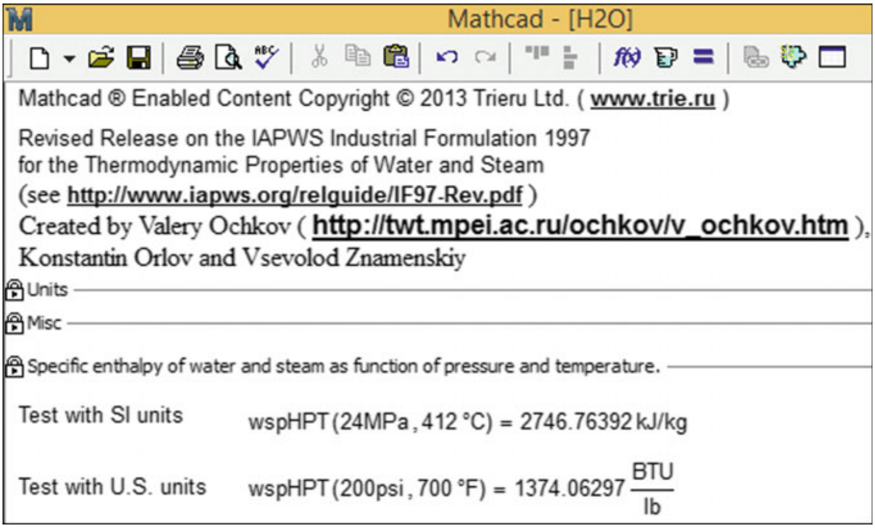


Fig. 4.4 "Cloud" Mathcad-functions on the properties of water/steam

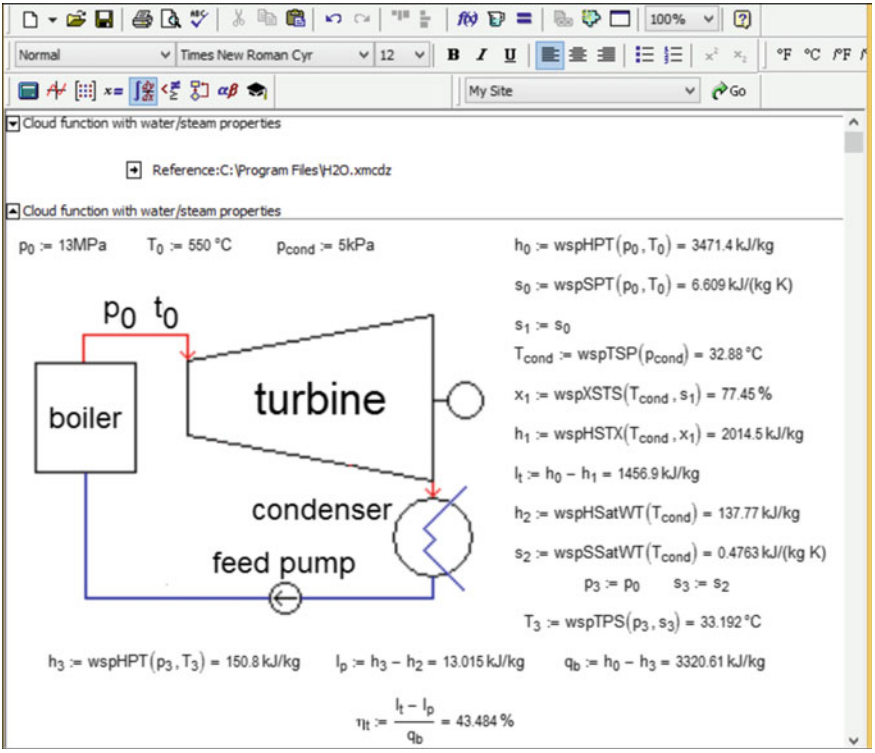


Fig. 4.5 Link to Mathcad-sheet that is stored on your computer

If you put the Mathcad-document with its functions shown in Fig. 4.6 for example, on your desktop, then the file can also be referenced (see Fig. 4.7).

Figure 4.7 does not provide a link to the file H2O.xmcdz—it is saved in the file named Eff-Ideal-Simple-PTU.xmcdz.

The technology of creating user functions and save them on your computer, local network or on the Internet is an effective technique for solving engineering problems including thermal engineering as we have already mentioned in Chap. 1. Having at hand such functions it is also easy to build diagrams for thermal engineering cycles [34]. In the book the reader can find a calculation of the thermal efficiency of a simple cycle steam turbine plant (STP) with superheated steam, made in the environment SMath.

Many engineers, scientists and technical workers perform calculations in the spreadsheet-program MS Excel™, which was mentioned in the introduction. Figure 4.8 shows a table in MS Excel, to which is attached WaterSteamPro functions by DLL technology. The list of the functions can be seen in the dialog box, shown in the same figure (see also Fig. 4.3). It is possible to calculate the efficiency of the steam turbine cycle using the formulas shown in Fig. 4.2. The initial data is entered in cell **B2**, **B3** and **B4**. The remaining cells in the column are filled with formulas, which carry out calculations. These formulas are shown in Fig. 4.9. This spreadsheet is made visible after entering the command **Show Formula**, tab **Formula**.

Comparing Figs. 4.2 (Mathcad), 4.8 and 4.9 (MS Excel), you can once again see that the work with a spreadsheet-program is much more difficult and complicated than the work with scientific calculator Mathcad.

There is a tool for simplifying formulas in MS Excel, fixing each entry names to its values, which are stored in the cells. Figure 4.10 shows the Excel-calculation displayed in Figs. 4.2 (Mathcad), 4.8 and 4.9 (Excel), with the difference that in the formulas of column **B** instead of the coordinates of the cells (**B5**, **D2**, and so on), the names of these cells (**h0**, **p0**, and so on) appear, which have been pre-assigned by the command **name of the range**.

So the result of the table cell **B5** in the calculation shown in Fig. 4.10, has a name h_0 (subscript here invalid) and stores the specific enthalpy of steam entering the turbine. This parameter is calculated using the **wspHPT(p0, T0)**, the arguments of it are stored in the cells **D2** and **D3** named **P0** and **T0**. This association between cells is further displayed through the command **Trace Precedents**, tab **Formula**.

In these calculations (and they can be quite complex) are not only formulas used in the cells, based on numeric values stored in other cells, as shown in our MS Excel-tables, but also software modules written in the programming language VBA (Visual Basic for Application). These calculations are practically impossible to understand. But inserting them into your Mathcad-calculation is possible and necessary. How it's done—see below.

Some of the names of table cells, shown in Fig. 4.10, have an index: h_1, s_2 etc. It is done due to the fact that the cell names **h1**, **s2**, and so on, coincide with the "coordinates" **H1**, **S2**, and so on (**H1**—cell in column **H** and row 1).

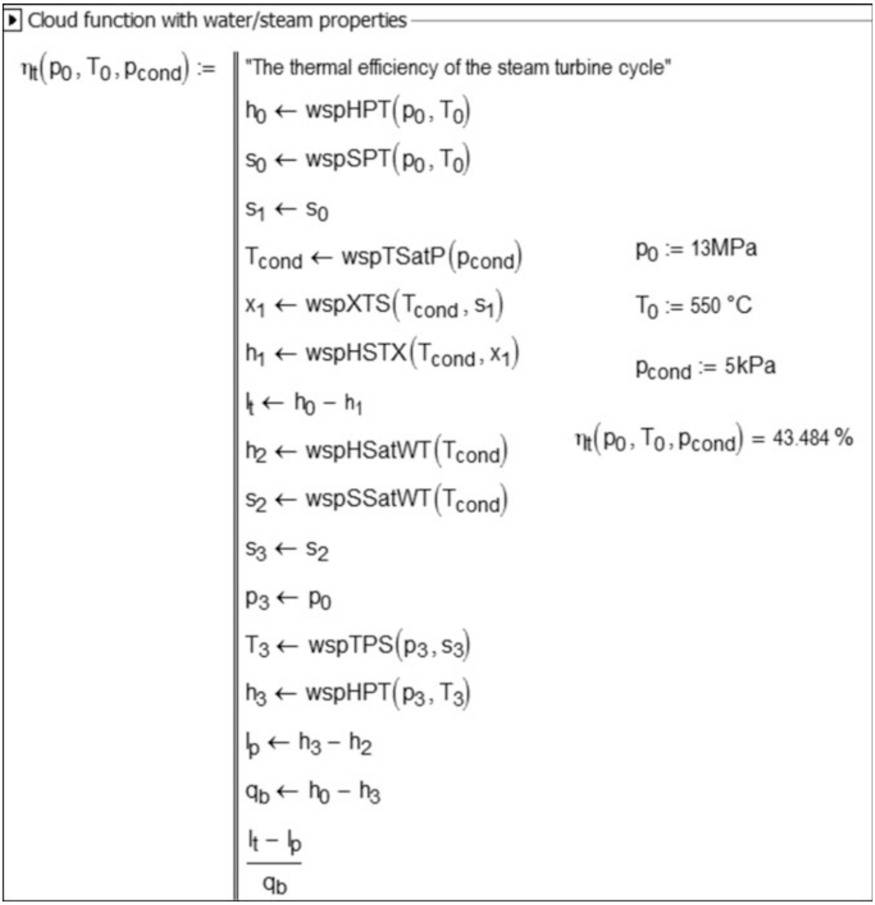


Fig. 4.6 User function in Mathcad

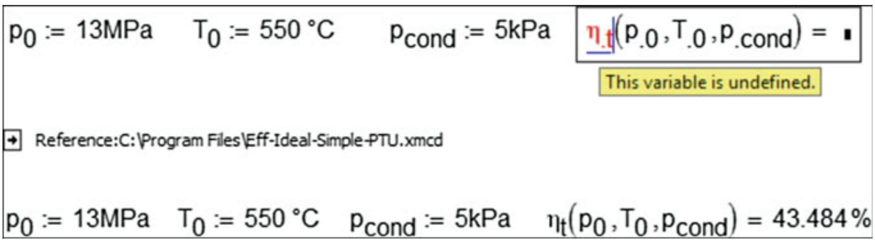


Fig. 4.7 Reference to the calculation of the efficiency of a steam turbine cycle

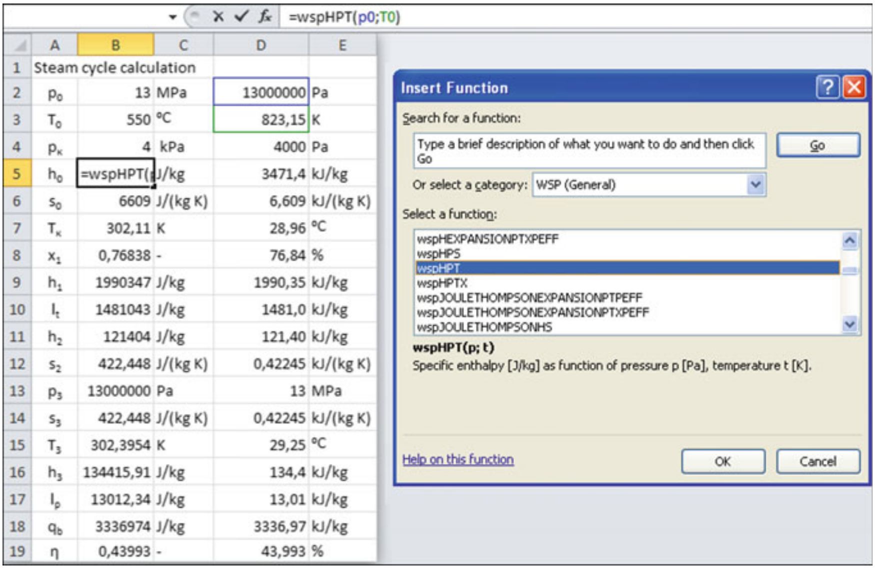
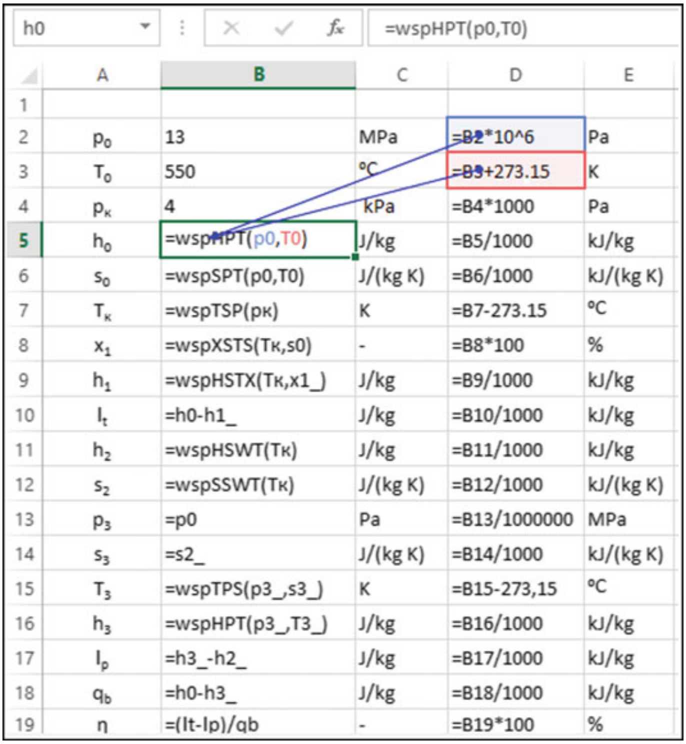


Fig. 4.8 The calculation of the efficiency of steam turbine cycle with MS Excel

	A	B	C	D	E
1					
2	p_0	13	MPa	=B2*10^6	Pa
3	T_0	550	°C	=B3+273.15	K
4	p_K	4	kPa	=B4*1000	Pa
5	h_0	=wspHPT(D2,D3)	J/kg	=B5/1000	kJ/kg
6	s_0	=wspSPT(D2,D3)	J/(kg K)	=B6/1000	kJ/(kg K)
7	T_K	=wspTSP(D4)	K	=B7-273.15	°C
8	x_1	=wspXSTS(B7,B6)	-	=B8*100	%
9	h_1	=wspHSTX(B7,B8)	J/kg	=B9/1000	kJ/kg
10	l_t	=B5-B9	J/kg	=B10/1000	kJ/kg
11	h_2	=wspHSWT(B7)	J/kg	=B11/1000	kJ/kg
12	s_2	=wspSSWT(B7)	J/(kg K)	=B12/1000	kJ/(kg K)
13	p_3	=D2	Pa	=B13/1000000	MPa
14	s_3	=B12	J/(kg K)	=B14/1000	kJ/(kg K)
15	T_3	=wspTPS(B13,B14)	K	=B15-273.15	°C
16	h_3	=wspHPT(B13,B15)	J/kg	=B16/1000	kJ/kg
17	l_p	=B16-B11	J/kg	=B17/1000	kJ/kg
18	q_b	=B5-B16	J/kg	=B18/1000	kJ/kg
19	η	=(B10-B17)/B18	-	=B19*100	%

Fig. 4.9 Calculation of the thermal efficiency of a steam turbine cycle with MS Excel: showing the hidden formulas



h0 | =wspHPT(p0,T0)

	A	B	C	D	E
1					
2	p_0	13	MPa	=B2*10^6	Pa
3	T_0	550	°C	=B3+273.15	K
4	$p_к$	4	kPa	=B4*1000	Pa
5	h_0	=wspHPT(p0,T0)	J/kg	=B5/1000	kJ/kg
6	s_0	=wspSPT(p0,T0)	J/(kg K)	=B6/1000	kJ/(kg K)
7	$T_к$	=wspTSP(pк)	K	=B7-273.15	°C
8	x_1	=wspXSTS(Tк,s0)	-	=B8*100	%
9	h_1	=wspHSTX(Tк,x1_)	J/kg	=B9/1000	kJ/kg
10	l_t	=h0-h1_	J/kg	=B10/1000	kJ/kg
11	h_2	=wspHSWT(Tк)	J/kg	=B11/1000	kJ/kg
12	s_2	=wspSSWT(Tк)	J/(kg K)	=B12/1000	kJ/(kg K)
13	p_3	=p0	Pa	=B13/1000000	MPa
14	s_3	=s2_	J/(kg K)	=B14/1000	kJ/(kg K)
15	T_3	=wspTPS(p3_,s3_)	K	=B15-273,15	°C
16	h_3	=wspHPT(p3_,T3_)	J/kg	=B16/1000	kJ/kg
17	l_p	=h3_-h2_	J/kg	=B17/1000	kJ/kg
18	q_b	=h0-h3_	J/kg	=B18/1000	kJ/kg
19	η	=(lt-lp)/qb	-	=B19*100	%

Fig. 4.10 The calculation of the thermal efficiency of steam turbine cycle with MS Excel: working with named cells

There is a very useful tool for inserting MS Excel sheets in Mathcad. This tool is used in two cases: firstly, when there is a ready-made calculation or a fragment of a calculation in MS Excel which you do not want to rewrite for Mathcad, and secondly, when you want to use in your Mathcad-document any specific tool from MS Excel—when, for example, you need to build a pie chart, which is not possible in Mathcad.

Figure 4.11 shows the calculation of the thermal efficiency of our simple steam turbine cycle in Mathcad with embedded MS Excel-components, which is displayed in Fig. 4.8.

Figure 4.12 shows a diagram of a simple steam turbine plant, for which is performed the calculation. Such schemes are given a lot in this book. They are usually drawn with the help of a graphical editor (using Paint, for example, a component of Windows), and inserted into a Mathcad document. The knowledge of schemes of thermal devices and their drawing "by hand" or a by computer—is a form of training and testing of students and energy specialists to raise their qualification. Alexey Ochkov developed a specialized program for the automation of the work [35]. Please visit http://twt.mpei.ac.ru/CALCULON [35–37].

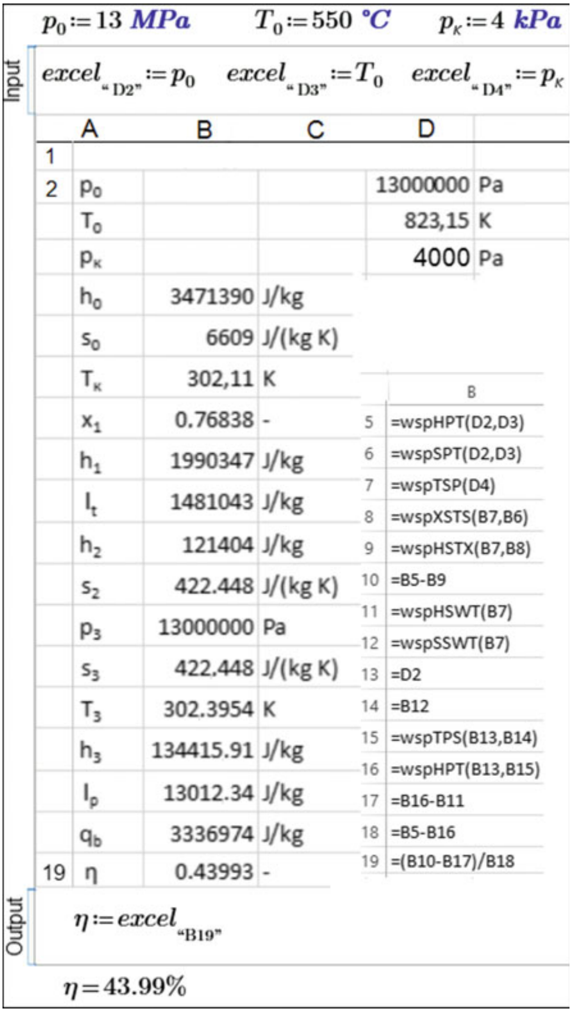


Fig. 4.11 The calculation of the thermal efficiency of steam turbine cycle in Mathcad with built-in MS Excel components

Figure 4.12 shows the window for testing the knowledge loaded with a blueprint (lower part) and a diagram drawn by the user (upper part). There is a direct link to this demo on the Internet at http://twt.mpei.ac.ru/CALCULON/TWTSchemeQuest_BOOK1/TWTSchemeQuest_BOOK1.htm. To run the demo, you must install a special multimedia shell TVT Shell, a reference to the installation of which (http://twt.mpei.ac.ru/twtshell) will automatically be offered on the site.

After drawing the user schema and clicking **Ready** the program compares with a special "grading" algorithm the scheme drawn by the user with the standard. In the case of error detection, the program will automatically indicate what is missing or what is messed up. For example, Fig. 4.13 shows the result of the program in the case of forgetting to draw a feed pump.

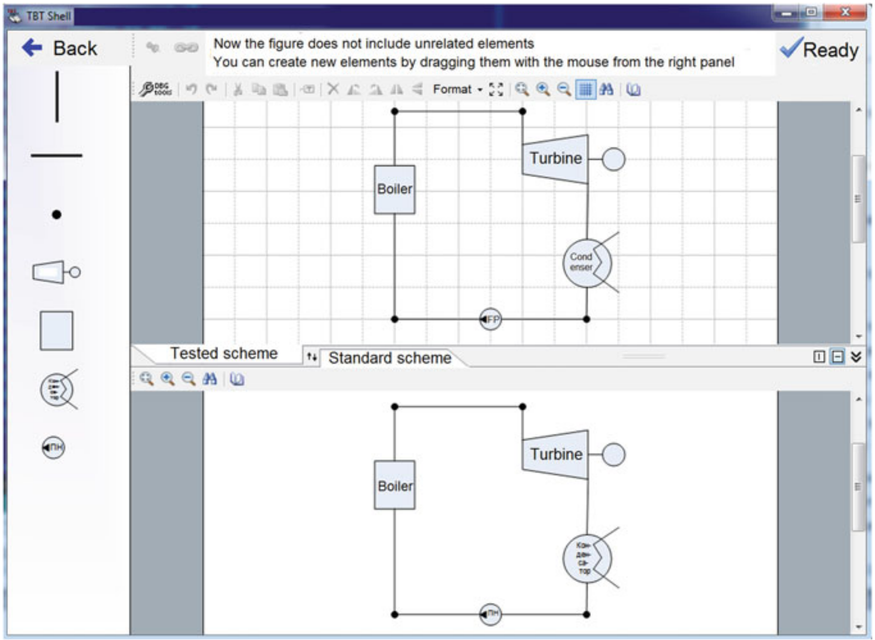


Fig. 4.12 Scheme testing

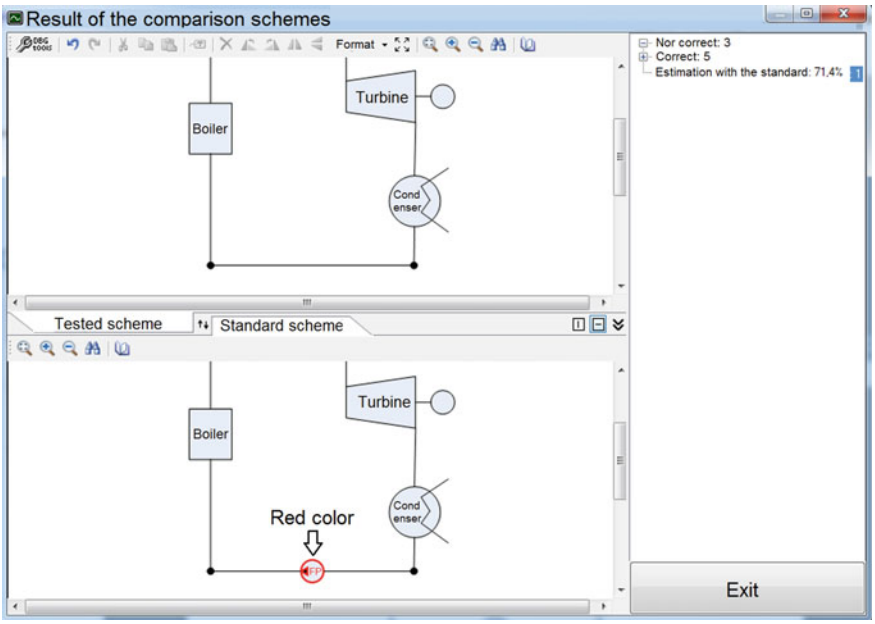


Fig. 4.13 Comparison: user-drawed steam turbine circuit compared with original version

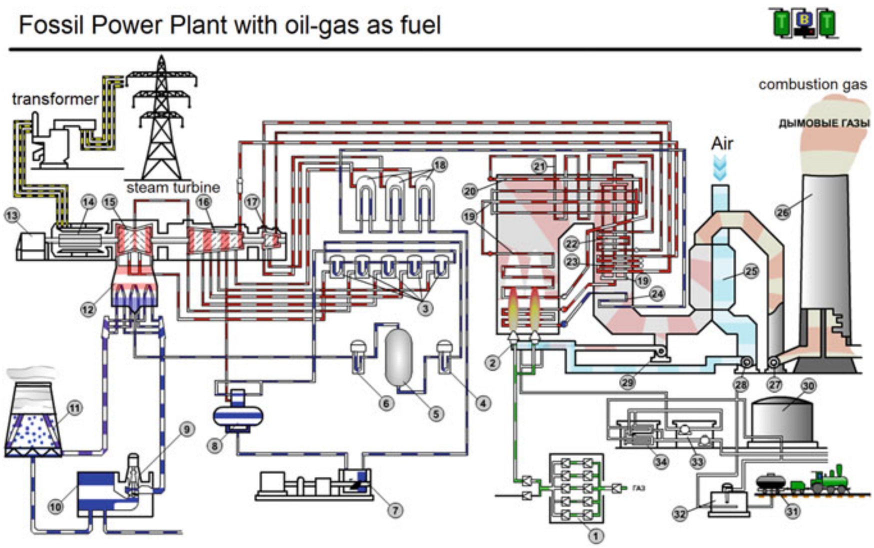


Fig. 4.14 One of the heat engineering schemes from the “Electronic Encyclopedia of Energy”

The comparison of the two schemes is the most important part of this program. A detailed discussion is beyond the scope of this book, but for readers, who are interested in it, the following information may be useful.

As a basis for the comparison algorithm the original Gemini II algorithm [36] is taken. The family of Gemini (The Twins) algorithms is based on the iterative calculation and comparison of the codes of individual elements and “networks” of the graph. It is widely used in the electronics industry to solve the LVS (Layout-Versus-Schematic, comparison of schematic diagrams and final PCB design or chip) problem.

In the process of developing an algorithm several changes in the original algorithm have been made, in particular, the problem of the impossibility of using “isopotential networks” has been solved, because sometimes the comparison of technological schemes requires an exact match topology (piping, pumps, valves, and so on).

The program for testing the knowledge of schemes is a part of the author’s “Electronic Encyclopedia of Energy” [37], (www.trie.ru), where you can find a lot of heat engineering schemes. One of such scheme is shown in Fig. 4.14.

Pressing the round button, shown in the lower part of Fig. 4.13, you can select individual parts of the heat power station. In addition, from this multimedia scheme you can quickly jump to the corresponding description in the encyclopedia.

The authors express their deep gratitude to Vsevolod Znamensky who wrote the main part of the H2O Mathcad-sheet.

Chapter 5
"Cloud" Thermal Engineering Algorithms

Valery Ochkov and Konstantin Orlov

Abstract This chapter explains how cloud algorithms can be used in the heat power processes calculations for the thermal power plants, for example, the processes of expansion or throttling of the steam, compression water in the pump. Information technology for the Internet libraries of equations is provided.

For thermodynamic calculations one shall know not only properties of working fluids (water, steam, air, flue gases etc.) [23, 38, 39, 40] and of thermal engineering related materials, but also of calculation formulas.

These formulas or sets of formulas with instructions for their application (formulations are taken from various reference literature sources, for example, from [41] or from Internet and are used for relevant calculations. Such work method is not quite convenient as it hinders execution of design calculations, and enhances occurrence of error risk in them. In view of the above, and also considering the fact that at present PC of almost all specialists making thermodynamic calculations and other engineering calculations are provided with continuous and high-speed Internet, we have developed new technology for work with formulas. It will be demonstrated with the use of simple standard examples typical for the engineering office (calculator) Mathcad [42], which is quite popular among the students, engineers and research engineers.

"Cloud" technology allowing work with formulas is a development of approach described in Chap. 4 [40] and related to use of function for calculation of energy

---

The site of the chapter: https://www.ptcusercommunity.com/message/423019.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_5

working fluid properties contained in Internet. We'll illustrate work with such functions by simple, but at the same time, quite typical examples.

When designing any facility, after introduction of initial data one shall put the link reference to Mathcad-document (contained in Internet (in the "clouds") at the address http://twt.mpei.ru/TTHB/H2O.xmcdz) into the calculation (see Fig. 4.1 in Chap. 4). After making this link reference all functions defined in Mathcad-document to which the link reference is made will become available in the working document. These are the functions converting heat-transfer properties of water and water steam to the following: specific enthalpy (**h**), specific entropy (**s**), temperature (**T**), dryness factor (**x**) etc. [23] as well as functions converting parameters (calculations) of some thermal processes (throttling in control valve or steam expansion (work) in steam turbine) into values of specific enthalpy and steam dryness factor at the end of the above processes. Names of these two functions have radicals in the name **EXPANSIONPTPEFF**[1]: **EXPANSION**-expansion, **PT**—steam pressure (**P**) and temperature (**T**) in the initial point, **P**—steam pressure in the end point and **EFF** (efficiency)—internal relative steam expansion process efficiency.

This approach is an alternative to application of dedicated software tools for calculation of working fluid properties [43], which can be downloaded and installed on PC for use. There is no need in downloading and installation of such software tools on the user's part if one uses "cloud" functions. As a clear benefit of "cloud" technology we can mention capability of on-line function updating: as a result the user always uses the most current calculation version.

Among the drawbacks of "cloud" functions we can mention the need for Internet access when making calculations. In this case as an alternative we can recommend usage of standard software tools. Thus, in the absence of Internet access one can install the software tool [43], and it will work for the examples provided in the present book. It is provided by similar names and arguments of functions used for calculations of working fluid properties. As another option we can recommend downloading of Mathcad-file under the name H2O.xmcdz, with its further installation on PC (for example, on the desktop, with link reference to it from the working calculation as it is shown in Fig. 5.1).

The use of "cloud" technologies for formulas and formula sets gives the user access to completed, ready calculations which can be directly apply in design efforts and which represent solutions of specific tasks.

Task 5.1. It is necessary to calculate water steam throttling in a control valve and its further expansion in a steam turbine. These processes shall be shown in *h, s*- and *T, s*-diagrams of water and water steam (Fig. 5.1).

[1]Let us remind you that the names of all functions are prefixed with **wsp**. It is an abbreviation of the WaterSteamPro software which can be downloaded from www.wsp.ru. The "cloud" functions described in this book are a small part of WaterSteamPro software.

Input data $p_1 := 24\text{MPa}$ $t_1 := 550\,°\text{C}$ $p_2 := 10\text{MPa}$ $p_3 := 5\text{kPa}$ $\eta_{i\_r\_1\text{-}2} := 5\%$

Link on cloud functions → Reference:http://twt.mpei.ru/TTHB/H2O.xmcdz $\eta_{i\_r\_2\text{-}3} := 75\%$

$h_1 := \text{wspHPT}(p_1, t_1) = 3350.89\ \text{kJ/kg}$ $s_1 := \text{wspSPT}(p_1, t_1) = 6.2116\ \text{kJ/(kg K)}$

$h_2 := \text{wspHEXPANSIONPTPEFF}(p_1, t_1, p_2, \eta_{i\_r\_1\text{-}2}) = 3338.13\ \text{kJ/kg}$

$t_2 := \text{wspTPH}(p_2, h_2) = 485.8\,°\text{C}$ $s_2 := \text{wspSPT}(p_2, t_2) = 6.5511\ \text{kJ/(kg K)}$

$x_3 := \text{wspXEXPANSIONPTPEFF}(p_2, t_2, p_3, \eta_{i\_r\_2\text{-}3}) = 90.56\,\%$

$t_3 := \text{wspTSP}(p_3) = 32.9\,°\text{C}$

$h_3 := \text{wspHEXPANSIONPTPEFF}(p_2, t_2, p_3, \eta_{i\_r\_2\text{-}3}) = 2332.15\ \text{kJ/kg}$

$s_3 := x_3\,\text{wspSSST}(t_3) + (1 - x_3)\,\text{wspSSWT}(t_3) = 7.6469\ \text{kJ/(kg K)}$

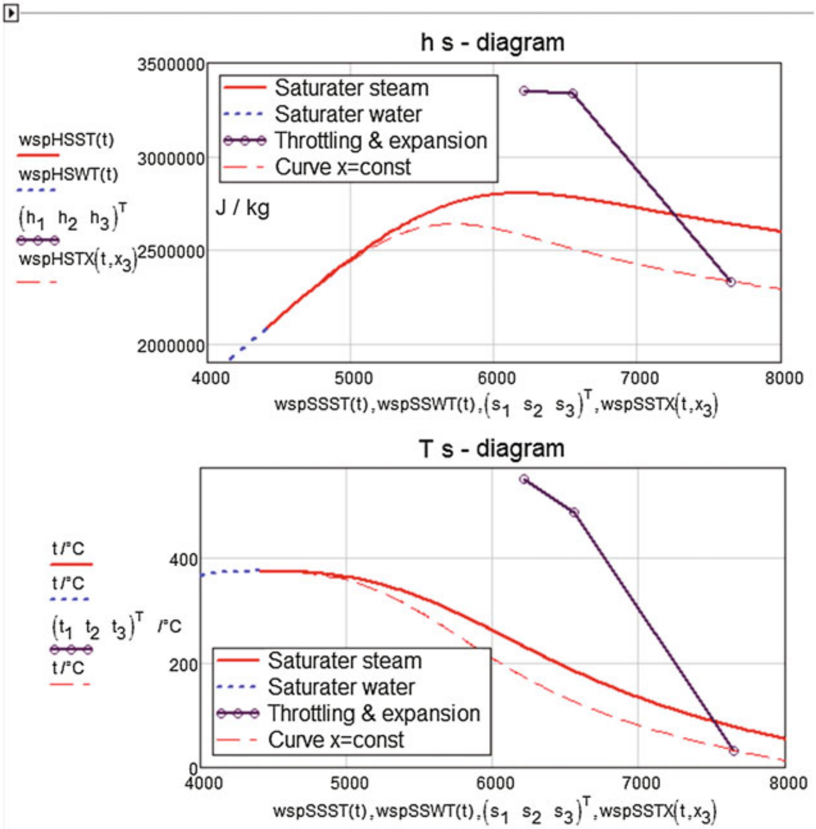


Fig. 5.1 Calculation process throttling and expansion of steam in the turbine

Initial data shall be entered as parameters of this calculation: pressure $\mathbf{p}_1$ and temperature $\mathbf{T}_1$ of fresh steam, pressure, to which steam is being throttled $\mathbf{p}_2$,[2] and pressure at the end of steam expansion process in the turbine (pressure in a condenser—$\mathbf{p}_3$). Additionally internal relative efficiencies of these processes shall be specified—$\boldsymbol{\eta}_{\mathbf{internal\ relative}}$. If that efficiency equals to 100 %, it is a perfect steam expansion (performance) process in the turbine where steam entropy value does not change. If that efficiency equals to 0 %, it is a "perfect" steam throttling process where enthalpy remains constant. With real throttling (for example, when steam goes along a steam pipeline) steam enthalpy slightly goes down which defines the value of "internal relative efficiency of throttling" (5 % in the calculation shown in Fig. 5.1) which is different from 0.

With handy ready-to-use functions one can quickly and unmistakably compute all required parameters of steam expansion (see Fig. 5.1) and show them on diagrams, where lines of saturation by water and water steam are additionally shown as well as a line of steam constant dryness (humidity) factor. All these lines are easily built on the diagrams with the following functions being visible in the calculation: **wspHSST**—specific enthalpy **H** of saturated (**S**—saturated) steam (**S**—steam) depending on temperature **T**, **wspHSWT**—the same for water (**W**—water) at the saturation line. If the letter H is to replace with **S**, in these two functions' names, they will result in specific entropies **s** of water/steam on the saturation line. Two functions are used to build the line of steam constant humidity—**wspHSTX** and **wspSSTX**, converting the specific enthalpy **h** and specific entropy **s** of wet steam depending on its temperature **T** and dryness factor **x**. On *h, s*-diagram parametric diagrams are built: parameter **T** (in Fig. 5.1 this area is hidden, while in Fig. 5.2 it is visible) is specified in a range between water critical point to its triple point (these constants also become visible in the calculation after the reference described above), while the axes fix three pairs of functions of **T**. On *T, s*-diagram argument (temperature) values are set on *y* axis and function values are set on *x* axis. About the temperature on the diagrams, by the way. Muliplier/**°C** has been introduced to the latest versions of Mathcad which allows for Celsius scale on diagram axes (Fig. 5.2). Kelvin scale is used by default on temperature diagram axes (see the second diagram on Fig. 5.1).

Task 5.2. A pump builds up water pressure from $\mathbf{p}_1$ value (water saturation line) to $\mathbf{p}_2$ value with internal relative efficiency of the process of $\eta_{\mathrm{internal\ relative}}$. Then water isobarically heats up to boiling point. Parameters of these processes shall be defined and shown on *T, h*-diagram.

Figure 5.2 shows this calculation using "cloud" functions. It is different from the calculation given in Fig. 5.1 due to only two details. First, function **wspEXPANSIONPTPXEFE** rather than **wspEXPANSIONPTPEFF** is used for computing the process of pressure build-up which has an additional argument

[2]We set very large pressure difference (24–10 MPa) in order to show that process on diagrams of Fig. 5.1. At real Thermal Power Plants pressure loss in a steam pipeline from the boiler to the turbine accounts for near 5 %. If $\mathbf{p}_2$ is to be entered less than $\mathbf{p}_1$ for those 5 %, then points 1 and 2 will coincide.

Input data $p_1 := 5\text{kPa}$ $p_2 := 13\text{MPa}$ $\eta_{\text{internal relative}} := 73\%$

Link on cloud function Reference:http://twt.mpei.ac.ru/TTHB/H2O.xmcdz

$T_1 := \text{wspTSP}(p_1) = 32.9\,°\text{C}$

$h_1 := \text{wspHSWT}(T_1) = 137.8\,\text{kJ/kg}$ $s_1 := \text{wspSSWT}(T_1) = 0.476\,\text{kJ/(kg K)}$

$n := 3000$ $i := 0..n$ $vp_{1\text{-}2_i} := p_1 + \frac{p_2 - p_1}{n} i$

$vh_{1\text{-}2_i} := \text{wspHEXPANSIONPTXPEFF}\left(p_1, T_1, 0, vp_{1\text{-}2_i}, \frac{1}{\eta_{\text{internal relative}}}\right)$ $h_2 := vh_{1\text{-}2_n} = 155.6\,\text{kJ/kg}$

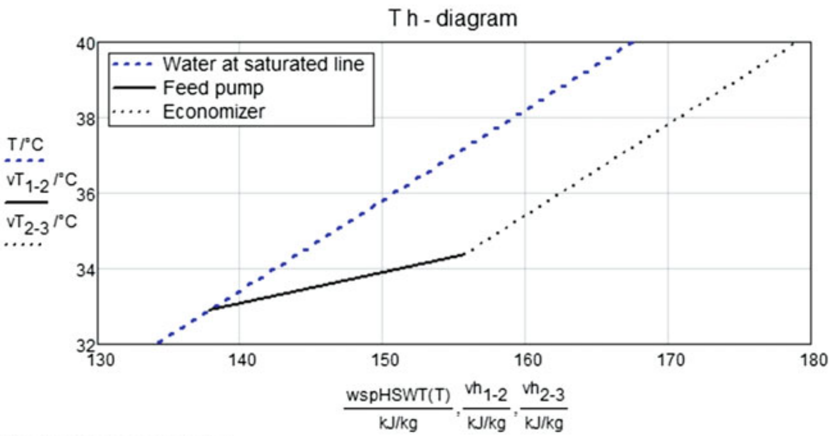
$vT_{1\text{-}2_i} := \text{wspT1PH}\left(vp_{1\text{-}2_i}, vh_{1\text{-}2_i}\right)$ $T_2 := vT_{1\text{-}2_n} = 34.4\,°\text{C}$

$vs_{1\text{-}2_i} := \text{wspS1PT}\left(vp_{1\text{-}2_i}, vT_{1\text{-}2_i}\right)$ $s_2 := vs_{1\text{-}2_n} = 0.492\,\text{kJ/(kg K)}$

$p_3 := p_2$ $T_3 := \text{wspTSP}(p_3) = 330.9\,°\text{C}$ $vT_{2\text{-}3_i} := T_2 + \frac{T_3 - T_2}{n} i$

$vh_{2\text{-}3_i} := \text{wspHPT}\left(p_2, vT_{2\text{-}3_i}\right)$ $h_3 := vh_{1\text{-}2_n} = 155.59\,\text{kJ/kg}$

$$T := T_{cr}, T_{cr} - \frac{T_{cr} - T_t}{n} .. T_t$$



Creating one reverse function

$$\text{wspPTH}(t, h) := \left| \begin{array}{l} p \leftarrow 1\text{MPa} \\ \text{root}(\text{wspHPT}(p, t) - h, p) \end{array} \right.$$

$\text{wspPTH}(T_2, h_2) = 13\,\text{MPa}$ $\text{wspXTH}(T_2, h_2) = 0.481\,\%$ $\text{wspPST}(T_2) = 5.43\,\text{kPa}$

Fig. 5.2 Pump performance calculation

x—dryness factor. It equals to 0 for water and 1—for saturated steam. And second. In Fig. 5.1 diagram points 1, 2 and 3 were jointed with straight lines. In Fig. 5.2 lines standing for the processes of pressure building-up in the pump and water heating in the economizer are shown in the diagram the other way: for that purpose vectors **vp** (pressure), **vh** (specific enthalpy), **vT** (temperature) and **vs** (specific entropy) were generated for processes 1–2 and 2–3, which are drawn on the diagram of Fig. 5.2. That's the process in cases when the line in a diagram is not straight or close to straight. In principle the similar way shall be taken for diagrams of Fig. 5.1, but it's difficult to define signatures 1–2 (throttling with efficiency equaled to 5 %) and 2–3 (steam expansion in the turbine with internal relative efficiency of the process equaled to 75 %), as the values of local efficiency might differ for different areas of the process. Line 2–3 in Fig. 5.2 is not straight. One can make sure by expanding the diagram area. That is why vectors were used for building this line in the diagram. The same was done with line 1–2.

In Fig. 5.2 *T, h*-diagram was built rather than *hs*- or *T, s*-diagram. *T, h*-diagrams are rarely used for demonstrating steam-turbine cycles, however they are used here for the following reason. At present students specializing in heat and power engineering and keening on computers have started building not only conventional "planar" diagrams of steam-turbine cycles (*hs*- and *T, s*-diagrams—see Fig. 5.1), but and 3D, dimensional diagrams. e.g., *h, T, s*-diagrams [34]. Two planes of such a dimensional diagram are shown in Fig. 5.1, while the third one (*Th*)—in Fig. 5.2. Looking at Fig. 5.2, one could mistakenly assume that the process of water pressure build-up "gets into" two-phase area to the right of the saturated water line. That's not the case. The area to the right of the water saturation line shows two areas of thermodynamic surface of water in *h*-, *T*- and *S*-coordinates. At point 2 water has the following parameters: p = 13 MPa, T = 34.4 °C and h = 156.6 kJ/kg. But the same parameters by **T** and **h** are applied for the "twin" of point 2—two-phase liquid: water with steam content of 0.481 %, temperature of 34.4 °C and pressure of 5.34 kPa.

A Mathcad user can broaden the list of "cloud" Mathcad-functions contained in the document **H2O.xmcdz** located at http://twt.mpei.ru/TTHB/H2O, using the tools of equations and system solutions or zero function searches. So, in the bottom of Fig. 5.2 creation and calling for **wspPTH** function is shown which converts water/steam pressure depending on temperature and specific enthalpy (one of two backward functions **wspHPT** of "cloud" functions package located at http://twt.mpei.ru/TTHB/H2O.xmcdz). The created function **wspPTH** uses the Mathcad-inbuilt function **root** and converts the value of pressure by temperature and specific enthalpy close to the point of first approximation by pressure (**1 MPa**). We do not include the complete list of functions as it is being continuously enlarged.

And a little bit more about the pump.

In Chap. 2 we gave the calculation of human heart power (see Fig. 2.10) as a pump transporting incompressible fluid. In that calculation we multiplied the value of volume flow rate of blood by pressure difference in the heart. Sometimes this simplified formula is applied by mistake to calculate apparatus transporting

Reference:http://twt.mpei.ru/tthb/H2O.xmcdz

Input Data

$$q := 70\,\frac{mL}{s} \qquad p_{in} := 80\,torr \qquad p_{out} := 120\,torr$$

Specific work of the heart

$$l_h := \int_{p_{in}}^{p_{out}} wspVPT(p, 36.6\,°C)\,dp = 5.368\,\frac{J}{kg}$$

Density of blood

$$\rho := \frac{1}{wspVPT\left(\frac{p_{out} + p_{in}}{2}, 36.6\,°C\right)} = 993.44\,\frac{kg}{m^3}$$

Power of the heart

$$W_h := l_h \cdot q \cdot \rho = 0.373\,W$$

or

$$W_h := q \cdot (p_{out} - p_{in}) = 0.373\,W$$

Fig. 5.3 Thermodynamic parameters of a human heart

compressible media—compressors—which is wrong as the value of volume flow rate here is not constant and depends on pressure. Figure 5.3 shows the right approach to calculation of power for pressure build-up. Power is be calculated by a definite integral of specific volume change function. Integration limits are values of inlet and outlet pressure.

If a working medium volume is not changed with pressure change (e.g., in the pump), integration element is practically constant and it can be taken out of the integral as a constant to derive the simplified formula above. If this is not the case (as in a compressor), then the simplified formula shown in Fig. 2.10 cannot be used because it will result in significant error in computations.

Afterpiece. Once the author gave classes to employees of Mosenergo, upgrading their skills. In these classes, among other things, the issues of this book were being considered—advanced computer means of solving thermodynamic problems. In particular, the attendees asked the author to explain how water mass flow rate can be calculated when volume flow rate is known. We solved that task in different software environments: MS Excel (Fig. 5.4), Mathcad (Fig. 5.5) and SMath (Fig. 5.6), and also posted in the Internet (Fig. 5.7).

There was a radio show "Let's learn a song", where fragments of a song were played by various musical instruments. It is unclear why it was done from a practical point of view. Out task related to water flow rate also is solved by various "tools".

	A	B	C		C
1	**Mass flow of water**				
2					
3	**Input data**				
4	Volume flow of water	m^3/hr	100		100
5	Pressure of water	atm	7		7
6	Temperature of water	°C	90		90
7					
8	**Intermediate values**				
9	Volume flow of water	m^3/sec	0,02778		=C4/3600
10	Pressure of water	Pa	709275		=C5*101325
11	Temperature of water	K	363,15		=C6+273,15
12	Density of water	kg/m^3	965,319		965,319
13					
14	**Results**				
15	Mass flow of water	kg/sec	26,814		=C9*C12
16	Mass flow of water	t/hr	96,532		=C15*3600/1000

Fig. 5.4 Calculation of water mass flow in MS Excel (demonstration of calculated values and formulas used; work with connected WaterSteamPro)

Fig. 5.5 Calculation of water flow rate in Mathcad: usage of a reference to a "cloud" function

Mass flow water calculation

Input data

$$q_{volume} := 100 \frac{m^3}{hr}$$

$$p := 1atm \qquad t := 90\,°C$$

Link on cloud function

Reference:http://twt.mpei.ru/tthb/H2O.xmcdz

$$\rho := wspDPT(p, t) = 965.319 \frac{kg}{m^3}$$

Answer

$$q_{mass} := q_{volume} \cdot \rho = 96.532 \cdot \frac{tonne}{hr}$$

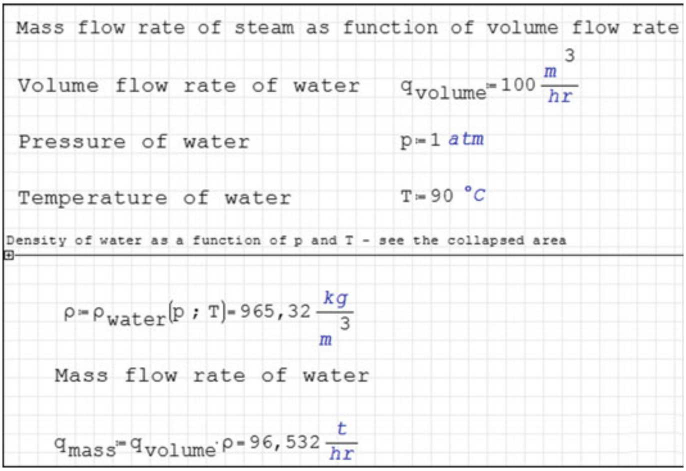


Fig. 5.6 Calculation of water mass flow rate in SMath: operation with a function in hidden area

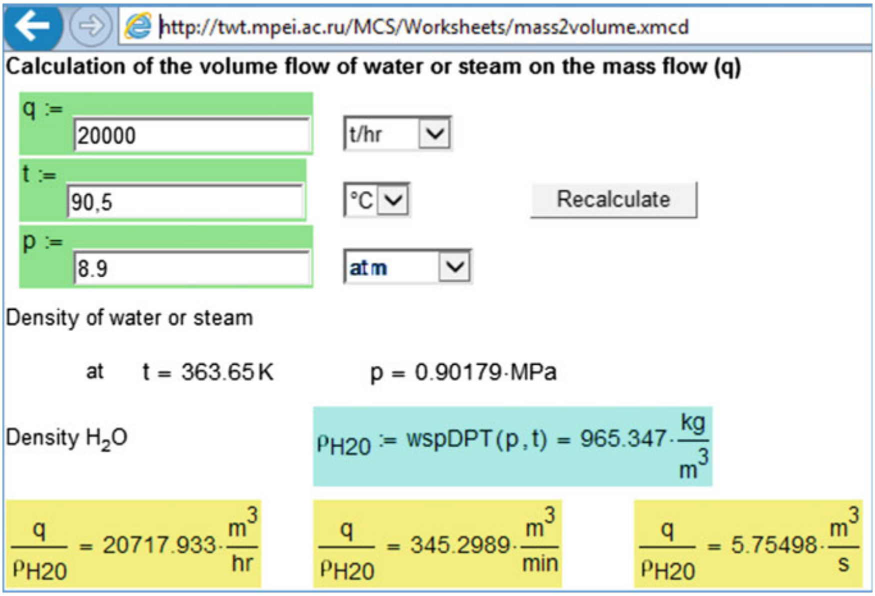


Fig. 5.7 Network calculation of water flow rate (a task which is inverse to the ones shown in Figs. 5.4, 5.5 and 5.6)

And one more afterpiece. The name of the previous chapter in the book was “My first thermo-technical Mathcad-calculation”. A book extract given further on can be called as “My first thermo-technical calculation at Elsevier/Knovel web-site”.

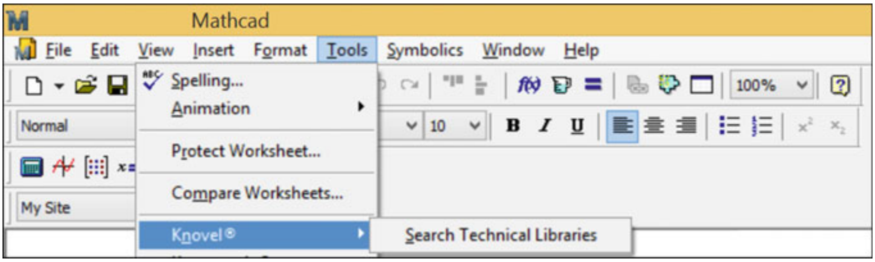


Fig. 5.8 Link to www.knovel.com from Mathcad 15

Elsevier (www.elsevier.com) is the largest e-publishing house which publishes magazines and books for many fields of expertise in its web-site.[3] Knovel (www.knovel.com)—is an e-publishing house too, but it is oriented at publication of technical literature with a possibility of interactive handling of diagrams, tables and formulas. Knovel allows its web-site users not only to read technical guides but also to download calculations made, e.g., by Mathcad. There is a direct link to www.knovel.com from Mathcad 15 (Fig. 5.8).

Mathcad users can search for a ready-made solution at Knovel web-site without leaving Mathcad before they solve a new task.

The author of this book collaborates with Knovel and also handed over some of his calculations related also to heat engineering to be posted at www.knovel.com (Fig. 5.9).

In 2012 Knovel became a part of Elsevier. Figure 5.10 gives a reference to www.knovel.com in the list of Elsevier's online tools.

In-house online-tools with so called Interactive Equations (Fig. 5.11) are included into Knovel's web-site.

Interactive Equations of Knovel allow not only looking over those that formulas but interactively work with them almost the same way as in Mathcad Calculation Server, which is supported by the author and which is much talked about in this book. In December 2015 the Interactive Equations of Knovel numbered 1170 formulas: 51 Civil Engineering & Construction Materials, 30 Geotechnical Engineering, 39269 in chemistry and chemical technologies, 51 Civil Engineering & Construction Materials, 30 Geotechnical Engineering, 1 Earth Anchors, 17 Retaining Walls, 12 Soil Mechanics, 9 Lateral Earth Pressure, 3 Surcharge Loads, 21 Structural Engineering, 12 Deep Foundations, 4 Micropile Foundation Analysis, 3 Pile Foundation Analysis, 5 Pole Foundation Analysis, 9 Shallow Foundations, 1 Mat Foundation Analysis, 8 Spread Footing Analysis, 50 Electrical & Power Engineering (one creator Valery Ochkov), 69 Electronics & Semiconductors, 100 General

[3]A tool of Web of Science scientists publishing activity evaluation is one of developments of Elsevier.

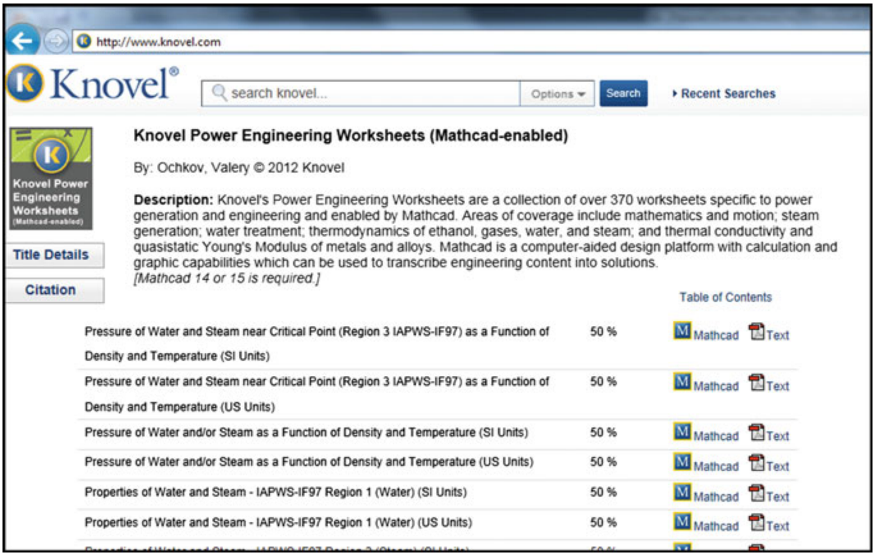


Fig. 5.9 Authoring Mathcad-calculations at Elsevier/Knovel web-sites

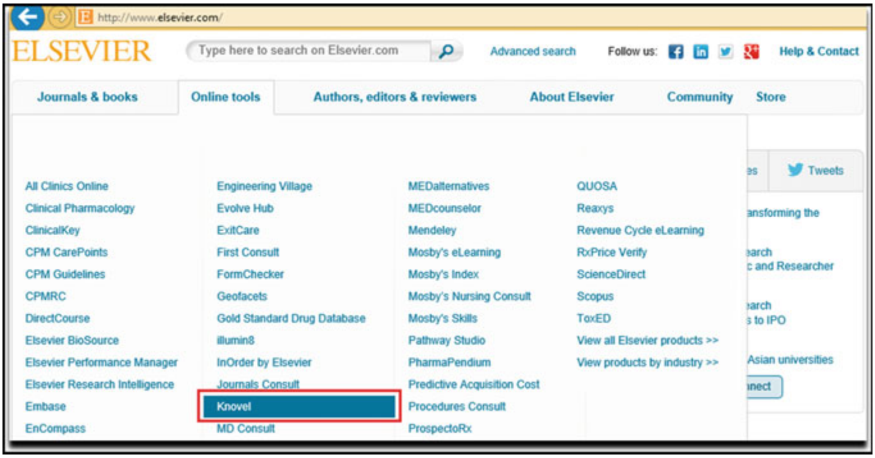


Fig. 5.10 Home page of e-publishing house Elsevier with a list of online tools

Engineering & Project Administration, 100 Mathematical Equations, 290 Mechanics & Mechanical Engineering, 20 Metals & Metallurgy, and 77 Oil & Gas Engineering.

The author of the book has great plans on collaboration with Knovel in terms of filling the library with thermodynamic and other calculations. Publication of pump power calculation as a function of water mass flow rate and pressure difference in

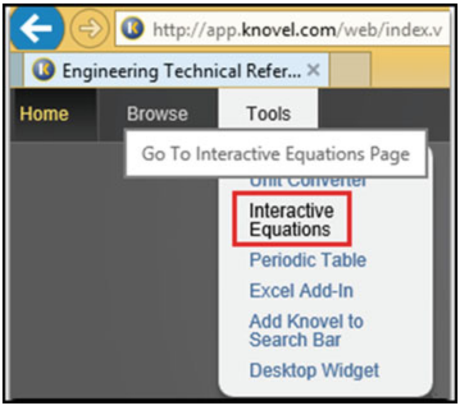


Fig. 5.11 Interactive tools from Knovel

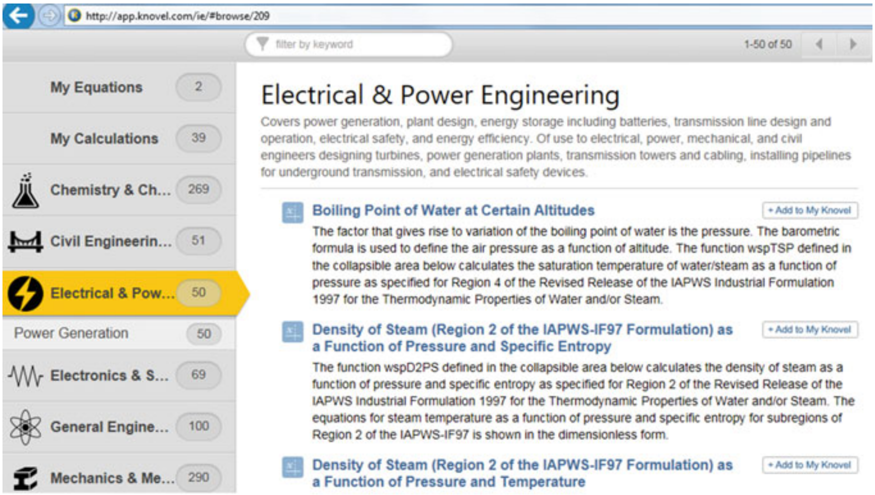


Fig. 5.12 Home page of Knovel's Interactive Equations

the pump (Figs. 5.13, 5.14, 5.15 and 5.16) relying on the function converting water density as a function of pressure and temperature, made the start to that work.

The calculation given in Fig. 5.14 was made in SMath which can be freely downloaded from www.smath.info. And that's the main point. Knovel did not place a stake on Mathcad due to the fact that it is a pretty expensive commercial product of another company—PTC.

In interactive SMath-calculation shown in Fig. 5.14 there is a hidden area with a function which converts water density as a function of pressure and temperature. This area is concealed and is partially shown in Fig. 5.15. This function and lots of

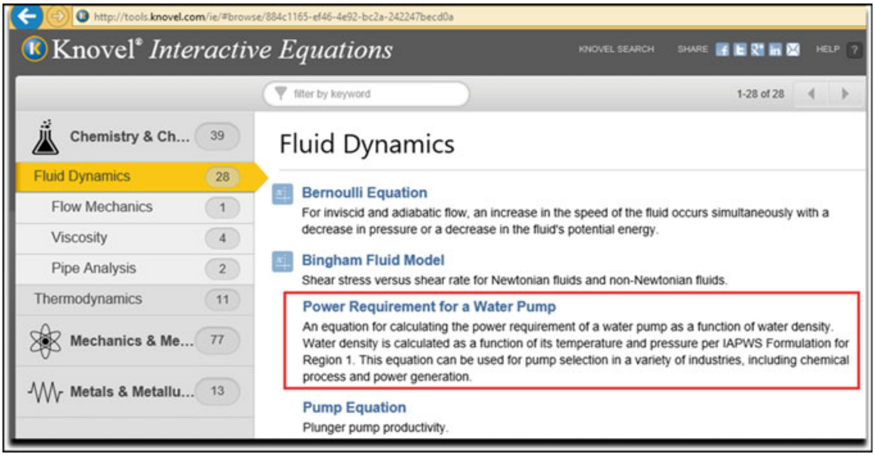


Fig. 5.13 Reference to pump power calculation in Interactive Equations of Knovel

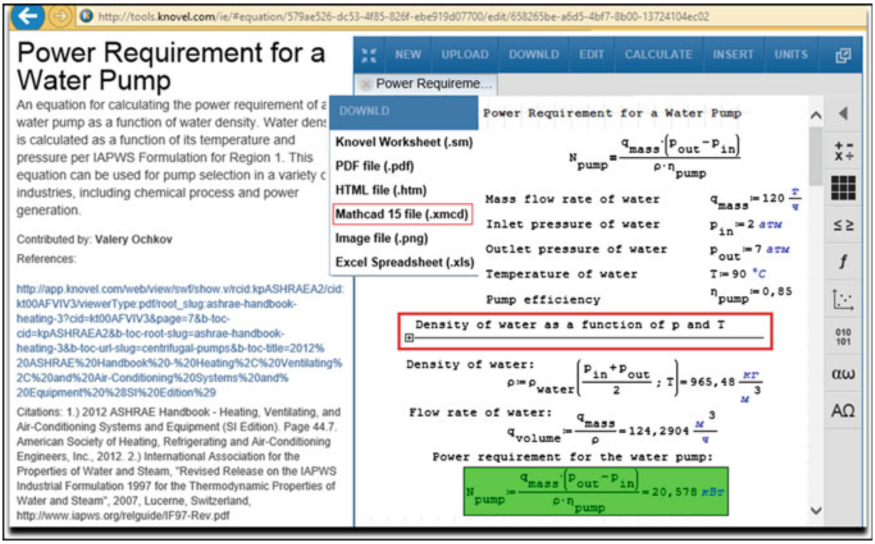


Fig. 5.14 Interactive calculation of pump power at Knovel’s web-site

other similar functions can be downloaded from the authoring web-site http://twt.mpei.ac.ru/rbtpp/Region. It has the name **wspD1PT** there. If 1 is to replace with 2 in the name of this function, it will convert the density of water steam rather than water and the properties of water steam will be defined by the second area of formulation IAPWS-IF97 rather than the first one. If there is 3 in the functions name, it will refer to near-critical region. If there is 5 there, it will refer to high temperature water steam region. The fourth region of formulation IAPWS-IF97

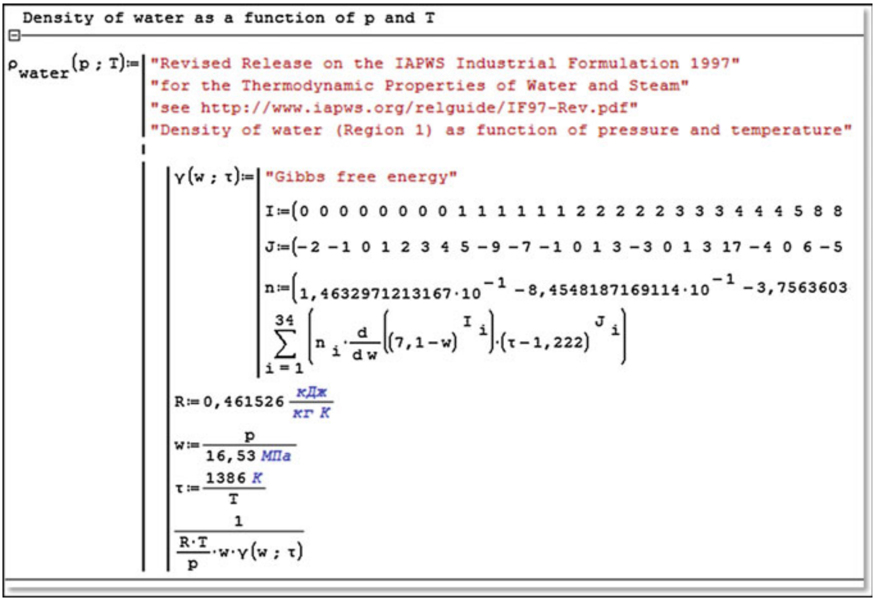
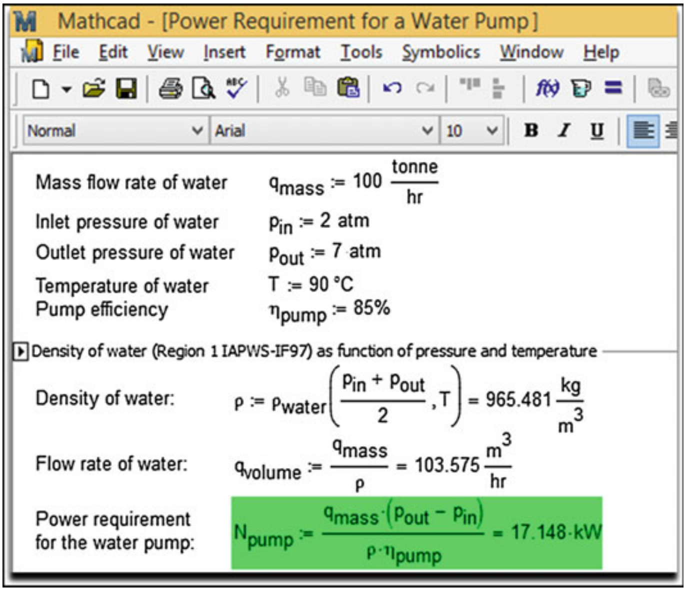


Fig. 5.15 Function of Knovel's Interactive Equations converting water density (a part of the function is shown)

defines water and water steam properties at the saturation line between water triple point to critical point. Mathcad-file with H2O name and extension of mcd, mcdx or mcdxz being stored in the Internet at http://twt.mpei.ru and widely used by us in this book (see example, Fig. 5.5) includes functions with names **wspD1PT**, **wspD2PT**, **wspD3PT** and **wspD5PT**. Besides, there is a function there with name **wspDPT**, which defines the formulation region IAPWS-IF97 with parameters (pressure and temperature) and calls the required numbered function.

One can also see a list of menu commands **DOWNLD** in Fig. 5.14 allowing downloading files of calculating software SMath (Knovel Worksheet), Mathcad and MS Excel. Figure 5.16 shows a respective Mathcad-calculation of water pump power.

Its efficiency appears in the formula of pump power calculation. It's easy to prove without any measurements that this value is always less one (one hundred per cent). It remains only to come closer to any running pump (compressor, turbine, etc.) to hear its noise and to feel its vibration. A part of pump drive energy is used for that noise and vibration. A perfect pump must run silently and without the least vibration. Noise is transmitted by air, metal and water after all. Noise level is measured in decibel which we mentioned in Chap. 2 (see Figs. 2.10 and 2.11). Noise problem at power plants and other industrial facilities is a big problem. The author's server includes a section with examples of calculations related to noise control at power plants (see http://twt.mpei.ac.ru/TTHB/1/Tupov.html) [44].



$$q_{mass} := 100\,\frac{\text{tonne}}{\text{hr}}$$

$$p_{in} := 2\ \text{atm}$$

$$p_{out} := 7\ \text{atm}$$

$$T := 90\ {}^{\circ}\text{C}$$

$$\eta_{pump} := 85\%$$

$$\rho := \rho_{water}\left(\frac{p_{in} + p_{out}}{2}, T\right) = 965.481\,\frac{\text{kg}}{\text{m}^3}$$

$$q_{volume} := \frac{q_{mass}}{\rho} = 103.575\,\frac{\text{m}^3}{\text{hr}}$$

$$N_{pump} := \frac{q_{mass}\cdot(p_{out} - p_{in})}{\rho\cdot\eta_{pump}} = 17.148\cdot\text{kW}$$

Fig. 5.16 Mathcad-calculation of pump power downloaded from Knovel's Interactive Equations

By the way, about acoustic speed in water. Water and water steam properties are split to thermodynamic (enthalpy, entropy, internal power, etc.) and physical (viscosity, thermal conductivity, etc.). The term "thermal and physical properties of water" unites these parameters. So, acoustic speed in water is neither physical nor transport parameter as one might think at first sight, but a thermodynamic parameter. One can say the same about water density.

There are future plans to place functions of working media, heat carriers and materials of heat, nuclear and industrial power engineering in Knovel's Interactive Equations. In particular, the data base of IVTANTERMO on thermal and physical properties of individual substances [45]. As it is, e-version of this guide with online calculations and a downloading option of Mathcad 15-, Mathcad Prime-, Mathcad Express-, SMath- and Excel-files can be found at the authoring web-site http://twt.mpei.ac.ru/TTHB/2/OIVT/IVTANThermo/Rus/index.htm (Fig. 5.17).

Properties of individual substances will be exemplified with their use similar to what have been done to water density during calculation of pump power.

Technology of inverse calculations will be used when, e.g., water density and its temperature are known and pressure is to be found [46].

Figure 5.18 shows a page of a web-site with calculation of heat capacity of gaseous ozone and image of a working point at the diagram. Scaled Gibbs energy, entropy and entropy difference for that substance are also calculated at the same web-site. Formulas used in calculation can both be reviewed (that can be done by referring to paper guide-book, of course), copied and pasted to your own calculation. The only thing to do is to download MS Excel table (see link to it in Fig. 5.17)

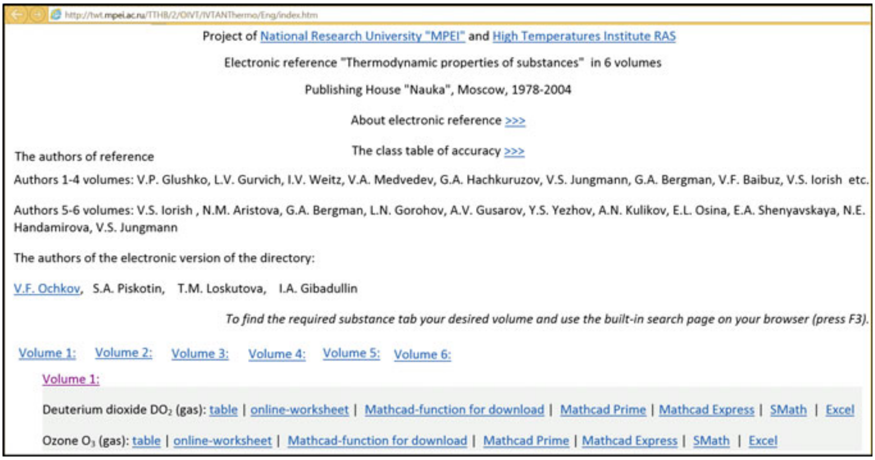


Fig. 5.17 E-guide on thermal and physical properties of individual substances

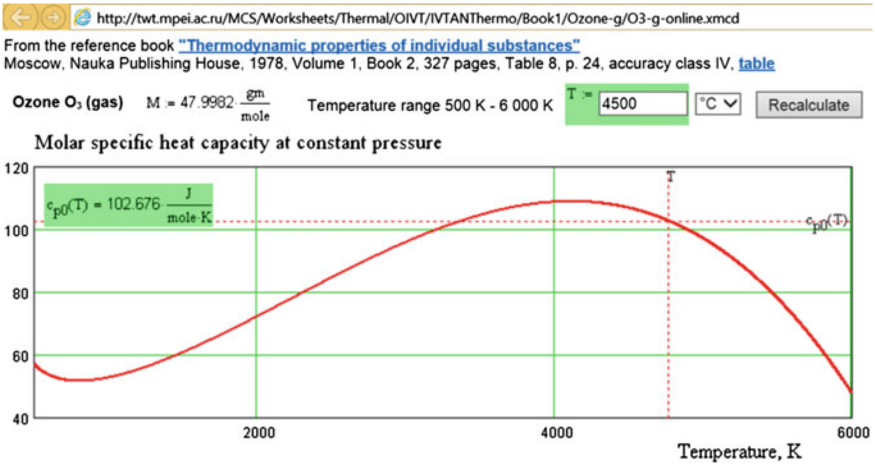


Fig. 5.18 Interactive network calculation of specific heat capacity of gaseous ozone

with that formula in text format ready to carry over to programs created by programming languages FORTRAN, C, Pascal, BASIC, etc. This is a specific example of “open interactive network calculation” principle realization which is described in this book.

Properties of individual substances at Elsevier/Knovel web-site (see Figs. 5.10, 5.11, 5.12, 5.13, 5.14, 5.15 and 5.16) shall be shown with reference to their use the same way as it was done to water density in calculation of the pump power (see Fig. 5.16). Technology of inverse calculations will be used when, e.g., water density and its temperature are known and pressure is to be found [46].

Chapter 6
Symbolic Mathematics and 3D-Graphics in Thermal Engineering

Valery Ochkov

Abstract A typical optimization problem with the objective function, optimization parameters and constraints solved in this chapter. The search for optimal values of the pressure of the steam extraction from the turbine for feedwater heating to improve the thermal efficiency of the Rankine cycle is used. Symbolic mathematics and graphical tools to solve this problem are described.

In Chap. 4 we calculated the thermal efficiency of the simplest ideal steam turbine cycle. Chapter 5 demonstrates how to calculate the "non-ideality" (irreversibility) steam expansion process in the turbine and in raising water pressure in the feed water pump. There we saw at the same time how these processes can be displayed in 2D graphics. Now we complicate the steam turbine plant by adding two steam extraction points for regeneration to increase the thermal efficiency of the power plant.

Figure 6.1 shows a Mathcad calculation of the thermal efficiency of a steam turbine cycle with two mixing regenerative heaters H-1 and H-2.

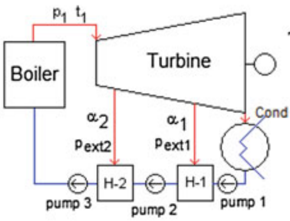
After the reference to the cloud Mathcad-sheet with the name H2O and entry of input data (see the cycle diagram and its parameters), we introduce into the calculation a system of two equations of material enthalpy balance in heaters. By means of Mathcad's symbolic mathematics, this is solved for the unknowns α_1 (which is the fraction of steam extracted from the turbine to the first heater) and α_2 (the fraction of steam extracted from the turbine to the second heater). After this analytical solution by the help of the **solve** operator it is easy to calculate the cycle's

The site of the chapter: https://www.ptcusercommunity.com/message/423020.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_6

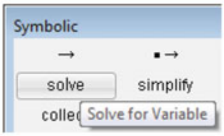


→ Reference:http://twt.mpei.ru/tthb/H2O.xmcdz

The parameters of steam turbine unit with two steam extraction

$p_1 := 13\text{MPa}$ $\quad p_{2ext} := 7\text{MPa}$ $\quad p_{1ext} := 6\text{MPa}$ $\quad p_{cond} := 4\,\text{kPa}$

$t_1 := 540\,°\text{C}$ $\quad \eta_{i\,t} := 85\,\%$ $\quad \eta_{i\,p} := 75\,\%$



Material and heat balance of steam extraction (solution of the system of two equations)

$$\begin{bmatrix} (1-\alpha_1-\alpha_2)\cdot h_{p1out}+\alpha_1\cdot h_{1ext}=(1-\alpha_2)\cdot h_{p2in} \\ (1-\alpha_2)\cdot h_{p2out}+\alpha_2\cdot h_{2ext}=h_{p3in} \end{bmatrix} \text{solve}, \begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} \rightarrow \cdots$$

$$\cdots \rightarrow \left(-\frac{h_{p2in}h_{p3in}-h_{p2in}h_{2ext}-h_{p3in}h_{p1out}+h_{2ext}h_{p1out}}{h_{1ext}h_{2ext}-h_{1ext}h_{p2out}-h_{2ext}h_{p1out}+h_{p1out}h_{p2out}} \quad \frac{h_{p3in}-h_{p2out}}{h_{2ext}-h_{p2out}} \right)$$

$h_1 := \text{wspHPT}(p_1, t_1) = 3445.05\,\text{kJ/kg}$ $\qquad s_1 := \text{wspSPT}(p_1, t_1) = 6.577\,\text{kJ/(kg K)}$

$t_{p1in} := \text{wspTSP}(p_{cond}) = 28.96\,°\text{C}$

$h_{cond} := \text{wspHEXPANSIONPTPEFF}(p_1, t_1, p_{cond}, \eta_{i\,t}) = 2200.29\,\text{kJ/kg}$

$h_{p1in} := \text{wspHSWT}(t_{p1in}) = 121.4\,\text{kJ/kg}$ $\qquad s_{p1in} := \text{wspSSWT}(t_{p1in}) = 0.422\,\text{kJ/(kg K)}$

$$h_{p1out} := \text{wspHEXPANSIONPTXPEFF}\left(p_{cond}, t_{p1in}, 0, p_{1ext}, \frac{1}{\eta_{i\,p}}\right) = 129.41\,\text{kJ/kg}$$

$t_{p2in} := \text{wspTSP}(p_{1ext}) = 275.59\,°\text{C}$ $\qquad h_{p2in} := \text{wspHSWT}(t_{p2in}) = 1213.73\,\text{kJ/kg}$

$$h_{p2out} := \text{wspHEXPANSIONPTXPEFF}\left(p_{1ext}, t_{p2in}, 0, p_{2ext}, \frac{1}{\eta_{i\,p}}\right) = 1215.49\,\text{kJ/kg}$$

$t_{p3in} := \text{wspTSP}(p_{2ext}) = 285.83\,°\text{C}$ $\qquad h_{p3in} := \text{wspHSWT}(t_{p3in}) = 1267.44\,\text{kJ/kg}$

$$h_{p3out} := \text{wspHEXPANSIONPTXPEFF}\left(p_{2ext}, t_{p3in}, 0, p_1, \frac{1}{\eta_{i\,p}}\right) = 1278.2\,\text{kJ/kg}$$

$h_{1ext} := \text{wspHEXPANSIONPTPEFF}(p_1, t_1, p_{1ext}, \eta_{i\,t}) = 3237.49\,\text{kJ/kg}$

$h_{2ext} := \text{wspHEXPANSIONPTPEFF}(p_1, t_1, p_{2ext}, \eta_{i\,t}) = 3276.12\,\text{kJ/kg}$

$$\alpha_1 := \frac{h_{2ext}\cdot h_{p2in}-h_{2ext}\cdot h_{p1out}-h_{p2in}\cdot h_{p3in}+h_{p3in}\cdot h_{p1out}}{h_{1ext}\cdot h_{2ext}-h_{1ext}\cdot h_{p2out}-h_{2ext}\cdot h_{p1out}+h_{p1out}\cdot h_{p2out}} = 34.01\,\%$$

$$\alpha_2 := \frac{h_{p3in}-h_{p2out}}{h_{2ext}-h_{p2out}} = 2.52\,\%$$

$$l_t := (h_1-h_{2ext})+(1-\alpha_2)\cdot(h_{2ext}-h_{1ext})+(1-\alpha_2-\alpha_1)\cdot(h_{1ext}-h_{cond}) = 864.92\,\text{kJ/kg}$$

$$l_p := (1-\alpha_2-\alpha_1)\cdot(h_{p1out}-h_{p1in})+(1-\alpha_2)\cdot(h_{p2out}-h_{p2in})+(h_{p3out}-h_{p3in}) = 17.558\,\text{kJ/kg}$$

$q_1 := h_1 - h_{p3out} = 2166.85\,\text{kJ/kg}$ $\qquad \eta_T := \dfrac{l_t - l_p}{q_1} = 39.11\,\%$

Fig. 6.1 Calculation of cycle thermal efficiency of steam turbine with two steam extractions

$\eta_t(p_1, t_1, p_{cond}, p_{1ext}, p_{2ext}, \eta_{i\,t}, \eta_{i\,p}) :=$ "Calculation of the efficiency of steam turbine with two heaters"

$$h_1 \leftarrow wspHPT(p_1, t_1)$$
$$s_1 \leftarrow wspSPT(p_1, t_1)$$
$$t_{p1in} \leftarrow wspTSP(p_{cond})$$
$$h_{cond} \leftarrow wspHEXPANSIONPTPEFF(p_1, t_1, p_{cond}, \eta_{i\,t})$$
$$h_{p1in} \leftarrow wspHSWT(t_{p1in})$$
$$s_{p1in} \leftarrow wspSSWT(t_{p1in})$$
$$h_{p1out} \leftarrow wspHEXPANSIONPTXPEFF\left(p_{cond}, t_{p1in}, 0, p_{1ext}, \frac{1}{\eta_{i\,p}}\right)$$
$$t_{p2in} \leftarrow wspTSP(p_{1ext})$$
$$h_{p2in} \leftarrow wspHSWT(t_{p2in})$$
$$h_{p2out} \leftarrow wspHEXPANSIONPTXPEFF\left(p_{1ext}, t_{p2in}, 0, p_{2ext}, \frac{1}{\eta_{i\,p}}\right)$$
$$t_{p3in} \leftarrow wspTSP(p_{2ext})$$
$$h_{p3in} \leftarrow wspHSWT(t_{p3in})$$
$$h_{p3out} \leftarrow wspHEXPANSIONPTXPEFF\left(p_{2ext}, t_{p3in}, 0, p_1, \frac{1}{\eta_{i\,p}}\right)$$
$$h_{1ext} \leftarrow wspHEXPANSIONPTPEFF(p_1, t_1, p_{1ext}, \eta_{i\,t})$$
$$h_{2ext} \leftarrow wspHEXPANSIONPTPEFF(p_1, t_1, p_{2ext}, \eta_{i\,t})$$
$$\alpha_1 \leftarrow \frac{h_{2ext} \cdot h_{p2in} - h_{2ext} \cdot h_{p1out} - h_{p2in} \cdot h_{p3in} + h_{p3in} \cdot h_{p1out}}{h_{1ext} \cdot h_{2ext} - h_{1ext} \cdot h_{p2out} - h_{2ext} \cdot h_{p1out} + h_{p1out} \cdot h_{p2out}}$$
$$\alpha_2 \leftarrow \frac{h_{p3in} - h_{p2out}}{h_{2ext} - h_{p2out}}$$
$$l_t \leftarrow (h_1 - h_{2ext}) + (1 - \alpha_2) \cdot (h_{2ext} - h_{1ext}) + (1 - \alpha_2 - \alpha_1) \cdot (h_{1ext} - h_{cond})$$
$$l_p \leftarrow (1 - \alpha_2 - \alpha_1) \cdot (h_{p1out} - h_{p1in}) + (1 - \alpha_2) \cdot (h_{p2out} - h_{p2in}) + (h_{p3out} - h_{p3in})$$
$$q_1 \leftarrow h_1 - h_{p3out}$$
$$\text{return } \frac{l_t - l_p}{q_1}$$

Call the function (see the Fig above too) $\eta_t(13MPa, 540\,°C, 4kPa, 6MPa, 7MPa, 85\%, 75\%) = 39.11\,\%$

Fig. 6.2 Function of cycle thermal efficiency of steam turbine with two steam extractions

thermal efficiency (see Fig. 6.1) by determining the parameters of steam at its key points and performing simple balance calculations.

You can manually change the turbine extraction pressure ($\mathbf{p}_{1ext}$ and $\mathbf{p}_{2ext}$) and observe how the thermal efficiency of the cycle is changing; although you could entrust the work to a computer. This requires you to create a user function that returns the thermal cycle efficiency, depending on its seven initial parameters listed in the upper part of Fig. 6.1. This function is shown in Fig. 6.2.

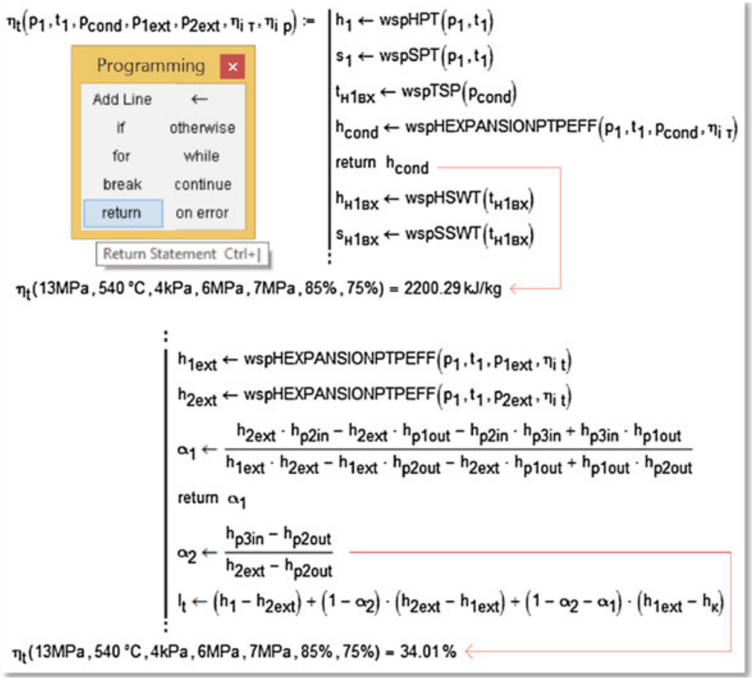


Fig. 6.3 Debugging the function, shown on the Fig. 6.2

Calculations with intermediate calculation results shown in Fig. 6.1 can be regarded as a debug phase in the creation of the user function shown in Fig. 6.2 as we did in Chap. 4 as well.

Writing a Mathcad-worksheet with intermediate calculation results, as shown in Fig. 6.1, can be regarded as a debug phase in the creation of the user function shown in Fig. 6.2. By the way, we did this in Chap. 4 as well. The function created can be debugged in another way: by inserting an operator **return** after the next calculation of the operator, with an operand storing the calculated value, and calling the unfinished function—see Fig. 6.3.

Mathcad has tools to automatically change the function arguments, allowing it to take the maximum or minimum value near the point of the first approximation to the solution (guess values). Figure 6.4 shows how the built-in Mathcad function named **Maximize** solved our optimization problem of finding the turbine extraction pressure values for which the cycle thermal efficiency will be maximal.

The collapsed area in Fig. 6.4 contains a function shown in Fig. 6.5.

Figure 6.5 shows how the solution of our optimization problem is checked using the 3D-surface and the same level lines (a counter plot).

The solution of such an optimization problem for the cycle of a steam turbine plant with one steam extraction from its turbine is typically illustrated with a graph of the cycle's thermal efficiency and feed water temperature.

Reference:http://twt.mpei.ru/tthb/H2O.xmcdz

$p_1 := 13\,\text{MPa} \quad t_1 := 540\,°\text{C} \quad p_{cond} := 4\,\text{kPa} \quad \eta_{i\,t} := 85\,\% \quad \eta_{i\,p} := 75\,\%$

$p_{1ext} := 6\,\text{MPa} \quad p_{2ext} := 7\,\text{MPa}$

$\eta_t(p_1, t_1, p_{cond}, p_{1ext}, p_{2ext}, \eta_{i\,t}, \eta_{i\,p}) = 39.11\,\%$

$\eta(p_{1ext}, p_{2ext}) := \eta_t(p_1, t_1, p_{cond}, p_{1ext}, p_{2ext}, \eta_{i\,t}, \eta_{i\,p})$

$$\begin{pmatrix} p_{1opt} \\ p_{2opt} \end{pmatrix} := \text{Maximize}(\eta, p_{1ext}, p_{2ext}) = \begin{pmatrix} 0.2348 \\ 2.787 \end{pmatrix} \text{MPa}$$

$\eta_{tMax} := \eta(p_{1opt}, p_{2opt}) = 41.37\,\%$

Fig. 6.4 Optimization of the steam turbine cycle with two extractions points

Figure 6.6 shows a complementary function, created using the function shown in Fig. 6.2. With these two functions: the main function—Fig. 6.2 and complementary function—Fig. 6.6, we can construct the desired graph and with a tracing tool mark on it the point of maximum thermal efficiency.

The y-axis of the curve in Fig. 6.5 records the output of the function for two steam extraction points from the turbine. However with $\mathbf{p}_{1\text{ext}} = \mathbf{p}_{2\text{ext}}$ this function is also suitable for the case of one steam extraction point. Indeed the curve itself is parametric: an independent variable in calculations and optimization will be based on the pressure extraction $\mathbf{p}_{\text{ext}}$, which changes depending on the pressure in the condenser p_{cond} and the live steam pressure $\mathbf{p}_1$, but the curve is based on two functions of $\mathbf{p}_{\text{ext}}$.

From the calculations shown in this chapter, it is clear that the optimized single steam extraction from turbine to regeneration increases the power plant thermal efficiency from 39.11 % (see Fig. 6.1) up to 40.125 % (see Fig. 6.6). The transition to the two regeneration steam extractions with pressure optimization improves thermal efficiency up to 41.372 % (see Fig. 6.4).

On the site of the book you may find calculations for steam turbine units with three or four regeneration extractions performed by the method described above. Figures 6.6 and 6.7 show the operators of the symbolic solution of fluid and heat balance equations of three and four mixing heaters of a steam turbine unit, whose diagram is shown in Fig. 6.1. This shows only part of the solution since it is too long and written in the form of a matrix with one row and three (Fig. 6.7) or four (Fig. 6.8) elements.

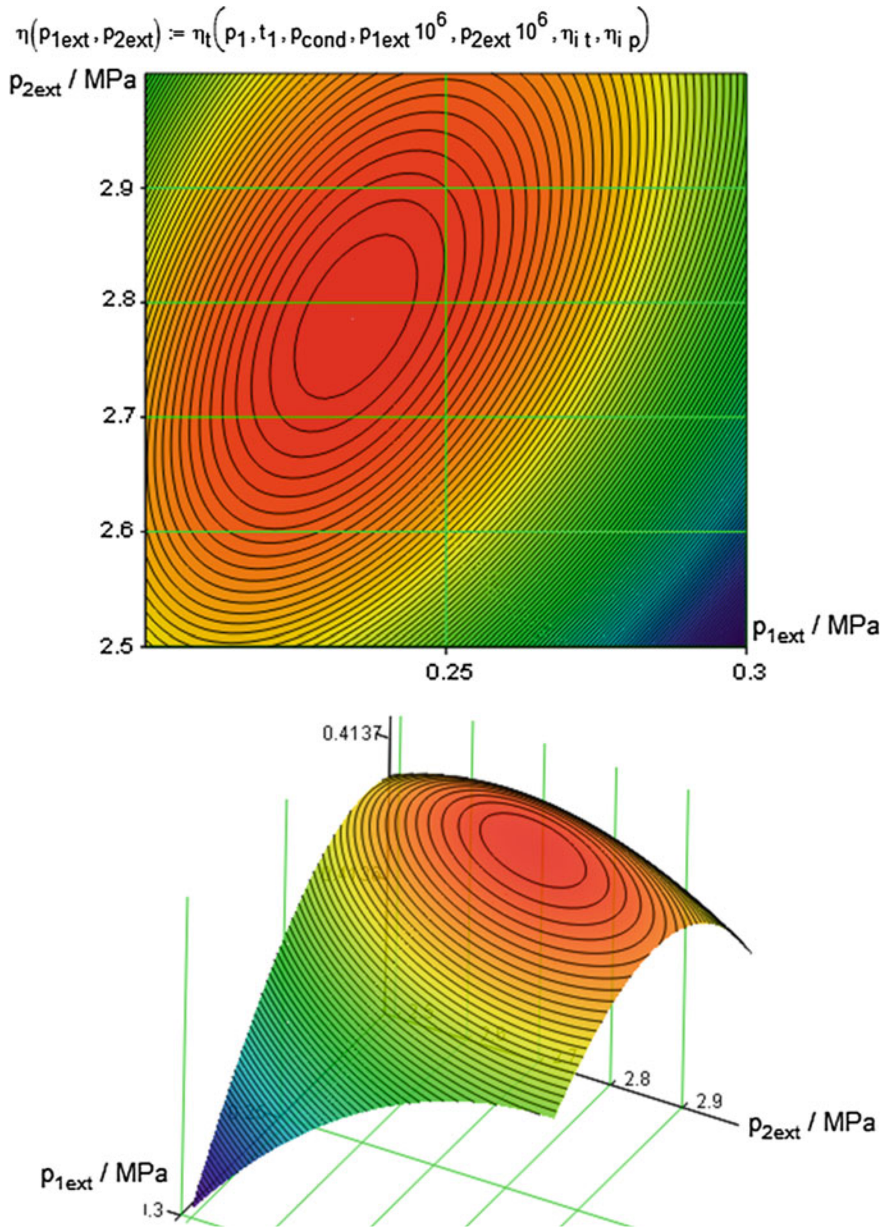


Fig. 6.5 Graphic illustration of the optimization problem

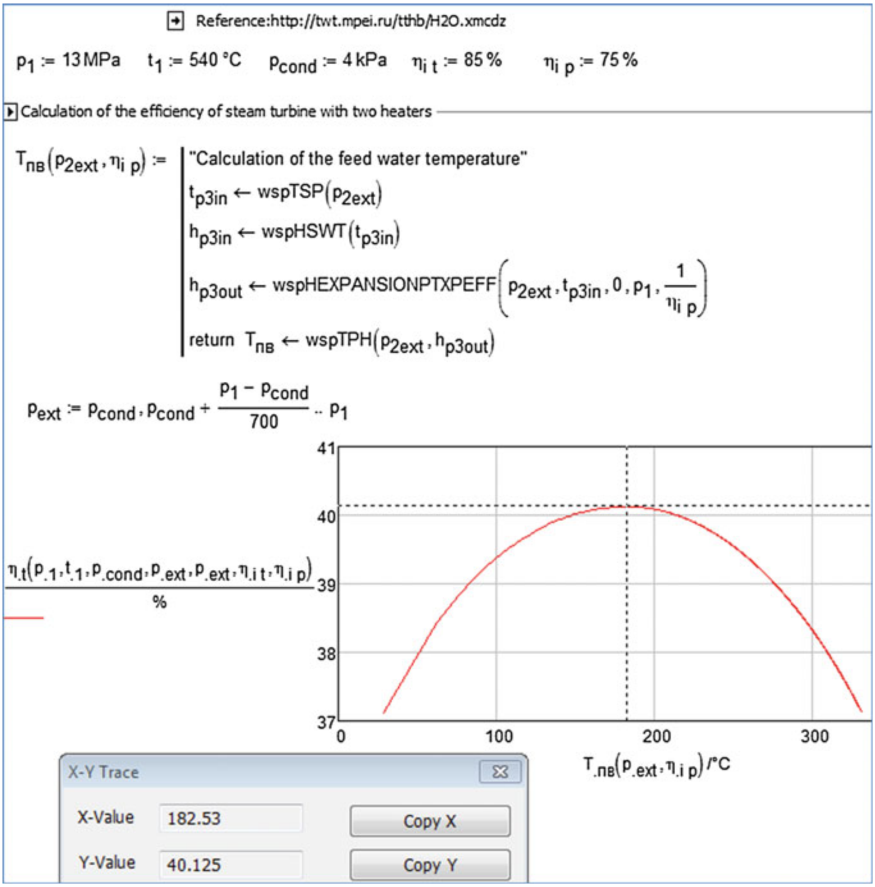


Fig. 6.6 Optimization of steam turbine cycle with one steam extraction point

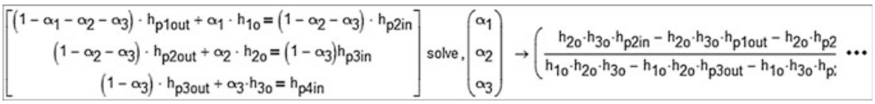


Fig. 6.7 Solution of the equations of material and heat balances of three mixing heaters of steam turbine unit

$$\begin{bmatrix} (1-\alpha_1-\alpha_2-\alpha_3-\alpha_4)\cdot h_{p1out}+\alpha_1\cdot h_{1ext}=(1-\alpha_2-\alpha_3-\alpha_4)\cdot h_{p2in} \\ (1-\alpha_2-\alpha_3-\alpha_4)\cdot h_{p2out}+\alpha_2\cdot h_{2ext}=(1-\alpha_3-\alpha_4)h_{p3in} \\ (1-\alpha_3-\alpha_4)\cdot h_{p3out}+\alpha_3\cdot h_{3ext}=(1-\alpha_4)h_{p4in} \\ (1-\alpha_4)\cdot h_{p4out}+\alpha_4\cdot h_{4ext}=h_{p5in} \end{bmatrix} \text{solve}, \begin{pmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\\ \alpha_4\end{pmatrix} \rightarrow \left(-\frac{h_{p2in}\cdot h_{p3in}\cdot h_{p4in}}{h_{1ext}\cdot h_{2ext}\cdot h_{3ext}\cdot h_{4ext}-h_{1ext}\cdot h_{;}} \cdots\right.$$

Fig. 6.8 Solution of the equations of material and heat balances of four mixing heaters of steam turbine unit

On the site http://twt.mpei.ac.ru/TTHB/2/tdc.html, you can find the calculation of a real steam turbine unit with eight steam extractions to regenerative heaters. On the site http://twt.mpei.ru/MCS/Worksheets/PTU/pnd.xmcd you can find the calculation of a low-pressure regenerative heater, and on the site http://twt.mpei.ru/MCS/Worksheets/PTU/pvd.xmcd—the calculation of a high-pressure regenerative heater, however not mixing one (as shown in Fig. 6.1), but a surface type.

Chapter 7
Thermal Engineers' Gift to Water Chemists

Valery Ochkov

Abstract This chapter continues the analysis of problems of optimization techniques using the example of staged evaporation of steam in the boiler drum to reduce the concentration of impurities in the steam from the boiler. The described techniques use mathematical packages and spreadsheets to solve the paradoxical problems of optimization.

In the previous Studies, when calculating steam turbine cycles, we did not consider impurities that may be contained in water and steam. Concentration of these impurities is very small and cannot significantly affect the thermal parameters of the working medium. But the impurities can significantly affect the performance of power units.

The initial natural water at thermal and nuclear power plants is treated (desalinated, etc.) in order to avoid build-up of deposits in the steam turbine cycle. Impurities may be transferred from boiler water to steam, and deposited in the turbine steam path, which decreases the turbine's efficiency and increases the load on the thrust bearing,[1] and the turbine is fraught with "under-scale" corrosion. The main sources of boiler water impurities are:

The site of the chapter: https://www.ptcusercommunity.com/message/423022.

[1]To the extent that bearings cannot withstand such above-the-limit loads and fall apart. In times of old (now we can already say this about the beginning of the era of "big" power engineering), an operator on patrol in a turbine hall of a power plant used a stethoscope to listen to the bearings and other "critical" parts of turbines, pumps and fans. Nowadays such "preventive medicine" control is automated.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_7

- make-up water, which is actually carefully treated, and it can bring impurities into the boiler only due to an oversight of the staff of the chemical department of the power plant who are involved in servicing the water treatment plants;
- leaks into a turbine condenser, which may bring a "dirty" water from a cooling tower, a cooling pond (service water supply recirculation system with spray ponds, which, for example, will be considered in Chap. 18) or from a river (flow systems) into a "clean" condensate, and then into the boiler and the steam turbine;
- corrosion processes in steam turbine circuits "enriching" water and steam with oxides of iron, copper or other metals included in constructional materials (alloys) of the mechanical equipment.

To reduce the accumulation of impurities in the boiler water, build-up of deposits in the boiler, and their entrainment into the steam turbine, one should take-up various measures: purify circuit water in a polishing plant, dose various antiscale additives (e.g. phosphates) into the boiler, etc. Here, heat engineers come to help to chemists conditioning make-up water, servicing a polishing plant and preparing reagents for dosing into a boiler—see the title of this chapter. What is the essence of this "gift" [47, 48]?

The impurities, entering with feed water in a *drum* boiler, are concentrated in the boiler and removed from it with a *blowdown*, which makes 1–3 % steam flow (the **Pr** variable). But a small part of the impurities goes into the steam, and this leads to the above complications. The ratio of impurities concentration in the steam to impurities concentration in the boiler water is conventionally called a *cumulative entrainment rate* which consists of two different processes—entrainment of impurities with drops of moisture into saturated steam and dissolution of impurities in the steam. But blowdown of water from a boiler cannot be smooth, the process should be *staged*, i.e. "thermal engineers' gift to bad[2] chemists" is required to be implemented.

As you know, a gift horse is not to be looked in the mouth, but we shall not only accept the "gift of the heat engineers" but also try to "look in its mouth" and optimize it. Figure 7.1 shows the website, by logging on to which you can not only simulate, but also optimize the process of three-stage evaporation of water in a drum boiler: Water from the first (pure) chamber is partially evaporated, and partially blown into the second ("dirty" salt) chamber, and from it in turn, again is partially evaporated and partially blown into the third chamber, which is normally designed as a remote cyclone separator.[3]

[2]We say "bad" not in the sense of no good, but in the sense that chemists in power engineering at some point can not, by reasons beyond their control, ensure an acceptable quality of a boiler feed water and the quality of steam leaving a boiler.

[3]The boiler drum is equipped with not only partitions separating the evaporation surface into chambers, but also with other boiler internals (steam washing screens, moisture separators, etc.), which are not shown in Fig. 7.1.

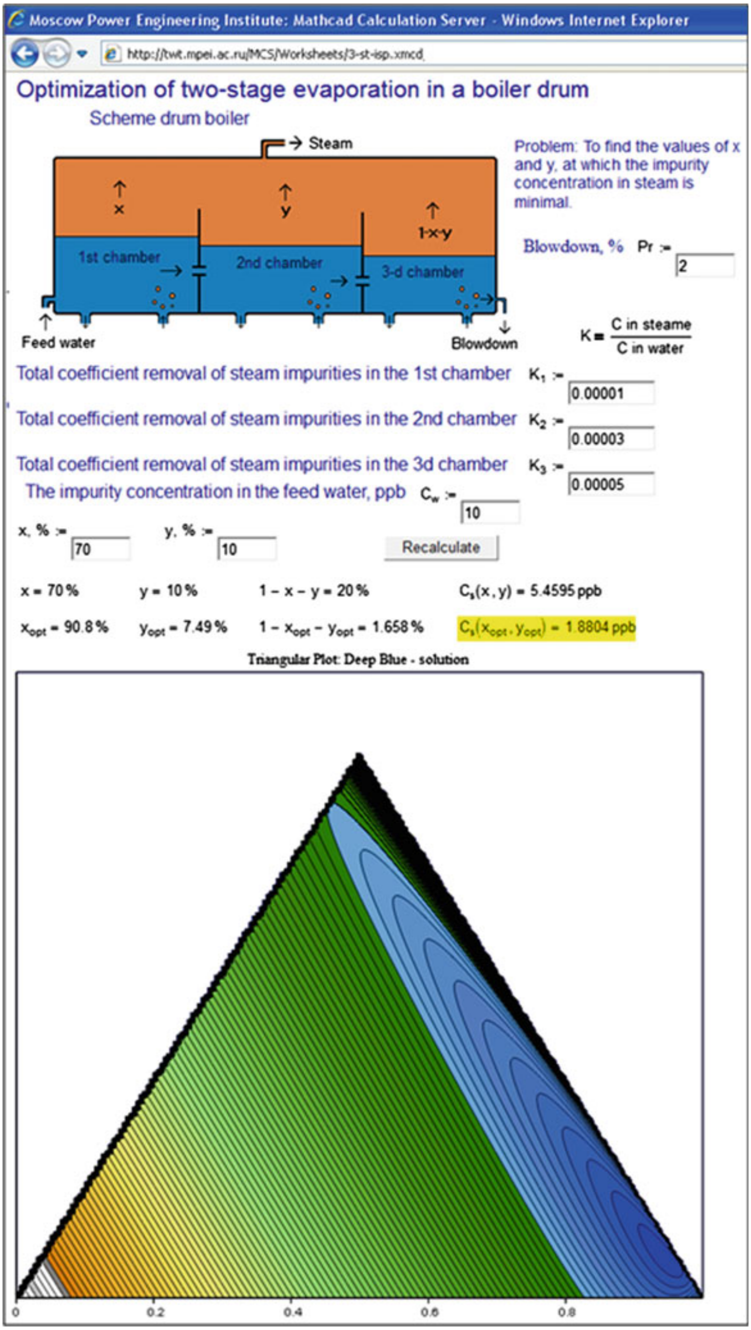


Fig. 7.1 Site for calculating staged evaporation in a drum boiler

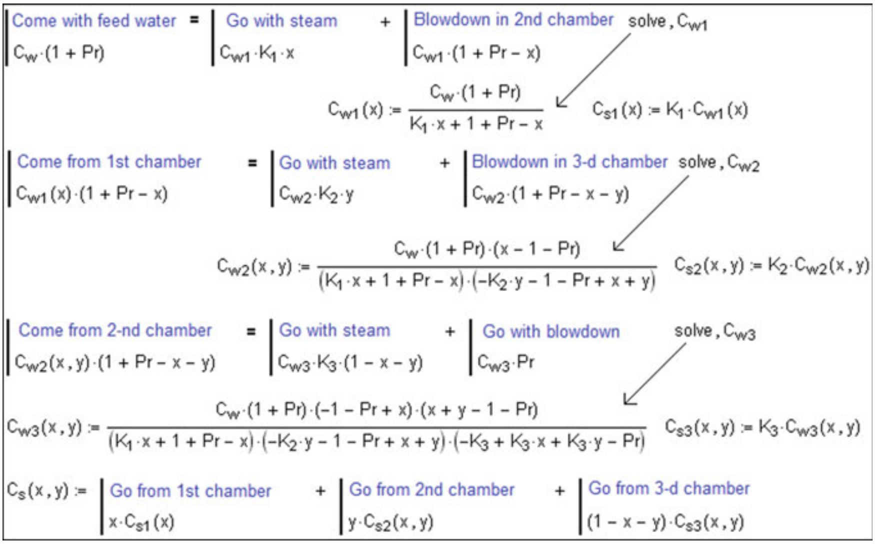


Fig. 7.2 Create user functions using Mathcad symbolic mathematics

On the site, shown in Fig. 7.1, you can adjust the input data, press the **Recalculate** button and get an answer—the concentration of impurities in the steam at the set proportions of steam generated in the first (pure) chamber (optimization variable **x**) and in the second ("dirty") chamber (**y**), and the optimum values with which the concentration of impurities in the steam will be minimal.

This is a classic optimization problem, which includes three elements:

- Objective function—concentration of impurities in the steam to be minimized; Fig. 7.2 shows a fragment of a Mathcad worksheet, where using the symbolic operator solve one could solve algebraic equations for balance of impurities in each of the three chambers and form the objective function of two arguments $\mathbf{C}_s$ with forming the intermediate functions **C** with subscripts **w1**, **w2**, **w3** (boiler water in the three chambers) and **s1**, **s2**, **s3** (steam generated in the three chambers);
- Optimization variables **x** and **y**—steam fractions generated in the first and second chambers of the boiler drum; Fig. 7.1 shows that a site visitor can change the values in the same-name text fields and see that the objective function $\mathbf{C}_s$ returns; with $\mathbf{x := 1}$ and $\mathbf{y = 0}$ we have a smooth evaporation; with $\mathbf{x + y = 1}$ we have a two-stage evaporation ($\mathbf{x < 1}$), and with $\mathbf{x + y < 1}$—a three-stage evaporation; a visitor can "play" with the **x** and **y** values and watch how $\mathbf{C}_s$ value varies approaching or moving away from its minimum value;
- Constraints; one of them we have already mentioned, this is $\mathbf{x + y < 1}$; the two other constraints are $\mathbf{x \geq 0}$ and $\mathbf{y \geq 0}$; these constraints lead to the fact that the area of visualization of the objective function has the form of an equilateral triangle (see Fig. 7.1). More precisely. This area has the form of an isosceles

The objective function

$$\eta(x, y) := \frac{C_s(1, 0) - C_s(x, y)}{C_s(1, 0)}$$

The accuracy of finding the minimum

$CTOL := 10^{-17}$ $TOL := 10^{-17}$

Constrains

Given $0 < x < 1$ $0 < y < 1$ $x + y < 1$

$$\begin{pmatrix} x_{opt} \\ y_{opt} \end{pmatrix} := \text{Maximize}(\eta, x, y) = \begin{pmatrix} 90.848 \\ 7.494 \end{pmatrix} \%$$

Fig. 7.3 Numerical search of the optimum

triangle (half a square), but we have transformed it into an equilateral triangle and displayed in the Cartesian coordinates. Animation of such a transformation can be seen here—https://www.ptcusercommunity.com/videos/1427.

An equilateral triangle is the basis of visualization of *three-component mixtures* (alloys): the surface above such a triangle indicates a parameter (density of the alloy, for example, or its melting point), and the sides of the triangle refer to the percentage of each of the three components. The angles of the triangle—one of the three pure metals, sides—a two-component alloy, and the "inside" of the triangle—a three-component alloy. Very often here also works the principle that two is a company and three is a crowd. Thus, for example, solder used for soldering is an alloy of lead and tin in an optimum proportion, having the minimum (again optimization) melting point. If one adds to the solder any third metal (e.g. cadmium or bismuth) it would only impair this key performance indicator, or, conversely, improve it, making the solder more high-melting. In this case, the third stage of evaporation produces a very little effect of reducing concentration of impurities in the steam.

The **x** and **y** values, set by visitors of the website, as shown in Fig. 7.1, serve as a first approximation in the numerical search for the optimal **x** and **y** values, minimizing the object function $\mathbf{C}_s$ (Fig. 7.3).[4]

Naturally, a real boiler operator cannot arbitrarily change in the wide range the proportions of steam generated in the boiler chambers. The optimum x and y[5] values, which strongly depend on the blowdown **Pr** value, are laid in the boiler design. However the boiler operator can slightly adjust x and y values. The point is that the boiler's downcomer and riser pipes in the first and second evaporation stages can be "serviced" by different burners. By changing the flow of fuel to these burners, you can change x and y values, thereby reducing the amount of impurities in the steam—"making a gift to the bad chemists" and not just "giving a horse" but "optimizing", "looking it in the mouth."

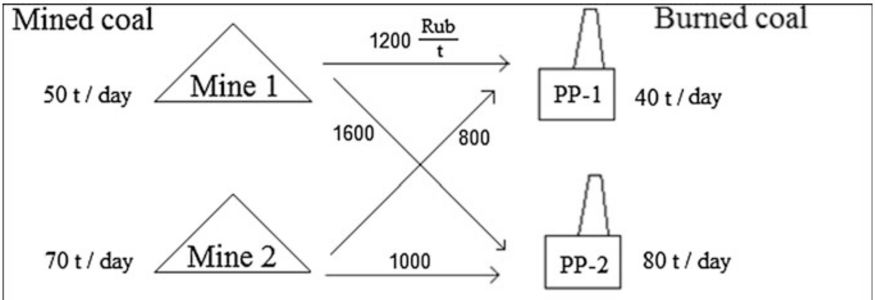
[4]Figure 7.3 shows a numerical (approximated) search of the maximum of function $\eta(x, y)$—the efficiency of the staged evaporation process.

[5]The fact, that among other things, makes this problem interesting is that it can be any number of optimization variables—from 1 to infinity. But we have to repeat that in practice they are limited to 2–3 evaporation stages. This problem is common with another problem considered in Chap. 6 here.

If we talk about water treatment for boilers, we can mention here that the author has developed and placed on the server some calculations relating to the process (see http://twt.mpei.ac.ru/ochkov/VPU_Book_New/mas/watertreatment.html) and even opened a special website with a short name www.vpu.ru (vpu means water treatment plant). On this site in particular, one can find calculations of the properties of Dow Chemical ion exchange resins (www.dow.com). Information about the properties of ion exchangers is supplied to consumers in two ways: on CD-ROM as a “closed” tool to calculate water treatment process diagrams (ROSA, IxCalc etc.), and as “open” technical documentation on paper or on the Internet sites. Advantages and disadvantages of these methods for publishing information have been already discussed here. The first method (“closed” tools on disks) can be hardly called a publication. After starting the program, located on the disk or downloaded from the Internet, the user is usually only given the opportunity to enter the input data and get the final answer: no intermediate results, and much less the formula used to obtain the answers, are given. Such “closed” information technology is inconvenient for the manufacturer of ion exchangers: if the developer of the program will leave the company, then it becomes very difficult, and sometimes impossible to make changes and additions in the program (“nightmare” of legacy software, which are described in the introduction). It is worth also mentioning the fact that these programs are very difficult to master. Even an experienced expert, well-versed in computers and water treatment, has to spend a lot of effort and time (several months) to learn to work with programs such as ROSA, IxCalc etc. After such “mastering”, an expert is often “hooked” on the program and can not switch to other more up-to-date and flexible design tools. The weak point of the second method for publishing technical information is that it is “dead”—for it is impossible to calculate at once, after entering the required input data.

Moscow Power Engineering Institute (www.mpei.ru) and Ltd Triery (www.trie.ru) developed a process of publishing technical information with the possibility of making immediate calculations using such information. A visitor to the author’s websites (http://twt.mpei.ac.ru/TTHB/1/Dow/index.html and http://twt.mpei.ac.ru/TTHB/1/RH/index.html) is given an opportunity to automatically calculate using such important process parameters of ion exchange resins as the value of slip of a removed ion and the working exchange capacity [49].

Small Divertissement after Studies 6 and 7. One of the author’s interests, directly related to his main job, is collection and animation of numerical methods for solving mathematical problems. This “collection” concerning optimization is posted on the website https://www.ptcusercommunity.com/groups/optimisation-with-mathcad. A special place in the “collection” is given to the examples which can be described as “obvious and incredible.” One such example, adapted to the subject of the book is as follows. There are two mines where coal is mined, and two thermal power plants, where the coal is burned on “just-in-time” basis, i.e. without storage. The first mine “gives to the surface” 50 tons of coal per day, while the second—70. The first power plant burns 40 tons of coal per day, while the second—80. Transportation of coal from the first mine to the first power plant (the variable **m1pp1**) costs 1200 rubles per tonne, from the first mine to the second power plant



$$40\,\frac{t}{day}\cdot 800\cdot\frac{Rub}{t}+0\cdot\frac{t}{day}\cdot 1200\cdot\frac{Rub}{t}+50\cdot\frac{t}{day}\cdot 1600\cdot\frac{Rub}{t}+(80-50)\cdot\frac{t}{day}\cdot 1000\cdot\frac{Rub}{t}=142000\,\frac{Rub}{t}$$

Fig. 7.4 Erroneous manual solution of the transportation problem

(the variable **m1pp2**)—1600, from the second mine to the first power plant (the variable **m2pp1**)—800 and from the second mine to the second power plant (the variable **m2pp2**)—1000. It is necessary to organize the transport of coal in such a way that its cost was minimal. Figure 7.4 shows the condition of the transportation problem and its "manual" solution: the cheapest route (the second mine—the first thermal power plant: 800 rubles per tonne of coal transported) is fully loaded (see the first term in the expression in the bottom in Fig. 7.4), and the remaining coal is "scattered" on other routes.

Figure 7.5 shows the solution of the optimization problem using Mathcad 15 standard features—form the objective function (**ObFunc**—the cost of transportation), provide a first approximation (we took the values found through the manual solution of the problem, as shown in Fig. 7.4), after the keyword **Given** the constraints are formed, and function Minimize finds the unknowns to minimize the objective function when the restrictions are applied. Automatic solution of the problem turned out to be paradoxical—nothing should be transported at the cheapest route.

The problem of coal transportation—probably the only one in the book which is more appropriate to be solved in MS Excel environment, rather than in Mathcad's environment. After all, this is purely an accounting program where the money appear. MS Excel spreadsheet are designed specifically for economic calculations, as noted in the *introduction* here.

Figure 7.6 shows calculations in MS Excel environment for the problem of transportation of coal, which repeats the suboptimal "manual" calculations in Mathcad's environment, as shown in Fig. 7.4. The objective function (cost of transportation) is stored in cell **B15** and displayed in the formula bar.

The user can change the transport plan (cell **B11**, **B12**, **B13** and **B14**) to meet the constraints for the exit of coal from the mines and bringing it to the thermal power plant with minimum cost of transportation. But the job could be charged to a computer by calling a command (dialog box) **Solver** from the **Data** tab (Fig. 7.7).

$$ObFunc(m1pp1, m1pp2, m2pp1, m2pp2) := 1200 \cdot \frac{RUB}{tonne} \cdot m1pp1 + 1600 \cdot \frac{RUB}{tonne} \cdot m1pp2 \; ... \; + 800 \cdot \frac{RUB}{tonne} \cdot m2pp1 + 1000 \cdot \frac{RUB}{tonne} \cdot m2pp2$$

Guess values $\begin{pmatrix} m1pp1 \\ m1pp2 \\ m2pp1 \\ m2pp2 \end{pmatrix} := \begin{pmatrix} 0 \\ 50 \\ 40 \\ 30 \end{pmatrix} \cdot \frac{tonne}{day}$ Data from the Fig. 7.4

$ObFunc(m1pp1, m1pp2, m2pp1, m2pp2) = 142000 \frac{RUB}{day}$ See the Fig. 7.4

Given

$m1pp1 + m1pp2 = 50 \frac{tonne}{day}$ $m2pp1 + m2pp2 = 70 \frac{tonne}{day}$ Mined coal

$m1pp1 + m2pp1 = 40 \frac{tonne}{day}$ $m1pp2 + m2pp2 = 80 \frac{tonne}{day}$ Burned coal

$\begin{pmatrix} m1pp1 \\ m1pp2 \\ m2pp1 \\ m2pp2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \frac{tonne}{day}$ No coal from Power Plants to mines

$$\begin{pmatrix} m1pp1 \\ m1pp2 \\ m2pp1 \\ m2pp2 \end{pmatrix} := Minimize(ObFunc, m1pp1, m1pp2, m2pp1, m2pp2) = \begin{pmatrix} 40 \\ 10 \\ 0 \\ 70 \end{pmatrix} \frac{tonne}{day}$$

$$ObFunc(m1pp1, m1pp2, m2pp1, m2pp2) = 134000 \frac{RUB}{day}$$

Fig. 7.5 Proper computer solution of the transportation problem

This command is available after uploading a computational Add-in **Solver** (see Fig. 8.6 in Chap. 8).

When you click **Solver** in the **Solver Parameters** dialog box (see Fig. 7.7), MS Excel tables will give us a paradoxical solution, such as shown in Fig. 7.5: nothing should be transported at the cheapest route…

One of the most complex optimization problems in the power sector is as follows. In the dark winter morning there is a maximum power consumption in the grid. Then, with the dawn electricity demand falls. Grid manager (human or computer) is to give commands to different power plants to reduce the load, to do

B15 | f_x =B11*B6+B12*B7+B13*B8+B14*B9

	A	B
1	Output of 1st coal mine, tonne/day	50
2	Output of 2nd coal mine, tonne/day	70
3	Coal fired at TPP-1, tonne/day	40
4	Coal fired at TPP-2, tonne/day	80
5	Transportation cost, Rub/tonne	
6	From 1st coal mine to TPP-1	1200
7	From 1st coal mine to TPP-2	1600
8	From 2nd coal mine to TPP-1	800
9	From 2nd coal mine to TPP-2	1000
10	Transportation scheme, tonne/day	
11	From 1st coal mine to TPP-1	0
12	From 1st coal mine to TPP-2	50
13	From 2nd coal mine to TPP-1	40
14	From 2nd coal mine to TPP-2	30
15	Total transportation cost per day	142 000,00 Rub

Fig. 7.6 The transportation problem in MS Excel environment

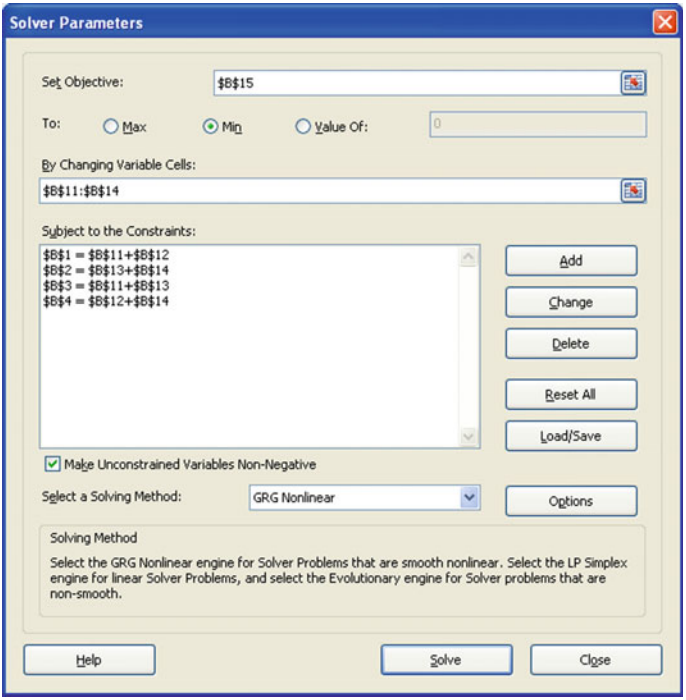


Fig. 7.7 Optimize the transportation problem in MS Excel environment

this he should be guided with some objective function—an average specific fuel consumption for electricity generation. One can select here either to turn off the whole power units, or reduce the load on some of them. And on what units? On the least or the most economical? One might find that reducing the load on the most uneconomic unit will not be the optimal solution. At the end of the day, transporting coal at the cheapest route leads to the cumulative rise in price of the coal transportation (see Fig. 7.4)… Another small paradox in this problem, associated with a number of variables, is mentioned in Chap. 17.

Grand Divertissement after Chaps. 6 and 7. The author showed the typescript of the book to many people for whom the book could be of interest. One of them expressed the opinion that the book lacks optimization problems. And our whole life is nothing but optimization. This is an old problem, which the author put in a ready book to fulfill the wishes of the reviewer.

The paint problem. Russian writer Mikhail Zhvanetsky is often asked questions where he takes themes for his sketches. "Look out the window and listen to the talk on the street"—this is the answer of the great satirist. "And how do you memorize all this?"—It is another question. "But I can not forget!"

Everyday stories are also worth collecting for writing *computer studies*, which is a hobby of the author of this book.

By profession the author is Professor in University (Moscow Power Engineering Institute—www.mpei.ru), where he gives a course of lectures on information technology (see the course outline at: http://twt.mpei.ac.ru/ochkov/Potoki.htm) and related disciplines (http://twt.mpei.ac.ru/ochkov/Potoki_MOpt.htm), and manages a group of process engineers and programmers, developing training programs and computer simulators for thermal and nuclear power plants (www.trie.ru). Power plants and power systems need our programs, but the notorious payments crisis prevents them from buying that. That's what a computer *study* appeared in March 1997 [50].

Joint-stock company "Tambovenergo" without free money, however, has expressed a desire to buy our software. Kotovsk paint factory (KPF, Tambov region) needs electricity for its operations. Moscow Power Engineering Institute needs paint to repair its classrooms. The author's scientific group needs a new computer hardware, tools and, of course, the salary. To solve such problems the humanity since the dawn of civilization has come up with the *money*. The transition of Russia from socialism to the market has revived barter. In the above commodity chain one link was missing to make it closed. Fortunately, MPEI received a batch of computers, a couple of which, as we have agreed, could be exchanged for paint.

This combination was only part of the described *computer composition*, if you remember the interpretation of the expression "chess composition"—solution to the puzzle by drawing up a chain of moves.

The second part of the *computer composition* took place in Tambov and Kotovsk—on KPF factory. In "Tambovenergo", after take-over of the software, I (the author proceeds to the story in the first person) was given a power of attorney to receive the paint products for the amount of 14 million, of course, old rubles[6] against the factory's debt for electricity and I left for Kotovsk. KPF Sales department first flatly refused to let the paint for some incomprehensible debt out there, and not for real money, but after the threat of blackouts and heat disconnection, though with difficulty, they agreed. Paint, which suits me, or rather not me, but MPEI purchasing department, costs 14,600 rubles per liter and is bottled in containers (jars, if you follow chandlery jargon, which I picked up in Kotovsk) each 15 and 55 l in volume. The cost of an empty jar was 24 and 30 thousand rubles respectively. An employee of the KPF sales department (her name was Olga), when typing an invoice on a computer, asked me a question what paint containers I would take. Instinct of a longtime collector of computer studies immediately prompted that here lies a typical and, most importantly, the real problem of *linear programming*, where the *objective function* to be maximized—the total amount of paint (or the cost of the paint), the *variables*—the number of jars filled with paint, 15 and 55 l in volume, to be collected, and three constraints:

- the cost of paint should not exceed the 14 million rubles agreed with "Tambovenergo" one can not take an incomplete jar (constraint for the integer variables);
- number of jars of different sizes should not be a negative number.

Olga offered to help in solving the *optimization problem* and then using a calculator figured that I needed to take 16 large and 2 small jars with a capacity of 910 l of paint for the amount of 13 million 814 thousand rubles. Remembering how desperately I traded at "Tambovenergo" and yet increased the price of programs from 12 to 14 million rubles, I asked Olga, if it would be possible not to loose 186,00 thousand—do not leave it in favor of "Tambovenergo". She said no, because she had to solve these problems almost every day, optimizing not only the cost of paint, but its canning in containers of different sizes, and that she was a dab hand at solving such problems.

Watching the "dance" of Olga's fingers on the keys of the calculator and the numbers on its display, I realized that Olga used a so-called "workers' and peasants' "algorithm to solve the problem: first she selected paint in a large container, and then the rest of the money (or the volume of the container) was filled with paint in a small container. It goes something like this when we pack a suitcase when going on a trip—first put the big things inside, and then cram every little thing into the empty space. I asked Olga, why she did not use to solve these problems a computer and MS Excel spreadsheet, which was put on her computer screen as if on purpose. I immediately offered to show how it's done. MS Excel environment has a solver procedure which dialog box is called by **Solver** command from the **Data** tab. In this window, the user specifies the cell storing the objective function, which is necessary to maximize, the cells with solver variables (at the beginning of optimization they

[6]In the Mathcad-worksheet and MS Excel spreadsheet solving this problem we cut-off three zeros.

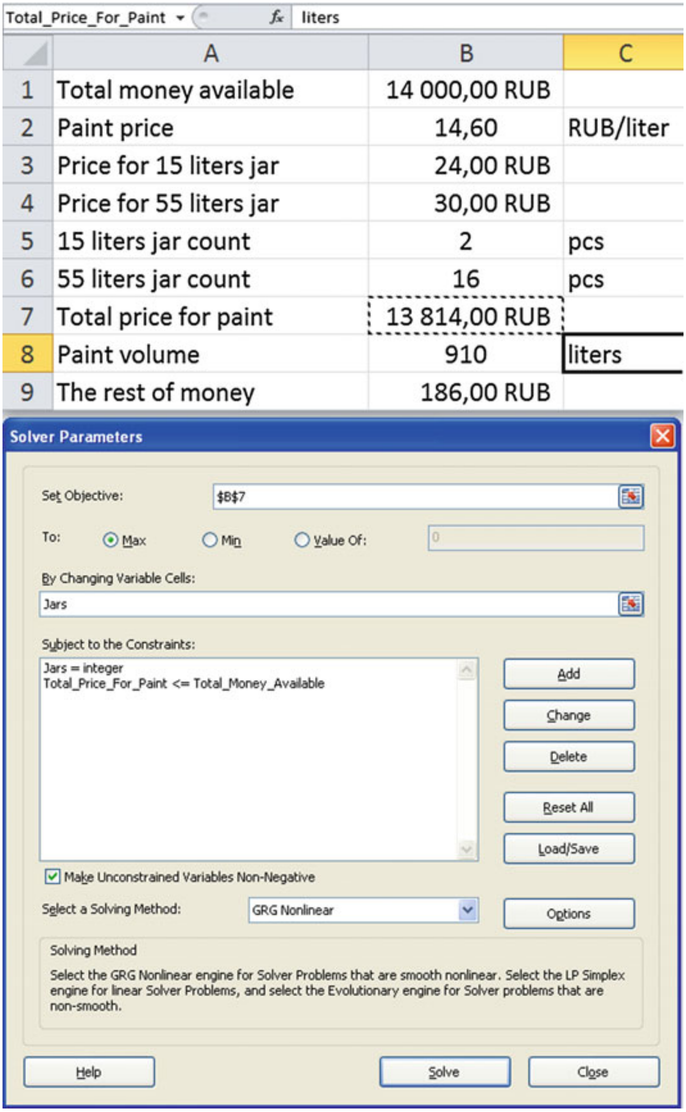


Fig. 7.8 Paint problem solved in MS Excel environment

are either empty, or store the values of the first approximation to the maximum) and the constraints (Fig. 7.8).

The optimization algorithm using the MS Excel solver procedure could be called “lazy”: the user creates a calculation table and says: “With a wave of a wand do so that… the objective function took maximum (minimum defined) value, but with that all the constraints should be met.” To do this, the user only has to click a

Solver. The MS Excel solver procedure gave us the old result—16 large and 2 small jars. But I did not want to give up.

A good rule—to verify the solution of a problem using not only other methods, but also other software tools. Also, do not forget about the KISS-programming principle. This has nothing to do with kisses, although its good attitude to the problem and the computer is visible. KISS—an abbreviation of the English phrase "Keep It Simple, Stupid!". It urges to solve the problem in the easiest way and only use sophisticated algorithms and techniques when simple methods are not suitable because of the length of time spent on computations or due to irrational use of other resources, human and/or computer.

The simplest way to solve the problem using a computer—it is to search through all the options and stop at the optimum. Fortunately, the options are not so many—1088: for the allotted 14 million rubles one could not take more than 63 small paint jars, or not more than 16 large jars. The brute force search could be called "workers' and peasants'" computer solver method. But among other things it can give a hundred percent certainty that the found solution is not only correct, but also unique, or demonstrate that there are several such solutions. A similar situation is not uncommon in the integer linear programming problems.

So, the brute force method. By following the above rule, a new method for solving the problem should be combined with a new software tool for its implementation. This, of course, could be also done in MS Excel environment by drawing up a table of all solutions and/or writing a brute program using built-in Visual Basic for Applications (VBA) language. But Olga also had Mathcad on her computer (the phenomenon of a rabbit pulled out of a hat). Mathcad quite successfully solves problems of various nature (including economic ones) without turning to pure programming (BASIC, C, Pascal, etc.). Moreover, that time I was working on one of my books on Mathcad. Example with *paint* could only give more colors to the book (unintentional pun).

Protocol of paint "check weighing" in Mathcad's environment is shown in Fig. 7.9. Comments explain what is happening in the formulas. In the Mathcad-worksheet a matrix is formed with the name **Price**, whose elements store the paint cost values depending on the combination of packaging. Then, some elements of the matrix **Price** are set to zero, if the packaging combinations are not in the value. The rest—sleight of hand and no mathematics: use function match to determine the line number (37) and column number (6) of the matrix **Price**, the element with the maximum value is located at their intersection. Olga was unpleasantly surprised with the answer (37 small and 6 large cans). She unwittingly deceived me for 175 thousand rubles.

So Mathcad saved me 175 thousand, though old, but rubles. Money is not so great, but if we add to it the new computer study placed in the book, a new lecture topic and a new laboratory work in computer science, the game was worth the candle.

When I came back home to Moscow from Tambov, in a relaxed atmosphere at my home computer I one more time analyzed the problem. And that are the results.

Input data

Rub	$Money$	$Ppice_{paint}$	$Price_{15}$	$Price_{55}$
(¤)	(Rub)	$\left(\frac{Rub}{L}\right)$	(Rub)	(Rub)
1	14000	14.60	24	30

Reviewing all variants for Solution $Jar_{15} := 0, 1..63$ $Jar_{55} := 0, 1..16$

Matrix of price variants packaging paint jars 15 and 55 liters

$$Price_{Jar_{15}, Jar_{55}} := Ppice_{paint} \cdot (15\ L \cdot Jar_{15} + 55\ L \cdot Jar_{55}) + Price_{15} \cdot Jar_{15} + Price_{55} \cdot Jar_{55}$$

Discard variants that do not pass on the money

$$Price_{Jar_{15}, Jar_{55}} := \text{if}\left(Price_{Jar_{15}, Jar_{55}} > Money, 0\ Rub, Price_{Jar_{15}, Jar_{55}}\right)$$

Finding the maximum price of the paint in jars

$$Price_{max} := \max(Price) = 13989\ Rub$$

Answer $\text{match}(Price_{max}, Price) = \left[\begin{bmatrix} 37 \\ 6 \end{bmatrix}\right]$

Fig. 7.9 Paint problem solved in Mathcad prime environment

Table 7.1 A complete analysis of the optimal solutions

Parameter	Solutions			
Packaging option (quantity of small cans/quantity of large cans)	2/16	6/15	13/13	37/6
Paint volume, l	910	915	910	885
Unused cash balance, rubles	186,000	47,000	12,000	11,000

Firstly, one could make the process of finding solutions in MS Excel to correctly "explain" the paint problem through changing the initial settings. And for this it was necessary to not be lazy and click the **Settings** key in the **Solver** dialog box. It was enough in the new dialog box **Solver Parameters** to reduce the permissible deviation from 1 to 0.01 %. After that, the right solution would be found. Frankly, in MS Excel it is not a solver procedure that is bad but its initial settings. Very few MS Excel users resorting to its services, click the **Parameters** key. Normally, those who understand the essence of optimization do not work with MS Excel. Hence the confusion. Table 7.1 provides a complete analysis of the optimal solutions. Secondly, when I showed this table in MPEI procurement department, they told me that the best option for me (for me important is money) and for MPEI (for them important is paint) will be the forth option: The "Tambovenergo" would spend almost all the money, and the 885 l of paint, as it may seem strange, is more than 910, and 915. The matter is that with the large packaging a lot of paint is lost due to

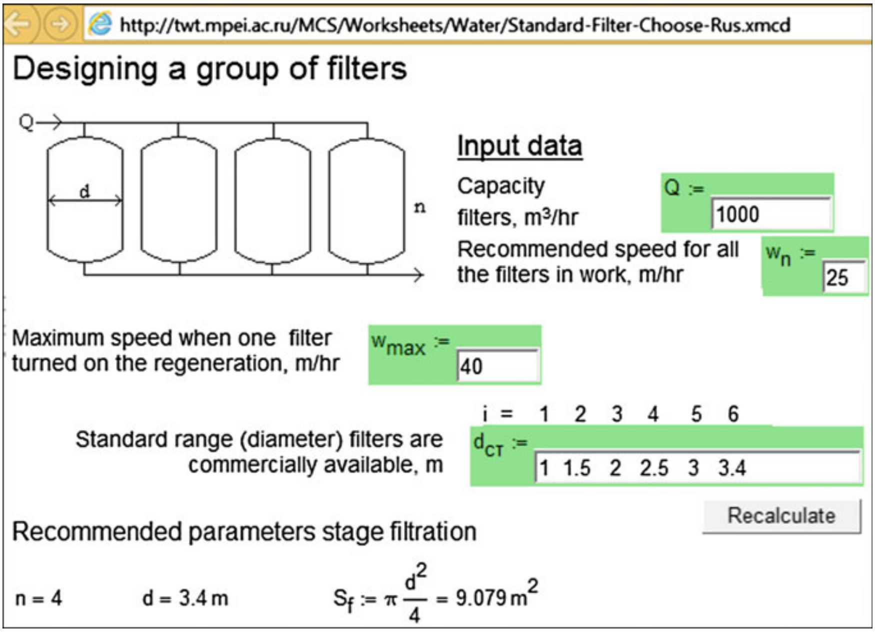


Fig. 7.10 The calculation for the group of filters used for treatment of make-up water of a steam turbine unit

overflows into a smaller container. One could take a 15-l jar to a room under repair and make full use of it.

The wrong solution of the problem is caused not only by the bad practices or defective software, but also by the fact that the user himself does not really know what he wants. All programs for solving the linear programming problems require a clear formulation of a single objective function. In solving the educational problems the goal is clear. What is the objective in real life? However this is not mathematics, but philosophy…

Computers in the optimization of thermal power facilities allow wide use of brute force method, which was applied in the paint problem. For example, when designing a make-up water treatment plant for steam turbine units, one can take many filters of small diameter or a few large diameter filters. The best option will be found through an exhaustive search [51].

Figure 7.10 shows a page of the website where you can set the water flow through the filters, high-speed characteristics (range of possible changes in the filtration rate) of one filter, and diameters of commercial standard filters. After clicking on **Recalculate** button, all the options will be searched through by the number of installed filters and their diameters to choose an optimum option whose real rate of filtration is as close to a specified recommended rate as possible.

On the website http://twt.mpei.ac.ru/MCS/Worksheets/Water/N-D-H-Price.xmcd. one could perform an on-line calculation, similar to the one shown in Fig. 7.10, with the complicating difference to reflect the cost of the filter material, filters and valves on the filters. The best option will be the one whose cost would be minimal.

Chapter 8
Maximum Water Density

Valery Ochkov and Konstantin Orlov

Abstract This chapter shows how to solve thermal engineering problem and to involve for this the audience of the specialized forums in Internet. Discussed the problem of the density of water depending on its parameters.

Water has the unique property—its density has a maximum value at a temperature slightly above zero Celsius. Therefore, and due to the fact that solid phase of water (ice) has a density lower than water, in winter basins generally do not freeze to the bottom—the temperature at the bottom of a basin has a value of about +4 °C.

The properties of water, as a practically incompressible fluid, are largely dependent on the temperature and little depends on the pressure. Nevertheless, we will try to develop the maximum density water temperature/pressure function.

Figure 8.1 shows a corresponding calculation made in Mathcad 15 environment.

After reference to file **H2O.xmcdz**, one can see in the calculation the function **wspDPT**, returning water and steam density **D** as function of pressure **P** and temperature **T**. Then using this the "cloud" function create another function $\mathbf{t}_{max}(\mathbf{p})$ with one argument **p**. To find the functional relationship maximum, Mathcad's environment provides for built-in function **Maximize**, but it did not work in our problem (see the error message shown in Fig. 8.1). Probably because of the fact that our function **wspDPT** in function **Maximize** is called without listing the arguments in brackets.[1] We had to use another Mathcad's built-in function—**root**, which

The site of the chapter: https://www.ptcusercommunity.com/message/423023.

[1]It was the author's assumption, which turned out to be false (see below).

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_8

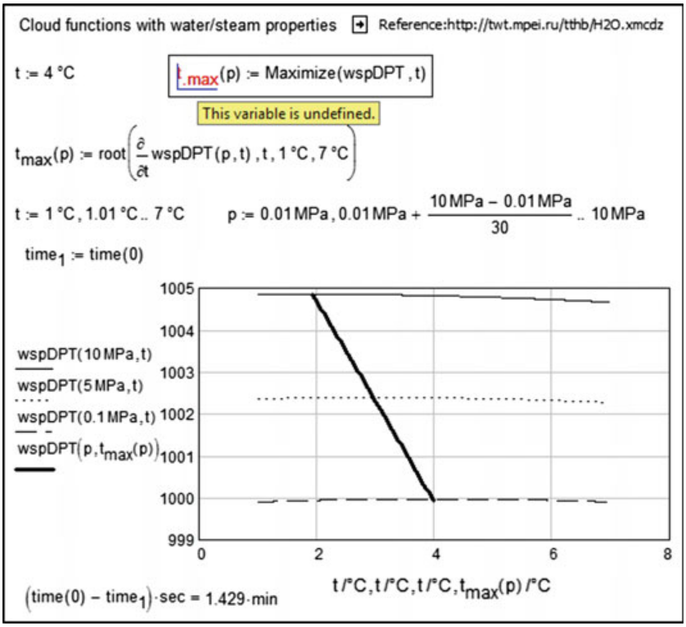


Fig. 8.1 Diagram of maximum density water temperature as a function of its pressure

returns the values of its second argument **t**, with which the first argument (partial derivative function **wspDPT** for **t** at fixed **p**) is equal to zero. The third (1 °C) and the forth (7 °C) arguments of the function **root**—are the boundaries within which the required temperature values are to be found.

Figure 8.1 shows the four parametric curves: three water density isobars as a function of temperature at p = 0.1, 0.5 and 10 MPa (thin line, dotted line and dots) and a solid line linking the points where the density is at its maximum. The diagram shows that the peak shifts to the left with an increase in water pressure.

However, the author is not satisfied with the solution shown in Fig. 8.1. First of all because creating the diagram took a very long time and because the diagram was not created at the most interesting point—at a pressure close to atmospheric.

Before and after the diagram in Fig. 8.1 there are operators available which include a Mathcad's built-in function **time**. It has a formal argument, whose value does not affect the result of calling the function (here we used zero), and returns some computer operating time in seconds. Of course this time is conveying not much,[2] but using the difference between the time values, you can calculate the execution time of individual operators of a Mathcad-document. The diagram shown

[2]The thermodynamic analogy: working medium's enthalpy and entropy values don't add-up too much: It is important to know the difference of these parameters at various points in the thermodynamic cycle.

in Fig. 8.1 took a long time—1 to 3 min, depending on the type and load of the computer on which these calculations are carried out. And this is understandable: the numerical definition of the derivative—is a very complex and not entirely correct operation in terms of “pure” mathematics. Because of this fact (the duration of calculation), such “slow” diagrams at the time of program debugging have to be disabled by the appropriate command which appears in the list of menu commands called by right-clicking, and then again included in the calculation.

The second thing that the author did not like in the solution shown in Fig. 8.1—is that the line of water maximum density as function of pressure was “torn” near the atmospheric pressure point: function $\mathbf{t}_{max}$ cannot calculate the temperature of maximum density of water at p = 0.1 MPa. Therefore, an attempt was made to return to the function **Maximize**.

All computer programs are supported by the developers, including through the users’ formula. If the user has some question, he can post it in the appropriate forum and look forward to hearing other forum visitors’ responses. The Mathcad **also** has its forum and not one. The main Mathcad forum (PTC Community) is located at https://www.ptcusercommunity.com/community/mathcad. One could also come to this forum directly from the Mathcad’s environment by choosing the appropriate command from the **Help** menu (Mathcad 15, in Mathcad Prime environment this command can be found on tab **Getting started**). In this forum without registration one can only view open resources. But after the registration it will be possible to ask questions and respond to other people.

In Mathcad forum the author asked a question whether it is possible for the problem shown in Fig. 8.1 to use not “capricious” and slow function **root** with a derivative as an argument, but function **Maximize**. One man from Austria Werner Exinger responded to this “call for help”. He showed that it is enough just to swap around pressure and temperature in the function, and it will work (Maximize function change only last argument of input function). Our dialog recorded in Fig. 8.2 [52].

Figure 8.3 shows the corrected calculation. It began to work much faster, despite the fact that the number of points on the diagram became not 30 (as in Fig. 8.1), but 300. And most importantly, the diagram has ceased to be torn—function $\mathbf{t}_{min}$ began to return a response throughout the specified range of pressure—from 0.01 to 10 MPa.

Conclusion: If, when you are working in Mathcad, difficulties arise, you should only ask a question on a forum (see Fig. 8.2). A positive response is likely to be obtained. Moreover, in this forum, one can create a subforum (Group), open to all, or only to invited users. In this “cyberspace” we can work together to solve problems, including heating engineering problems. The author have opened on a Mathcad forum his own Group for this book, where it is possible to discuss the book, add new or adjust old examples. It is an English-language book forum. Russian forum, as noted in the Preface, is located at www.thermal.ru. On these forums one could download “cloud” patterns and functions necessary for calculations.

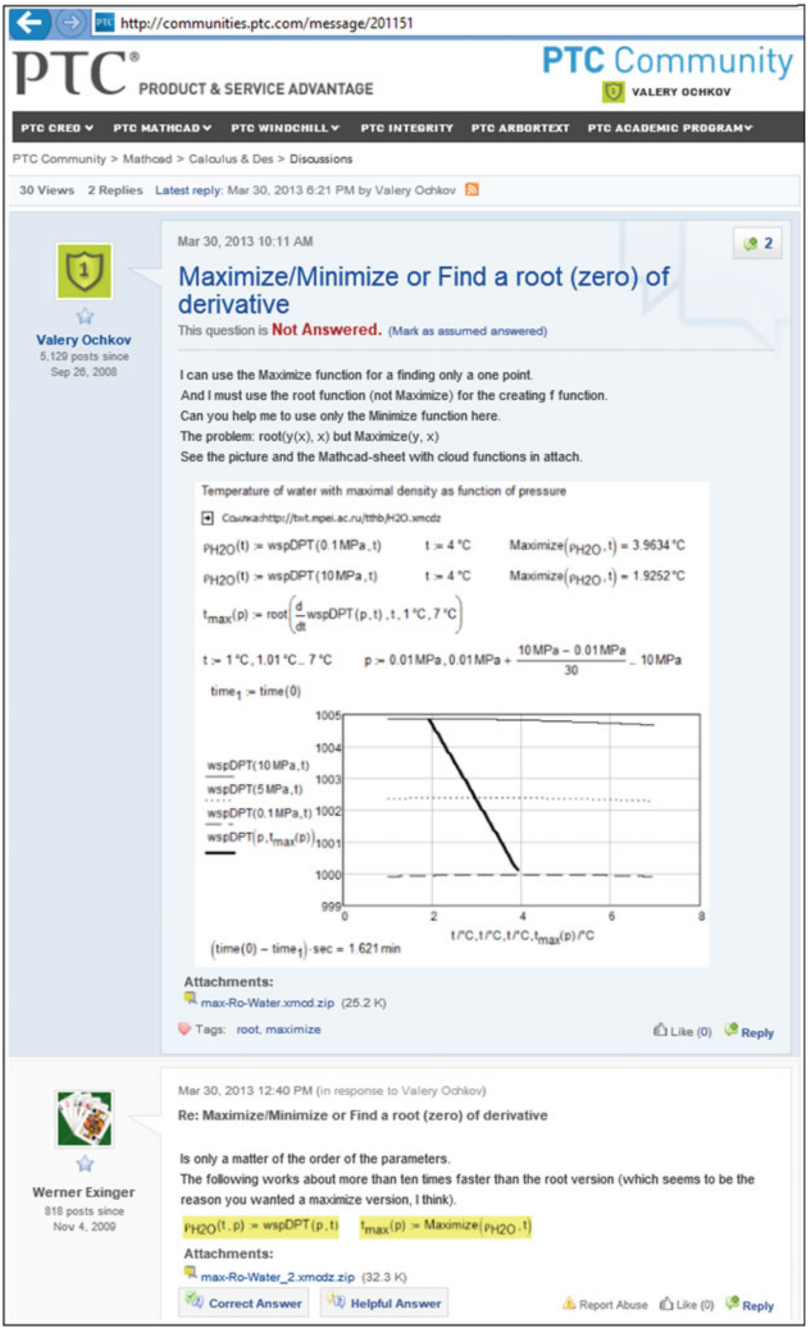


Fig. 8.2 Mathcad forum

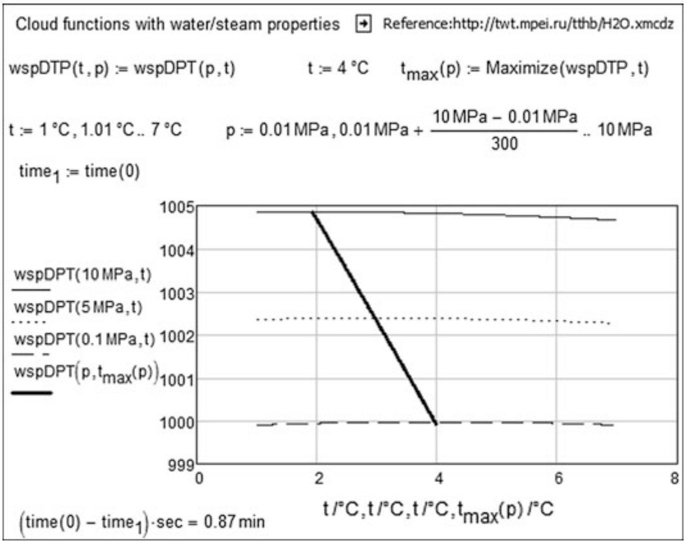


Fig. 8.3 Adjusted calculation of maximum density water temperature/its pressure function

On the site of the reference book [23, 38] one can find the water/steam properties "cloud" functions not only for Mathcad program, but also for its Russian clone—SMath (www.smath.info). Figure 8.4 shows a page of the reference book with water property functions (the first area of formulation IAPWS-IF97).

If you click on the link **SMath** in the water specific volume row after you have installed on your computer from the site www.smath.info the SMath program itself, a hidden area would appear in its environment with the relevant function **wspV1PT** (Fig. 8.5). An inverse function named **wspD1PT** (water density as function of pressure and temperature) has been used in Chap. 4 here in the description of the modern Internet technology for publishing calculations.

Using function **wspV1PT** one can create function **wspD1PT** (water density) and analyze it near the temperature of maximum density of water.

The problem of the maximum density of water at a fixed pressure is easy to solve in MS Excel as well, if in addition to the WaterSteamPro attach to these spreadsheets another add-in—**Solver** (Fig. 8.6).

Figure 8.7 shows MS Excel table, where in cell B2 is entered a first approximation value—the water temperature on the Celsius scale. In cell **B3**, this temperature is shown in Kelvin (**=B2 + 273.15**). Cell B4 has water pressure value entered in megapascals, which in cell B5 is converted into Pascals (**=B4 * 10 ^ 6**). Cell **B6** stores WaterSteamPro function **wspDPT (B5; B3)**, that returns the density of water as function of pressure and temperature. Figure 8.7 also shows **Solver Parameters** dialog box, which called from the **Data** tab, and stating that the computer has to automatically change the value in cell **B2**, so that the value in cell **B6** took its maximum.

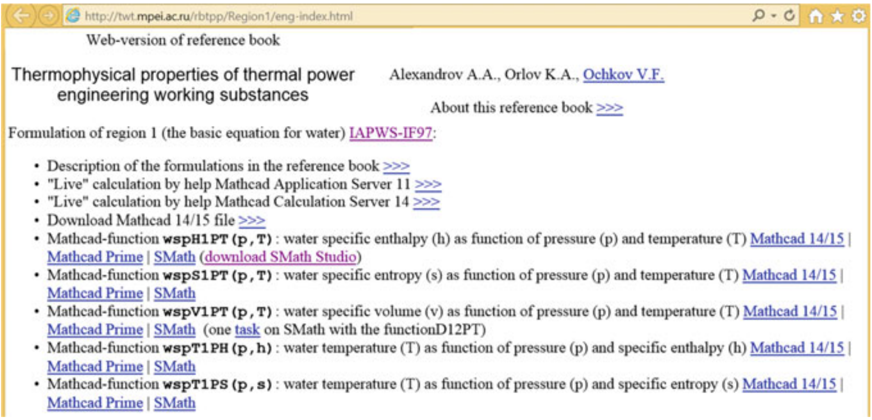


Fig. 8.4 Web-page of water properties reference book

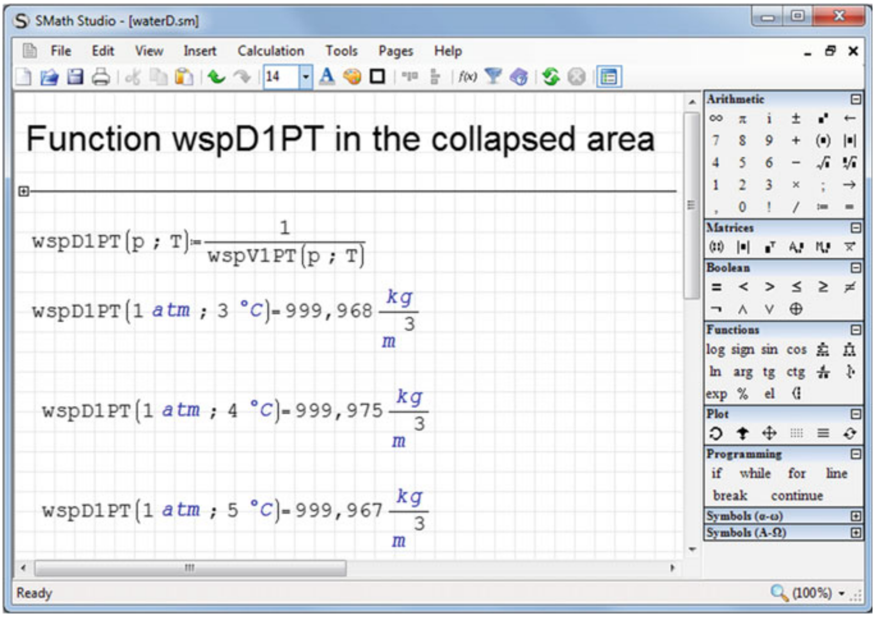


Fig. 8.5 Water density calculation in SMath

After pressing the **Solver** key in the **Solver Parameters** box (this key is not shown in Fig. 8.7, but one could see it in Fig. 7.8 in Chap. 7) in cell B6 appears the value 999.9748 (maximum density of water in kilograms per cubic meter at pressure 0.1 MPa), and in cell B2—3.96335 (maximum water density Celsius temperature).

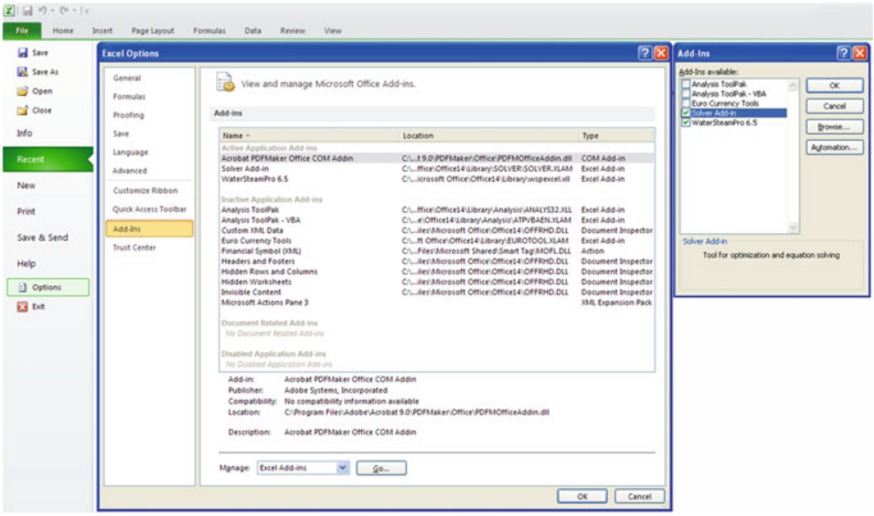
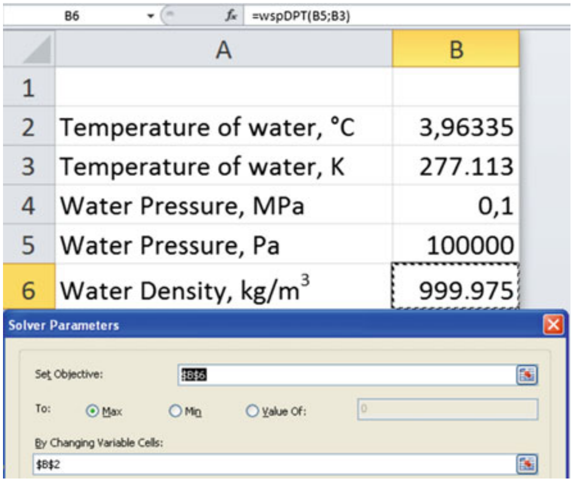


Fig. 8.6 List of MS Excel add-ins

Fig. 8.7 Calculation of maximum density of water in MS Excel environment



Water density as function of temperature and pressure can be displayed on the contour plot—the one level line graph. Figure 8.8 shows such a graph plotted using a very handy Mathcad built-in function **CreateMesh**—see at https://www.ptcusercommunity.com/message/258547. This function creates a mesh (in this case the size is 300 × 300) of triangles with the values of a two arguments function in a specified area, which can then be displayed using a surface or lines of one level (see Fig. 8.8). In our case this is function **wspD1PT** returning the water density

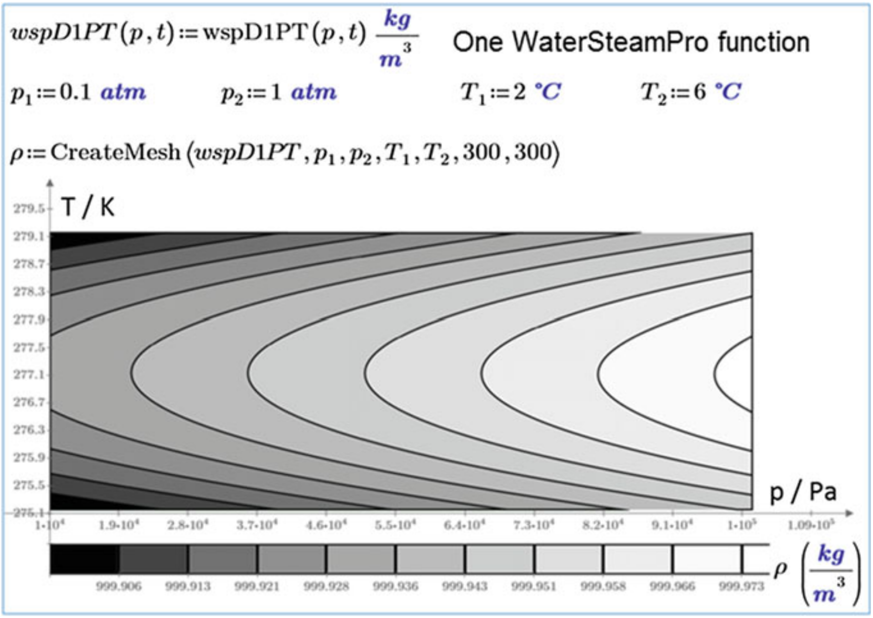


Fig. 8.8 Water density as a function of pressure and temperature: one level lines

(area 1 in formulation IAPWS-IF97—see Fig. 8.4) as function of pressure and temperature. In Mathcad Prime the function **CreateMesh** began to work with dimensional arguments, which is very convenient: set the desired range of pressure and temperature and receive without much hassle the right graph, a surface or a family of one level lines.

The graph (see Fig. 8.8) demonstrates that the maximum water density (the leftmost point of the level lines) is a function of water density. Function **CreateMesh**, by the way, have been used here to graphically illustrate that thermal efficiency of a steam turbine cycle is a function of pressures in two steam extraction points (see Fig. 6.4). In Fig. 8.8 the level lines are complemented with a scale showing how the graph fill colors are changing. We use ten shades of gray, marking a change in density from 999.906 to 999.973 kg/m^3.

Chapter 9
Nuclear Power Plant Steam Turbine Cycle

Valery Ochkov and Konstantin Orlov

Abstract This chapter is about calculation of steam turbine cycle of nuclear power plant operating on wet steam. You can learn about authors Internet sites that help to solve thermal engineering tasks.

Figure 9.1 shows a principle scheme of a steam turbine unit of a double circuit power reactor of a nuclear power plant (NPP) [53] (so called Pressure Water Reactor—PWR). Pressurized water flows from the nuclear reactor (NPP primary circuit) to the steam generator (SG) and transfers heat to the boiling water of the second circuit. Saturated steam is supplied from the steam generator to the high pressure cylinder (HPC) of the steam turbine. Part of the live (fresh) steam from the steam generator enters the superheater. Before being superheated, the exhaust steam from the HPC is partially dried in the separator where some water is separated from the wet steam, and then returned to the circuit with bypass of the low pressure cylinder (LPC) of the steam turbine.

This complication of the thermal circuit of a nuclear steam turbine cycle compared with a thermal power plant on fossil fuel (see, e.g., Fig. 4.2 in Chap. 4) is associated with the following NPP feature: if saturated steam is supplied to a turbine, whose outlet pressure is 5–4 kPa, moisture in the last stages of the steam

The site of the chapter: https://www.ptcusercommunity.com/message/423024.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_9

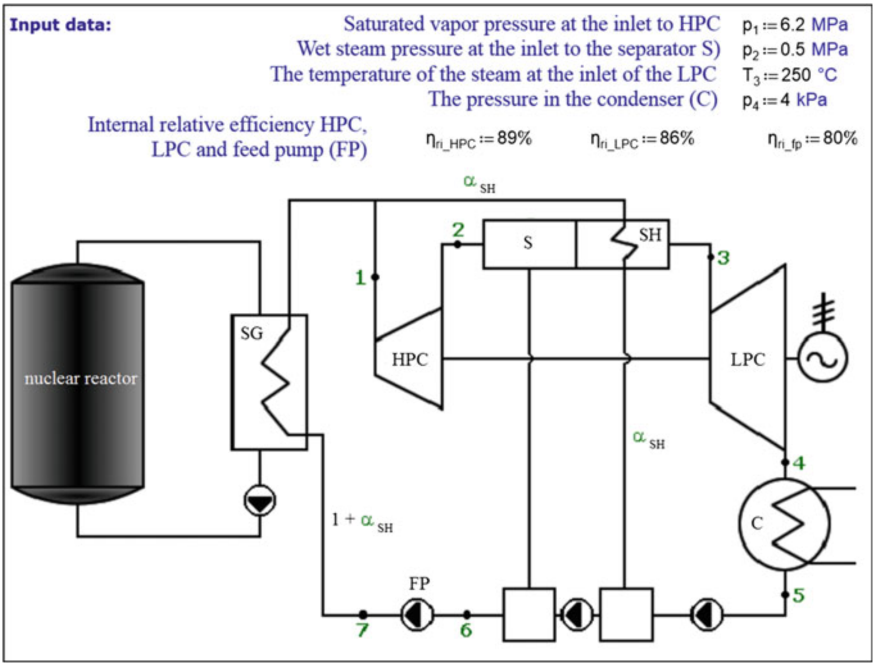


Fig. 9.1 Calculation of the thermal efficiency of the nuclear power unit with wet steam: Input data and the unit scheme

turbine will be unacceptably high which may lead to the steam turbine failure. Steam moisture at the outlet of a nuclear turbine without separator and superheater can be estimated by referring to the two Internet sites [23], as shown in Figs. 9.2 and 9.3.

First, steam pressure upstream turbine (6.2 MPa—see input data in Fig. 9.1) can be used to determine steam temperature, taking into account that this is saturated water/steam.

Then the site, shown in Fig. 9.3, allows you to calculate (in online mode) the turbine steam expansion process by setting the values of initial pressure, initial temperature, final pressure and relative internal efficiency of the process.

Note. The view of the site (URL: http://twt.mpei.ru/MCS/Worksheets/PTU/h-s-ExpEng.xmcd) slightly differs from what is shown in Fig. 9.3. The point is that the original site was copied to the clipboard by pressing <Print Screen> key, moved to the graphics editor and streamlined.

The calculation, shown in Fig. 9.3, demonstrates that the wet water steam, when expanding in the turbine from initial pressure of 6.2 MPa, at the condenser pressure of 4 kPa will consist of nearly one third water, which, as noted above, is unacceptable for the last stages of the turbine, since they cannot operate with such steam moisture.

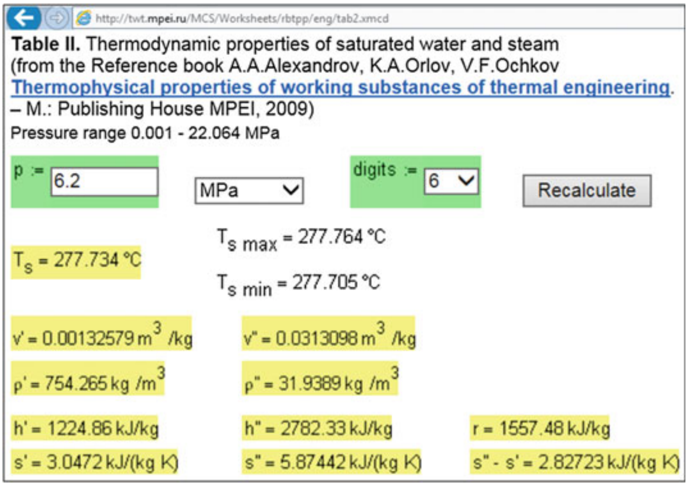


Fig. 9.2 Internet site to determine the water boiling point as function of pressure

Because of this feature of the steam turbine plant, the nuclear steam turbine is divided into two cylinders—high pressure cylinder (HPC) and low pressure cylinder (LPC). Steam, before it enters the LP cylinder, is dried in the separator and heated in steam-steam heater (superheater) using heat of live steam supplied from the steam generator (see the diagram in Fig. 9.1).

Figure 9.4 shows the site we have already referred to for calculation of steam expansion process in the turbine with another (not 4 kPa, but 0.5 MPa) pressure at the turbine outlet.

Comparing Figs. 9.2 and 9.3, you can see that with increasing pressure at HP cylinder outlet from 4 kPa to 0.5 MPa, moisture of steam leaving the turbine has dropped from 27.8 to 16.8 %. This percentage of steam moisture in the last stages of HP cylinder (16.8 %) is quite acceptable for trouble-free operation.

You can "by hand" calculate thermal efficiency of the steam turbine cycle shown schematically in Fig. 9.1, referring to the web-sites supporting the Reference Book [23] and shown in Figs. 9.2, 9.3 and 9.4. Thus, Fig. 9.5 represents an *h, s*-diagram of steam expansion process in LP cylinder after separating and superheating the steam. But we will proceed with our calculations on a computer with Mathcad, the beginning of computations is shown in Fig. 9.1.

As we have noted in Chap. 2, there are water and steam property functions Water-SteamPro (www.wsp.ru) that can be attached through DLL to Mathcad 15 (see Fig. 2.21) and Mathcad Prime 3 (see Fig. 2.23). In Mathcad Prime 3 (namely in this software we will calculate the NPP cycle, this was already started in Fig. 9.1), the work process has slightly changed compared to Mathcad 15. In Mathcad 15, the function arguments with suffix wsp could not be automatically resized for given dimension, so the arguments had to be left with no dimensions (see Fig. 2.24 in Chap. 2). In Mathcad Prime 3, this operation (de-dimensioning of the arguments) is

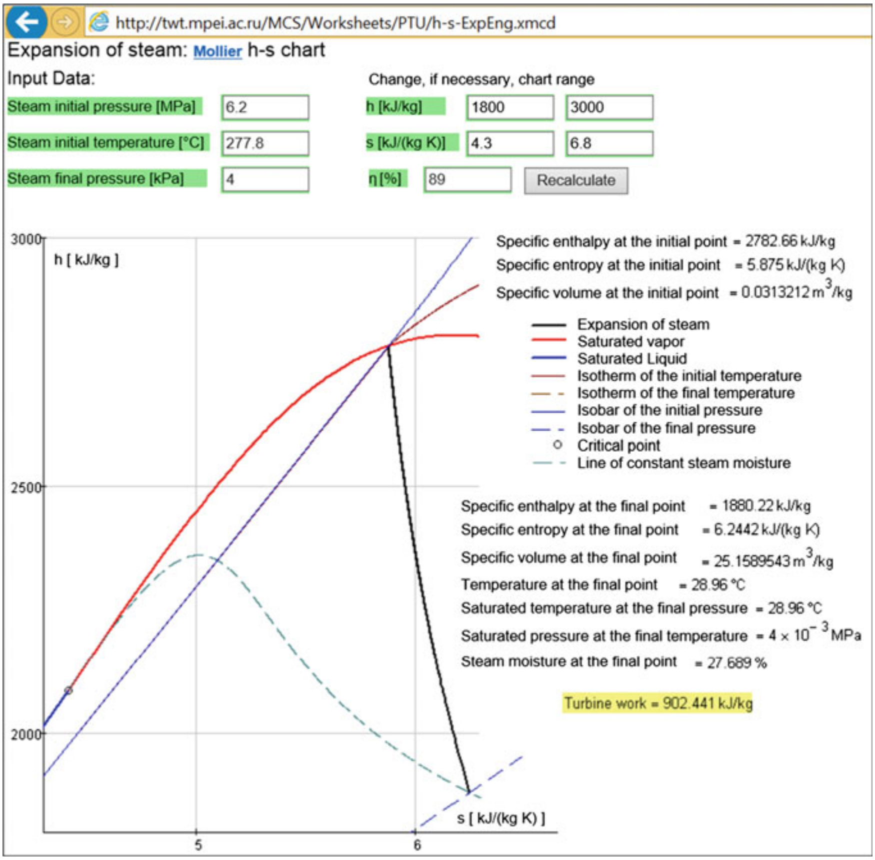


Fig. 9.3 Internet site for calculating the turbine steam expansion process

done automatically, but the WaterSteamPro functions continue (as in Mathcad 15) to give dimensionless quantities corresponding to the base SI units: pressure in pascals, and not in megapascals, temperature in degree Kelvin, and not by centigrade, etc.

To function properly in Mathcad Prime 3 with dimensional values, the WaterSteamPro functions should be overwritten as shown in Fig. 9.6. As in Chap. 2 we highly recommend to reference file watersteampro.xmcd for Mathcad 15 or redefine functions like it showed on Fig. 9.6. In this case all functions arguments and results will be automatically translated to desired dimensions of WaterSteamPro functions inputs and back with results.

In a Mathcad document the function overwriting operators are grouped into a separate area that can be minimized by clicking on the icon “minus”, shown in the upper left part of Fig. 9.6. Figure 9.10 shows this area minimized. In addition, this area contains user’s entered “one-line” (without division operators) units of specific

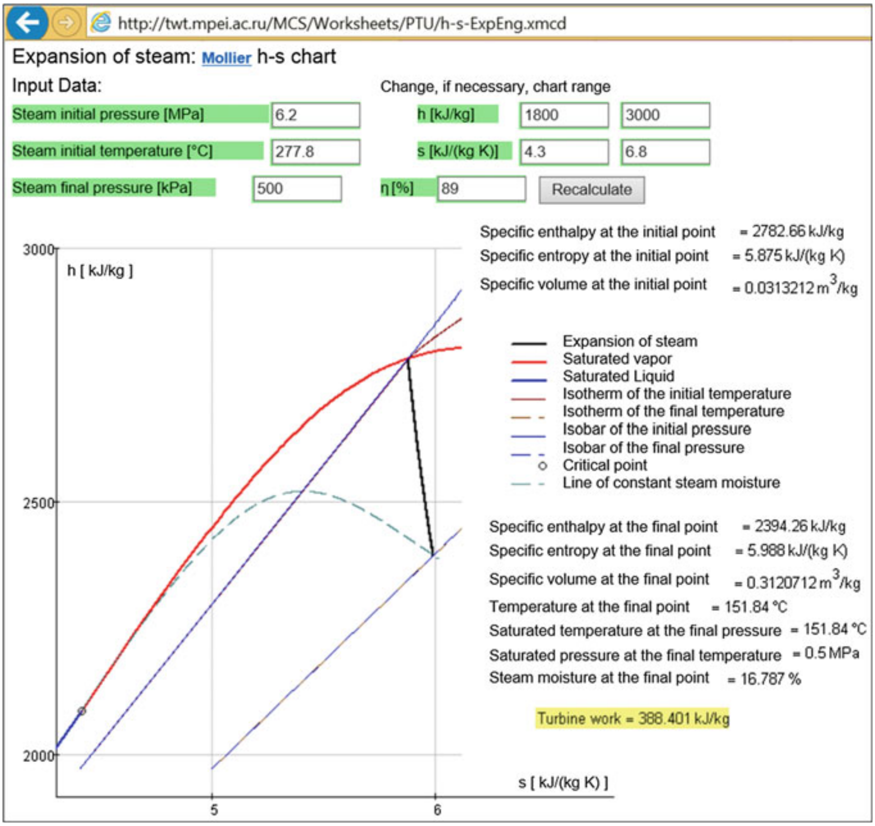


Fig. 9.4 Internet diagram of steam expansion process in HP cylinder of a NPP steam turbine

entropy (kJ/(kg_K) and J/(kg_K)) and specific enthalpy (kJ/kg and J/kg). Additionally, you can enter control of dimension of WaterSteamPro functions as shown in Fig. 9.7.

You can clearly see in Fig. 9.7, that call of the function **XEXPANSIONPTPEFF** with repositioned first two arguments (the original pressure and temperature) ends with an error message. (Function **XEXPANSIONPTPEFF** was not on the list, as shown in Fig. 9.6, since this function returns dimensionless quantity—steam moisture at the end of expansion process in the turbine, and it is not necessary to overwrite the function into dimensional form.)

After the WaterSteamPro functions became visible (through DLL) and dimensional (through overwriting functions), it is easy in Mathcad Prime 3 to calculate thermal efficiency of a nuclear steam turbine plant (see Fig. 9.7), that has been started in Fig. 9.1.

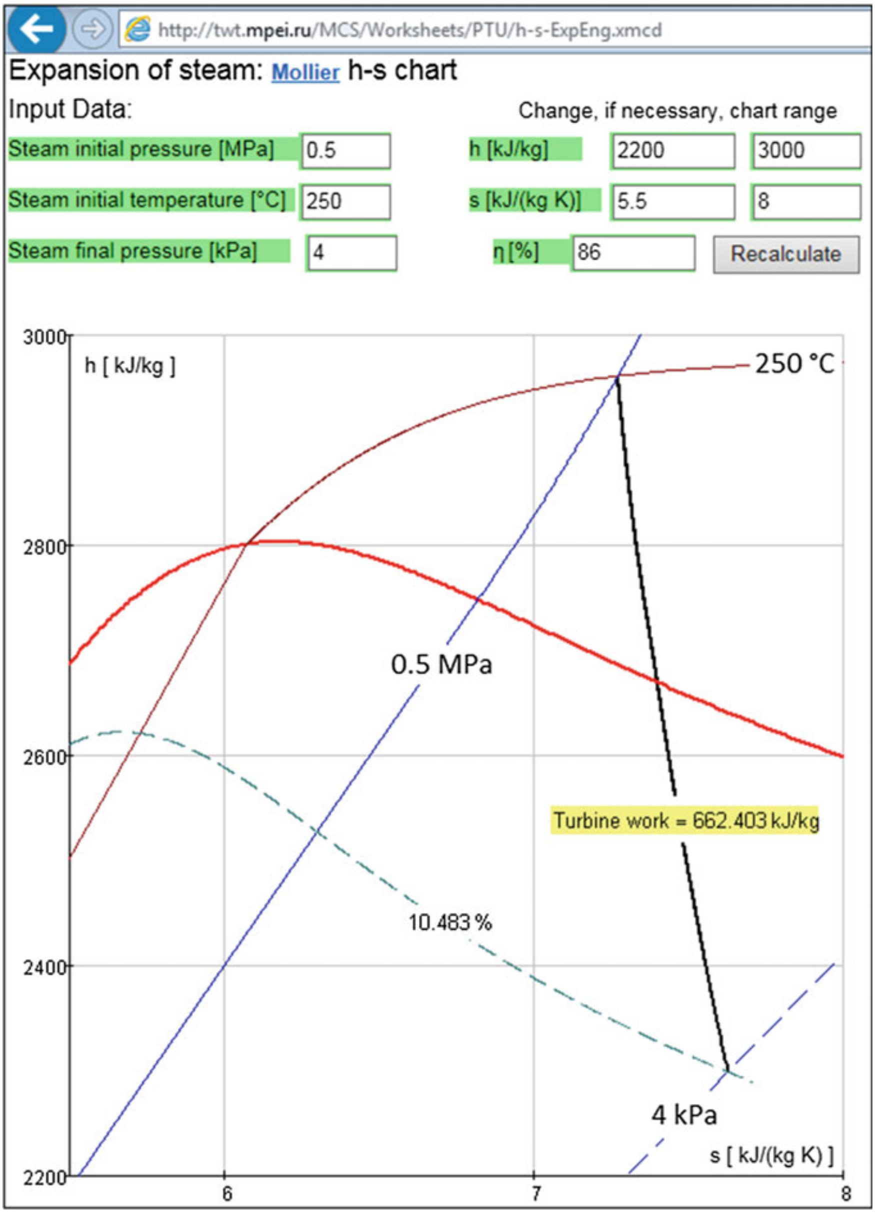


Fig. 9.5 Internet diagram of steam expansion process in LP cylinder of a NPP steam turbine

The calculation shown in Fig. 9.8 does not require special comments, and that is a very useful consumer quality of Mathcad, which we mentioned in the introduction and to which we will come back in Chap. 20.

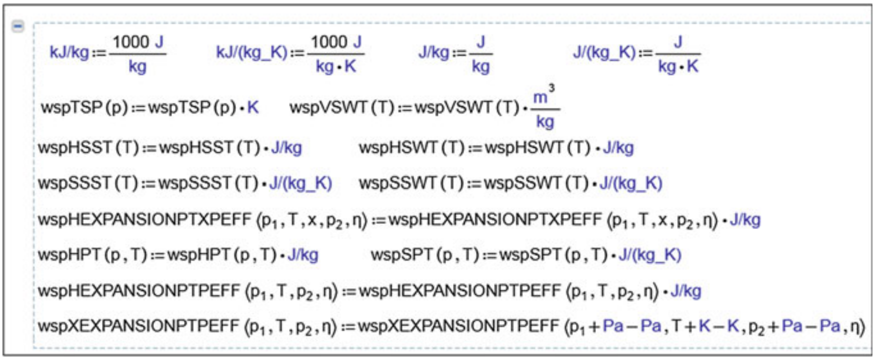


Fig. 9.6 Overwriting of WaterSteamPro functions into dimensional functions in Mathcad Prime 3

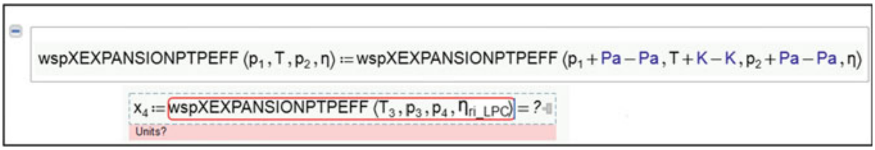


Fig. 9.7 Control of dimensions of WaterSteamPro function arguments

Divertissement. To determine a water steam moisture is a complex thermal engineering problem, especially in flow. One way of measuring this parameter is as follows. Wet steam with a known pressure (or a known temperature, these parameters are linked in the two-phase area, see Fig. 9.2) is diverted into a separate chamber, where steam becomes single phase, and where one can measure already "independent" pressure and temperature. Using these three measured parameters (wet steam pressure, temperature and throttled dry steam pressure) we can estimate the original steam moisture referring to the web-site as shown in Fig. 9.9.

Computations in the site as shown in Fig. 9.9 use three WaterSteamPro functions:

- **wspHPT**—specific enthalpy as a function of pressure and temperature;
- **wspTSP**—temperature at saturation line as a function of pressure;
- **wspXSTH**—steam moisture content as a function of temperature and specific enthalpy.

Figure 9.10 shows the corresponding calculations in Mathcad Prime 3 using WaterSteamPro functions.

On the book site the reader will find an example of the calculation of thermal efficiency of a nuclear steam turbine cycle made in MS Excel spreadsheets with WaterSteamPro connected. This calculation is not replicated here because it is almost impossible to understand. However, it is very easy to create this calculation on a computer in MS Excel, having at hand a Mathcad calculation, as shown in

1. Steam parameters at the inlet to HPC

$T_1 := \text{wspTSP}(p_1) = 277.7\ °C$ $h_1 := \text{wspHSST}(T_1) = 2782.3$ kJ/kg $s_1 := \text{wspSSST}(T_1) = 5.874$ kJ/(kg_K)

$h'_1 := \text{wspHSWT}(T_1) = 1224.9$ kJ/kg

2. Steam parameters on the inlet to the separator

$T_2 := \text{wspTSP}(p_2) = 151.8\ °C$ $h'_2 := \text{wspHSWT}(T_2) = 640.2$ kJ/kg $s_2 := s_1 = 5.8744$ kJ/(kg_K)

$h''_2 := \text{wspHSST}(T_2) = 2748.1$ kJ/kg

$h_2 := \text{wspHEXPANSIONPTXPEFF}(p_1, T_1, 1, p_2, \eta_{ri\_HPC}) = 2394.0$ kJ/kg

$x_2 := \text{wspXEXPANSIONPTXPEFF}(p_1, T_1, 1, p_2, \eta_{ri\_HPC}) = 83.201\%$

3. The parameters at the inlet to LPC

$p_3 := p_2 = 0.5$ MPa $h_3 := \text{wspHPT}(p_3, T_3) = 2961.1$ kJ/kg $s_3 := \text{wspSPT}(p_3, T_3) = 7.27$ kJ/(kg_K)

4. Parameters at the end of the expansion process of steam in the LPC

$T_4 := \text{wspTSP}(p_4) = 28.96\ °C$ $s_4 := s_3 = 7.27$ kJ/(kg_K)

$h_4 := \text{wspHEXPANSIONPTPEFF}(p_3, T_3, p_4, \eta_{ri\_LPC}) = 2298.7$ kJ/kg

$x_4 := \text{wspXEXPANSIONPTPEFF}(p_3, T_3, p_4, \eta_{ri\_LPC}) = 89.52\%$

5. Parameters on the outlet of the condenser

$h_5 := \text{wspHSWT}(T_4) = 121.4$ kJ/kg $s_5 := \text{wspSSWT}(T_4) = 0.42$ kJ/(kg_K)

6. The share of steam extraction in the separator

$$\alpha_{SH} := x_2 \cdot \frac{h_3 - h''_2}{h_1 - h'_1} = 11.38\%$$

7. The input parameters to the feed pump

$$h_6 := \frac{x_2 \cdot h_5 + \alpha_{SH} \cdot h'_1 + (1 - x_2) \cdot h'_2}{1 + \alpha_{SH}} = 312.39 \text{ kJ/kg}$$

8. The specific pump work

$$l_{FP} := \text{wspVSWT}(T_1) \cdot \frac{p_1}{\eta_{ri\_fp}} = 10.27 \text{ kJ/kg}$$

9. Parameters at the outlet of the feed pump $h_7 := h_6 + l_{FP} = 322.66$ kJ/kg

10. Specific work of HPC $l_{HPC} := h_1 - h_2 = 388.3$ kJ/kg

11. Specific work of LPC $l_{LPC} := x_2 \cdot (h_3 - h_4) = 551.1$ kJ/kg

12. The thermal efficiency of the cycle

$$\eta_{oi} := \frac{l_{HPC} + l_{LPC} - l_{FP}}{(1 + \alpha_{SH}) \cdot (h_1 - h_7)} = 33.917\%$$

Fig. 9.8 Calculation of thermal efficiency of steam turbine cycle of PWR NPP

Figs. 9.1 and 9.8, and checking if the formulas are entered correctly and correctness of references to the table cells. But without these tips it would be quite difficult to create even such a simple calculation table in MS Excel.

The author created a WaterSteamPro, which is often referred to in this book, in the C++ programming language. But before that, all of the individual functions of the package were written and debugged in Mathcad. It's allows to save time for development. Without this preliminary work, it would have been very difficult to write a program in C++. In conclusion: When getting to the creation of a larger project requiring the use of a programming language or an MS Excel spreadsheet, it

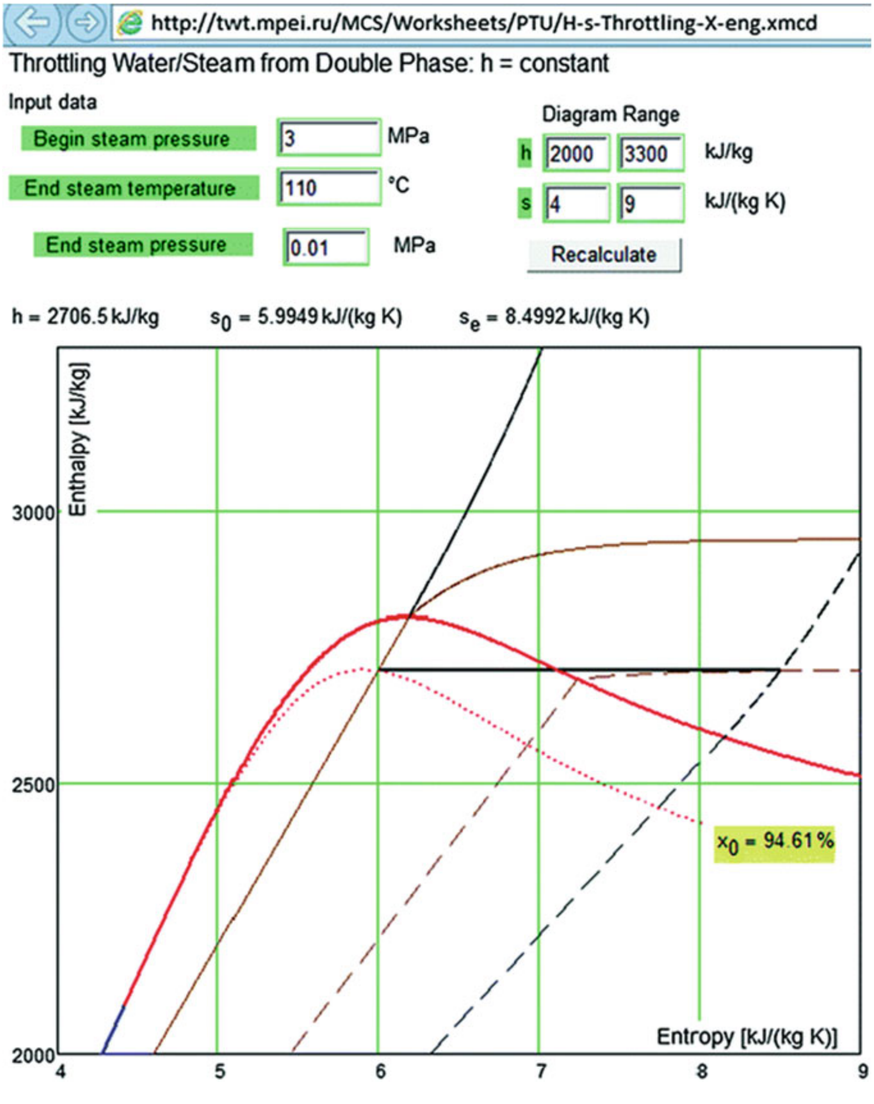


Fig. 9.9 Internet calculation of steam moisture throttling process

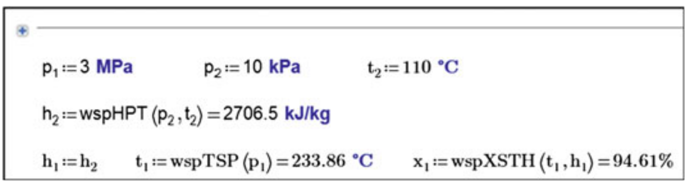


Fig. 9.10 Calculation of steam moisture content in Mathcad Prime 3

is useful to write and debug individual modules in Mathcad. It is much easier and expedites the work by eliminating many errors. One can, of course, execute whole project in Mathcad, but programming languages allow you to generate executable code that runs on a computer without the programming languages themselves. As for MS Excel (which, as a matter of fact, is installed on almost every computer), these tables are used by many end users who want that a ready program should be executed exactly in this software. The author of this book once developed a program to optimize the water chemistry of a condenser cooling system of a thermal power plant [3]. The program itself, minimizing the cost of prevention of scale deposits in the cooling system, was written in Mathcad. Then it was manually transcribed for MS Excel, because the staff of the power plant used to work with these spreadsheets and did not want to learn any other programs.

Chapter 10
Isobar, Isotherm, Isochor…

Valery Ochkov, Konstantin Orlov and Volodymyr Voloshchuk

Abstract This chapter describes how you can build contour lines (isobars, isotherms and isochores) and lines of phase transitions on the state diagrams of water and steam.

Different types of diagrams could illustrate the thermal processes: *T*-, *s*-, *h*-, *s*-, *p*-, *v*- and others. There are many of these diagrams in this book. Points on such diagrams are often fixed as crossing points of isolines, e.g. isobars and isotherms (see, e.g. Fig. 9.3 in Chap. 9). "The third time is a charm"—an isochor is missing here. Let us use this small chapter to see how we can build this "third time" for isolines on *h, s*-diagram [34].

Problem Superheated steam pressure and temperature values are given. It is necessary to build in Mathcad's environment on *h, s*-diagram, firstly, the water and steam saturation line, secondly, the crosshair of isobar, isotherm and isochor showing the specified pressure and temperature values. Figure 10.1 shows the solution of this problem in Mathcad Prime 3.0.

The diagram in Fig. 10.1 built in Mathcad Prime 3.0 with WaterSteamPro package connected. Without this package, you can build such a diagram in Mathcad 15 environment, making reference to the "cloud" file H2O.xmcdz, stored at

The site of the chapter: https://www.ptcusercommunity.com/message/423028.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Voloshchuk
National University of Water Management and Natural Resources, Rivne, Ukraine
e-mail: Vl.volodya@gmail.com

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_10

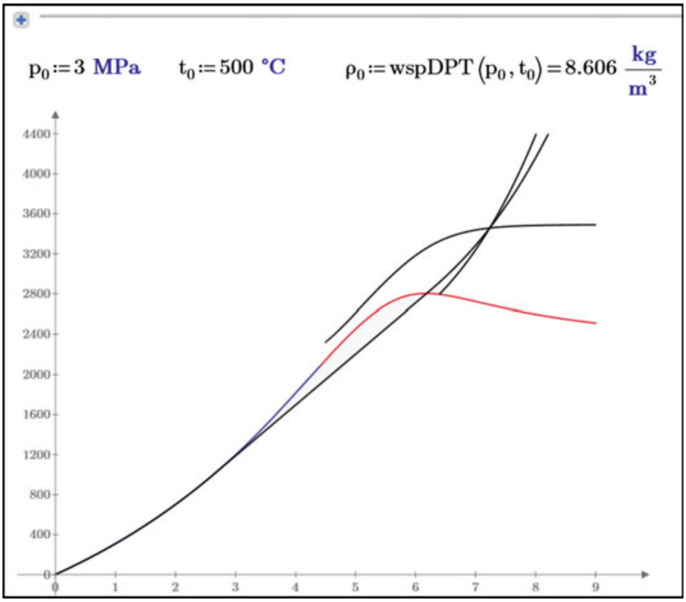


Fig. 10.1 Water/Steam isobar, isotherm and isochor on hs-diagram

http://twt.mpei.ru/tthb, we have done this more than once, when solving some problems in this book.

Figure 10.1 shows a collapsed Mathcad-area, which is further shown expanded in Fig. 10.2.

Operators shown in Fig. 10.2 used for the following.

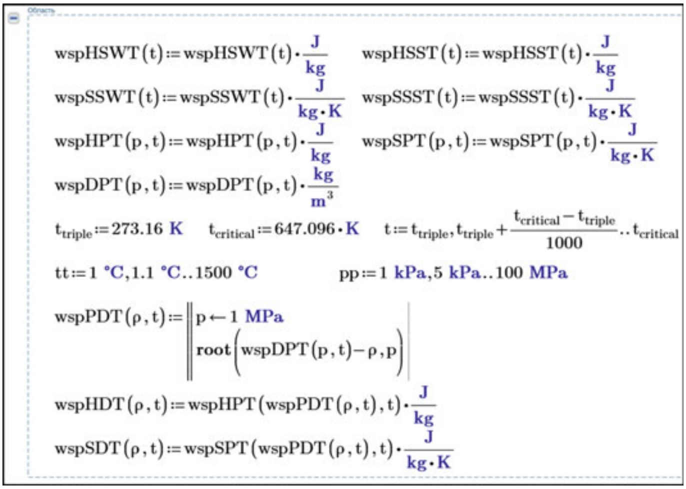


Fig. 10.2 Auxiliary operators in the problem of building the water/steam isobar, isotherm and isochor

First, WaterSteamPro functions are to be overwritten, so that they become redefined in Mathcad Prime 3.0 environment, since Mathcad Prime 3 does not allow defining user-function from C code with units. This operation described in Chap. 9 (see Fig. 9.6). Not all the functions of this package are to be overwritten, but only those that will be needed in solving our problem:

- **wspHSWT**—specific enthalpy (**H**) at saturation line (**S**) of water (**W**) as a function of temperature (**T**);
- **wspHSST**—specific enthalpy (**H**) at saturation line (**S**) of steam (**S**) as a function of temperature (**T**);
- **wspSSWT**—specific entropy (**S**) at a saturation line (**S**) of water (**W**) as a function of temperature (**T**);
- **wspSSST**—specific entropy (**S**) at a saturation line (**S**) of steam (**S**) as a function of temperature (**T**);
- **wspHPT**—specific enthalpy (**H**) of water/steam as a function of pressure (**P**) and temperature (**T**);
- **wspSPT**—specific entropy (**S**) of water/steam as a function of pressure (**P**) and temperature (**T**);
- **wspDPT**—density (**D**) of water/steam as a function of pressure (**P**) and temperature (**T**).

Second, there are three range variables introduced with the names **t**, **tt** and **pp**, storing discrete values of temperature and pressure curves for the construction of our *h, s*-diagram.

Range variable **t** stores 1000 discrete values of temperature from the triple point (273.16 K) to the critical point (647.096 K). This variable used as auxiliary variable in building water and steam saturation lines. Using this variable and using functions **wspHSWT(t)**, **wspHSST(t)**, **wspSSWT(t)** and **wspSSST(t)** one could calculate specific enthalpy and entropy values on the saturation line and construct a pair of parametric diagrams converging at the critical point.

The range variable **tt** is used as an auxiliary variable in building isobar and isochor in a temperature range from 1 to 1500 °C with an increment of 0.1 °C. Using this variable and using functions **wspHPT(p_0, tt)** and **wspSPT(p_0, tt)** one could calculate values of specific enthalpy and entropy for the isobar and build a parametric diagram. Functions **wspHDT(ρ_0, tt)** and **wspSDT(ρ_0, tt)** (discussed below) are used to calculate values of specific enthalpy and entropy for isochor and also to build a parametric diagram.

The range variable **pp** is used as an auxiliary variable in building isotherm in the pressure range from 1 kPa to 100 MPa. Using this variable and using functions **wspHPT(pp, t_0)** and **wspSPT(pp, t_0)** calculate values of specific enthalpy and entropy for isotherm and build a parametric diagram.

Isobar and isotherm in the single-phase region on *h*, *s*-diagram are easy to build, having at hand built-in (added) in Mathcad functions that return specific enthalpy and entropy as a function of pressure and temperature. To do this, we repeat, it is enough on the axes y and x of the diagram to write (insert) functions **wspHPT(p_0, tt)** and **wspSPT(p_0, tt)** for isobar and **wspHPT(pp, t_0)** and **wspSPT(pp, t_0)** for isotherm.

To build an isochor, one needs to have functions that return values of specific enthalpy and entropy as a function of density and temperature or pressure. However, there are no such functions on the functions list in WaterSteamPro. But one could create them, relying on the built-in Mathcad's function **root**. This operation described in Chap. 1 (see Fig. 1.10). This auxiliary problem could be solved by using the third group of operators, as shown in Fig. 10.2. First, create a function **wspPDT**, returning water/steam pressure as a function of density and temperature, and second, create two functions—**wspHDT** and **wspSDT**—using wsp-functions **wspHPT** and **wspSPT**, where the first argument will be not pressure, but the previously created function **wspPDT**.

This chapter is small in size but very important to learn how to use the tool *functions* when solving problems on a computer. Knowing this tool, one could easily build other types of diagrams and other isolines on them, e.g. lines where h = const (line of constant enthalpy, an isenthalpa) or s = const (line of constant entropy, an adiabat). It would be very useful to learn how to use the function tool from the point of view of making calculations without schedules. An expert (not only thermal engineer), creating a set of his own user-defined functions, linking different values, could easily solve complex problems by combining and inserting into each other various previously created and debugged functions.

Chapter 11
Construction of Forward and Backward Functions on Properties of Working Fluids on Tabular Data Base

Volodymyr Voloshchuk

Abstract This chapter shows how to create functions that return the values of the thermophysical properties of working fluids of power engineering, on the basis of tabular values in different areas of the state: single-phase region, two phase region, the saturation line.

For investigation, optimization, improvement and further development of heat and power systems thermophysical modeling should often be applied. For this purpose data about properties of working fluids, which are involved in energy transfer and conversion, are required. These data are partially published on paper or electronic media in the form of tables, graphs, formulations (sets of formulas and guidelines for their use) or computer programs.

If such thermophysical calculations are carried out "by hand" it is possible to use a table or a graph. But today this technology is used only in very simple cases.

For variant and more complex modeling special computer programs with a possibility of calculating working fluids properties are required. One of the most powerful programs among them is a **REFPROP** program (www.nist.gov/srd/nist23.htm) of the National Institute of Standards and Technology (NIST—www.nist.gov) supplied on disks or downloaded from the Internet.

But quite often such programs are not suitable for their direct application in further computer modeling. Properties calculated in those programs have to be manually entered in additional program which is impractical, unrealistic and error-prone.

Saving in one program or one computer data about all existing working substances is impossible and/or unreasonable too. Any reference base is modified and expanded over time. The real and rational way is to provide opportunities to receive

The site of the chapter: https://www.ptcusercommunity.com/message/423029.

V. Voloshchuk (✉)
National University of Water Management and Natural Resources, Rivne, Ukraine
e-mail: Vl.volodya@gmail.com

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_11

from popular database necessary information concerning functions on required properties of the substance. This information technology can be successfully implemented within the Mathcad.

The authors have developed and tested Internet technology of "cloud" resources which are arranged on the computation server of the Moscow Power Engineering University's (MPEI) National Research University (www.mpei.ru) and Ltd Trieru (www.trie.ru)—http://twt.mpei.ac.ru/ochkov/VPU_Book_New/mas. "Cloud" resources provide a possibility to calculate thermophysical properties of working fluids and are accessible and convenient in application.

The calculation server proposes three options of using its resources. Online option gives a possibility to do "cloud" calculations of the properties in an interactive mode with the possibility for each user to enter his own input data. Downloading enables using functions on thermophysical properties for calculations on the own computer, even without an Internet connection. The reference procedure gives a possibility to do a reference to the functions on thermophysical properties on the Internet server or in "clouds" for developing further individual thermophysical calculations on the own computer.

The proposed functions can be both forward and backward. For example, in addition to a function which returns the values of specific enthalpy versus pressure and temperature in a single phase (forward function) authors propose functions which return the values of pressure or temperature of the working fluid in a single phase versus enthalpy and temperature or pressure respectively.

Part of the "cloud" functions and technology of their application in calculations is described in [23–25, 55]. These functions are related to water and steam (which is the main working fluid of fossil and nuclear power plants) and are based on the formulations approved by the International Association for Properties of Water and Steam (IAPWS—www.iapws.org; authors of this book are members of this Association). These formulations are not based on table data but formulas. Fast calculation of the properties of different substances can be done by additional techniques [76, 77].

In many cases it's quite difficult to find formulations on properties of working substances and to somehow connect these databases to the working environment (Excel, Mathcad, Matlab, other programming languages and specialized programs, etc.). Additional time and resources are required to implement or convert formulations in specialized computer programs.

The authors propose new approaches to construction of both forward and backward functions on properties of working fluids on tabular databases. Below necessary programming blocks are listed.

In this case estimating tabular data with a continuous function is one of the first task. The type of this function can be determined by a variety of requirements.

If the graph of the function should pass through the points specified within the range of a discrete set of tabular data interpolation can be applied. Automatically estimating beyond the original tabular range results extrapolation.

If the goal of the function is not only to estimate relationship between values of the data in general but to minimize possible errors in the tabular data, methods of filtering and smoothing are applied.

Formulation process of not yet fully investigated or newly synthesized compounds is being carried out during two phases. Initially so called skeletal table on properties through experimental measurements is created. Then out of these discrete tabular data one common function or a set of functions is generated by various mathematical methods for different areas or applications, and predicting possible errors for the primarily use on computer programs—this is called the formulation. But often only tables are published without the mathematical processing.

In reality application of interpolation techniques for creating functions on thermophysical properties of working fluids is one of the most convenient. In this case all values specified within the range of a tabular data will lie on the graph of the function. The choice of this method is supported by the fact that modern software have built in tools for conducting different types of interpolation. It has been already mentioned in Chap. 1.

Functions created on tabular data cover only a limited range of arguments. But the task should be appointed and solved in such a way that it would be possible to extend the range of function calculations by additional inserting of tabular data.

The proposed approaches to creation of such functions are partially presented by the authors earlier including the works [27, 55].

Let's consider in a more detailed manner some aspects which were not covered before.

This is demonstrated on example of functions for calculation of some properties of refrigerants.

The tabular data base is taken from the program **REFPROP**. The available in this program options enable to change the step of discrete tabular data, the number of decimal places and others.

In a single-phase region properties of working fluids depend on two parameters. And if the task is to create a common function for both single-phase regions of subcooled liquid and superheated vapor, it should be remembered that these regions below sub-critical parameters are separated by a saturation region.

Figure 11.1 shows a fragment of the program for creating a function to calculate the entropy of the refrigerant R410a in the single-phase regions of subcooled liquid with two arguments—pressure (first row) and temperature (first column). The given function is created as a programming block which can be applied to other programs.

The function can be denoted in different ways. For convenience, it is proposed to denote it as **R410aSPT**—the same principle as in **WaterSteamPro**. Here the first five letters indicate the type of working substance (R410a). The sixth letter indicates the parameter, which is calculated (specific entropy **s**). The last two letters indicate arguments on which the specific entropy is calculated (pressure **p** and temperature **T**). These parameters (arguments of the function) are duplicated in the brackets after the name of the function.

As shown in Fig. 11.1 at the beginning of the program the arguments of the function lose their units **p** ← **p/MPa** and **T** ← **T/K**. Without this the built-in function **cspline** would not work in Mathcad 15 (it can work in the newest version, Mathcad Prime). Then control is made if arguments **p** and **T** exceed the specified range of tabular data. After this it is proposed to insert a matrix of thermophysical

```
R410aSPT(p,T) := | "Refrigerant R410a. Specific entropy as a function of temperature (t) and pressure (p) in a single phase"
                 | p ← p/MPa
                 | T ← T/K
                 | error("Pressure is low")  if  p < 0.35529
                 | error("Pressure is high")  if  p > 4.4501
                 | error("Temperature is low")  if  T < 250
                 | error("Temperature is high")  if  T > 450
                 | M1 ←
```

0	0.35529	0.4293	0.51458	0.61221	0.72329	0.84899	0.99052	...	4.4559
250	0.87106	0.87089	0.8707	0.87048	0.87023	0.86995	0.86964	...	0.86223
255	0.89949957	0.89933	0.89913	0.8989	0.89864	0.89834	0.89801	...	0.8902
260	0.927741265	0.927556958	0.92734	0.92709	0.92681	0.9265	0.92614	...	0.91788
265	0.955940241	0.955672495	0.955395676	0.95511	0.95481	0.95448	0.9541	...	0.94532
270	0.984045449	0.983729013	0.9833975	0.983054242	0.9827	0.98233	0.98193	...	0.97255
275	1.011405666	1.011228201	1.011014301	1.010758694	1.010456155	1.0101	1.0097	...	0.99963
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
340	4.350692362	4.220725982	4.07578784	3.916082214	3.742284594	3.555521857	3.357479135	...	1.4042

fictitious values

Fig. 11.1 Fragment of the program for creating a function to calculate the specific entropy of the refrigerant R410a in single-phase regions of subcooled liquid with two arguments—pressure (*first row*) and temperature (*first column*)

values (in this case specific entropy) which are functions of the determined arguments. The function arguments, pressure **p** and temperature **T**, are located in the first row ("hat") and first column (outset) respectively as published on paper media. If values of pressure and temperature overflow beyond the determined boundaries the values are not calculated. Thus, the user clearly sees the initial data applied for the calculation. In Fig. 11.1 the input matrix corresponds to a state of subcooled liquid. Below the critical pressure and temperature such matrixes are only partially filled. But to realize interpolation tool built in Mathcad it is required the matrix to be filled completely. That's why initial matrix should be added with fictitious values. Extrapolation can be used for this purpose. In this case it is also possible to calculate values beyond the scope of the input matrix and to use iterative methods. In Fig. 11.1 the region of fictitious values is *separated*. In the same manner the table is created for the superheated vapor region of the refrigerant (see Fig. 11.2). For convenience matrixes on properties of the working substance in single-phase regions without phase transition (at supercritical regions) can be inserted additionally. In this case the matrix will be filled completely and no fictitious regions will be required. As a result it is possible to insert several matrixes corresponding different argument ranges.

The next step of the program is choosing the matrix which includes the specified values of the arguments. As shown in Fig. 11.3 there are three different matrixes for three ranges of arguments. The first matrix $\mathbf{M}_1$ corresponds to the single phase region of subcooled liquid below critical pressure—so, for this matrix values of pressure which fit this region should be applied. Moreover the matrix $\mathbf{M}_1$ can be used for temperature lower than the saturation temperature at the given pressure. For this purpose a special function for calculating the saturation temperature depending on pressure (**R410aTSLP(p)**) is applied. Here the first five letters indicate the type of working substance (R410a). The sixth letter indicates the parameter, which is calculated (temperature **T**). The next two letters indicate that

$M_2 \leftarrow$

0	0.35407	0.42785	0.51285	0.61016	0.72089	0.84619	...	4.4501
250	1.8587	1.8272357782	1.7964381364	1.7624933563	1.715749281	1.6688919339	...	−34.1571018041
255	1.8773	1.8475	1.8169345341	1.7847773425	1.7451246405	1.7030525336	...	−29.1273865335
260	1.8951	1.8664	1.8368	1.8061193356	1.7713898562	1.7334262668	...	−24.5939886598
265	1.9121	1.8843	1.8559	1.8265	1.795022464	1.7605105067	...	−20.5309572165
270	1.9287	1.9015	1.8741	1.8459	1.8165	1.7848026267	...	−16.9123412371
275	1.9448	1.9181	1.8914	1.8643	1.8363	1.8068	...	−13.7121897552
⋮		⋮	⋮	⋮	⋮	⋮		⋮
340	2.1312	2.1076	2.0847	2.0622	2.0402	2.0185	...	1.6172

fictitious values

Fig. 11.2 Fragment of the matrix on values of entropy of the refrigerant R410a in a superheated vapor region as a function of two arguments—pressure (the *first row*) and temperature (the *first column*)

the function calculates values of the saturated liquid (**SL**). The last letter indicates argument on which the saturation temperature is calculated (pressure **p**). This parameter (argument of the function) is duplicated in the brackets after the name of the function. The smallest permitted value of temperature (**250 K**) and the range of permitted pressure have been specified above (see Fig. 11.1). It should be noted that refrigerant R410a is a zeotropic mixture. It is a non-ozone depleting blend of two HFC refrigerants (R32, R125) which have different saturation temperatures at the same pressure. It means that evaporation starts at the saturated liquid temperature of the most volatile component and the temperature progressively rises during evaporation until the saturated vapor temperature of the less volatile component is reached. That's why saturated liquid and vapor temperatures of the R410a are different. As a consequence functions on properties of this refrigerant at *saturated* liquid and *saturated* vapor *lines* will be different too. For example, for calculation of saturated vapor temperature for a given pressure a function **R410aTSVP** was created which differs from function **R410aTSLP** in denoting the eighth letter by **V** instead of **L**. In this case collocation SV points out that saturated vapor parameter is calculated. The matrix $\mathbf{M_2}$ corresponds to the single phase region of superheated vapor below critical pressure and temperature ranging from 250 to 340 K. In the given case this matrix can be used for temperatures lower than **340 K** and higher than saturated vapor temperature for a given pressure. The left region of the matrix can be expanded to the critical point. But in the critical region the values of the function are *rapidly changing and it is recommended to insert additional* matrix with smaller step. The matrix $\mathbf{M_3}$ corresponds to the single phase region of superheated vapor below critical pressure and temperature ranging from 340 to 450 K. There is no saturation line in this matrix. Matrixes $\mathbf{M_2}$ and $\mathbf{M_3}$ border each other along the isotherm T = 340 K. The border should be included into one of these matrixes. If the input value of the temperature **T** is not included in the scope of permissible values it can be located in the two-phase region or on the saturated line. In this case the program returns an error message "The point is in the double phase or saturated line" (see Fig. 11.3) using built in Mathcad operator **error(S)**, where **S**—the text of the error message.

```
M ← M1  if  R410aTSLP(pMPa) > T K
M ← M2  if  340K > T K > R410aTSVP(pMPa)
M ← M3  if  T K ≥ 340K
error("The point is in the double phase or saturated line")  otherwise
```

Fig. 11.3 Fragment of the program for creating a function to calculate the specific entropy of the refrigerant R410a in the single-phase region

The next fragment of the program for creating a function on the specific entropy of the refrigerant R410a in the single-phase region is shown in Fig. 11.4. With the help of built in Mathcad operator **submatrix** additional matrixes out of the chosen matrix **M** are extracted—the first row (vector of pressures **p′**) and the first column (vector of temperatures **T′**). The vector of pressures is transposed into a column with the special operator built in Mathcad. Then the operator **submatrix** forms matrix **s′**—values of specific entropy of working fluid, which is also transposed swapping rows and columns. In Mathcad the built-in variable **ORIGIN** is set to **0**. It means that the first row/column is designated with the index **0**.

In the program shown in Fig. 11.4 a method with double spline interpolation is used (see also Fig. 1.1). At first out of the columns of the entropy matrix **s′** an additional line (vector **s″**) for a given pressure is generated by spline interpolation. Values of vector **s″** correspond to the values of temperatures from the vector **T′**. Further again spline interpolation is used for calculating the specific entropy as a function of the temperature absent in the vector of temperatures **T′**. The Mathcad has its own operator of spline interpolation with the help of which it is possible to create a function of two arguments. But in this case there is one important limitation factor—the matrix should be square shaped. The proposed method of double spline interpolation can be used for rectangular matrix too. The last operator of the programming block shown in Fig. 11.4 returns the necessary unit of the entropy—kJ/(kg K).

```
p' ← submatrix(M, ORIGIN, ORIGIN, ORIGIN + 1, cols(M) − 1)^T
T' ← submatrix(M, ORIGIN + 1, rows(M) − 1, ORIGIN, ORIGIN)
s' ← submatrix(M, ORIGIN + 1, rows(M) − 1, ORIGIN + 1, cols(M) − 1)^T
for  i ∈ 0.. cols(s') − 1
   s"i ← interp(cspline(p', s'<i>), p', s'<i>, p)
interp(cspline(T', s"), T', s", T) · 10^3 J / (kg K)
```

Fig. 11.4 Fragment of the program for creating a function on the specific entropy of the refrigerant R410a in the single-phase region

Based on created in Mathcad functions with two arguments *sp*- и *sT*-diagrams have been plotted (see Fig. 11.5). On these diagrams three regions which correspond to matrixes $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$ are highlighted with different backgrounds. Figure 11.5 is copied from web-page address of which is also indicated in this figure. The Internet resource gives a possibility to do interactive calculations. The values of arguments, namely pressure **p** and temperature **T**, are entered from keyboard by a user. Units of arguments can also be chosen by a user. After clicking the button **Recalculate** the calculated value of specific entropy in different units and graphical illustrations in *sp*- and *sT*-phase diagrams of the calculated point are displayed.

Let's consider developed and tested by authors some approaches to creation of backward functions on properties of working fluids in single-phase regions.

If analytical expression of the function is known and it is known that in the specified region the backward function exists it is possible to find analytical expression for the latter function too.

Mathcad has built-in operators for solving equations and searching zeros of functions which enables to receive necessary data for finding backward functions. For example, the built-in function **root** solves the task of searching zeros functions by using bisection method or secant method.

After finding a range of roots of equations as a result of equating the values of the function to the specified numbers and receiving a tabular data of the dependent variables for which a function is defined the backward function can be created on the base of spline interpolation.

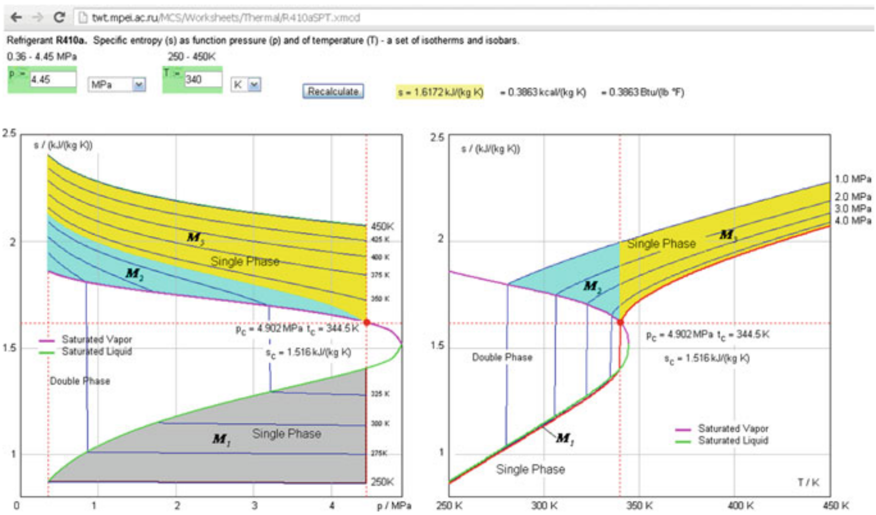


Fig. 11.5 Graphical illustration of the created function on specific entropy of the refrigerant R410a in *sp*- and *sT*-phase diagrams

If a function of two arguments is not specified analytically but in table the algorithm of creation the backward function is the same as in the case of the forward function. The difference is that values of the forward function are the arguments of the backward function. Hereby it is possible to use the tabular data in the same form as in the method of creating the forward function **R410aSPT(p, T)** for determining the specific entropy depending on the pressure **p** and temperature **T** (Fig. 11.6) which is described above.

At first arguments of the functions lose their units. After this control either the inputs does fall within the specified range of arguments, or not is performed.

R410aTPS(p, s) := "Temperature of refrigerant R410a as a function of pressure and specific entropy"

$p \leftarrow \frac{p}{MPa}$

$s \leftarrow \frac{s}{\frac{10^3 J}{kg K}}$

error("Pressure is low") if $p < 0.35529$

error("Pressure is high") if $p > 4.4501$

$M_1 \leftarrow$

0	0.35529	0.4293	0.51458	0.61221
250	0.87106	0.87089	0.8707	0.87048
255	0.89949957	0.89933	0.89913	0.8989
260	0.927741265	0.927556958	0.92734	0.92709
265	0.955940241	0.955672495	0.955395676	0.95511
270	0.984045449	0.983729013	0.9833975	0.983054242
275	1.011405666	1.011228201	1.011014301	1.010758694
280	1.03953457	1.039315248	1.039055235	1.038749549
285	1.072145748	1.070990114	1.069811019	1.068644378
290	1.095973754	1.095681097	1.095342267	1.094952795
295	1.123958502	1.123747759	1.123486042	1.123164305
300	1.162516546	1.161149364	1.159672418	1.15810482
305	1.1831027	1.182814757	1.182466966	1.182049007

s''_0

fictitious values

....

$M_2 \leftarrow$

290	1.991	1.9655	1.9403	1.9152
295	2.0058	1.9807	1.9558	1.9311

....

$M_3 \leftarrow$

390	2.259	2.2362	2.2142	2.1928
395	2.2714	2.2486	2.2266	2.2053

....

$M_2 \leftarrow$ stack(M_2, submatrix(M_3, ORIGIN + 2, rows(M_3) − 1, ORIGIN, cols(M_3) − 1))

Fig. 11.6 Fragment of the program for creating the backward function for calculating the temperature of the refrigerant R410a in the single-phase region depending on pressure and specific entropy

In the given case there are three matrixes regions which are graphically shown in Fig. 11.5. For convenience matrixes $\mathbf{M}_2$, $\mathbf{M}_3$ are merged because they belong to one region (see Fig. 11.5).

The next part of the programming block determines the matrix which contains the input arguments. If this is a single-phase region the specific entropy which is an argument should be less than specific entropy of saturated liquid at the specified pressure. For calculation of specific entropy on saturation line two additionally created functions are used. At first saturated liquid temperature is calculated with the function **R410aTSLP(p)**, and then this value of the temperature is used to calculate saturated liquid specific entropy with the additional function **R410aSSLT(T)**.

The singe phase region of the subcooled liquid of the refrigerant is specified by matrix $\mathbf{M}_1$ (see Fig. 11.5). Using the built-in operator **submatrix** similarly as in the case of creation of the function **R410aSPT(p, T)** three matrixes are generated: a matrix of pressures $\mathbf{p}'$ which is transposed into a column, a matrix of temperatures $\mathbf{T}'$ and a matrix of specific entropy which is also transposed swapping rows and columns (see Fig. 11.7).

Next, out of the columns of the specific entropy $\mathbf{s}'$ an additional line (vector $\mathbf{s}''$) for the given pressure is generated by spline interpolation. Values of vector $\mathbf{s}''$ correspond to the values of temperatures from the vector $\mathbf{T}'$.

In this case, unlike the previous example (see Fig. 11.4), the vector of the specific entropy $\mathbf{s}''$ is the arguments vector to perform the second spline interpolation. The value of the searched temperature is generated by interpolating the values of the matrixes $\mathbf{s}''$, $\mathbf{T}'$ for the given value of specific entropy $\mathbf{s}$.

One control is applied in Fig. 11.7. The value of the second argument, the specific entropy $\mathbf{s}$, for the given first argument—pressure $\mathbf{p}$, should remain within the values of generated vector $\mathbf{s}''$. It should be higher than the first value of the

```
if  s·(10^3 J)/(kg K) < R410aSSLT(R410aTSLP(p MPa))
    M ← M1
    p' ← submatrix(M, ORIGIN, ORIGIN, ORIGIN + 1, cols(M) − 1)^T
    T' ← submatrix(M, ORIGIN + 1, rows(M) − 1, ORIGIN, ORIGIN)
    s' ← submatrix(M, ORIGIN + 1, rows(M) − 1, ORIGIN + 1, cols(M) − 1)^T
    for  i ∈ 0 .. cols(s') − 1
        s"i ← interp(cspline(p', s'<i>), p', s'<i>, p)
    R410aTPS ← "out of range"   if  s < s"0
    R410aTPS ← interp(cspline(s", T'), s", T', s) K   otherwise
```

Fig. 11.7 Fragment of the program for creating an in backward function for calculating the temperature of the refrigerant R410a in the single-phase region of subcooled liquid

```
if  s·(10^3 J)/(kg K) > R410aSSVT(R410aTSVP(p MPa))
    M ← M2
    p' ← submatrix(M, ORIGIN, ORIGIN, ORIGIN + 1, cols(M) − 1)^T
    T' ← submatrix(M, ORIGIN + 1, rows(M) − 1, ORIGIN, ORIGIN)
    s' ← submatrix(M, ORIGIN + 1, rows(M) − 1, ORIGIN + 1, cols(M) − 1)^T
    for  i ∈ 0 .. cols(s') − 1
        s"_i ← interp(cspline(p', s'^<i>), p', s'^<i>, p)
    R410aTPS ← "out of range"  if  s > s"_rows(s")−1
    R410aTPS ← interp(cspline(s", T'), s", T', s) K  otherwise
```

Fig. 11.8 Fragment of the program for creating backward function for calculating the temperature of the refrigerant R410a in the single phase region of superheated vapor

```
R410aTPS ← [R410aTSLP(p MPa) + (s·(10^3 J)/(kg K) − R410aSSLT(R410aTSLP(p MPa))) / (R410aSSVT(R410aTSVP(p MPa)) − R410aSSLT(R410aTSLP(p MPa))) · (R410aTSVP(p MPa) − R410aTSLP(p MPa))]  otherwise
```

Fig. 11.9 Fragment of the program for creating the backward function for calculating the temperature of the refrigerant R410a in the double-phase region

matrix **s″**. If this control is not performed the created function will return a value of temperature by extrapolation, not interpolation.

The last operator in the programming block shown in Fig. 11.7 returns the desired unit of the temperature—**K**.

The similar fragment for creation of the backward function for calculation of temperature of the refrigerant R410a in the single phase region of superheated vapor is demonstrated (Fig. 11.8).

In the proposed backward function **R410aTPS(p, s)** calculation of the temperature in the double-phase region is also provided. Linear interpolation is used for this purpose (see Fig. 11.9).

A copy of the web-page for online calculation and graphical illustration of the created backward function **R410aTPS(p, s)** for computing the temperature of the refrigerant R410a depending on pressure and specific entropy is shown in Fig. 11.10.

Figure 11.11 shows an example of using created functions on thermodynamic properties of refrigerant R410a for calculating in Mathcad documents the amount of heat q_{1_2}, which is transferred to the working fluid in the isobaric process 1–2. In this case a technology of "cloud" reference, that gives a possibility to refer to "cloud" functions **R410aHPT(p, T)** and **R410aTPS(p, s)**, which are located on the internet-server http://twt.mpei.ac.ru/TTHB/2/R410aEng.html (see Fig. 11.12), is proposed.

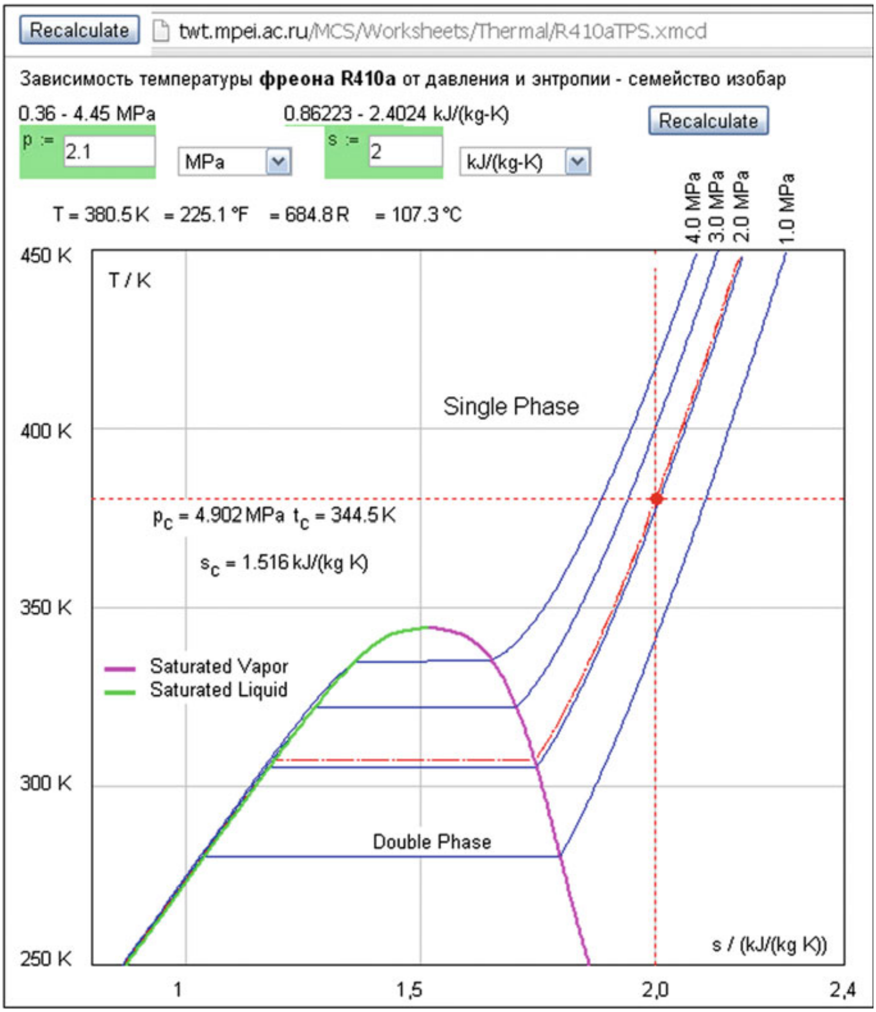


Fig. 11.10 A copy of the web-page for online calculation and graphical illustration of the created backward function **R410aTPS(p, s)**

For computing the heat energy q_{1_2} a well-known formula can be used: $\int_{s_1}^{s_2} T(p,s)ds$, where **T(p, s)**—is the backward function **R410aTPS(p, s)** for computing temperature of the refrigerant R410a depending on pressure and specific entropy. The proposed function can be directly used in integration and differentiation operations. It substantially simplifies solving tasks of computing processes in thermal engineering.

In the presented example it is possible to use the first law of thermodynamics—rate of heat transferred to the refrigerant in the isobaric process 1–2 is equal enthalpy change of the refrigerant.

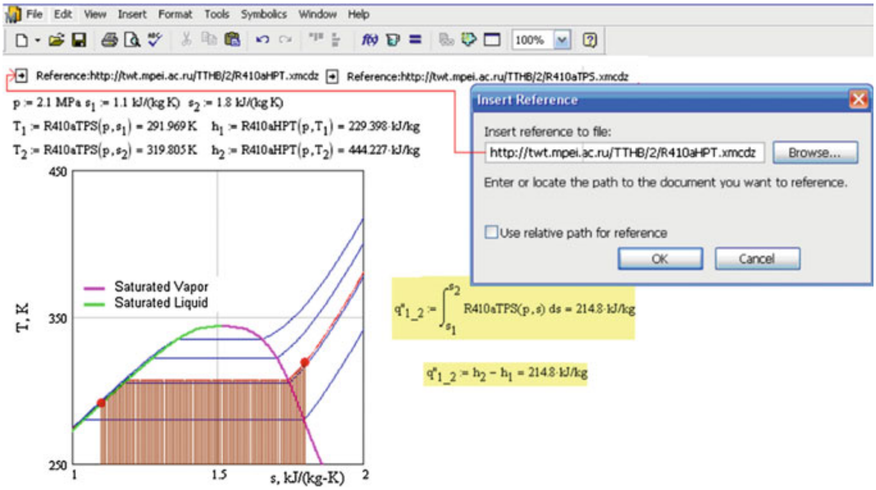


Fig. 11.11 Using of the created functions on thermodynamic properties of the refrigerant R410a for computing heat transferred to the refrigerant in isobaric process 1–2 with the help of reference to "clouds"

Fig. 11.12 The web-page of "cloud" functions on thermophysical properties of refrigerant R410a on saturated lines and single-phase regions

A graphical illustration of the isobaric process 1–2 in *T*, *s*-phase diagram is also shown in Fig. 11.1. The highlighted area is the amount of heat transferred to the refrigerant in the process.

Figure 11.12 shows a copy of the web-page on "cloud" functions on thermophysical properties of refrigerant R410a on saturated lines and single-phase regions with options of online computing, download and reference.

Chapter 12
Animation of Thermal Processes

Valery Ochkov

Abstract This chapter shows how to create animated clips illustrating various thermal processes.

At the website of the author of this book at http://twt.mpei.ac.ru/MCS/worksheets/PTU/h-s-ExpEng.xmcd, one may find the online calculation of steam expansion process in the turbine replicated on *h*, *s*-diagram (Fig. 12.1).

Opening the calculation shown in Fig. 12.1, one could vary, for example, the pressure at the turbine outlet (condenser pressure) and see how the steam expansion "drops", whether the end point is in the dry steam single-phase area (as shown in Fig. 12.2) or it falls into the two-phase wet steam area (see Fig. 12.1), and how the steam parameters get modified at the end of the expansion process, etc. One could define the final steam pressure in small "steps" from the initial pressure and "revive", animate the process. The Internet Explorer (namely, it is displayed in Fig. 12.1) stores in its buffer pictures of the process, and they can be quickly displayed on the screen by clicking on the arrows Forward and Backward (see in the upper left corner in Fig. 12.1). This will be a kind of pseudo-animation, since the frames of such animation are changing rather slowly. The real animation is achieved with the frame change rate of no less than 10 frames/s.

In Mathcad 15 environment there are tools available to create a "real" computer animation using a previously created and debugged calculation. They are shown in Fig. 12.2.

The site of the chapter: https://www.ptcusercommunity.com/message/423031.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_12

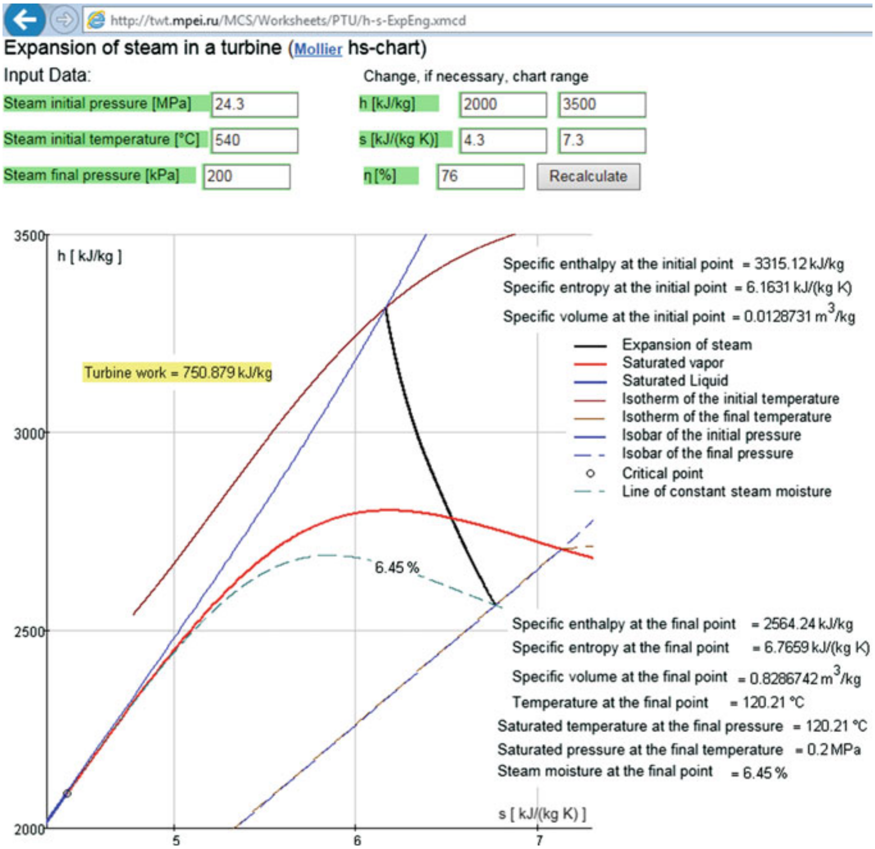


Fig. 12.1 Interactive open network calculation process of steam expansion in the turbine

Figure 12.2 shows the calculation process of steam expansion in the turbine, which is open not via the Internet (see Fig. 12.1), but directly in Mathcad 15 environment. In this calculation process, the operator intended for assigning pressure values in the end point $\mathbf{p_k}$ is replaced with a formular in which appears the system variable **FRAME**. If this variable is equal to zero, the steam pressure at the end point is equal to the condenser pressure (100 kPa). If **FRAME** variable takes the value 999, the steam pressure is equal to 24.3 MPa. Smooth change of the value of **FRAME** variable from 999 to 0 (in Fig. 12.2 it is equal to 99) makes it possible to create an animation and see how the steam is expanded in a turbine. The **FRAME** variable can be changed not only by Mathcad users—it can be automatically changed Mathcad itself. To do this it is necessary and sufficient to call a dialog box **Record Animation** of the eponymous command in **Tools** menu—see in the bottom of Fig. 12.2.

Before that, one should delete or disable the operator of manual setting of the variable **FRAME := 99**, which was used to manually debug the program—form individual animation frames. In the dialog box **Record Animation** specify the initial (999) and the end (0) values of the variable **FRAME** (we count the frames in a reverse order from 999 to 0) and the animation display rate (24 frames/s). In addition, you need to broach the computer mouse to select an animation area. It is limited by a dotted rectangle, as shown in Fig. 12.2. After that you should click **Animate**. Now wait a little bit (2–3 min) until Mathcad itself starts to change the value of **FRAME** variable, memorize the animation files and "put" them in the avi. "container". At the end of this process, the animation window will be displayed, two frames of which are shown in Fig. 12.3.

Animations created in this way could be saved in an avi. file (see the key **Save As …** in the dialog box in Fig. 12.2) and run without Mathcad. Animation could also be placed in various forums, such as Mathcad's user forum—PTC Community (Fig. 12.4).

The author also collects other animations of thermal processes on the forum PTC Community. Thus, in Fig. 12.5, you can see an animation of the processes occurring in the steam turbine cycle with steam boiler (economizer, a drum with downcomers and risers), steam turbine, condenser and feed pump.

The lower right corner of Fig. 12.5 includes references (**More Like This**) to the animation of the other thermal cycles: Otto cycle (Fig. 12.6) and Diesel cycle (Fig. 12.7). When creating this animation, a Mathcad opportunity was used to change the image (image of phase of the internal combustion engine cycle) in the calculation.

Animation in Mathcad environment can be used not only for visualization of various processes, but also for simple calculations. One disadvantage of Mathcad is that it cannot generate so-called executable files, for example, exe. files. Because of this, calculations created in Mathcad can only be run when it is installed on the computer. One solution to this problem is the Mathcad Calculation Server, which we have repeatedly mentioned. Another partial solution to this problem may be … animation. If our calculation depends only on one variable, we can link this variable to the system variable **FRAME**, create an appropriate animation, and then run it outside Mathcad using some video player, move the animation slider, set the value of the argument and read the value of the function. Figure 12.8 shows a frame of animation linking the temperature to the saturation pressure of water and steam. One can open the animation using the slider shown in the bottom of the figure, set the temperature with increment of 0.5 °C (K) and calculate the saturation pressure. The other pairs of steam and water thermal properties at saturation line can be added into this animation, such as water and steam density, specific enthalpy and entropy, thermal conductivity, viscosity, etc.

Another aspect of the use of animation: Fig. 12.9 shows the frame of the animation of R407c isotherm motion in coordinates "specific entropy—pressure" at a fixed pressure and varying temperature. Considering the behavior of the moving curve one can to some extent judge on the quality of the function created using the principle "Nature does not like sharp corners and… oscillation" (see Fig. 1.8 in Chap. 1).

Figures 12.10 and 12.11 shows a pair of frames of the animation for scanning isolines using one argument of a function that returns the value of the coefficient of friction f of the fluid in a circular tube, depending on the Reynolds number **Re** and

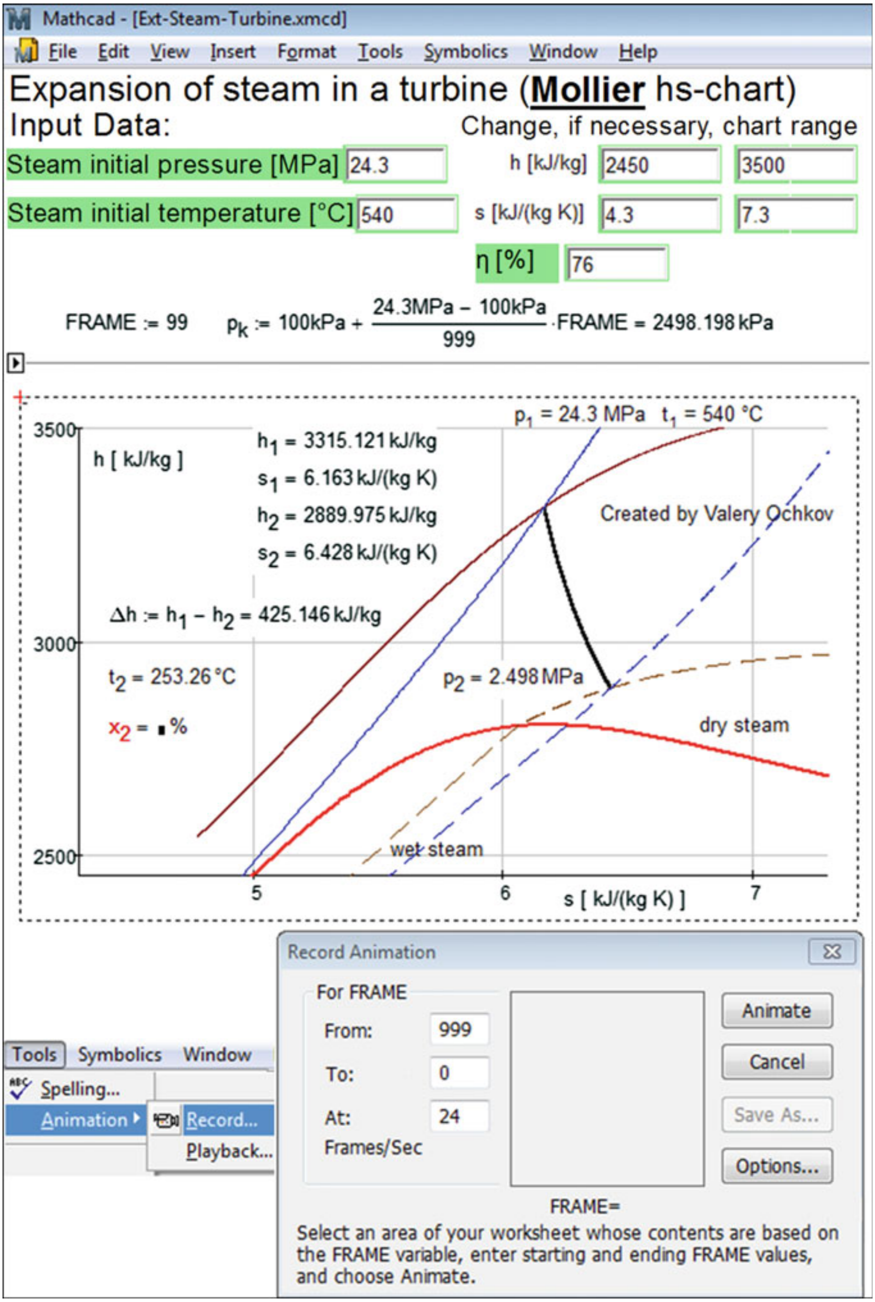


Fig. 12.2 Creating of animation in Mathcad 15 environment

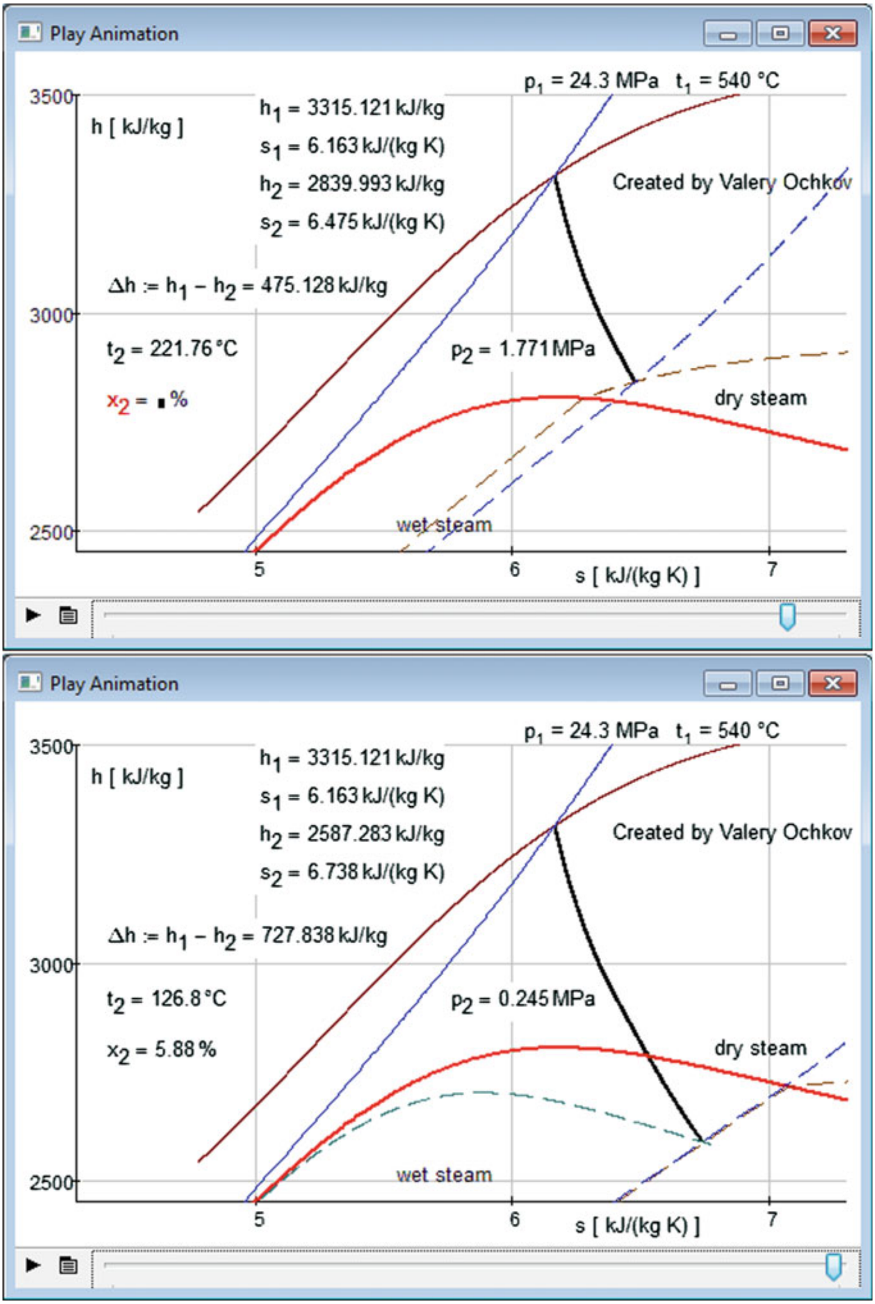


Fig. 12.3 Mathcad animation frames

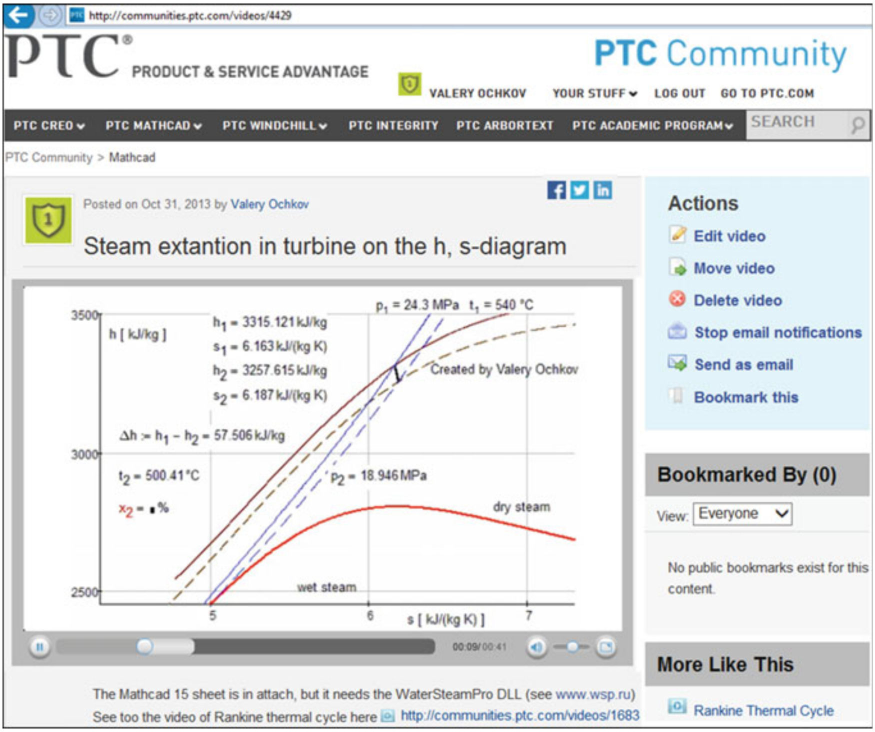


Fig. 12.4 Turbine steam expansion animation at PTC community website

the liquid flow and the relative roughness of the tube inner surface **Δ**. This problem will be discussed in detail in Chaps. 16 (see Fig. 16.7) and 17. The function **f (Re, Δ)** was created using the double interpolation method described in Chap. 1 (see Figs. 1.1 and 1.2). So, if you use two spline interpolations, the function **f (Re, Δ)** is obtained with oscillation (see Figs. 12.10 and 1.8). If a spline interpolation is replaced with a linear interpolation, the function **f (Re, Δ)** gets pretty smooth (see Fig. 12.11). Without animation this error couldn't have been noticed.

Divertisment English blacksmith Thomas Newcomen, in 1705, built a steam engine, which was used to pump water from mines—see the author's Mathcad-animation of this machine at https://www.ptcusercommunity.com/videos/2134. A legend is connected with this machine. A boy Humphrey Potter in 1712 got the job of operator of such a steam-driven pump. To implement the working cycle of the machine, the boy had to open the steam supply valve when the piston was in its lower position, and the water supply valve—at fully lifted piston. But this genius boy figured out how to rid yourself of the repetitive work. Using sticks and rope he coupled the handles of the valves with the stem of the steam engine.

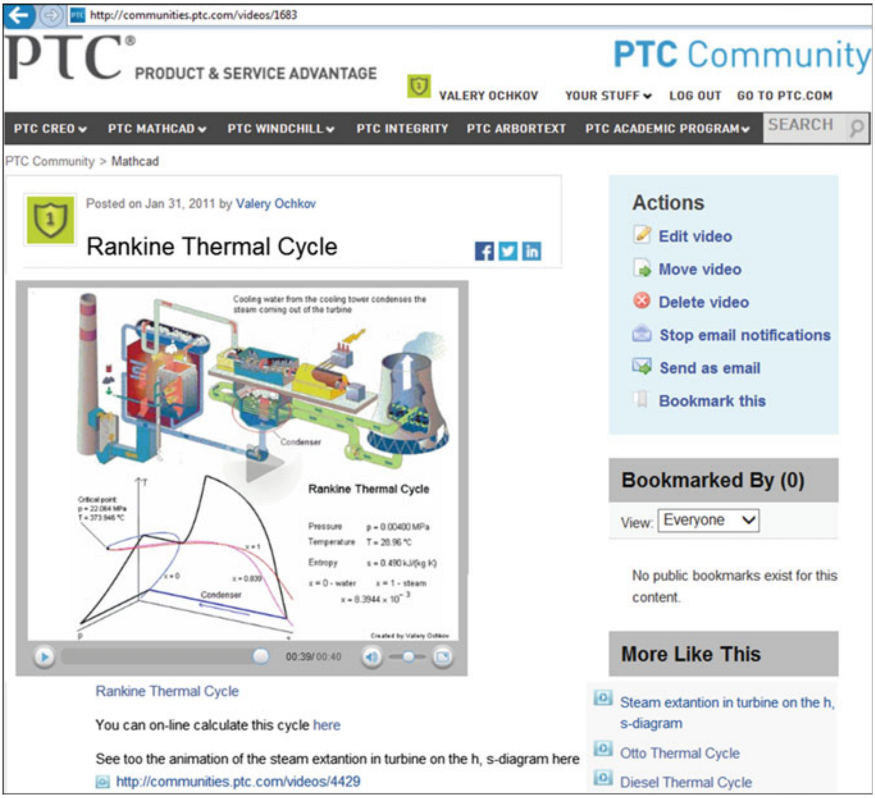


Fig. 12.5 Steam turbine cycle work animation

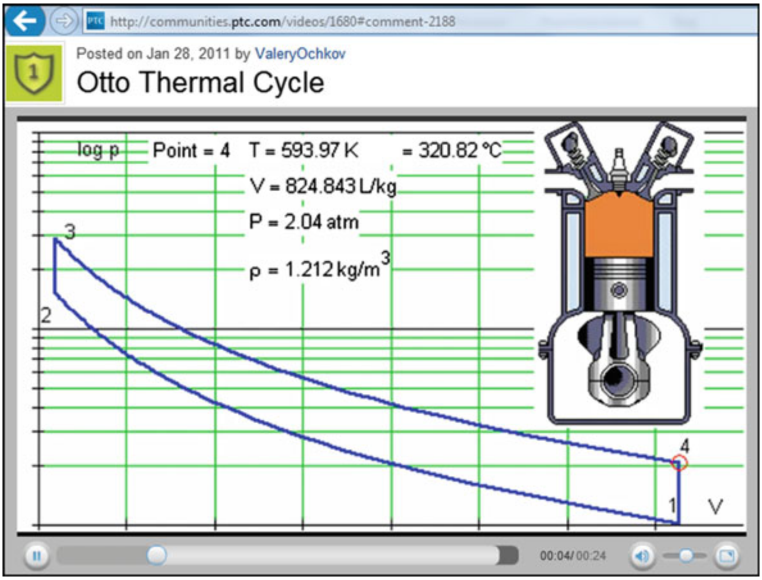


Fig. 12.6 Otto cycle work animation

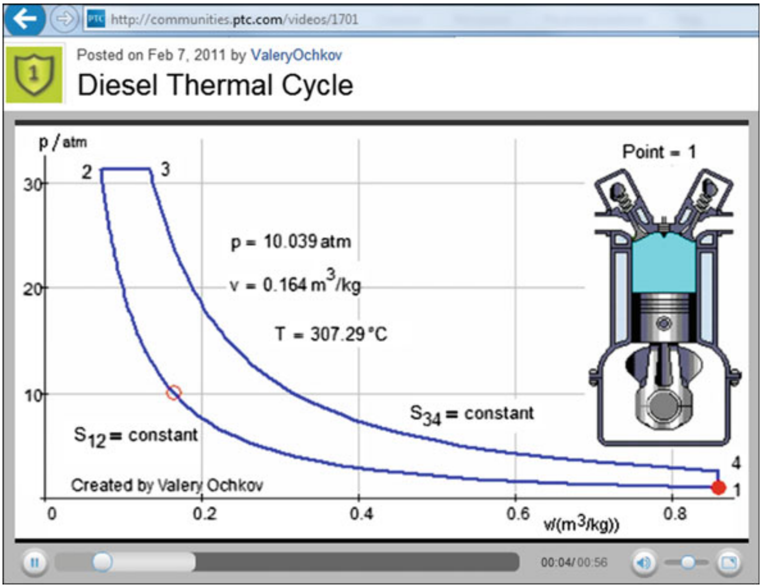


Fig. 12.7 Diesel cycle work animation

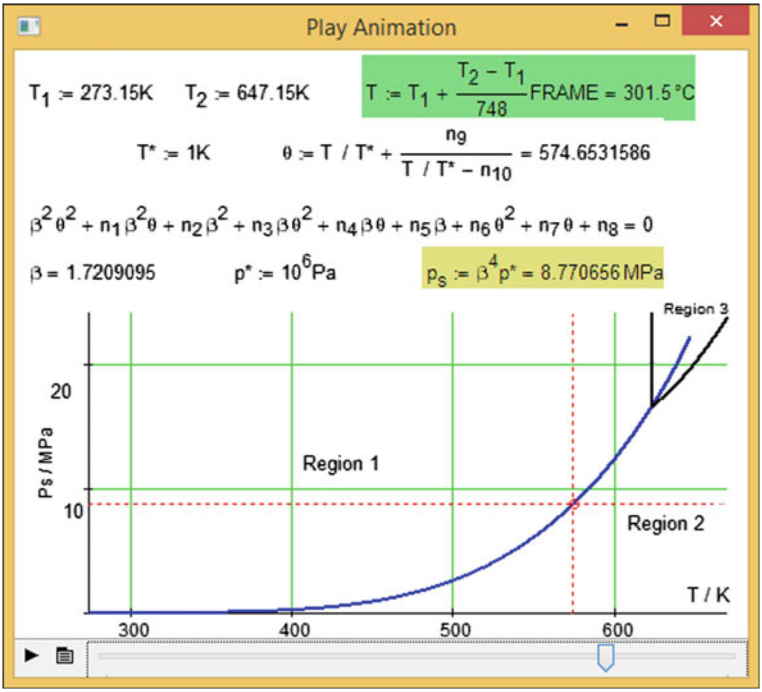


Fig. 12.8 Animation as exe-file

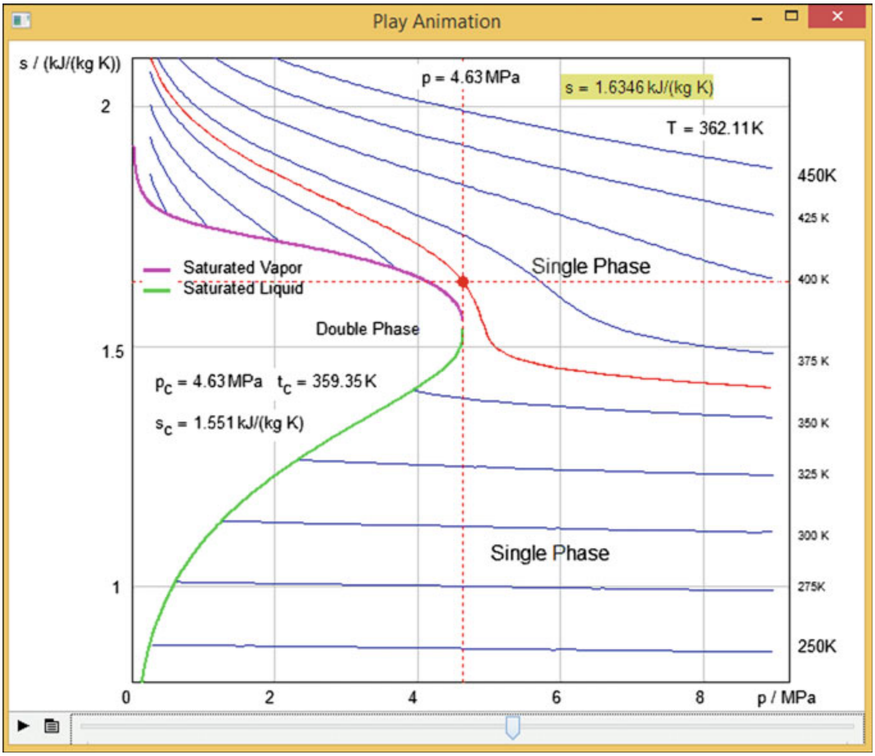


Fig. 12.9 Scanning with animation of a function created using the properties of substances

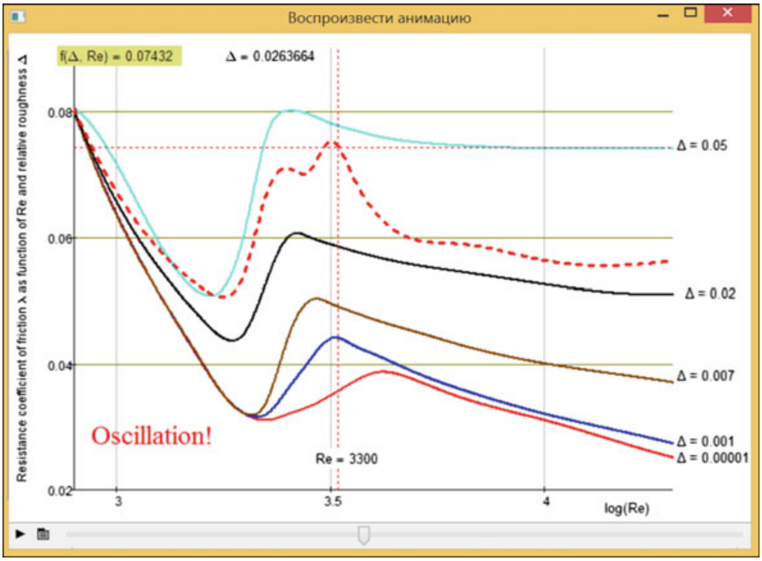


Fig. 12.10 Scanning of the created function: spline oscillation

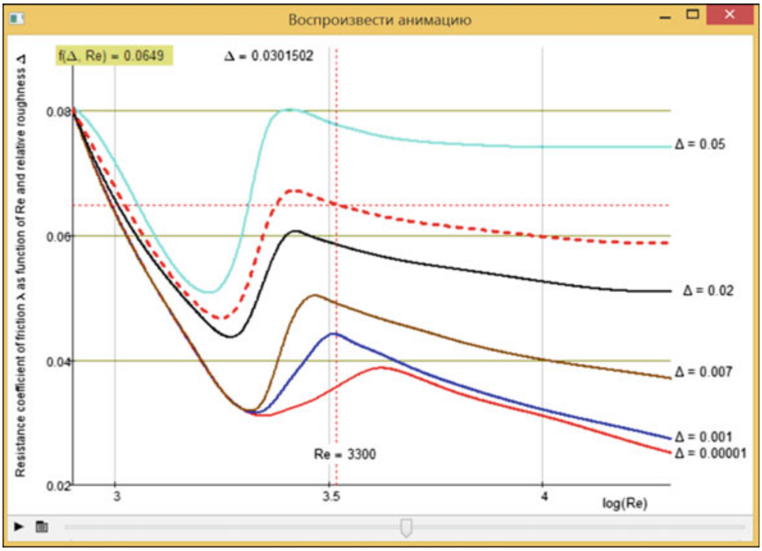


Fig. 12.11 Scanning of the created function: everything is OK

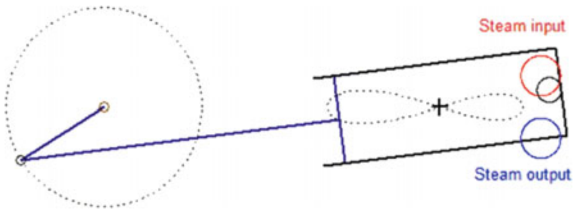


Fig. 12.12 Frame of the animation of the steam engine with oscillating cylinder

It turned out some sort of rocker mechanism. The valves were controlled by a piston itself, and the boy was able to play with his friends. Thus was invented the slide valve of steam engine—an automatic steam distribution device.

Now one could only see a steam engine hardware in a museum. On the Internet, one could find a lot of animations of steam engine alone or mounted on a steam train or boat. It is quite realistic to create an animation of the steam engine using a mathematical program Mathcad [57], and many heat engineering students take interest in this.

Figure 12.12 shows the animation frame of a rather unusual steam engine with oscillating cylinder. The animation itself is placed on the Planet PTC website https://www.ptcusercommunity.com/videos/3033. There one could download a Mathcad-calculation.

Chapter 13
Calculation of Gas Turbine Engine Cycle

Volodymyr Voloshchuk

Abstract This chapter describes in detail the process of calculating energy characteristics of the gas turbine cycle (Brayton cycle) with modeling of the composition and thermophysical properties of air, natural gas and products of combustion along the flow in gas turbine.

For actual thermal power plants the most comprehensive measure of their performance is the overall efficiency that considers all losses.

It can be shown that the thermodynamic cycle, which in the given conditions has the largest internal absolute efficiency of the cycle η_i, provides the highest overall efficiency of the plant. It gives a possibility for actual, internally irreversible, cycles of thermal power plants that produce electricity to accept as the main measure of their thermodynamic efficiency a parameter which is expressed by following relationship [81, 82]

$$\eta_i = \frac{l_{gte\_act}}{q_{1\_act}}, \tag{13.1}$$

where l_{gte_act}—net specific work output of the actual gas turbine engine cycle; q_{1_act}—actual specific heat addition from outside to the cycle.

Let's analyze influence of working fluid parameters (temperature and pressure) on the internal absolute efficiency of the cycle of simple gas turbine engine with continuous combustion of fuel at constant pressure sketch of which is shown in Fig. 13.1.

The site of the chapter: https://www.ptcusercommunity.com/message/422997.

V. Voloshchuk (✉)
National University of Water Management and Natural Resources, Rivne, Ukraine
e-mail: Vl.volodya@gmail.com

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_13

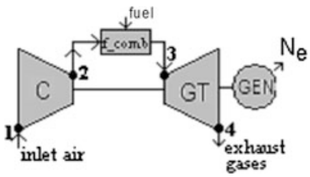


Fig. 13.1 Sketch of the simple gas turbine engine with continuous combustion of fuel at constant pressure: C—compressor; F_Comb—fuel combustor; GT—gas turbine; GEN—electric generator

It is supposed that processes of air compression and gas expansion are irreversible and can be evaluated by isentropic efficiencies.

The internal absolute efficiency of the cycle

$$\eta_i = \frac{l_{gte\_act}}{q_{1\_act}} = \frac{l_{gt\_act} - l_{c\_act}}{q_{1\_act}} = \frac{(h_3 - h_{4\_act}) - (h_{2\_act} - h_1)}{h_3 - h_{2\_act}} \quad (13.2)$$

Turbine efficiency:

$$\eta_{is\_gt} = \frac{l_{gt\_act}}{l_{gt}} = \frac{h_3 - h_{4\_act}}{h_3 - h_4}.$$

Compressor efficiency:

$$\eta_{is\_c} = \frac{l_c}{l_{c\_act}} = \frac{h_2 - h_1}{h_{2\_act} - h_1}.$$

Then

$$\eta_i = \frac{(h_3 - h_4) \cdot \eta_{is\_gt} - \frac{(h_2 - h_1)}{\eta_{is\_c}}}{h_3 - h_{2\_act}}, \quad (13.3)$$

where

$$h_{2\_act} = h_1 + \frac{(h_2 - h_1)}{\eta_{is\_c}}.$$

In the cycle of gas turbine engine work input required for compression of air is quite large and takes a significant share of the work output from the gas turbine. Therefore, in analysis of the internal absolute cycle efficiency the ratio of these values ($\phi = l_c/l_{gt}$) plays an important role.

Using this ratio the net specific work output of the actual gas turbine engine cycle can be expressed as

$$l_{gte\_act} = l_{gt} \cdot \left(\eta_{is\_gt} - \frac{\phi}{\eta_{is\_c}} \right).$$

As a result it can be seen that gas turbine engine has network output ($l_{gte\_act} > 0$) only if $(\eta_{is\_gt} - \phi/\eta_{is\_c}) > 0$ or $\eta_{is\_gt} \cdot \eta_{is\_c} > \phi$. It means that a quality of both compressors and gas turbines which is characterized by their efficiencies plays an important role for such engines.

For further analysis it may be expressed as

$$\eta_i = \frac{l_{gte\_act}}{q_{1\_act}} = \frac{l_{gt\_act} - l_{c\_act}}{l_{gt} - l_c} \cdot \frac{l_{gt} - l_c}{q_1} \cdot \frac{q_1}{q_{1\_act}} = \eta_{i\_gte} \cdot \eta_t \cdot \frac{q_1}{q_{1\_act}}. \tag{13.4}$$

Here a thermal efficiency of reversible cycle is

$$\eta_t = \frac{q_1 - q_2}{q_1} = \frac{l_{gt} - l_c}{q_1},$$

and internal relative efficiency of the turbine-compressor set is

$$\eta_{i\_gte} = \frac{l_{gt} \cdot \left(\eta_{is\_gt} - \frac{\phi}{\eta_{is\_c}} \right)}{l_{gt} \cdot (1 - \phi)} = \frac{\eta_{is\_gt} - \frac{\phi}{\eta_{is\_c}}}{(1 - \phi)}. \tag{13.5}$$

Each of the terms in Eq. (13.5) has its own dependence on the parameters of the cycle. Therefore, internal relative efficiency of the actual cycle η_{i_gte} is greatly complicated by many conditions.

Calculations of different types of gas turbine engines are located on the web-site http://twt.mpei.ru/ochkov/VPU_Book_New/mas/index.html. In the given study approaches which were used in creation of programs for computing thermodynamic cycles of gas turbine engines will be analyzed.

The calculated scheme of simple open gas turbine engine consisting on the compressor, fuel combustor and gas turbine is shown in Fig. 13.1.

For determining thermodynamic parameters of working fluids **WaterSteamPro** (www.wsp.ru) was used.

The calculation begins from interactive procedure of data input. The number of entered parameters can be different. It depends on how detailed calculation should be. An example of input data block, which is prepared on the technology of Mathcad Calculation Server, is shown in Fig. 13.2. Such set of data enables to conduct rather detailed thermodynamic calculation of the given gas turbine engine cycle. In the lower part of the Fig. 13.2 operators of interaction with a user during creation of Internet MathCad documents are presented.

For convenience acceptable units are chosen (for example, kilojoule per kilogram of gas—kJ/(kg g)) (see Fig. 13.3). Additional procedures during calculation (for example, procedure of unit selection, reference to the WaterSteamPro and so on)

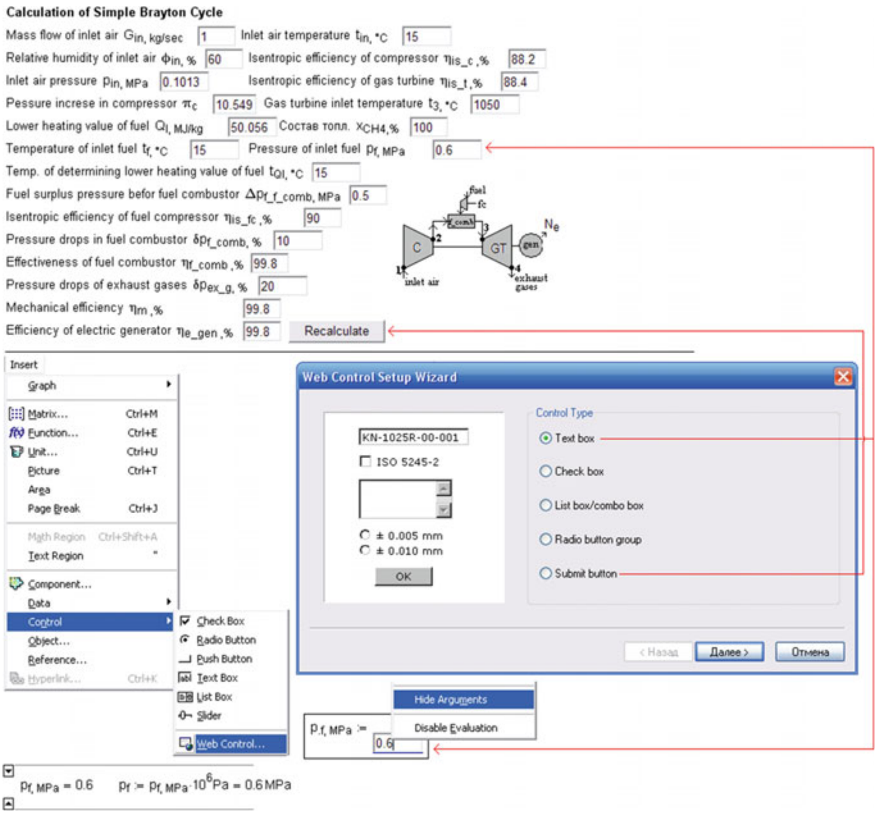


Fig. 13.2 Input data block of the Internet document prepared on technology of Mathcad Calculation Server

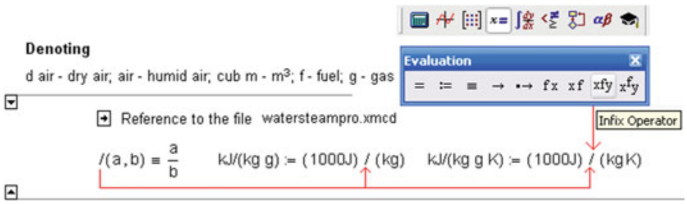


Fig. 13.3 The procedure of units choosing

can be hidden in Areas (**Collapse-Expand**) (see Fig. 13.3). Explanations to the reductions can be included separately.

One of the first task of the given calculation is a task of determining inlet air properties which in different parts of such type of engine are different. Air goes through the compressor, a mixture of combustion products and substantial excess air for reducing the combustion temperature to well below because of the metallurgical limits on the gas turbine operating temperature go through the fuel combustor and the gas turbine.

The compressor is the first element of the calculation which operates with moist air. At first humidity should be determined. Relative humidity at the specified temperature is the input data for this. Marginal pressure of water vapor of inlet air is calculated. At temperatures lower the triple point (0.01 °C) this pressure is computed as sublimation pressure, at temperatures higher 0.01 °C—as saturation pressure (see Fig. 13.4). These values of pressure are calculated with the help of **WaterSteamPro** functions—**wspPSUBT(T)** and **wspPST(T)** respectively.

Then mass and molecular humidity ratio are calculated on the base of thermodynamic equations (see Fig. 13.4). The calculation mixture of the compressor inlet

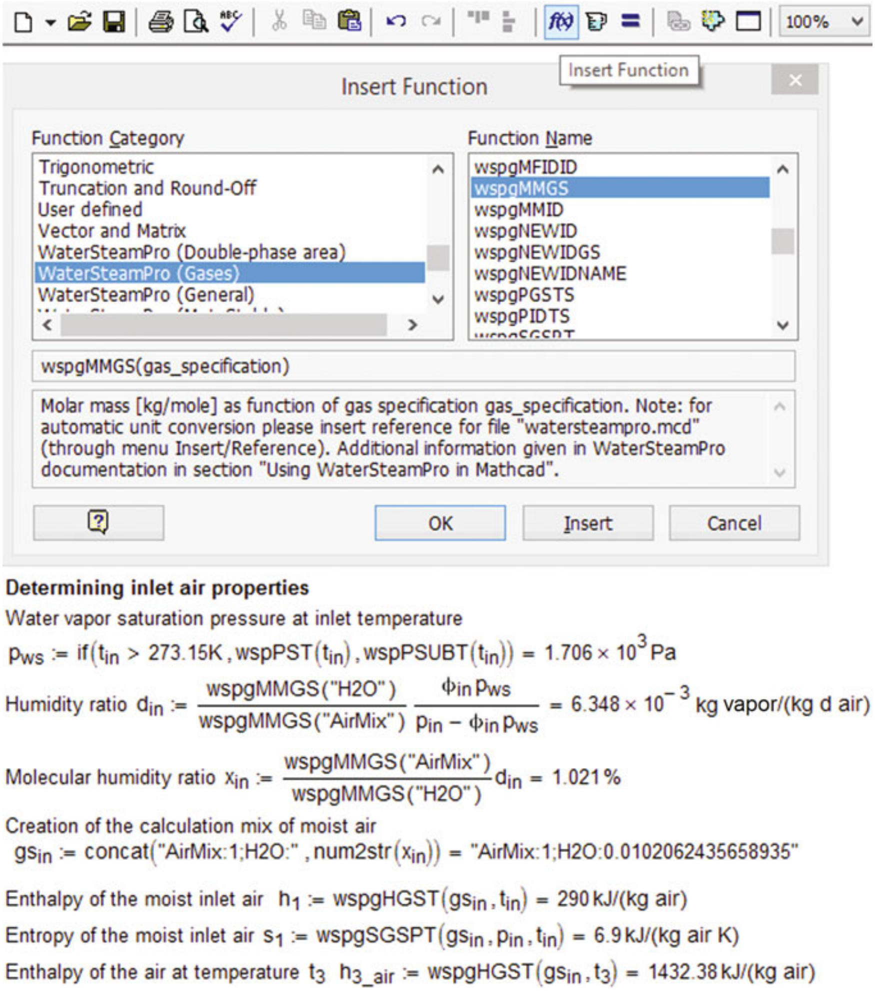


Fig. 13.4 Creation of the calculation mixture of the working fluid in the compressor inlet

temperature can be derived from knowing humidity ratio of the air. For this purpose a buil-in MathCad function **concat(s_1, s_2,…)** (returning the string formed by concatenating strings s_1, s_2, and so on) and **num2str(x)** (returning the number x to a string) are used. The result string (in **WaterSteamPro** this is a gas specification) is used for calculating thermodynamic properties of working fluid in the compressor. In Fig. 13.4 with the help of a buil-in MathCad function **wspgHGST(gs, T**), where gs—gas specification, and T—temperature, specific enthalpy of the moist inlet air is calculated.

Functions in **WaterSteamPro** for calculating properties of ideals gases and ideal mixtures have the prefix wspg, then calculated parameter (in the given case H—enthalpy) and a list of arguments (gas specification GS and temperature T) are indicated.

Both chemical denotation of single gases (CO_2, H_2, O_2 and so on) and their mixtures can be used as gas specification indication. Mixtures specifying is set by rather simple method. Components of mixture are separated with semicolon and the amount of the component in the mixture is pointed out after the component indication followed by colons. The number of moles or mass but not fractions should be used in the amount specifying. In case of default the value means mole but if the denotation “M” is used mass units are applied. So, the indication “AirMix:1; H_2O: 0.0102062435658935” means that one mole of dry air “AirMix” and 0.0102062435658935 moles of water vapor are selected.

After creation of the mixture of the working fluid calculation of the compression is performed. For this purpose, using **WaterSteamPro** functions, thermodynamic parameters of the air at inlet and outlet of the compressor are computed. Isentropic and actual works for the air compression are also determined (see Fig. 13.5).

For calculation of thermodynamic properties of moist air the following functions of **WaterSteamPro** are used:

- **wspgTGSPS(gs, p, s)**—temperature as function of gas specification gs, pressure **p**, specific entropy s;
- **wspgHGST(gs, T)**—specific enthalpy as function of gas specification gs, pressure **p**, temperature **T**;

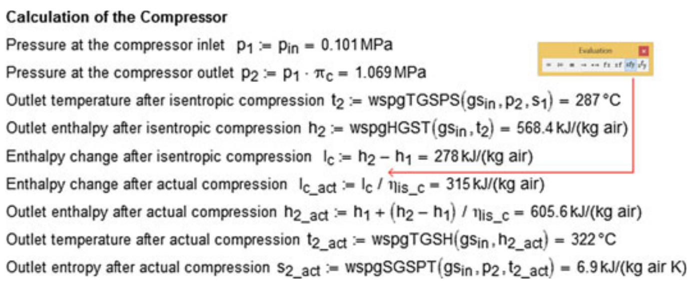


Fig. 13.5 Calculation of the compressor

- **wspgTGSH(gs, h)**—temperature as function of gas specification gs, specific enthalpy h;
- **wspgSGSPT(gs, p, T)**—specific entropy as function of gas specification gs, pressure **p**, temperature **T**.

Gaseous or liquid fuels can be used for burning in gas turbine engines. To ensure stability of the combustion process pressure of the fuel at the fuel combustor inlet should be 0.3–0.5 MPa higher than the maximum pressure of the air directed from the compressor to the fuel combustor. If this requirement is not followed a booster fuel compressor should be used.

When calculating the fuel combustor it is convenient to use natural gas consisting of pure methane (CH_4 = 100 %). It simplifies comparison of different types of gas turbine engines during calculation. The primary document for calculating thermodynamic properties of methane is available in the Internet on the web-page http://twt.mpei.ac.ru/orlov/gases/methan_functions.mcd. The given file can be downloaded and inserted into user's document (see Fig. 13.6) or it is possible to do a reference to it from the Mathcad worksheet.

Using function on thermodynamic properties of methane (see Fig. 13.6) it is possible to calculate the fuel (booster) compressor, namely, work for compression and inlet/outlet enthalpy (see Fig. 13.7).

Calculation of the fuel combustor is started from determining theoretical amount of air (volume, mass) necessary for burning one unit of fuel (for the gas it is 1 m^3 under standard conditions)—the air-fuel ratio. Theoretical or stoichiometric reaction is used for this purpose. Again, it is convenient to apply **WaterSteamPro** functions for evaluating the amount of oxidant (oxygen) contained in the moist air (Fig. 13.8).

On the base of the gas specification of the moist air mixture (see Fig. 13.4) the **WaterSteamPro** function **wspgVFGSGS(gs1, gs2)** calculates percentage of the

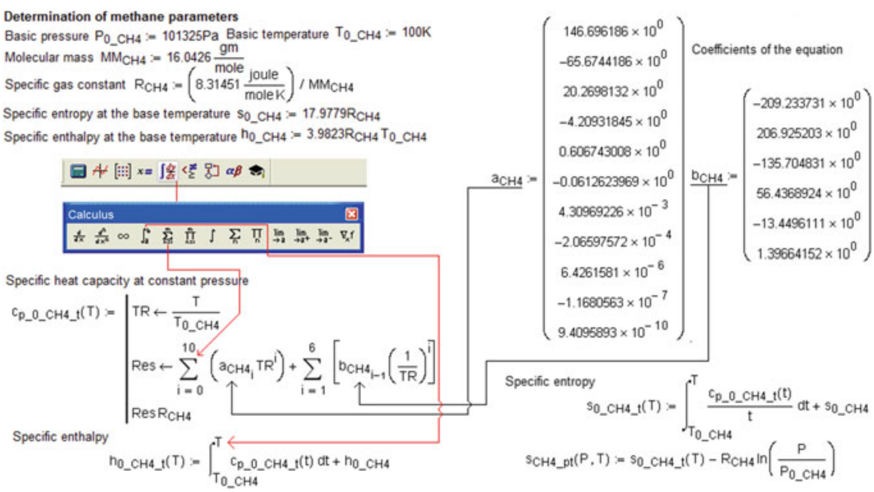


Fig. 13.6 A fragment of calculating thermodynamic properties of methane

Calculation of the Fuel Compressor (FC)

Enthalpy at the fuel compressor inlet $h_{in\_fc} := h_{0\_CH4\_t}(t_f) = 602\ \text{kJ/(kg f)}$

Entropy at the fuel compressor inlet $s_{in\_fc} := s_{CH4\_pt}(p_f, t_f) = 10.6\ \text{kJ/(kg f K)}$

Pressure at the fuel compressor outlet $p_{out\_fc} := p_2 + \Delta p_{f\_f\_comb} = 1.569\ \text{MPa}$

Temperature at the fuel compressor outlet after isentropic compression $t_{out\_fc} := t_{CH4\_ps}(p_{out\_fc}, s_{in\_fc}) = 85\ °\text{C}$

Enthalpy at the fuel compressor outlet after isentropic compression $h_{out\_fc} := h_{0\_CH4\_t}(t_{out\_fc}) = 763\ \text{kJ/(kg f)}$

Enhtalpy change in the fuel compressor after isentropic compression $l_{fc} := h_{out\_fc} - h_{in\_fc} = 161\ \text{kJ/(kg f)}$

Enhtalpy change in the fuel compressor after actual compression $l_{fc\_a} := l_{fc} / \eta_{is\_fc} = 178\ \text{kJ/(kg f)}$

Enthalpy at the fuel compressor outlet after actual compression $h_{out\_fc\_a} := h_{in\_fc} + l_{fc\_a} = 781\ \text{kJ/(kg f)}$

Temperature at the fuel compressor outlet after actual compression $t_{out\_fc\_a} := t_{CH4\_h0}(h_{out\_fc\_a}) = 92\ °\text{C}$

Fig. 13.7 A fragment of calculating fuel (booster) compressor

Calculation of the Fuel Combustor (F_Comb)

Persentage of oxygen in the inlet air $x_{O2} := \text{wspgVFGSGS}(gs_{in}, \text{"O2"}) = 20.778\ \%$

Persentage of hydrogen in the inlet air $x_{H2} := \text{wspgVFGSGS}(gs_{in}, \text{"H2"}) = 9.9 \times 10^{-3}\ \%$

Persentage of water wapor in the inlet air $x_{H2O} := \text{wspgVFGSGS}(gs_{in}, \text{"H2O"}) = 1.01\ \%$

Theoretical volume of the air required for stoichiometric combustion of a unit volume of the fuel

$$V_0 := \frac{2x_{CH4}}{x_{O2} - 0.5x_{H2}} = 9.628\ \text{cub m/cub m}$$

Theoretical nitrogen volume in the products from the combustion per cubic meter of the fuel
$V_{0\_N2} := \text{wspgVFGSGS}(gs_{in}, \text{"N2"})\, V_0 = 7.437\ \text{cub m/cub m}$

Theoretical CO2 volume in the products from the combustion per cubic meter of the fuel
$V_{0\_CO2} := \text{wspgVFGSGS}(gs_{in}, \text{"CO2"})\, V_0 + x_{CH4} = 1.0029\ \text{cub m/cub m}$

Theoretical H2O volume in the products from the combustion per cubic meter of the fuel
$V_{0\_H2O} := x_{H2} V_0 + 2x_{CH4} + x_{H2O} V_0 = 2.098\ \text{cub m/cub m}$

Theoretical argon volume in the products from the combustion per cubic meter of the fuel
$V_{0\_Ar} := \text{wspgVFGSGS}(gs_{in}, \text{"Ar"})\, V_0 = 0.0896\ \text{cub m/cub m}$

Theoretical mass of air required for stoichiometric combustion of a unit mass of the fuel

$$L_0 := V_0 \frac{\text{wspgMMGS}(gs_{in})}{MM_{CH4}} = 17.316\ \text{kg air/(kg f)}$$

Fig. 13.8 A fragment of calculating the fuel combustor

component gs2 in the mixture gs1. Oxygen O_2 is a component and moist air is a mixture. So, the amount of moist air required for burning 1 m^3 of fuel is evaluated. Then, using coefficients of stoichiometric combustion reaction theoretical volumes of gases (components of combustion products) are calculated. The MathCad functions **concat(s_1, s_2,…)** and **num2str(x)** create the combustion products mixture (see Fig. 13.9).

The required temperature of the working fluid at gas turbine inlet should be provided in the fuel combustor. Combustion with excess air, i.e. more air than is required for stoichiometric reaction and calculated above (see Fig. 13.8), is applied for this purpose.

Creation of the combustion products mixture

$gs_0 := \text{concat}(\text{"N2:"}, \text{num2str}(V_{0\_N2}), \text{";CO2:"}, \text{num2str}(V_{0\_CO2}), \text{";H2O:"}, \text{num2str}(V_{0\_H2O}), \text{";Ar:"}, \text{num2str}(V_{0\_Ar}))$

$gs_0 =$ "N2:7.43674052894925;CO2:1.00285918513224;H2O:2.09822486124273;Ar:0.0895878008101024"

Enthalpy of the combustion products at temperature t_3 $h_{3comb\_p} := \text{wspgHGST}(gs_0, t_3) = 1609.1\ \text{kJ/(kg g)}$

Fig. 13.9 Creation of the combustion products mixture

Comparing Enthalpies with the Temperature of Determining Q_l

$h_{1\_Ql} := h_1 - \text{wspgHGST}(gs_{in}, t_{Ql}) = 0\ \text{kJ/(kg air)}$

$h_{2\_act\_Ql} := h_{2\_act} - \text{wspgHGST}(gs_{in}, t_{Ql}) = 315.481\ \text{kJ/(kg air)}$

$h_{3comb\_p\_Ql} := h_{3comb\_p} - \text{wspgHGST}(gs_0, t_{Ql}) = 1296.2\ \text{kJ/(kg g)}$

$h_{out\_fc\_a\_Ql} := h_{out\_fc\_a} - h_{0\_CH4\_t}(t_{Ql}) = 178.417\ \text{kJ/(kg f)}$

$h_{in\_fc\_Ql} := h_{in\_fc} - h_{0\_CH4\_t}(t_{Ql}) = 0\ \text{kJ/(kg f)}$

$h_{3\_air\_Ql} := h_{3\_air} - \text{wspgHGST}(gs_{in}, t_{Ql}) = 1142.285\ \text{kJ/(kg air)}$

Fig. 13.10 Comparing enthalpies of working fluids with temperature of determining lower heating value of fuel

Knowing values of enthalpies of all flows entering and leaving the fuel combustor it is possible to apply a heat balance in evaluating air excess for providing the required temperature at gas turbine inlet. It should be taken into account that the temperature of determining lower heating value of fuel can be different. That's why values of enthalpies of all flows entering and leaving the fuel combustor are to be compared with this temperature (see Fig. 13.10).

The total heat balance of the fuel combustor consists of heat input with mass of air, mass of fuel, energy release with combustion reaction and heat output with mass of combustion products and excessive air. The heat balance equation of the fuel combustor is presented in Fig. 13.11. Evaluation of the amount of excessive air

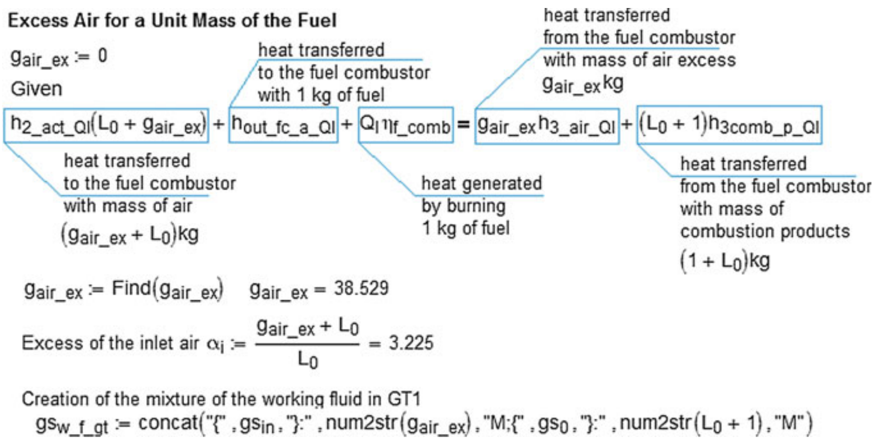


Fig. 13.11 Evaluation of the amount of excessive air and the contents of the turbine working fluid

made with the help of MathCad calculation block **Given—Find** is also shown in Fig. 13.11. The MathCad functions **concat(s1, s2,…)** and **num2str(x)** are used for creation of the turbine working fluid contents in the format of the **WaterSteamPro** gas specification (see Fig. 13.11). The working fluid in the gas turbine is a mixture of combustion products and excessive air for cooling. The components of the mixture are placed in curly braces using gas specification denoting. Then after colons the amounts of the components in kg are indicated. In the calculation shown in Fig. 13.11 a mixture of 38.529 kg of moist excessive air and combustion products consisting of 1 kg of fuel and of 17.316 kg of moist air as the oxidizer is used for creation of **WaterSteamPro** gas specification.

During calculation of the gas turbine inlet and outlet pressures should be determined. The gas turbine inlet pressure is calculated using the value of the compressor outlet pressure taking into account pressure drops in the path "compressor-fuel combustor-gas turbine inlet". The gas turbine outlet pressure is set to be equal to the atmospheric pressure minus pressure drops in output path of the engine. The next evaluation of parameters of the gas turbine is performed using **WaterSteamPro** functions (see Fig. 13.12) similarly as for the compressor but taking into account that the working fluid is expanded.

Knowing the amount of the air needed for stoichiometric burning of 1 kg of fuel, the amount of the excessive air for 1 kg of fuel it is possible to calculate mass flow rate of the fuel and the working fluid through the gas turbine (see Fig. 13.13).

The total amount of the heat transfer to the gas turbine engine from outside consists of heat transferred with mass of inlet air, inlet fuel and from burning fuel.

The results of gas turbine engine calculation may be: the compressor power, the fuel compressor power, the gas turbine power, the output power, the overall efficiency (see Fig. 13.13).

For visualizing the gas turbine engine cycle it is necessary to prepare appropriate data—vectors elements of which are parameters of working fluid in different points of the Brayton cycle. These vectors are prepared with Mathcad-operators according

Calculation of the Gas Turbine (GT)

Inlet pressure $p_3 := p_2(1 - \delta p_{f\_comb}) = 0.962\,\text{MPa}$

Inlet entropy $s_3 := wspgSGSPT(gs_{w\_f\_gt}, p_3, t_3) = 8.1\,\text{kJ/(kg g K)}$

Outlet pressure $p_4 := p_{in} / (1 - \delta p_{ex\_g}) = 0.1266\,\text{MPa}$

Outlet temperature after isentropic expansion $t_4 := wspgTGSPS(gs_{w\_f\_gt}, p_4, s_3) = 537\,°\text{C}$

Outlet enthalpy after isentropic expansion $h_4 := wspgHGST(gs_{w\_f\_gt}, t_4) = 867\,\text{kJ/(kg g)}$

Enhtalpy change after isentropic expansion $l_{gt} := h_3 - h_4 = 622\,\text{kJ/(kg g)}$

Enhtalpy change after actual expansion $l_{gt\_act} := l_{gt}\,\eta_{is\_t} = 550\,\text{kJ/(kg g)}$

Outlet enthalpy after actual expansion $h_{4\_act} := h_3 - l_{gt\_act} = 940\,\text{kJ/(kg g)}$

Outlet temperature after actual expansion $t_{4\_act} := wspgTGSH(gs_{w\_f\_gt}, h_{4\_act}) = 599\,°\text{C}$

Outlet entropy after actual expansion $s_{4\_act} := wspgSGSPT(gs_{w\_f\_gt}, p_4, t_{4\_act}) = 8.1\,\text{kJ/(kg g K)}$

Fig. 13.12 Calculation of the gas turbine

Fig. 13.13 Result characteristics of the Brayton cycle

Result Characteristics of the Brayton Cycle

Mass flow rate of the working fluid through the GT

$$G_{gt} := \left(\frac{g_{air\_ex} + L_0 + 1}{g_{air\_ex} + L_0}\right) G_{in} = 1.018\ \text{kg g/sec}$$

Fuel mass flow rate $$B_f := \left(\frac{1}{g_{air\_ex} + L_0}\right) G_{in} = 0.0179\ \text{kg f/sec}$$

Heat rate transfered to the gas turbine engine

$$Q_1 := \left[h_{1\_Ql} G_{in} + \left(h_{in\_fc\_Ql} + Q_l \eta_{f\_comb}\right) B_f\right] = 0.895\ \text{MBT}$$

heat transferred to the gas turbine engine with inlet air

heat transferred to the gas turbine engine with mass of fuel

heat transferred to the gas turbine engine with fuel burning

Compressor power $N_c := \left(\frac{l_{c\_act}}{\eta_m}\right) G_{in} = 0.316\ \text{MBT}$

Fuel compressor power $N_{fc} := \frac{l_{fc\_a}}{\eta_m} B_f = 0.003\ \text{MW}$

Gas turbine power $N_{gt} := \left(l_{gt\_act} G_{gt}\right) \eta_m = 0.559\ \text{MBT}$

Electric generator power $N_e := \left(N_{gt} - N_c\right) \eta_{e\_gen} - N_{fc} = 0.239\ \text{MW}$

Electric efficiency of gas turbine engine $\eta_e := \frac{N_e}{Q_1} = 26.687\ \%$

to the procedure shown in Fig. 13.14. The vectors should be created for all four processes of the cycle (compression, heat addition, expansion and heat rejection). A cycle is formed by a sequence of processes such that the final state of the final process is the same as the initial state of the initial process. The **stack** function links together vectors belonging to four processes. It enables to represent the cycle in different types of property diagrams (*Ts*-diagram, *hs*-diagram and so on). The rate of the vectors is set by a user. In the presented example there are 300 elements in one initial vector numbered from 0 to 299. In the Mathcad environment it is possible to create vectors in two ways: through element method (based on the vectors with index *i*: $\mathbf{p}_{c_i}$, $\mathbf{p}_{f\_c_i}$ and so on) or by means of the **vectorize** operator (vector indicator over the entire expression) (see Fig. 13.14).

In the example shown in the Fig. 13.14 the following order of vector forming is introduced. At first vectors of pressures values are calculated in the compressor, fuel combustor, gas turbine and engine outlet (respectively $\mathbf{p}_{c_i}$, $\mathbf{p}_{f\_c_i}$, $\mathbf{p}_{t\_a_i}$, $\mathbf{p}_{cool_i}$). These values changes in a uniform manner from the beginning figure to the end one within one element of the engine (for example, for the compressor the beginning pressure is p_1—inlet pressure and the final one p_2—outlet pressure compressor).

$n := 300 \quad i := 0..n-1$

$p_{c_i} := p_1 + \frac{p_2 - p_1}{n-1} i \quad p_{fc\_a_i} := p_2 + \frac{p_3 - p_2}{n-1} i \quad p_{t\_a_i} := p_3 - \frac{p_3 - p_4}{n-1} i \quad p_{cool_i} := p_4 - \frac{p_4 - p_{in}}{n-1} i$

$t_c := \overrightarrow{wspgTGSPS(gs_{in}, p_c, s_1)} \quad h_c := \overrightarrow{wspgHGST(gs_{in}, t_c)}$

$l_c := h_c - h_1 \quad l_{c\_a} := \frac{l_c}{\eta_{is\_c}} \quad h_{c\_a} := h_1 + l_{c\_a}$

$t_{c\_a} := \overrightarrow{wspgTGSH(gs_{in}, h_{c\_a})} \quad s_{c\_a} := \overrightarrow{(wspgSGSPT(gs_{in}, p_c, t_{c\_a}))}$

$t_{fc\_a_i} := t_{2\_act} + \frac{t_3 - t_{2\_act}}{n-1} i \quad s_{fc\_a} := \overrightarrow{(wspgSGSPT(gs_{in}, p_{fc\_a}, t_{fc\_a}))}$

Calculation of the gas turbine with air as a working fluid

Enthalpy of the working fluid at temperature t_3 $s_{3\_air} := wspgSGSPT(gs_{in}, p_3, t_3) = 7.86$ kJ/(kg air K)

Gas turbine outlet temperature after isentropic expansion $t_{4\_air} := wspgTGSPS(gs_{in}, p_4, s_{3\_air}) = 524$ °C

Gas turbine outlet enthalpy after isentropic expansion $h_{4\_air} := wspgHGST(gs_{in}, t_{4\_air}) = 824$ kJ/(kg air)

Enhtalpy change after isentropic expansion $l_{gt\_air} := h_{3\_air} - h_{4\_air} = 609$ kJ/(kg air)

Enhtalpy change after actual expansion $l_{gt\_act\_air} := l_{gt\_air} \eta_{is\_t} = 538$ kJ/(kg air)

Gas turbine outlet enthalpy after actual expansion $h_{4\_act\_air} := h_{3\_air} - l_{gt\_act\_air} = 894$ kJ/(kg air)

Gas turbine outlet temperature after actual expansion $t_{4\_act\_air} := wspgTGSH(gs_{in}, h_{4\_act\_air}) = 587$ °C

Gas turbine outlet entropy after actual expansion $s_{4\_act\_air} := wspgSGSPT(gs_{in}, p_4, t_{4\_act\_air}) = 8$ kJ/(kg air K)

$t_t := \overrightarrow{wspgTGSPS(gs_{in}, p_{t\_a}, s_{3\_air})} \quad h_t := \overrightarrow{wspgHGST(gs_{in}, t_t)}$

$l_t := h_{3\_air} - h_t \quad l_{t\_a} := l_t \eta_{is\_t} \quad h_{t\_a} := h_{3\_air} - l_{t\_a}$

$t_{t\_a} := \overrightarrow{wspgTGSH(gs_{in}, h_{t\_a})} \quad s_{t\_a} := \overrightarrow{(wspgSGSPT(gs_{in}, p_{t\_a}, t_{t\_a}))}$

$t_{cool_i} := t_{4\_act\_air} - \frac{t_{4\_act\_air} - t_{in}}{n-1} i \quad s_{cool} := \overrightarrow{wspgSGSPT(gs_{in}, p_{cool}, t_{cool})}$

$t_{act} := stack(t_{c\_a}, t_{fc\_a}, t_{t\_a}, t_{cool}) \quad s_{act} := stack(s_{c\_a}, s_{fc\_a}, s_{t\_a}, s_{cool})$

Fig. 13.14 The procedure for creating vectors used in sketching the Brayton cycle

Change of the mass (it is about 2 %) and contents of the working fluid will not be taken into account when plotting the Brayton cycle. The cycle will be sketched for 1 kg of air. So, recalculation of the elements in the gas turbine engine where air is not a working fluid (heat addition, expansion, heat rejection) should be done.

Besides it should be noted that Mathcad calculates entropy in J/(kg K). But if we represent the entropy in the diagram in kJ/(kg K) we need to divide the values of y axis by kJ/(kg K). In Mathcad temperature is calculated automatically in Kelvins. If we want to represent the temperature in units of degrees Celsius we need to subtract from the values of x axis 273.15 K (see Fig. 13.15).

Isobars can be plotted on the *Ts*-diagram which enables to evaluate visually pressures in typical points of the Brayton cycle (compressor outlet, gas turbine inlet

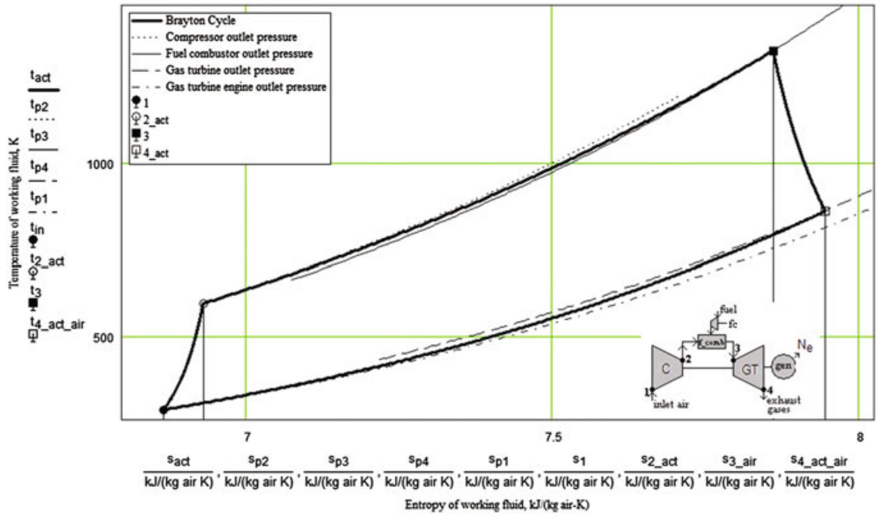


Fig. 13.15 Brayton cycle on the *Ts*-diagram

$$t_{p1_i} := t_{in} + 2t_{in} - \frac{t_{in} + t_{in} - (t_{in} - 0.8t_{in})}{n-1} i \quad s_{p1} := \overrightarrow{(wspgSGSPT(gs_{in}, p_{in}, t_{p1}))}$$

$$t_{p2_i} := t_{2\_act} + 1t_{2\_act} - \frac{t_{2\_act} + 0.9t_{2\_act} - (t_{2\_act} - 0.1t_{2\_act})}{n-1} i \quad s_{p2} := \overrightarrow{(wspgSGSPT(gs_{in}, p_2, t_{p2}))}$$

$$t_{p3_i} := t_3 + 0.2t_3 - \frac{t_3 + 0.2t_3 - (t_3 - 0.5t_3)}{n-1} i \quad s_{p3} := \overrightarrow{[wspgSGSPT[gs_{in}, p_2(1 - \delta p_{f\_comb}), t_{p3}]]}$$

$$t_{p4_i} := t_{4\_act\_air} + 0.2t_{4\_act\_air} - \frac{t_{4\_act\_air} + 0.2t_{4\_act\_air} - (t_{4\_act\_air} - 0.5t_{4\_act\_air})}{n-1} i$$

$$s_{p4} := \overrightarrow{(wspgSGSPT(gs_{in}, p_4, t_{p4}))}$$

Fig. 13.16 Forming vectors for plotting isobars in *Ts*-diagram

and so on) and pressure drops on the components of the cycle (see Fig. 13.15). The necessary vectors have to be formed for this purpose (see Fig. 13.16).

Let’s consider an algorithm of creation of Mathcad document for calculation of open Brayton cycle with three-stage compression and tree-stage expansion of working fluid.

In this case block of input data is formed similarly as in the previous example (simple open Brayton cycle). But additional parameters characterizing three-stage compression with intermediate cooling—pressure increase during compression

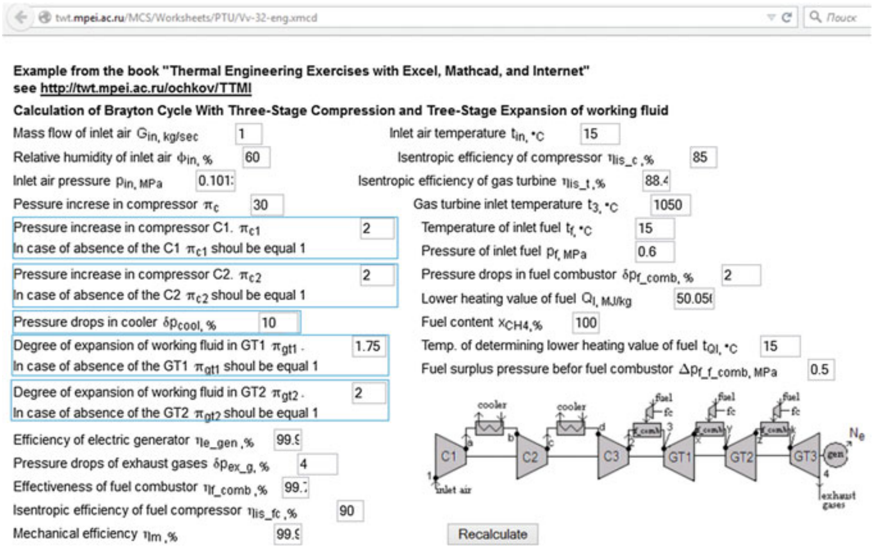


Fig. 13.17 Block of initial data and sketch of the open Brayton cycle with three-stage compression and tree-stage expansion of working fluid prepared on technology of Mathcad Calculation Server

before cooling and pressure drops in the cooler; and parameters characterizing tree-stage expansion—degree of expansion of working fluid before intermediate heat addition (see Fig. 13.17).

In the given case it is supposed that after partial compression air is cooled to the temperature of outside. But in general this temperature can also be specified as initial data. It is also supposed that after partial expansion the working fluid is heated in the fuel combustors F_Comb2 and F_Comb3 to the same temperature t_3 as in the F_Comb1. But again these temperatures can also be specified in initial data.

The total work of the compressor is equal to the sum of works during compression before and after air cooling (see Fig. 13.18).

Pressure drops in intermediate coolers are taken into account (see Fig. 13.19).

Calculation of the first fuel combustor (F_Comb1) and gas turbine (GT1) is the same as in the simple gas turbine engine and described above (see Figs. 13.7, 13.8, 13.9, 13.10, 13.11 and 13.12). It is only necessary to indicate correctly the GT1 inlet and outlet pressures (see Fig. 13.20).

Calculation of the Compressor C1

Compressor outlet pressure $p_a := \text{if}(p_{in}\pi_{c1} \le 5\text{MPa}, p_{in}\pi_{c1}$, "WSP calculculates properties of gases as ideal working fluids at pressures lower than 5 MПa ") = 0.203 MPa

Outlet entropy after isentropic compression $s_a := s_1 = 6.9$ kJ/(kg air K)

Outlet temperature after isentropic compression $t_a :=$ wspgTGSPS$(gs_{in}, p_a, s_a) = 78$ °C

Outlet enthalpy after isentropic compression $h_a :=$ wspgHGST$(gs_{in}, t_a) = 353.7$ kJ/(kg air)

Outlet enthalpy after actual compression $h_{a\_act} := h_1 + (h_a - h_1) / \eta_{is\_c} = 365$ kJ/(kg air)

Outlet temperature after actual compression $t_{a\_act} :=$ wspgTGSH$(gs_{in}, h_{a_act}) = 89$ °C

Outlet entropy after actual compression $s_{a\_act} :=$ wspgSGSPT$(gs_{in}, p_a, t_{a_act}) = 6.9$ kJ/(kg air K)

Enthalpy change after isentropic compression $l_{c1} := h_a - h_1 = 64$ kJ/(kg air)

Enthalpy change after actual compression $l_{c1\_act} := h_{a\_act} - h_1 = 75$ kJ/(kg air)

Calculation of the Compressor C2

Cooler outlet pressure (C2 inlet) $p_b := \begin{vmatrix} p_a(1-\delta p_{cool}) \text{ if } \pi_{c1} \ne 1 \\ p_a \text{ otherwise} \end{vmatrix} = 0$ MPa

Inlet temperature $t_b := t_{in} = 15$ °C

Inlet enthalpy $h_b :=$ wspgHGST$(gs_{in}, t_b) = 290$ kJ/(kg air)

Inlet entropy $s_b :=$ wspgSGSPT$(gs_{in}, p_b, t_b) = 7$ kJ/(kg air K)

Outlet pressure $p_c := \text{if}(p_b\pi_{c2} \le 5\text{MPa}, p_b\pi_{c2}$, "WSP calculculates properties of gases as ideal working fluids at pressures lower than 5 MПa ") = 0.365 MPa

Outlet entropy after isentropic compression $s_c := s_b = 6.7$ kJ/(kg air K)

Outlet temperature after isentropic compression $t_c :=$ wspgTGSPS$(gs_{in}, p_c, s_c) = 78$ °C

Outlet enthalpy after isentropic compression $h_c :=$ wspgHGST$(gs_{in}, t_c) = 353.7$ kJ/(kg air)

Outlet enthalpy after actual compression $h_{c\_act} := h_b + (h_c - h_b) / \eta_{is\_c} = 365$ kJ/(kg air)

Outlet temperature after actual compression $t_{c\_act} :=$ wspgTGSH$(gs_{in}, h_{c_act}) = 89$ °C

Outlet entropy after actual compression $s_{c\_act} :=$ wspgSGSPT$(gs_{in}, p_c, t_{c_act}) = 6.7$ kJ/(kg air K)

Enthalpy change after isentropic compression $l_{c2} := h_c - h_b = 64$ kJ/(kg air)

Enthalpy change after actual compression $l_{c2\_act} := h_{c\_act} - h_b = 75$ kJ/(kg air)

Calculation of the Compressor C3

Cooler outlet pressure (C3 inlet) $p_d := \begin{vmatrix} p_c(1-\delta p_{cool}) \text{ if } \pi_{c2} \ne 1 \\ p_c \text{ otherwise} \end{vmatrix} = 0.3$ MPa

Inlet temperature $t_d := t_{in} = 15$ °C

Inlet enthalpy $h_d :=$ wspgHGST$(gs_{in}, t_d) = 290$ kJ/(kg air)

Inlet entropy $s_d :=$ wspgSGSPT$(gs_{in}, p_d, t_d) = 7$ kJ/(kg air K)

Outlet pressure $p_2 := \text{if}(p_{in}\pi_c \le 5\text{MPa}, p_{in}\pi_c$, "WSP calculculates properties of gases as ideal working fluids at pressures lower than 5 MПa ") = 3.039 MPa

Outlet entropy after isentropic compression $s_2 := s_d = 6.5$ kJ/(kg air K)

Outlet temperature after isentropic compression $t_2 :=$ wspgTGSPS$(gs_{in}, p_2, s_2) = 267$ °C

Outlet enthalpy after isentropic compression $h_2 :=$ wspgHGST$(gs_{in}, t_2) = 547.7$ kJ/(kg air)

Outlet enthalpy after actual compression $h_{2\_act} := h_d + (h_2 - h_d) / \eta_{is\_c} = 593.1$ kJ/(kg air)

Outlet temperature after actual compression $t_{2\_act} :=$ wspgTGSH$(gs_{in}, h_{2_act}) = 310.2$ °C

$t_{2\_act} := \text{if}(t_3 > t_{2\_act}, t_{2\_act}$, "Temperature t3 must be higher than t3_act") = 310 °C

Outlet entropy after actual compression $s_{2\_act} :=$ wspgSGSPT$(gs_{in}, p_2, t_{2_act}) = 6.6$ kJ/(kg air K)

Enthalpy change after isentropic compression $l_{c3} := h_2 - h_d = 258$ kJ/(kg air)

Enthalpy change after actual compression $l_{c3\_act} := h_{2\_act} - h_d = 303$ kJ/(kg air)

Total work of isentropic compression $l_c := l_{c1} + l_{c2} + l_{c3} = 385$ kJ/(kg air)

Total work of actual compression $l_{c\_act} := l_{c1\_act} + l_{c2\_act} + l_{c3\_act} = 453$ kJ/(kg air)

Fig. 13.18 Calculation of the air compression with intermediate cooling

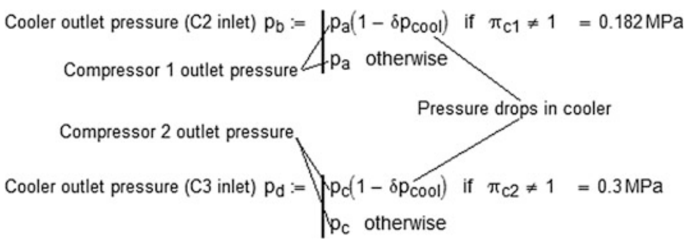


Fig. 13.19 Calculation of pressures in typical points of compression and cooling

Inlet pressure $p_3 := p_2(1 - \delta p_{f\_comb}) = 2.978\ \text{MPa}$

Compressor 3 outlet pressure

Pressure drops in fuel combustor

Fig. 13.20 Calculation of the GT1 inlet pressure

Calculation of the Fuel Combustor 2 (F_Comb2)

Theoretical volume of the GT1 outlet gas required for stoichiometric combustion of a unit volume of the fuel

$$V_{0\_f\_comb2} := \frac{1}{x_{O2\_3} - 0.5x_{H2\_3}}(2x_{CH4}) = 14.489\ \text{cub m/cub m}$$

Theoretical nitrogen volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_N2\_f\_comb2} := \text{wspgVFGSGS}(gs_{w\_f\_gt1}, \text{"N2"})V_{0\_f\_comb2} = 10.838\ \text{cub m/cub m}$$

Theoretical CO2 volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_CO2\_f\_comb2} := \text{wspgVFGSGS}(gs_{w\_f\_gt1}, \text{"CO2"})V_{0\_f\_comb2} + x_{CH4} = 1.4615\ \text{cub m/cub m}$$

Theoretical H2O volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_H2O\_f\_comb2} := x_{H2\_3}V_{0\_f\_comb2} + 2x_{CH4} + x_{H2O\_3}V_{0\_f\_comb2} = 3.058\ \text{cub m/cub m}$$

Theoretical argon volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_Ar\_f\_comb2} := \text{wspgVFGSGS}(gs_{w\_f\_gt1}, \text{"Ar"})V_{0\_f\_comb2} = 0.1306\ \text{cub m/cub m}$$

Theoretical mass of gas required for stoichiometric combustion of a unit mass of the fuel

$$L_{0\_f\_comb2} := V_{0\_f\_comb2}\frac{\text{wspgMMGS}(gs_{w\_f\_gt1})}{MM_{CH4}} = 25.694\ \text{kg g/(kg f)}$$

Creation of the combustion products mixture

$$gs_{0p\_f\_comb2} := \text{concat}(\text{"N2:"}, \text{num2str}(V_{0\_N2\_f\_comb2}), \text{";CO2:"}, \text{num2str}(V_{0\_CO2\_f\_comb2}), \text{";H2O:"}, \text{num2str}(V_{0\_H2O\_f\_comb2}), \text{";Ar:"}, \text{num2str}(V_{0\_Ar\_f\_comb2}))$$

$$gs_{0p\_f\_comb2} = \text{"N2:10.8381661770147;CO2:1.4615481686228;H2O:3.05791356231511;Ar:0.13056358075604"}$$

Fig. 13.21 Calculation of the second fuel combustor

Calculation of the Fuel Combustor 3 (F_Comb3)

Theoretical volume of the GT2 outlet gas required for stoichiometric combustion of a unit volume of the fuel

$$V_{0\_f\_comb3} := \frac{1}{x_{O2\_y} - 0.5x_{H2\_y}}(2x_{CH4}) = 16.2\ \text{cub m/cub m}$$

Theoretical nitrogen volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_N2\_f\_comb3} := \text{wspgVFGSGS}(gs_{w\_f\_gt2}, \text{"N2"})V_{0\_f\_comb3} = 12.036\ \text{cub m/cub m}$$

Theoretical CO2 volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_CO2\_f\_comb3} := \text{wspgVFGSGS}(gs_{w\_f\_gt2}, \text{"CO2"})V_{0\_f\_comb3} + x_{CH4} = 1.6231\ \text{cub m/cub m}$$

Theoretical H2O volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_H2O\_f\_comb3} := x_{H2\_y}V_{0\_f\_comb3} + 2x_{CH4} + x_{H2O\_y}V_{0\_f\_comb3} = 3.396\ \text{cub m/cub m}$$

Theoretical argon volume in the products from the combustion per cubic meter of the fuel

$$V_{0\_Ar\_f\_comb3} := \text{wspgVFGSGS}(gs_{w\_f\_gt2}, \text{"Ar"})V_{0\_f\_comb3} = 0.145\ \text{cub m/cub m}$$

Theoretical mass of gas required for stoichiometric combustion of a unit mass of the fuel

$$L_{0\_f\_comb3} := V_{0\_f\_comb3}\frac{\text{wspgMMGS}(gs_{w\_f\_gt2})}{MM_{CH4}} = 28.644\ \text{kg g/(kg f)}$$

Creation of the combustion products mixture

$$gs_{0p\_f\_comb3} := \text{concat}(\text{"N2:"}, \text{num2str}(V_{0\_N2\_f\_comb3}), \text{";CO2:"}, \text{num2str}(V_{0\_CO2\_f\_comb3}), \text{";H2O:"}, \text{num2str}(V_{0\_H2O\_f\_comb3}), \text{";Ar:"}, \text{num2str}(V_{0\_Ar\_f\_comb3}))$$

$$gs_{0p\_f\_comb3} = \text{"N2:12.0358813234023;CO2:1.62306242759837;H2O:3.39584059998097;Ar:0.144992034397004"}$$

Fig. 13.22 Calculation of the third fuel combustor

During calculation of the second and third fuel combustors (F_Comb2 and F_Comb3) it should be remembered that fuel and combustion products after GT1 and GT2 respectively are supplied into these fuel combustors. That's why calculation of F_Comb2 and F_Comb3 is performed taking into account that oxidant for fuel burning in these combustors is a mixture of excessive air and combustion products leaving respectively GT1 and GT2 (see Figs. 13.21 and 13.22).

Enthalpy of the combustion products at temperature t_y $h_{ycomb\_p} := wspgHGST(gs_{0p\_f\_comb2}, t_y) = 1609.1\ \text{kJ/(kg g)}$

Enthalpy of the GT1 outlet gas at temperature t_y $h_{g\_gt1} := wspgHGST(gs_{w\_f\_gt1}, t_y) = 1490\ \text{kJ/(kg g)}$

Comparing Enthalpies with the Temperature of Determining Q_l

$h_{x\_act\_Ql} := h_{x\_act} - wspgHGST(gs_{w\_f\_gt1}, t_{Ql}) = 1013.128\ \text{kJ/(kg g)}$

$h_{ycomb\_p\_Ql} := h_{ycomb\_p} - wspgHGST(gs_{0p\_f\_comb2}, t_{Ql}) = 1296.2\ \text{kJ/(kg g)}$

$h_{g\_gt1\_Ql} := h_{g\_gt1} - wspgHGST(gs_{w\_f\_gt1}, t_{Ql}) = 1192.47\ \text{kJ/(kg g)}$

Excess GT1 Outlet Gas for a Unit Mass of the Fuel

$$g_{gt1\_ex} := \begin{cases} \infty & \text{if } h_{g\_gt1\_Ql} = h_{x\_act\_Ql} \\ \dfrac{h_{x\_act\_Ql} L_{0\_f\_comb2} - (L_{0\_f\_comb2} + 1) h_{ycomb\_p\_Ql} + h_{out\_fc\_a\_Ql} + Q_l \eta_{f\_comb}}{h_{g\_gt1\_Ql} - h_{x\_act\_Ql}} & \text{otherwise} \end{cases} = 232.506$$

Excess of the GT1 outlet gas $\alpha_{gt1} := \dfrac{g_{gt1\_ex} + L_{0\_f\_comb2}}{L_{0\_f\_comb2}} = 10.049$

Creation of the mixture of the working fluid in GT1

$$gs_{w\_f\_gt2} := \begin{cases} gs_{w\_f\_gt1} & \text{if } \pi_{gt1} = 1 \\ concat("\{", gs_{w\_f\_gt1}, "\}:", num2str(g_{gt1\_ex}), "M;\{", gs_{0p\_f\_comb2}, "\}:", num2str(L_{0\_f\_comb2} + 1), "M") & \text{otherwise} \end{cases}$$

Fig. 13.23 Evaluation of the amount of excessive exhausted gases after GT1 for providing the required temperature at GT2 inlet

Enthalpy of the combustion products at temperature t_k $h_{kcomb\_p} := wspgHGST(gs_{0p\_f\_comb3}, t_k) = 1609.1\ \text{kJ/(kg g)}$

Enthalpy of the GT2 outlet gas at temperature t_k $h_{g\_gt2} := wspgHGST(gs_{w\_f\_gt2}, t_k) = 1502.3\ \text{kJ/(kg g)}$

Comparing Enthalpies with the Temperature of Determining Q_l

$h_{z\_act\_Ql} := h_{z\_act} - wspgHGST(gs_{w\_f\_gt2}, t_{Ql}) = 983.617\ \text{kJ/(kg g)}$

$h_{kcomb\_p\_Ql} := h_{kcomb\_p} - wspgHGST(gs_{0p\_f\_comb3}, t_{Ql}) = 1296.2\ \text{kJ/(kg g)}$

$h_{g\_gt2\_Ql} := h_{g\_gt2} - wspgHGST(gs_{w\_f\_gt2}, t_{Ql}) = 1203.152\ \text{kJ/(kg g)}$

Excess GT2 Outlet Gas for a Unit Mass of the Fuel

$$g_{gt2\_ex} := \begin{cases} \infty & \text{if } h_{g\_gt2\_Ql} = h_{z\_act\_Ql} \\ \dfrac{h_{z\_act\_Ql} L_{0\_f\_comb3} - (L_{0\_f\_comb3} + 1) h_{kcomb\_p\_Ql} + h_{out\_fc\_a\_Ql} + Q_l \eta_{f\_comb}}{h_{g\_gt2\_Ql} - h_{z\_act\_Ql}} & \text{otherwise} \end{cases} = 182.283$$

Excess of the GT2 outlet gas $\alpha_{gt2} := \dfrac{g_{gt2\_ex} + L_{0\_f\_comb3}}{L_{0\_f\_comb3}} = 7.364$

Creation of the mixture of the working fluid in GT3

$$gs_{w\_f\_gt3} := \begin{cases} gs_{w\_f\_gt1} & \text{if } \pi_{gt2} = 1 \\ concat("\{", gs_{w\_f\_gt2}, "\}:", num2str(g_{gt2\_ex}), "M;\{", gs_{0p\_f\_comb3}, "\}:", num2str(L_{0\_f\_comb3} + 1), "M") & \text{otherwise} \end{cases}$$

Fig. 13.24 Evaluation of the amount of excessive exhausted gases after GT2 for providing the required temperature at GT3 inlet

Evaluation of the amount of excessive exhausted gases leaving GT1 per 1 kg of fuel is made according to heat balance of the second fuel combustor (see Fig. 13.23). The heat with mass of exhausted gases after the GT1, mass of fuel, energy release with combustion reaction are the inlet terms of the F_Comb2 heat

Result Characteristics of the Brayton Cycle

Mass rate of the working fluid through the GT1 $G_{gt\_1} := \left(\frac{g_{air\_ex} + L_{0\_f\_comb1} + 1}{g_{air\_ex} + L_{0\_f\_comb1}}\right) G_{in} = 1.018\,\text{kg g/sec}$

Mass rate of the working fluid through the GT2 $G_{gt\_2} := \left(\frac{g_{gt1\_ex} + L_{0\_f\_comb2} + 1}{g_{gt1\_ex} + L_{0\_f\_comb2}}\right) G_{gt\_1} = 1.022\,\text{kg g/sec}$

Mass rate of the working fluid through the GT3 $G_{gt\_3} := \left(\frac{g_{gt2\_ex} + L_{0\_f\_comb3} + 1}{g_{gt2\_ex} + L_{0\_f\_comb3}}\right) G_{gt\_2} = 1.027\,\text{kg g/sec}$

Fuel rate in the F_Comb1 $B_{f\_comb1} := \left(\frac{1}{g_{air\_ex} + L_{0\_f\_comb1}}\right) G_{in} = 0.0181\,\text{kg f/sec}$

Fuel rate in the F_Comb2 $B_{f\_comb2} := \left(\frac{1}{g_{gt1\_ex} + L_{0\_f\_comb2}}\right) G_{gt\_1} = 0.0039\,\text{kg f/sec}$

Fuel rate in the F_Comb3 $B_{f\_comb3} := \left(\frac{1}{g_{gt2\_ex} + L_{0\_f\_comb3}}\right) G_{gt\_2} = 0.0048\,\text{kg f/sec}$

Total fuel rate $B_f := B_{f\_comb1} + B_{f\_comb2} + B_{f\_comb3} = 0.0269\,\text{kg f/sec}$

Heat rate transfered to the gas turbine engine

$Q_1 := h_{1\_Ql} G_{in} + (h_{in\_fc\_Ql} + Q_l) B_{f\_comb1} + (h_{in\_fc\_Ql} + Q_l) B_{f\_comb2} + (h_{in\_fc\_Ql} + Q_l) B_{f\_comb3} =$ 1.347 MBт

Compressor power $N_c := \left(\frac{l_{c\_act}}{\eta_m}\right) G_{in} =$ 0.453 MBт

Fuel compressor power $N_{fc} := \frac{l_{fc\_a}}{\eta_m} B_{f\_comb1} + \frac{l_{fc\_a}}{\eta_m} B_{f\_comb2} + \frac{l_{fc\_a}}{\eta_m} B_{f\_comb3} = 0.01\,\text{MW}$

Gas turbine power $N_{gt} := (l_{gt1\_act} G_{gt\_1} + l_{gt2\_act} G_{gt\_2} + l_{gt3\_act} G_{gt\_3}) \eta_m =$ 0.981 MBт

Electric generator power $N_e := (N_{gt} - N_c) \eta_{e\_gen} - N_{fc} = 0.517\,\text{MW}$

Electric efficiency of gas turbine engine $\eta_e := \frac{N_e}{Q_1} = 38.391\,\%$

Fig. 13.25 Result characteristics of the open Brayton cycle with three-stage compression and tree-stage expansion of working fluid

balance. The heat with mass of combustion products and excessive amount of exhausted gases after the GT1 are outlet terms of the F_Comb2 heat balance.

Evaluation of the amount of excessive exhausted gases leaving GT2 per 1 kg of fuel is performed in the similar manner (see Fig. 13.24).

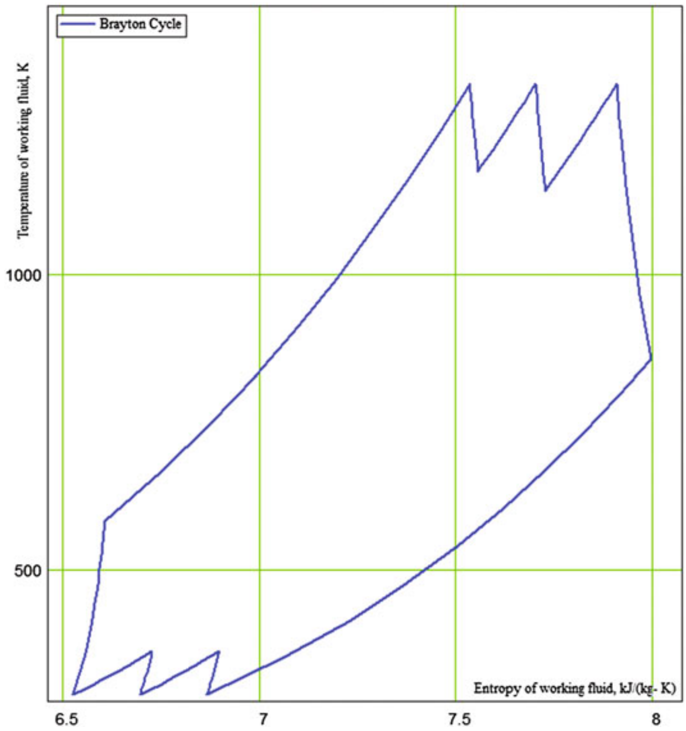


Fig. 13.26 Open Brayton cycle with three-stage compression and tree-stage expansion of working fluid on the *Ts*-diagram

An example of resulting characteristics of the analyzed gas turbine engine is shown in Fig. 13.25. Cycle visualizing is presented in Fig. 13.26.

Chapter 14
Calculation of Combined (Binary) Cycle

Valery Ochkov, Konstantin Orlov and Volodymyr Voloshchuk

Abstract This chapter describes approaches to calculate the thermal efficiency of a binary power cycle consisting gas and steam turbines. Links to authors' web sites with calculations such binary cycles of different types are provided.

Figures 14.1, 14.2 and 14.3 shows Mathcad calculation of thermal efficiency for the simplest ideal combined (binary) cycle, i.e. steam turbine cycle with use of superheated steam. At this unit the boiler burner is replaced with gas turbine (Brayton cycle) equipped with air compressor (c), combustion chamber (cc), gas turbine (gt) and electric generator [26].

The initial part of calculation in Fig. 14.1 contains a scheme of the combined cycle plant. There is a *T, s*-diagrams of thermodynamic cycles (gas turbine and water steam cycles) on the site of the book used for the calculation. This diagram is "dead" in the sense that changing of initial plant parameters (see assignment operators to the right and to the left from the cycle scheme in Fig. 14.1) does not result in corresponding changes in the diagram.

The site of the chapter: https://www.ptcusercommunity.com/message/423032.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Voloshchuk
National University of Water Management and Natural Resources, Rivne, Ukraine
e-mail: Vl.volodya@gmail.com

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_14

Water and steam properties

Reference:http://twt.mpei.ru/tthb/H2O.xmcdz $kgws := kg$ $kJ/kgws := \frac{1000J}{kgws}$ $kJ/(kgws\ K) := \frac{1000J}{kgws\ K}$

Gas turbine

Working medium $wm := "Air"$

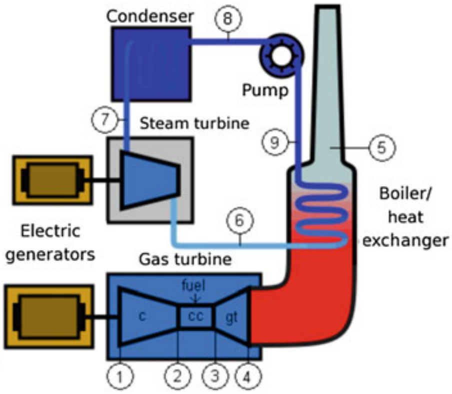
$p_1 := 0.1MPa$

$T_1 := 15\,°C$

$T_3 := 1100\,°C$

$T_5 := 130\,°C$

c - compressor
cc - combustion chamber
gt - gas turbine



Steam turbine

$p_6 := 9MPa$

$T_6 := 480\,°C$

$\Delta T_{c\text{-}r} := 17K$

Notation:
h - specific enthalpy
s - specific entropy
q - specific heat
l - specific work
η - thermal efficiency
kgg - kg of gas
kgws - kg of water and steam

Ideal Steam turbine calculation

$h_6 := wspHPT(p_6, T_6) = 3336.33\ kJ/kgws$ $s_6 := wspSPT(p_6, T_6) = 6.593\ kJ/(kgws\ K)$

$T_7 := T_1 + \Delta T_{c\text{-}r} = 32\,°C$ $p_7 := wspPST(T_7) = 4.76\ kPa$ $s_7 := s_6$ $x_7 := wspXSTS(T_7, s_7) = 77.12\,\%$

$h_7 := wspHSTX(T_7, x_7) = 2004.37\ kJ/kgws$ $T_8 := T_7$ $h_8 := wspHSWT(T_8) = 134.11\ kJ/kgws$

$p_9 := p_6$ $s_8 := wspSSWT(T_8) = 0.464\ kJ/(kgws\ K)$ $s_9 := s_8$ $h_9 := wspHPS(p_9, s_9) = 143.12\ kJ/kgws$

$q_{boiler} := h_6 - h_9 = 3193.22\ kJ/kgws$ $l_{steam\ turb} := h_6 - h_7 = 1331.96\ kJ/kgws$ $l_{pump} := h_9 - h_8 = 9.011\ kJ/kgws$

The **thermal efficiently** of the steam turbine cycle is: $\eta_{steam\ turb} := \frac{l_{steam\ turb} - l_{pump}}{q_{boiler}} = 41.43\,\%$

Fig. 14.1 Calculation of steam turbine part of the combined cycle

Gas (Air) properties

Reference:http://twt.mpei.ru/tthb/gas.xmcdz $kgg := kg$ $kJ/kgg := \frac{1000J}{kgg}$ $kJ/(kgg\ K) := \frac{1000J}{kgg\ K}$

Ideal Gas turbine calculation

$wm = "Air"$ $h_1 := wspgHGST(wm, T_1) = 288.56\ kJ/kgg$ $s_1 := wspgSGSPT(wm, p_1, T_1) = 6.83\ kJ/(kgg\ K)$

$s_2 := s_1$ $T_2 := wspgTGSPS(wm, p_2, s_2) = 279.02\,°C$ $h_2 := wspgHGST(wm, T_2) = 557.39\ kJ/kgg$

$p_3 := p_2$ $s_3 := wspgSGSPT(wm, p_3, T_3) = 7.847\ kJ/(kgg\ K)$ $h_3 := wspgHGST(wm, T_3) = 1483.61\ kJ/kgg$

$p_4 := p_1$ $s_4 := s_3$ $T_4 := wspgTGSPS(wm, p_4, s_4) = 498.08\,°C$

$h_4 := if(T_4 > T_6, wspgHGST(wm, T_4), "Error") = 790.87\ kJ/kgg$

$q_{gt} := h_3 - h_2 = 926.22\ kJ/kgg$ $l_{gas\ turb} := h_3 - h_4 = 692.74\ kJ/kgg$ $l_{compressor} := h_2 - h_1 = 268.83\ kJ/kgg$

The **thermal efficiently** of the gas turbine cycle is: $\eta_{gas\ turb} := \frac{l_{gas\ turb} - l_{compressor}}{q_{gt}} = 45.77\,\%$

Fig. 14.2 Calculation of gas turbine part of the combined cycle

Ideal Combined cycle calculation

$h_5 := \text{wspgHGST}(\text{wm}, T_5) = 404.51\ \text{kJ/kgg}$

Heat balance of the boiler $m \cdot (h_4 - h_5) = h_6 - h_9$ $\qquad m := \dfrac{h_6 - h_9}{h_4 - h_5} = 8.26 \dfrac{\text{kgg}}{\text{kgws}}$

$q_{cc} := m \cdot (h_3 - h_2) = 7655.16\ \text{kJ/kgws}$ $\qquad l_{gas\ turb} := (h_3 - h_4) - (h_2 - h_1) = 423.91\ \text{kJ/kgg}$

$l_{steam\ turb} := (h_6 - h_7) - (h_9 - h_8) = 1322.95\ \text{kJ/kgws}$ $\qquad \dfrac{m \cdot l_{gas\ turb}}{l_{steam\ turb}} = 2.648$

The thermal efficiently of the combined cycle is:

$\eta_{cc} := \dfrac{m \cdot l_{gas\ turb} + l_{steam\ turb}}{q_{cc}} = 63.05\,\%$ where $\qquad m = 8.26497 \dfrac{\text{kgg}}{\text{kgws}}$

$l_{gas\ turb} = 423.913\ \text{kJ/kgg}$

$m \cdot l_{gas\ turb} = 3503.63\ \text{kJ/kgws}$

$l_{steam\ turb} = 1322.95\ \text{kJ/kgws}$

$m \cdot l_{gas\ turb} + l_{steam\ turb} = 4826.58\ \text{kJ/kgws}$

$q_{cc} = 7655.16\ \text{kJ/kgws}$

Fig. 14.3 Calculation of thermal efficiency for the simplest ideal combined cycle

But in the author’s Internet site http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Rankine.xmcd (steam turbine cycle) and http://twt.mpei.ac.ru/MCS/Worksheets/PTU/GTU.xmcd (gas turbine cycle) there are provided “live” diagrams that change their view in case of initial cycle parameter changing. Moreover, the site of steam turbine cycle provides for selection of diagram axis and for possibility of creating not only 2-D but also 3-D diagrams, for example, adding pressure to temperature and specific entropy. However, it is difficult to show two cycles (steam and gas turbine cycles) in one “live” diagram due to the fact that working media of the above cycles (steam and gas mixture) have different base points for pressure and temperature at calculation of enthalpy and entropy.

As it was already underlined in Chap. 1, it is impossible to perform calculation of thermodynamic cycles without information relevant to the properties of working media used in these cycles. As it was mentioned before, all functions computing thermodynamic properties of water and steam and required for calculation of steam turbine cycle using superheated steam are presented in the author’s site www.trie.ru . These functions can be downloaded from the site and used in calculation by copying and pasting. These functions can be also loaded into the work computer or into the local networks. After that one can make link references to the relevant Mathcad-files in order to make these functions viewable in the calculations as it was already described. However, we’d like to repeat once again that it is possible to proceed the following way: to make a link reference not to your own computer and not to your own local network but to the relevant site.

It was done this way in the calculation shown in Fig. 14.1. Link reference was made to the file **H2O.xmcdz** stored in the author's site before calculation of specific enthalpy $\mathbf{h}_6$ and specific entropy $\mathbf{s}_6$ of the superheated water steam. We have discussed it in Chap. 4, where similar calculation was created. After such link reference the functions relevant to water and water steam properties became viewable. As it was mentioned earlier, files and function names related to water and water steam properties are provided with wsp prefix.

Similar way "cloud" link reference was made to the file **GAS.xmcdz** which stores functions with **wspg** prefix computing thermodynamic properties of the second working medium of our binary cycle (see Fig. 14.2). In this case calculation is more complicated. As a matter of fact, the first working medium (water and steam) as it circulates in the cycle changes only its parameters and phase state. The working medium of the second cycle at the binary plant also changes its composition: air—air mixture with fuel—flue gases. The author's site contains calculations of gas turbine cycles with various degrees of complexity taking into account changes in the working medium composition, initial air humidity degree and other aspects. One can find one of such calculations in the site at the following address http://twtmas.mpei.ac.ru/mas/Worksheets/orlov/gases/simple_gtu.mcd and in the Chap. 13. The calculation shown in the Fig. 14.2 has as the working medium air with standard composition (see the first assignment operator in the Figure: **wm: = "Air"**). One can put symbols of other gases (H_2, N_2, O_2 etc.) and gas mixtures into the given string constant.

Figures 14.1 and 14.2 show calculations of thermal efficiencies for the steam turbine cycle and gas turbine cycle (Brayton cycle) for the case when they are separately operated with initial parameters shown in the initial part of Fig. 14.1.

Figure 14.3 provides for calculation of binary cycle thermal efficiency (see its scheme in the Fig. 14.1), where part of gas turbine exhaust heat is used in the heat recovery steam generator for generation of main superheated steam (point 6) out of feed water (point 9). The control over theoretical possibility of this process is executed by if operator inserted into the gas turbine calculation (see the fifth string in the bottom part of Fig. 14.2): temperature of exhaust gases from the gas turbine $\mathbf{T}_4$ shall be higher than the temperature of main water steam $\mathbf{T}_6$, fed to the steam turbine. Otherwise, variable $\mathbf{h}_4$ (specific enthalpy of gas leaving the gas turbine) is assigned string value ("Error") instead of numeric value thus interrupting further calculation by error message.

The calculations shown in Figs. 14.1, 14.2 and 14.3 introduce measuring units of water and/or water/steam kilogram (kgws) and gas kilogram (kgg). This type of calculation with separated physical quantities was discussed in the Chap. 2—See Fig. 2.24.

The book site contains calculations of more complicated binary cycles which diagrams are presented in Figs. 14.4, 14.5, 14.6, 14.7, 14.8, 14.9, 14.10, 14.11, 14.12 and 14.13. Internet addresses of the calculations are given in the address strings and on the site of the book.

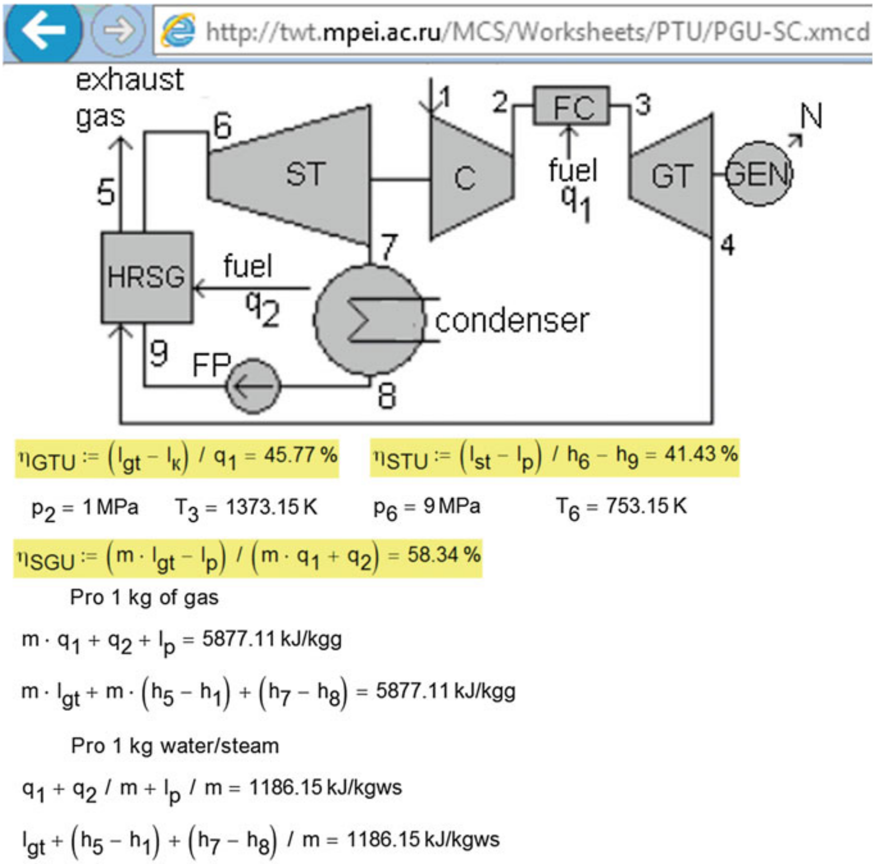


Fig. 14.4 CCPP with compressor driven by steam turbine

Honestly speaking, combined cycle power plants in Russian power industry were introduced as in its time the potato was in Russia, often to the prejudice of traditional cultural products. The Chief Engineer of one major Russian power plant complained to the author about the fact that they were forced to build a combined cycle unit at the power station where for a long time six traditional steam turbine units had been in operation demonstrating perfect results. The specialists at the above power plant were going to construct the seventh similar unit and made foundation for it. But instead of it they were instructed to construct the seventh unit basing on combined cycle diagram. This decision required formation of new operation, maintenance and other services as well as demolition of already built foundation. But the main point here lies in the fact that this new combined cycle

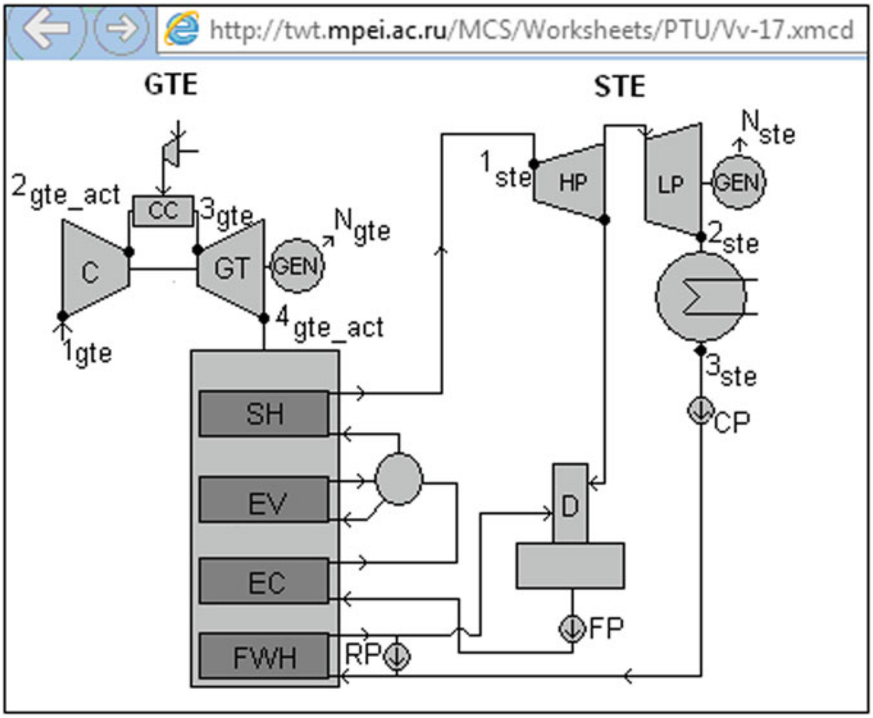


Fig. 14.5 CCPP with single pressure heat recovery steam generator

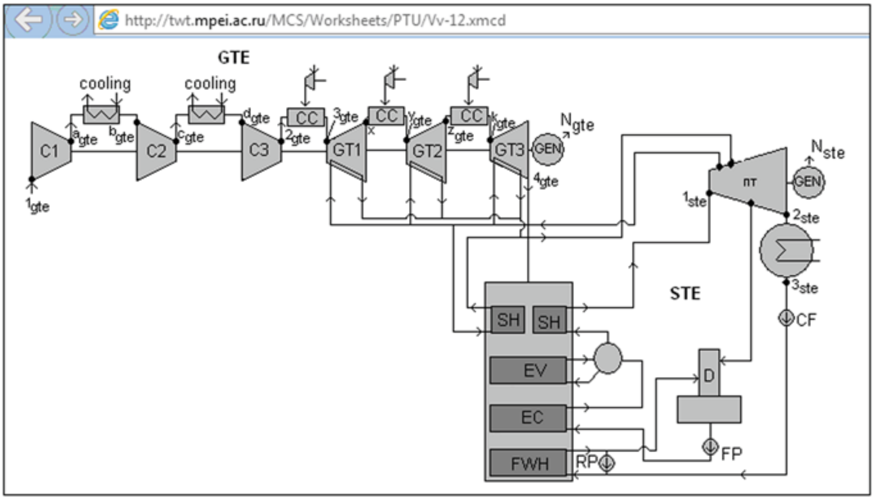


Fig. 14.6 CCPP with single pressure heat recovery steam generator with reheat and multistage air compression

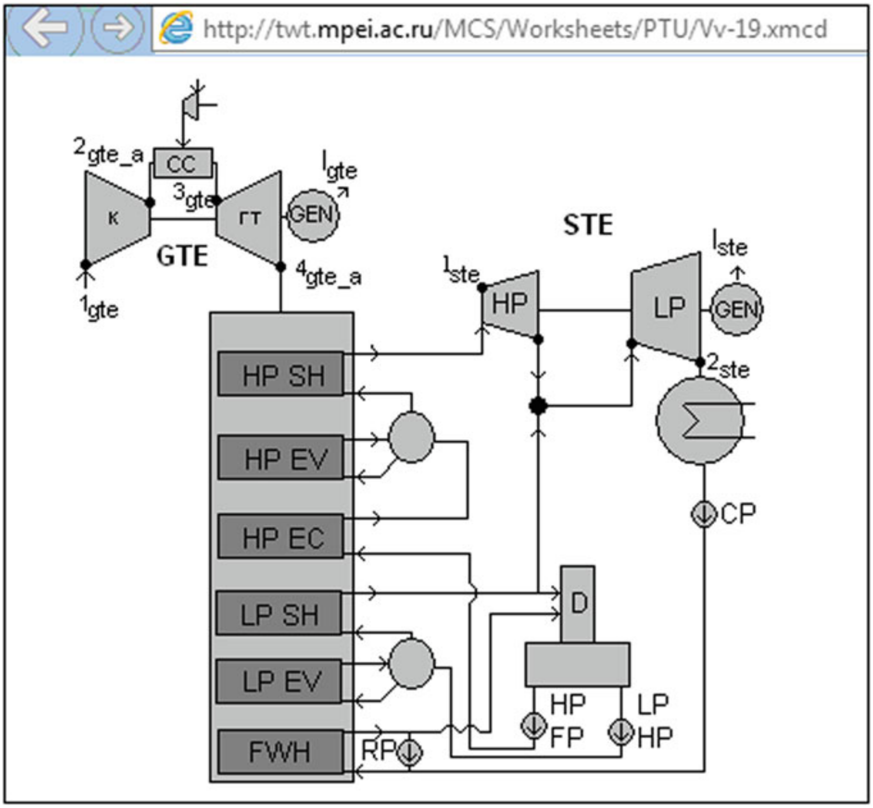


Fig. 14.7 CCPP with double pressure heat recovery steam generator

unit will not provide for any special cost saving. Despite of the fact that fuel consumption for kwhr production is reduced due to the higher combined cycle efficiency, all financial profit will be used for paying for after-sales servicing of gas turbine supplied by foreign company (and the above servicing may be interrupted any time due to the sanctions). Besides the turbine itself is quite expensive as it is a customized product unlike aero gas turbines representing mass production articles. Customized products are always expensive and prankish. It's no coincidence that more and more often voices are raised against application of high capacity gas turbines at power plants.

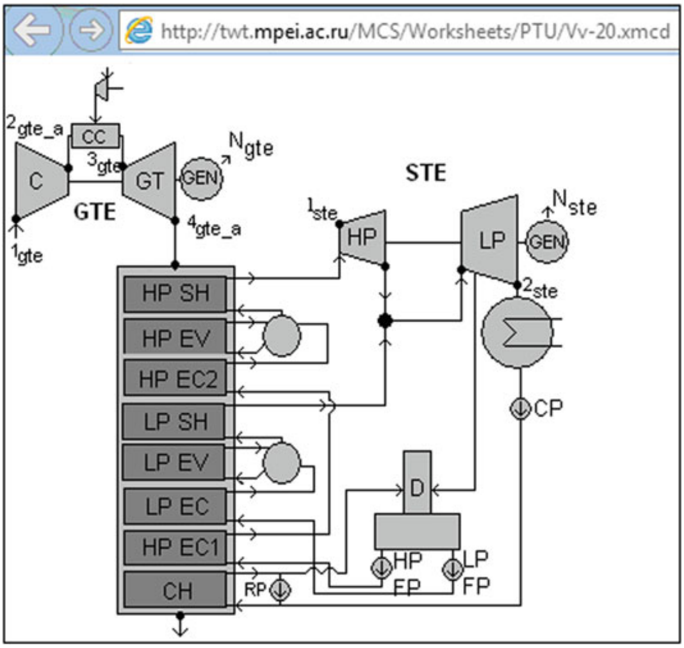


Fig. 14.8 CCPP with double pressure heat recovery steam generator and application of two-stage feed water heating in the economizer

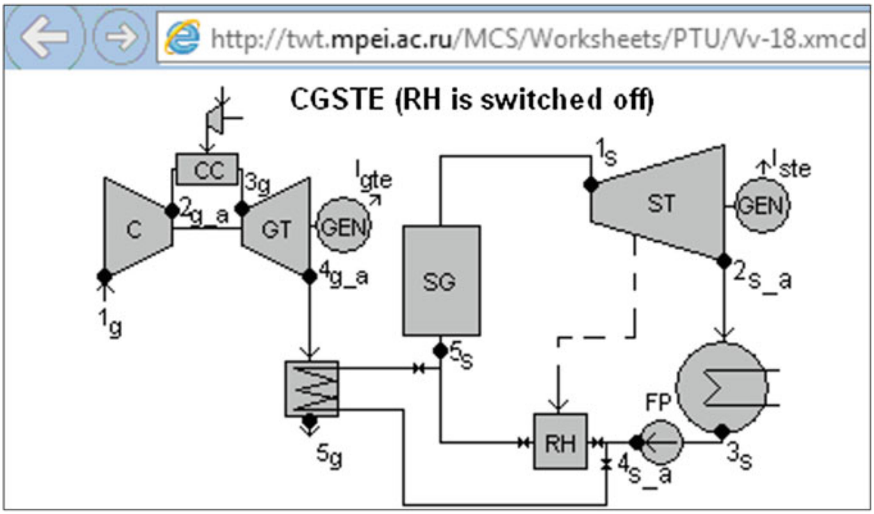


Fig. 14.9 CCPP with semi-dependent diagram (with regeneration displacement)

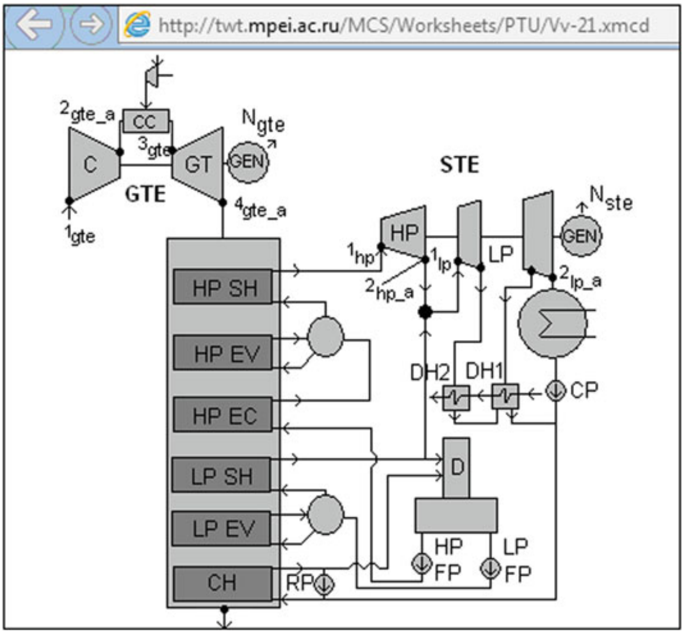


Fig. 14.10 CCPP with extraction steam turbine

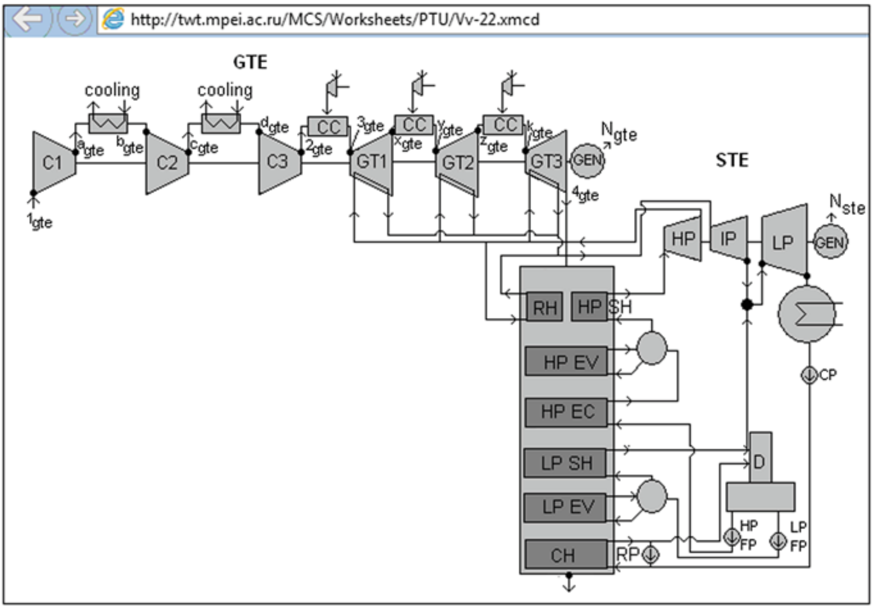


Fig. 14.11 CCPP with multistage air compression and double pressure reheat heat recovery steam generator

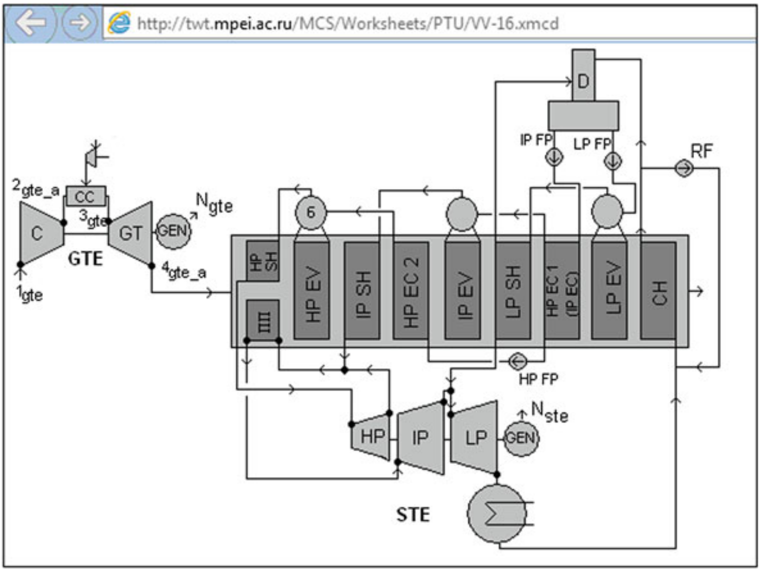


Fig. 14.12 CCPP with triple-pressure reheat heat recovery steam generator

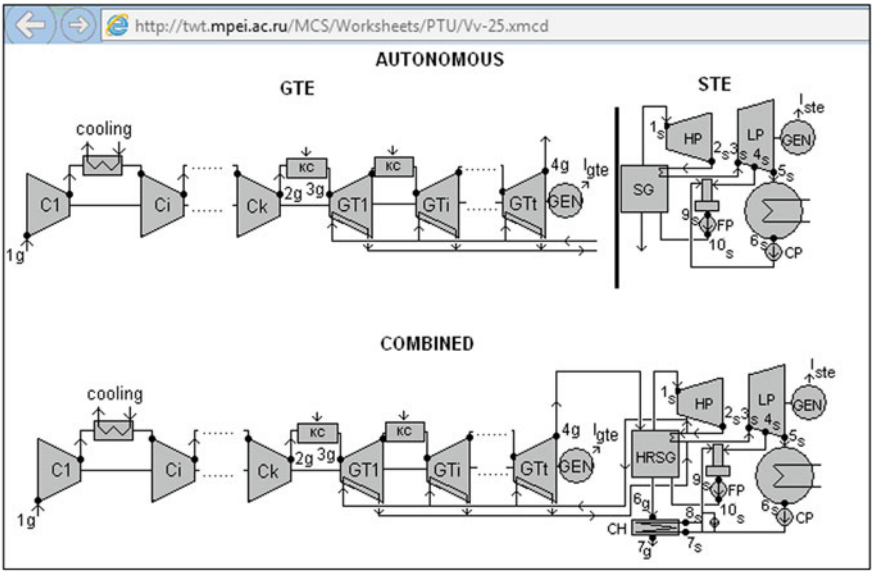


Fig. 14.13 Comparison of gas turbine power plant and steam turbine power plant with CCPP

Chapter 15
Otto Cycle or What Is Behind the Simplified Formula

Valery Ochkov and Konstantin Orlov

Abstract This chapter explains how the usual simplified formula to calculate thermal efficiency of an internal combustion engine is appears. Described Internet sites with the "reverse" design challenge—you must not to enter raw data and get a response, but you must enter formulas for calculations and verify their correctness (educational task).

Otto cycle is one of the thermodynamic cycles used in the internal combustion engines. It comprises (in its ideal version) two isochors (combustion of air-fuel mixture in the engine cylinder and gas exhaust) and two isentropes (compression of fuel-air mixture in the cylinder and expansion of gases in a working stroke of a piston). In reference books and textbooks on thermodynamics one can find a formula used to calculate the Otto cycle thermal efficiency. Figure 15.1 displays a fragment of a web page with this formula for its interactive use: one can change the compression ratio **r** (ratio of the cylinder maximum volume to its minimum volume), $\mathbf{C_p}$ and $\mathbf{C_v}$ (isobaric and isochoric) heat capacity of the working fluid in Otto cycle linked to the universal gas constant **R**), and obtain a new η_t value to fix the points on the efficiency/compression rate function diagram [58].

The site of the chapter: https://www.ptcusercommunity.com/message/423036.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_15

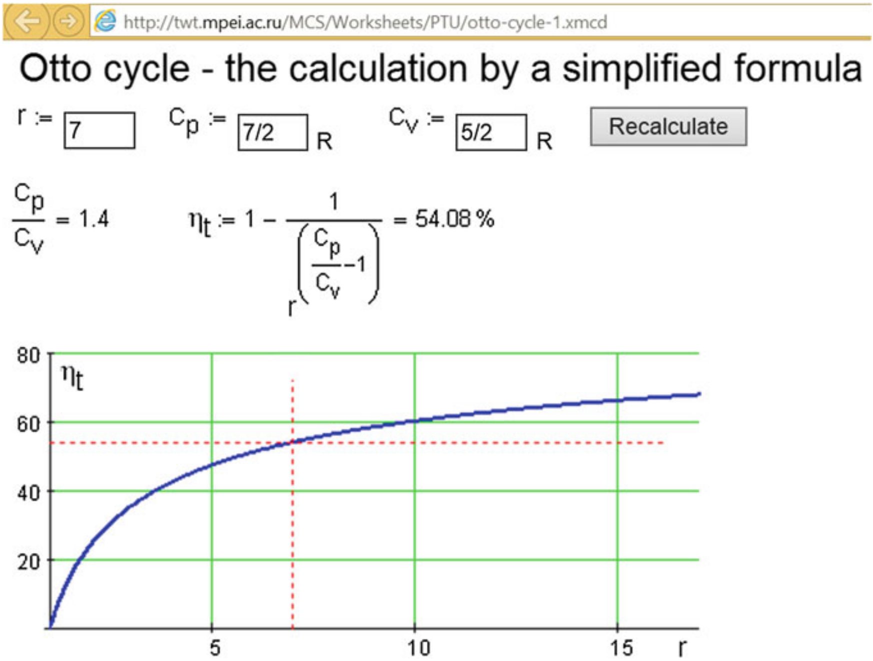


Fig. 15.1 Otto cycle thermal efficiency

Similar formulas for other cycles of internal combustion engines (Diesel, Brayton, Trinkler, Kalina, etc.) can be found in the Internet (see e.g. http://en.wikipedia.org/wiki/Thermodynamic_cycle).

What is the origin of the formula in Fig. 15.1? Figure 15.2 shows the calculation of gas parameters in four nodal points of the Otto cycle using the basic concepts (definitions) of thermodynamics: what is entropy **S** and what is the enthalpy **U**.

Specific entropy is the sum of two integrals: the integral temperature from $\mathbf{T_0}$ (you can take any) to **T** (the temperature of the working fluid at the point under consideration) and integral pressure from $\mathbf{p_0}$ (again take any value) to **p** (pressure at the consideration point).

The specific enthalpy is the integral of the specific isochoric heat capacity at temperatures from $\mathbf{T_0}$ to **T**.

In the calculation shown in Fig. 15.2, in points 2 (the end of the compression of the fuel-gasoline mixture) and 4 (the end of the piston working stroke), the equation is solved where the unknowns are the integration limits corresponding to temperature and pressure values. At the end of the calculation shown in Fig. 15.2, there are two answers: received through pointwise calculation (see it in the frame), and using a formula obtained in the result of preliminary symbolic integration. The two results match.

If the integrand contains a constant factor—a constant (and in our calculation it is assumed that $\mathbf{C_p}$ and $\mathbf{C_v}$ are independent of temperature), then this constant can

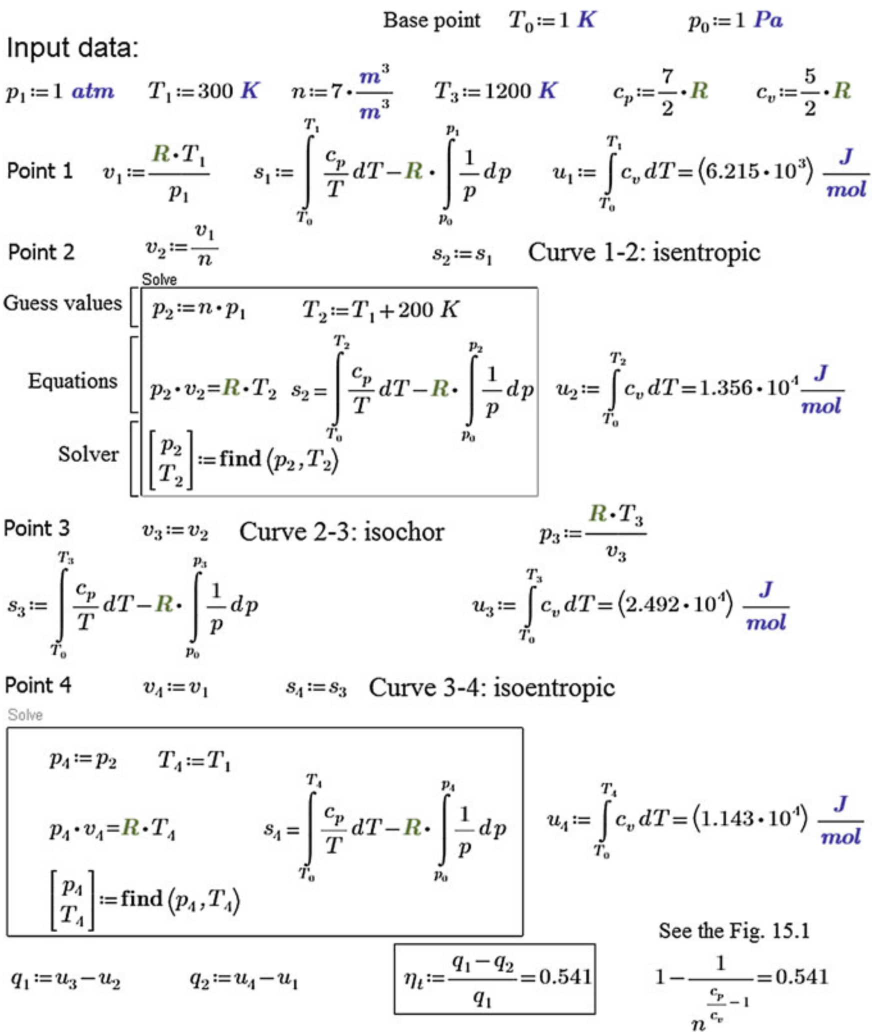


Fig. 15.2 "Pointwise" calculation of Otto cycle thermal efficiency

be taken outside the integral sign and we take it without a constant, and then multiply by this constant. Figures 15.3 and 15.4 show how Mathcad symbolic processor did that.

The specific entropy calculation is of interest in terms of measuring units used for temperature and pressure (see Chap. 2). Mathcad symbolic processor gave a response containing a difference between the logarithms. To use this formula for numerical calculations, it is necessary to recall that the difference between the logarithms is equal to the logarithm of the fraction (see Fig. 15.3), and adjust the formula.

$$\int_{t_0}^{t} \frac{c_p}{T}\,dT - R\int_{p_0}^{p} \frac{1}{P}\,dP \rightarrow c_p\cdot(\ln(t)-\ln(t_0)) - R\cdot(\ln(p)-\ln(p_0))$$

$$c_p := \frac{7}{2}R = 29.101\ \frac{J}{mole\cdot K} \qquad t_0 := K \qquad p_0 := Pa$$

$$t := 18\ °C \qquad p := 1\ atm$$

$$s := c_p\cdot(\ln(t)-\ln(t_0)) - R\cdot(\ln(p)-\ln(p_0))$$

Units?

$$s := c_p\cdot\ln\left(\frac{t}{t_0}\right) - R\cdot\ln\left(\frac{p}{p_0}\right) = 69.279\ \frac{J}{mole\cdot K}$$

Fig. 15.3 Ideal gas specific entropy calculation

If in the calculation shown in Fig. 15.2 we proceed with transformation and simplification, such as those shown in Figs. 15.3 and 15.4, we finally obtain the formula, as shown in Fig. 15.1.

But it should be remembered that the specific isobaric and isochoric heat capacity of the real working fluid in Otto cycle is a function of temperature and pressure, and the composition of the working fluid changes. Therefore, the calculation shown in Fig. 15.2, although complicated, but is more interesting in terms of its improvement and approximation to the real thermodynamic process—the cycle of the internal combustion engine, Otto cycle.

On the site of the book the reader will find other interactive pages on the Otto cycle calculation. Figure 15.5 shows the website where visitors can:

Fig. 15.4 Ideal gas enthalpy calculation

$$\int_{t_0}^{t} c_v\,dT \rightarrow c_v\cdot(t-t_0)$$

$$c_v := \frac{5}{2}R = 20.786\ \frac{J}{mole\cdot K}$$

$$t_0 := K$$

$$t := 18\ °C$$

$$u := c_v\cdot(t-t_0) = 6031.1\ \frac{J}{mole}$$

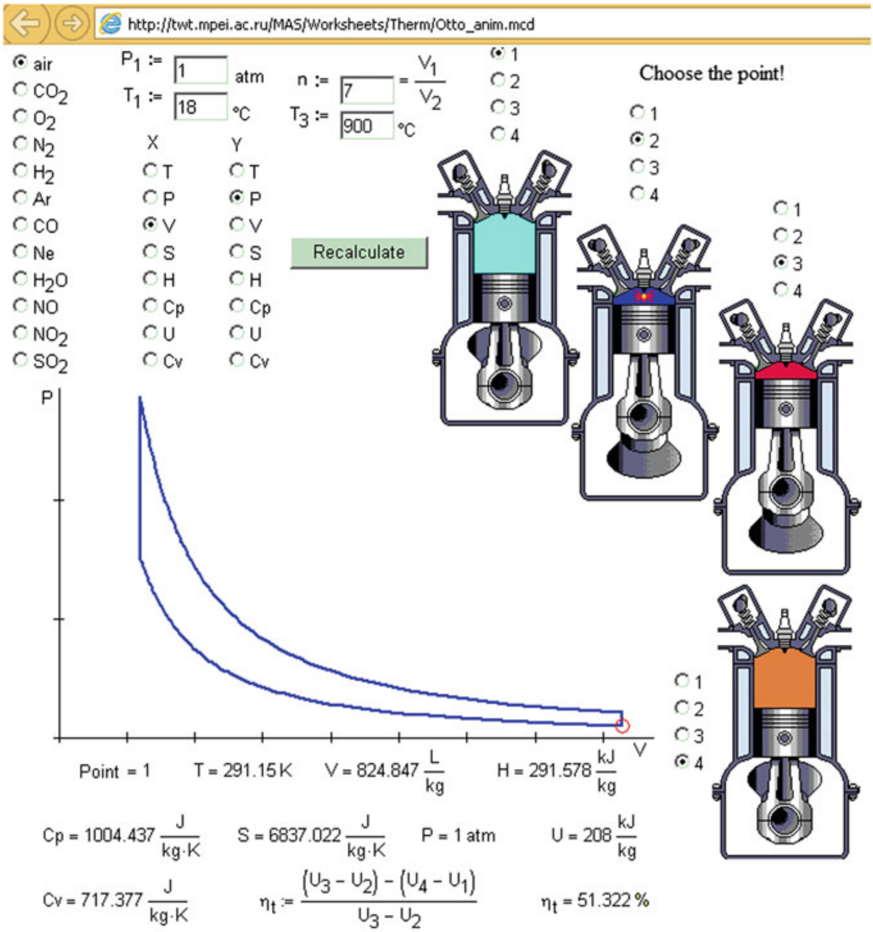


Fig. 15.5 The site with the calculation and graphical display of the Otto cycle

- select the working fluid of the cycle;
- specify its parameters ($\mathbf{p}_1$, $\mathbf{T}_1$, **n** и $\mathbf{T}_3$);
- choose the type of chart that displays the process;
- specify the point at which the piston position will be shown (here one can see all the drawings, but the website will display one of them showing the corresponding point on the Otto cycle diagram).

Figure 12.6 in Chap. 12, which describes the techniques for creating animation in Mathcad's environment, shows the Otto cycle animation, which is placed in the Internet on Mathcad PTC Community forum.

The calculation in Fig. 15.2 can be transformed so that it became a kind of test task for students (Figs. 15.6 and 15.7).

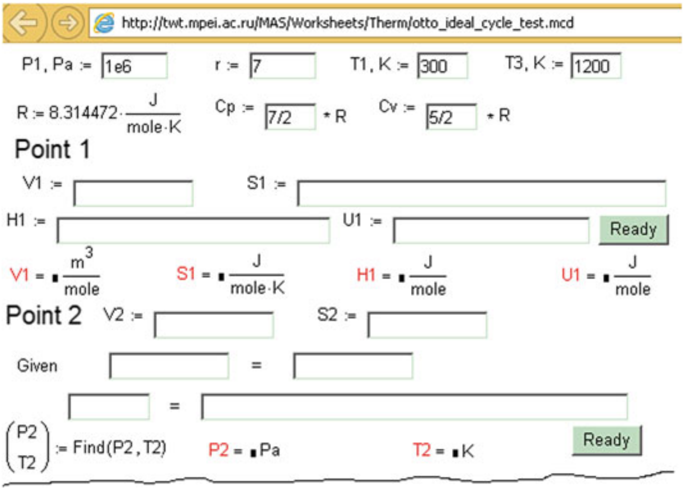


Fig. 15.6 Start of the test on knowledge of the calculation of the Otto cycle

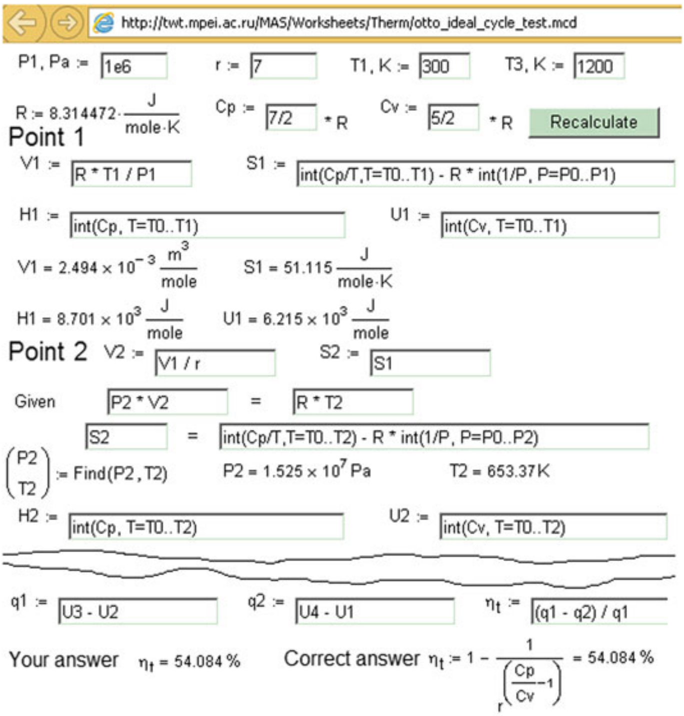


Fig. 15.7 End of the test on knowledge of the calculation of the Otto cycle

If, calculation in Fig. 15.2 demonstrates all the formulas, then the website in Fig. 15.7, instead of formulas offers empty text fields to be filled with the relevant formulas to be used to calculate the thermal efficiency of the ideal Otto cycle. If you fill out all of the formulas and press **Ready** key, at the end of the online test (see Fig. 15.7) you will be given an answer (thermal efficiency of the Otto cycle), which can be compared with the response issued by the simplified formula. Formulas entered in the fields shown in Figs. 15.6 and 15.7, are written not by the operator (via operator of the definite integral, for example, as shown in Figs. 15.2, 15.3 and 15.4), but in the form of text functions—int, / (division), * (multiplication), etc.

Chapter 16
Calculation of Pressure Losses in the Tube

Valery Ochkov and Konstantin Orlov

Abstract This chapter explains how to calculate the pressure loss in the pipeline using cloud functions for thermodynamic and transport properties of the fluid. A description is given of finite element methods for the solving of important engineering problems.

This chapter is designed to show how one can use "cloud" functions to solve a simple but typical hydrotechnical problem i.e. the problem of the loss of water head in the horizontal tubing. Such problem occurs in the design of a (cooling) water supply loop from the cooling tower to the steam turbine condenser with a heated water return to the cooling tower.

Loss of head (pressure) takes place also in the steam supply tubing from the boiler to the turbine. Chapter 5 (Fig. 5.1) calculates and shows the process of steam "delivery" from the boiler to the turbine in hs- and Ts-diagram on assumption that the initial pressure is 24 MPa and the final pressure is 10 MPa.

In the author's website there is a web page making possible to calculate the process of steam expansion (throttling) and to show it in the hs-diagram (Fig. 16.1). If we set the value of relative internal efficiency of the steam expansion process equal to 100 %, then we obtain the process of ideal steam operation in the turbine while steam entropy remains constant ("vertical" expansion of steam). But if we set

The site of the chapter: https://www.ptcusercommunity.com/message/423038.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_16

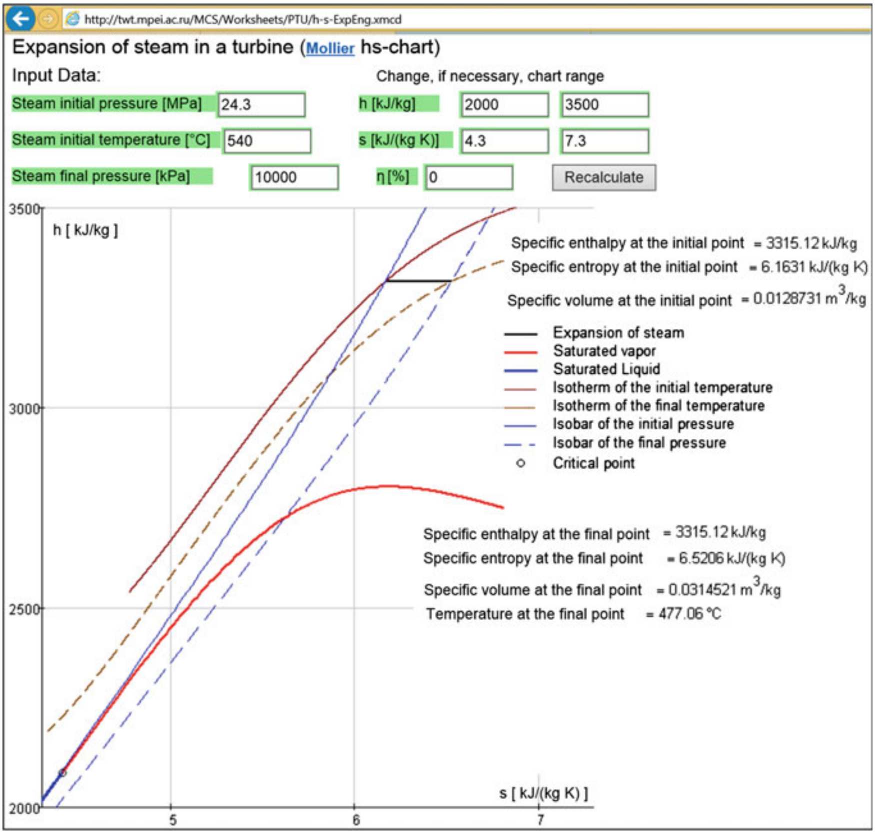


Fig. 16.1 Throttling of steam in the web site hs-diagram

the value of relative internal efficiency of the steam expansion process equal to 0, then we obtain the process of ideal throttling of steam while steam enthalpy remains constant ("horizontal" expansion of steam). In the actual steam operation in the turbine the value of internal efficiency of the process is close to 100 % while in the actual throttling it is close to zero. A visitor of the web site shown in Fig. 16.1 can, at his(her) discretion, change the value of the relative internal efficiency of the process and obtain the required steam expansion curves.

Figure 9.9 in Chap. 9 displays a diagram of wet steam throttling making possible to evaluate the rate of steam dryness in the initial point if one knows the initial pressure, the pressure and the temperature in the final point.

As we have noted before, a great number of computer programs have been created to calculate tubes of various degree of complexity and details. Such programs operate on the principle of a "black box" into which one "puts" initial data array, "closes the cover of the box" (by pressing the key "**Recalculate**"), "opens the cover" and "takes out" of the box the solution i.e. the parameters of the designed or operated tubing. But as it has been already noted in the introduction, one always wants to know what it is inside of such "black box" and whether the calculations are being correctly made. In addition, it is useful to slightly open the cover of such "box" for the purposes of education and self-education—for the study of mathematical models built in these or those calculation programs. Here we will now consider such particular mathematical model!

In order to solve the problem of the loss of head in a tube an engineer will recall or find in the reference books (in the printed reference books or in the Web-versions of reference books), a set of the corresponding formulae and rules of their application—formulations. In addition, the engineer must know some properties of water, in particular for this problem—its kinematic viscosity **ν**. Values of such water parameters are given in many various reference books in the form of tables, graphs or empirical formulae related water viscosity to its temperature. As a rule pressure is assumed to not affect this property. However, data taken from the tables and formulae of rather fat reference books can considerably differ. In addition, as we have already noted many times, when transferring figures from the tables to the particular calculation an engineer can make a mistake connected with incorrect interpolation and even with incorrect entry of digits on the keyboard and/or incorrect interpretation of measurement units of viscosity and their multipliers. All these things can complicate and hinder calculations and increase the possibility of errors.

Dynamic viscosity of water is calculated by the formulation developed and approved by International Association of Properties of Water and Steam (IAPWS—www.iapws.org) with which the author of this book closely cooperates. Based on the IAPWS formulations reference books are issued all over the world with the tables keeping discrete values of water and steam parameters as functions of temperature and pressure. Such tables are published in our country as well. The most recent reference book [23] is added with the Web-version (http://twt.mpei.ac.ru/rbtpp) to make the use of the reference book easier and faster. One of the pages from the reference book at the URL address http://twt.mpei.ru/MCS/Worksheets/rbtpp/tab9.xmcd is shown in Fig. 16.2.

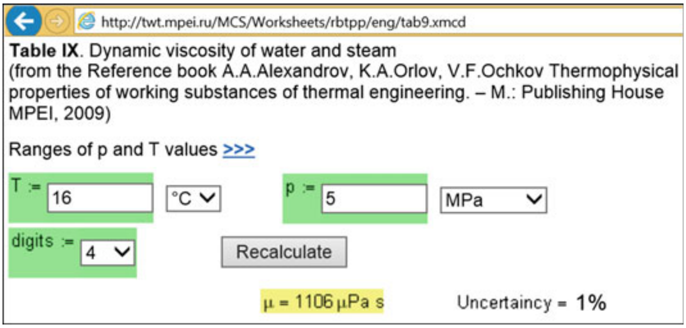


Fig. 16.2 Web page with the on-line calculation of dynamic viscosity of water

By getting into the Web site shown in Fig. 16.2, it is possible to change the temperature and/or pressure values selecting the required measurement units of these initial values, set the number of significant figures in the response, press the key **Recalculate** and obtain not only the required value of water or steam dynamic viscosity but as well the value of relative error of this figure that is different in different pressure and temperature ranges. It is possible to slightly open such “black box” for calculations and to learn from this Web-version of the reference book the formulae used for calculations of water and/or steam dynamic viscosity.

As we have already noted, in Moscow Power Engineering Institute at the Power Plant department they have created the new Internet technology to operate with water and steam properties excluding the manual transfer of data from the printed [23] or Web-version of the reference (Fig. 16.2) book.

The reference book [23] compiles not only on-line calculation of water and/or steam properties but also the corresponding programmed functions that one can directly use in the calculations by setting e.g. not water dynamic viscosity but water temperature and water. The beginning of a list of such functions (“cloud” functions—Web cloud-based functions) is shown in Fig. 16.3.

Such list contains the water density and dynamic viscosity values required to calculate the loss of water head in the tubing. In order to make the water density value versus temperature and pressure values accessible (or visible as programming specialists say) in the Mathcad-calculation, it is necessary to know the URL of this “cloud” function http://twt.mpei.ac.ru/rbtpp/MC-WSP/M15/wspDYNVISCTD.xmcdz. For this purpose it is enough to point the cursor to the necessary function (see the underlined names in Fig. 16.3), right-click and choose the command **Properties**. Such action will open the following window with the URL of the required function (Fig. 16.4).

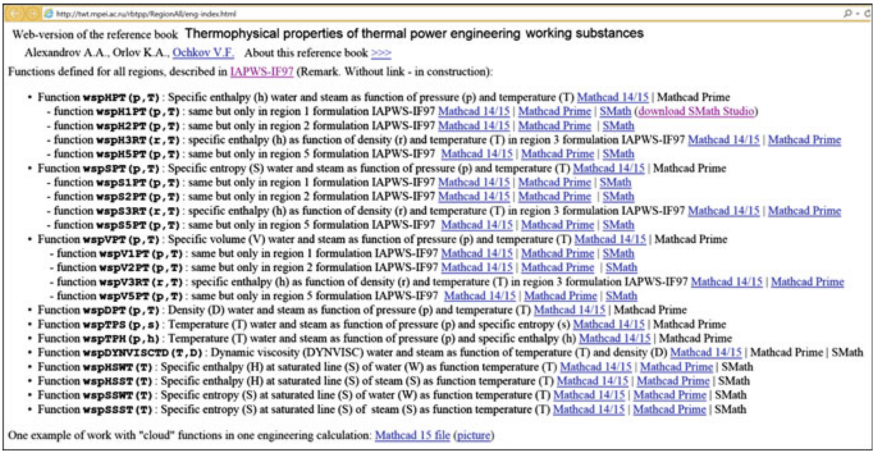


Fig. 16.3 List of “cloud” functions of properties of water and steam

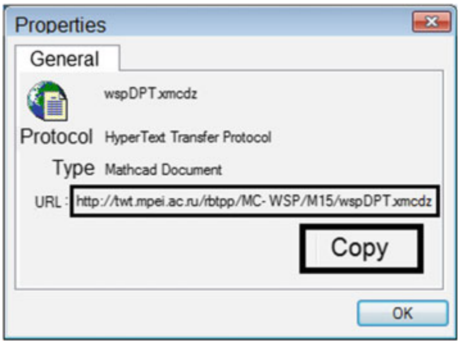


Fig. 16.4 URL of the "cloud" function "Water and steam dynamic viscosity"

It is necessary to copy this URL address and insert it in the dialog box **Insert Reference in the** Mathcad program which calculates the loss of water head in the pipeline or makes any other calculation requiring the knowledge of the water density value ρ (Fig. 16.5).

The same way it is possible to insert in the calculation the function returning the water dynamic viscosity μ as a function of temperature and pressure. The URL of this "cloud" function is http://twt.mpei.ac.ru/rbtpp/MC-WSP/M15/wspDYNVISCTD.xmcdz.

The value of kinematic viscosity ν is determined by the values of water density and its dynamic viscosity in the calculation shown in Fig. 16.6. In the previous calculation the internal tube diameter value d was entered and the value of water velocity in the pipe v was calculated. These three parameters (velocity, diameter and viscosity) are used to calculate very important dimensionless criterion—the

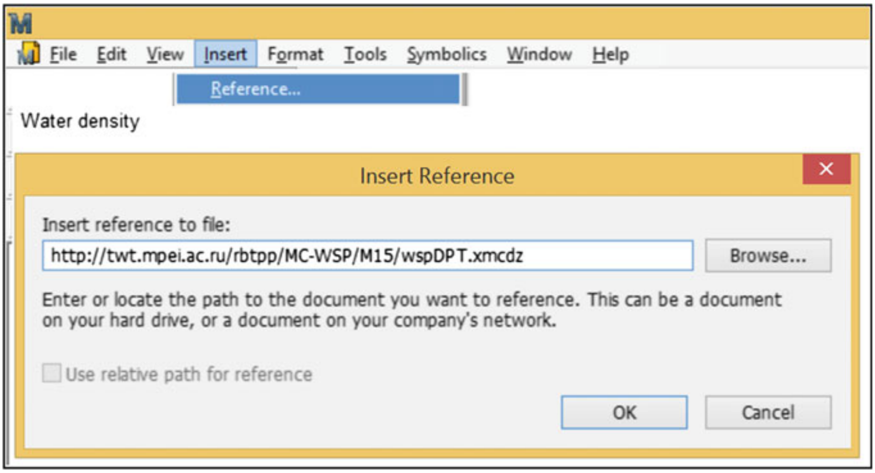


Fig. 16.5 Inserting reference to the "cloud" function of water density

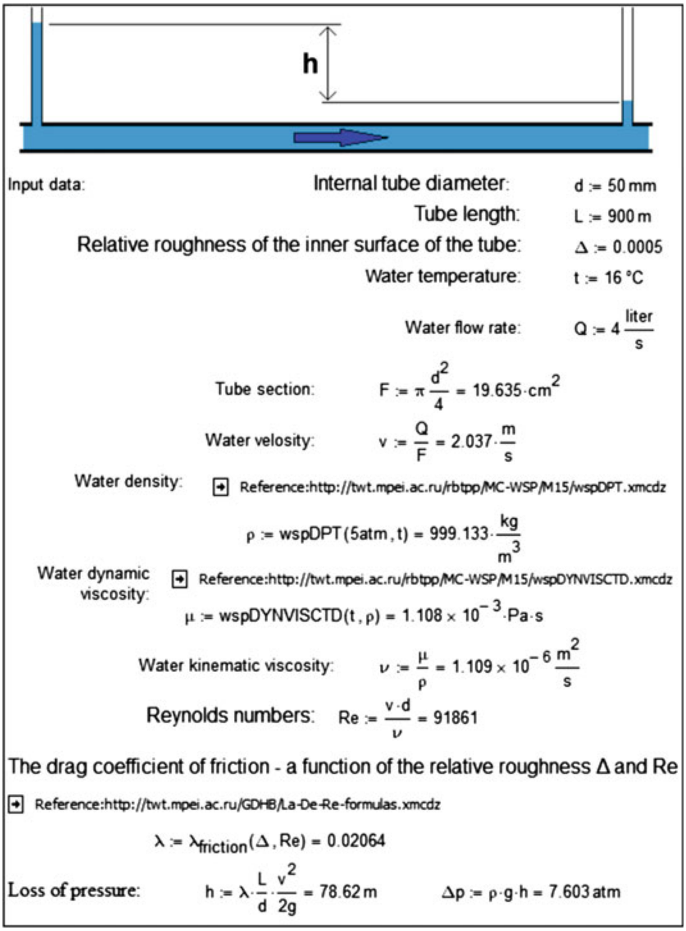


Fig. 16.6 Calculation of loss of pressure in the tubing

criterion of liquid flow in the tube i.e. Reynolds' number Re, which determinates not only the tube inner surface roughness but as well the desired value of the loss of water head due to wall friction.

Reference books show the relationship between the relative dimensionless resistance factor and the mode of liquid flow in a round tube λ either in the graphs or by a set of formulae. Figure 16.7 shows a fragment of the Web page at URL http://twt.mpei.ac.ru/MCS/Worksheets/Hydro/La-De-Re.xmcd displaying the "live" nomograph (Nikuradse curve) relating the resistance factor λ under our consideration with the Reynolds' number Re and the relative roughness of surface **Δ** (relationship between medium height of asperities (roughness) of the tube inner surface ant its internal diameter).

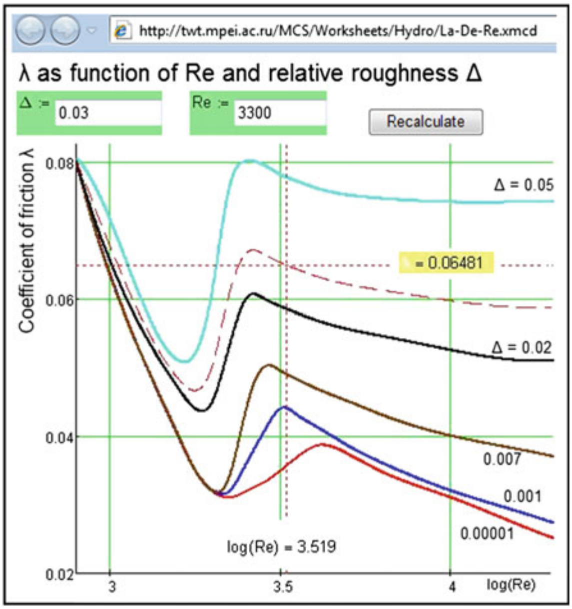


Fig. 16.7 Characteristic curve of relative coefficient of friction and parameters of water flow in the round tube

A visitor of the web site (Fig. 16.7) can change the value **Δ** and **Re**, press the key **Recalculate** and obtain (read) not only the desired figure but also can see the character of this relationship in the most "interesting" area—in the transition area from laminar flow to turbulent flow—the area where it is not recommended to operate tubing. The areas to the left (liquid laminar flow) and to the right (enhanced turbulence) of the curves shown in Fig. 16.7 are calculated by the following formulae in Fig. 16.8. At the calculation portal where the web sites described in the books are located, there is a fluid and gas dynamics reference book.

In this Web-reference book the "cloud" function is stored that returns the value of the required friction coefficient as a function of the Reynolds' number in the wide range and relative roughness of the internal surface of the round tube. This function

Fig. 16.8 Set of formulae for calculation of the relative coefficient of friction versus parameters of water flow in the round tube

$$\lambda_{friction}(\Delta, Re) := \begin{array}{l} \text{return } \frac{64}{Re} \text{ if } Re \le 2300 \\ \text{return NaN if } 2300 < Re \le 4000 \\ \frac{0.3164}{Re^{0.25}} \text{ return if } 4000 < Re \le \frac{10}{\Delta} \\ 0.11 \cdot \left(\Delta + \frac{68}{Re}\right)^{0.25} \text{ return if } \frac{10}{\Delta} < Re < \frac{560}{\Delta} \\ 0.11 \cdot \Delta^{0.25} \text{ otherwise} \end{array}$$

$$\lambda_{friction}(0.0005, 91861) = 0.02064$$

is shown in Fig. 16.8 and entered in the calculation shown in Fig. 16.6 by means of the above tool of the reference to the "cloud" function.

In the literature one can find various formulae for calculation of the friction coefficient as a function of **Re** and **Δ** (Fig. 16.8 and table in Fig. 17.10 in Chap. 17). There are other formulae for these ranges and one can take, in principle, any formula from this set of formulae without thinking about their accuracy. The matter is that the initial parameter i.e. roughness of the tube is very roughly estimated.

One can only approximately measure roughness of the tube. Usually it is done in the process of hydraulic tests of tubes. Reference literature gives this important parameter within some limits depending on the material of the tubes (metal, plastic, ceramics, copper etc.), the method of their manufacturing (a welded tube, a seamless drawn tube etc.) and other manufacturing features. In addition, roughness of the tube inner surface can change in the process of its operation or at a standstill due to corrosion and erosion or due to deposits in the tubes. In principle, when calculating the loss of head one shall not enter absolute values of the relative roughness **Δ** as initial data but certain linguistic constants ("a seamless drawn tube", "a plastic tube" etc.) which can be used by the method of fuzzy sets for evaluation of the value **Δ** in a certain range determining along with other uncertainties (i.e. uncertainty of the range of Reynolds' number values and the formulae for calculation of **λ** in Fig. 16.8) the expected range of possible change of the calculated value of the loss of head.

In the calculation shown in Fig. 16.6 water parameters along the length of the tube are determined by the given temperature of 16 °C and the pressure equal to 5 atm. This is rather rough approximation which we will try to solve now.

Water pressure along the length of the tube varies, of course. As a rule, it goes down along the liquid flow but it can also rise due to hydrostatic pressure if the liquid in the tube is fed from bottom to top. But properties of liquids, water in particular, weakly depend on pressure. They greatly depend on temperature. And how will the liquid temperature and pressure change along the water flow? If water at a temperature higher than ambient air temperature is pumped by the tube, then it is necessary to consider in the calculation the loss of heat through the tube wall. The attempt to solve the problem is shown in Figs. 16.9 and 16.10.

The initial data will be the following values i.e. internal diameter of the tube **d**, length of the tube **L**, tube wall thickness $\boldsymbol{\Delta}_T$, relative roughness of the inner surface of the tube **Δ**, water temperature **t** and pressure **p** at the inlet of the tube, water mass flow **Q**, tube wall material heat conduction, **λ**, heat-transfer coefficient from water to the inner surface of the tube $\boldsymbol{\alpha}_1$ and from the outer surface of the tube to ambient environment $\boldsymbol{\alpha}_2$ as well as ambient temperature tout.

For calculation the tube is divided into n sections (finite elements). Calculation of the initial (zero) section is shown in Fig. 16.9.

This figure shows the successive calculation of the following values after the input of the initial data: area of the inner section of the tube **F**, length of the tube finite element **Δl**, its inner **r** and outer **R** radius. During the program debugging n (the number of finite elements of the tube sections) can be set as small (10–100) and after the program is ready it can be increased to 1000–10,000.

$$d := 20\,\text{mm} \quad \Delta\tau := 5\,\text{mm} \quad Q := 15\frac{\text{kg}}{\text{min}} \quad t := 230\,°\text{C} \quad tout := 17\,°\text{C} \quad p := 4.5\,\text{MPa}$$

$$L := 150\,\text{m} \quad Q := 50\frac{\text{kg}}{\text{min}} \quad \lambda := 0.15\frac{\text{W}}{\text{m}\cdot\text{K}} \quad \alpha1 := 500\frac{\text{W}}{\text{m}^2\cdot\text{K}} \quad \alpha2 := 20\frac{\text{W}}{\text{m}^2\cdot\text{K}} \quad n := 1000$$

$$F := \pi\frac{d^2}{4} = 3.142\cdot\text{cm}^2 \quad \Delta l := \frac{L}{n} = 0.15\cdot\text{m} \quad r := \frac{d}{2} = 10\cdot\text{mm} \quad R := r + \Delta\tau = 15\,\text{mm}$$

$$T_0 := t = 230\cdot°\text{C} \quad P_0 := p = 4.5\cdot\text{MPa} \quad L_0 := 0\text{m}$$

$$N_0 := \frac{2\pi\cdot\Delta l\cdot(T_0 - tout)}{\frac{1}{\alpha1\cdot r} + \frac{1}{\lambda}\cdot\ln\left(\frac{R}{r}\right) + \frac{1}{\alpha2\cdot R}}\cdot\frac{1}{Q} = 38.63\cdot\frac{\text{J}}{\text{kg}} \quad \rho_0 := \text{wspDPT}(P_0, T_0) = 828.758\frac{\text{kg}}{\text{m}^3}$$

$$v_0 := \frac{Q}{F}\cdot\frac{1}{\rho_0} = 3.2\frac{\text{m}}{\text{s}} \quad h_0 := \text{wspHPT}(P_0, T_0) = 990.536\,\text{kJ/kg}$$

$$\mu_0 := \text{wspDYNVISPT}(P_0, T_0) = 1.167\times10^{-4}\cdot\text{Pa s} \quad Re_0 := \frac{\rho_0\cdot v_0\cdot d}{\mu_0} = 4.548\times10^5$$

$$\Delta H_0 := \lambda_{friction}(\Delta, Re_0)\cdot\frac{\Delta l}{d}\cdot\frac{(v_0)^2}{2g} + \text{if}(Var = 1, 0\text{m}, \text{if}(Var = 2, -\Delta l, \Delta l)) = 68.791\,\text{mm}$$

$$\Delta P_0 := \Delta H_0\cdot g\cdot\rho_0 = 0.559\cdot\text{kPa}$$

Fig. 16.9 Calculation of the initial section of the tubing (**Var = 1**: horizontal tubing)

After the value n is set, the length of the tube finite element Δl is calculated. Then it is necessary to enter the initial (zero) elements of vectors storing the following values on this finite elements: **T**—water temperature, **P**—water pressure, **N**—specific loss of heat (enthalpy), **ρ**—water density, **v**—water velocity, **h**—water specific enthalpy, **μ**—water dynamic viscosity, **Re**—Reynolds number, **ΔH**—loss of head and **ΔP**—loss of pressure.

The formula determining the loss of head **ΔH** contains the summand with if function and additional parameter Var i.e. variant. There are three variants here:

- **Var = 1**: horizontal tubing;
- **Var = 2**: vertical tubing, pump at the top;
- **Var = 3**: vertical tubing, pump at the bottom.

Here we consider some extreme cases. Actual tubing can contain various sections hugging the terrain and/or the tubing geometry at different process plants.

The most complicated formula is the formula for determination of water specific loss of heat (enthalpy) N. This is a standard formula developed for solving the problem of steady heat conduction of the coiled cylindrical tubing if there are no internal heat sources. The boundary conditions of this problem will be the values of the heat-transfer coefficient on the inner α_1 and outer α_2 surfaces of the tube. Determination of $\boldsymbol{\alpha}_1$- and $\boldsymbol{\alpha}_2$-values is an individual and not least complicated problem. For the time being we will set these values as constants. However, in the

$$M := \text{for } i \in 1..n$$

$$h_i \leftarrow h_{i-1} - N_{i-1}$$

$$P_i \leftarrow P_{i-1} - \Delta P_{i-1}$$

$$T_i \leftarrow \mathrm{wspTPH}(P_i, h_i)$$

$$N_i \leftarrow \frac{2\pi \cdot \Delta l \cdot (T_i - tout)}{\frac{1}{\alpha 1 \cdot r} + \frac{1}{\lambda} \cdot \ln\left(\frac{R}{r}\right) + \frac{1}{\alpha 2 \cdot R}} \cdot \frac{1}{Q}$$

$$\rho_i \leftarrow \mathrm{wspDPT}(P_i, T_i)$$

$$v_i \leftarrow \frac{Q}{F} \cdot \frac{1}{\rho_i}$$

$$\mu_i \leftarrow \mathrm{wspDYNVISPT}(P_i, T_i)$$

$$Re_i \leftarrow \frac{\rho_i \cdot v_i \cdot d}{\mu_i}$$

$$\Delta H_i \leftarrow \lambda_{friction}(\Delta, Re_i) \cdot \frac{\Delta l}{d} \cdot \frac{(v_i)^2}{2g} + \mathrm{if}(Var = 1, 0m, \mathrm{if}(Var = 2, -\Delta l, \Delta l))$$

$$\Delta P_i \leftarrow \Delta H_i \cdot g \cdot \rho_i$$

$$L_i \leftarrow L_{i-1} + \Delta l$$

$$\mathrm{augment}\left(\frac{P}{Pa}, \frac{T}{K}, \frac{L}{m}\right)$$

$$P := M^{\langle 0 \rangle} \cdot Pa \qquad T := M^{\langle 1 \rangle} \cdot K \qquad L := M^{\langle 2 \rangle} \cdot m$$

Fig. 16.10 Element-by-element calculation of the tubing

process of calculation we can enter in it the formulae for determination of $\boldsymbol{\alpha}_1$- and $\boldsymbol{\alpha}_2$-values depending on the mode of water flow in the tube and other factors.

Figure 16.10 shows Mathcad operators entering the rest elements of the above mentioned vectors.

Figure 16.10 shows the enumeration cycle of all tubing sections starting from the first one. Zero (initial) tubing section has been calculated before (Fig. 16.9) outside of the cycle including for the purpose of debugging of the calculation program.

The cycle in Fig. 16.10 shows the calculation of the successive *i* section of the tubing with the use of the data for the previous (i – 1) section. The matrix **M** is added with three columns i.e. data for change of water pressure **P** and water temperature **T** as well as vector **L** storing the values of distances from the beginning of the tube to the point *i*. During the derivation (ref. to the last operator in the program in Fig. 16.10 with the function augment[1]) these data are to be firstly

[1]This function built in Mathcad joins individual vectors in a matrix where these vectors become columns.

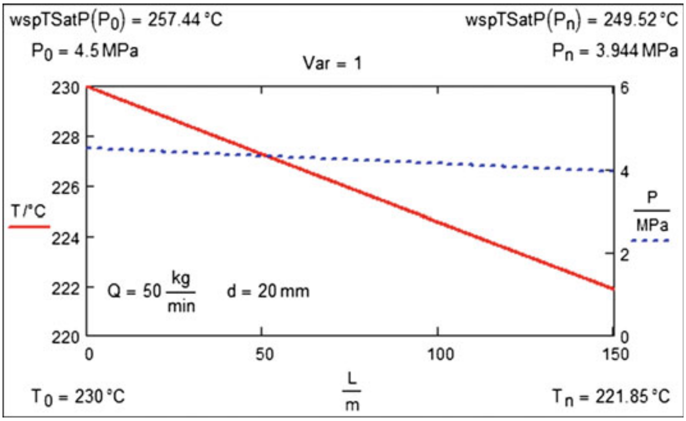


Fig. 16.11 Water temperature and pressure curves in the horizontal tubing

deprived of dimensions of pressure, temperature and length and then one has to return them in new vectors **P**, **T** and **L** generated outside of the program (ref. to the last three operators in Fig. 16.10). As we have already noted many times, this is connected with the main disadvantage of Mathcad 15 (as the calculation shown in Figs. 16.9 and 16.10 has been created in this program) i.e. impossibility of storing data of different dimensions in the arrays.

Figures 16.11, 16.12 and 16.13 show temperature and pressure curves in our tubing located horizontally (Fig. 16.11), vertically with water feed from bottom to top (Fig. 16.12) and from top to bottom (Fig. 16.13).

The curves also show the values of water saturation at the initial and final pressure in the tubing calculated by means of the function **wspTSatP**. And this is being done for some reason. The fact is that the calculation shown in Figs. 16.9,

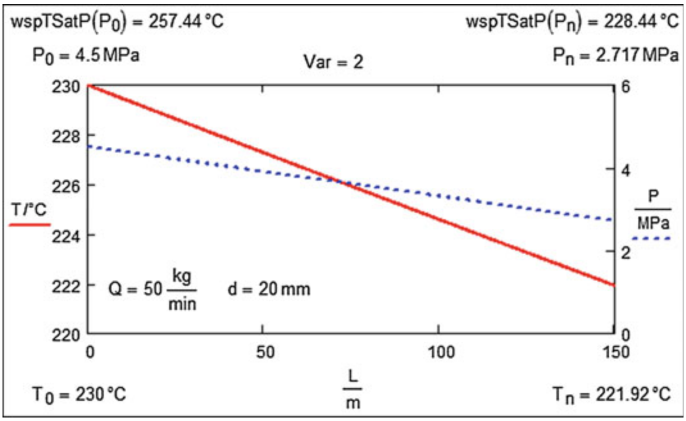


Fig. 16.12 Water temperature and pressure curves in the vertical tubing (pump at the top)

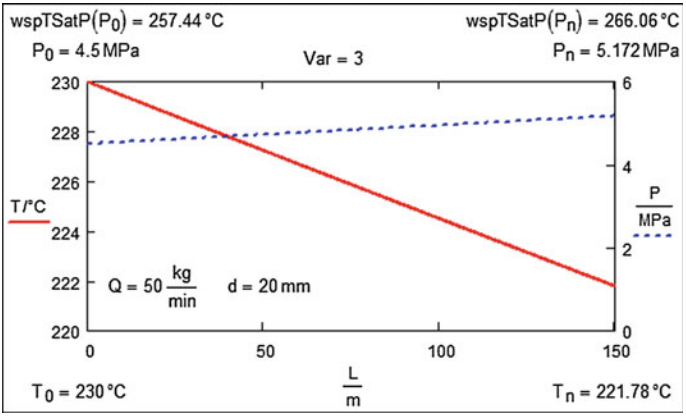


Fig. 16.13 Water temperature and pressure curves in the vertical tubing (pump at the bottom)

16.10, 16.11, 16.12 and 16.13 is the attempt to simulate the process of pressure hot water feed by a rubber hose for fire extinguishing. At the end of the hose there is a nozzle through which the water that has passed into wet steam cloud extinguishes fire.

When feeding such water it is important to avoid its early boiling inside of the rubber tube. The most risky case here is water feed from top to bottom. If in the two first cases (horizontal tubing and vertical tubing with water feed from bottom to top) the loss of pressure can be not so considerable, then in the third case the loss of head of water can cause sharp loss of pressure and flashing of water in the rubber tube.

On-line calculation of water parameters in the tube can be made at http://twt.mpei.ac.ru/MCS/Worksheets/Hydro/TubeMChS.xmcd.

It is also possible to make on-line evaluation of the change of temperature inside the tube wall. For this purpose it is enough to get in the Web site http://twt.mpei.ac.ru/MCS/Worksheets/Thermal/Heat-Flow-1radius.xmcd, enter the required initial data and obtain the result (Fig. 16.14).

One would always like to simplify the key formula for calculation of the loss of head (this formula is in the box in Fig. 16.6) by transferring e.g. constants 2 and g (gravity acceleration) from the denominator to the coefficient of resistance, **λ**. But we cannot do this for several reasons. Firstly, the gravity acceleration g (9.81 m/s^2) is not a constant, strictly speaking, but a function of other values and, mainly, a function of the altitude above sea level and terrestrial latitude. If one makes reference in Mathcad-document to the "cloud" function at http://twt.mpei.ac.ru/TTHB/g-h-psi.xmcdz, then in the Mathcad environment the constant **g** will turn into the function **g(h, φ)**, where **h** is the altitude above sea level and **φ** is the terrestrial latitude (0°—the Equator, 90°—Earth poles). Besides this, one should remember that some time in future tubing can be designed and built e.g. in the … Moon or Mars where the value of g is absolutely not 9.81 m/s^2! And this is the

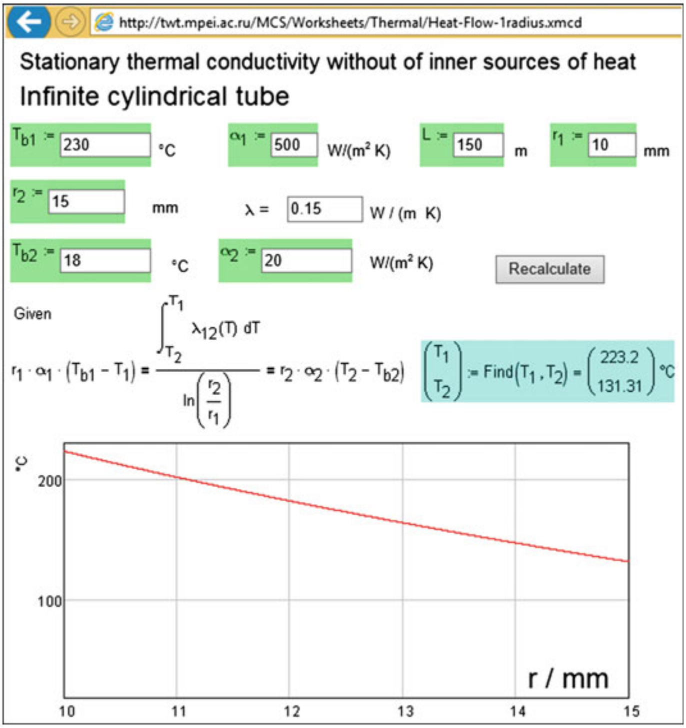


Fig. 16.14 Temperature profile in the cylindrical wall

interesting question: is our formula applicable for determination of the loss of head in a water supply line or fuel supply line at the space station where g = 0? Incidentally, in many calculations g is taken equal to 10 m/s^2 and water density is taken as 1000 kg/m^3 to simplify the counting and with due regard to the fact that calculations are rather rough. We also can use the loss of head we found in Fig. 16.2 for rather rough estimation of pressure drop in the environments required for pumping water by the tube. For this purpose one shall divide the head value by ten.

So, the constant g cannot be removed from the formula of calculation of the loss of head in the tubing and transferred to the coefficient λ. And what can we do with "two" in the denominator of this formula? It also cannot be transferred to the coefficient of resistance, λ, but for other reasons. Looking at two and velocity squared in the formula of the loss of head, one can recall an old joke. A student was in a hurry to the examination in physics, he became thoughtful and… suddenly run into a post. "That's good that it's divided in half!" exclaimed the student. "What's in half?" his comrade asked him. "m·v squared is divided in half!" the student answered.

R_1 (mm)	R_2 (mm)	λ_0 $\left(\frac{mW}{m \cdot K}\right)$	β $\left(\frac{1}{K}\right)$	q_v $\left(\frac{W}{m^3}\right)$	T_{c1} (°C)	T_{c2} (°C)
200	500	50	0.1	100	100	20

$$\lambda(T) := \lambda_0 \cdot (1 + \beta \cdot T) \qquad \lambda(100\ °C) = 1.916\ \frac{W}{m \cdot K} \qquad \lambda(20\ °C) = 1.516\ \frac{W}{m \cdot K}$$

Solve

$$T(R_1) = T_{c1} \quad T(R_2) = T_{c2} \qquad \frac{1}{R} \cdot \frac{d}{dR}\left(R \cdot \lambda(T(R)) \cdot \frac{d}{dR} T(R)\right) + q_v = 0\ \frac{W}{m^3}$$

Unknown error : ode%_function%_illegal%_context.

$$T := \text{odesolve}\,(T(R), R_1, R_2)$$

Fig. 16.15 Solving the problem of the temperature profile in the cylindrical wall (beginning)

There is another approach to calculation of the temperature profile inside of the cylindrical wall e.g. the use of the difference equation and shooting method [59, 60].

Figure 16.15 shows a box of initial data, input of functional relationship between wall material heat conduction and temperature **λ(T)** and the attempt to solve the differential equation of steady heat conduction for the case of the coiled cylindrical wall with the internal source of heat. This can be e.g. a nuclear reactor pressure vessel in the cylindrical wall of which neutrons are moderated giving their energy to the wall metal, or it can be a hollow tube aired from the inside and outside through which electric current is passed. And this is an example of common life i.e. a glass of water located in the microwave oven.

The attempt to solve the problem by means of the function odesolve (to be described in more details in Chap. 18) proved to be unsuccessful and this is connected with the non-standard record of conditions of the problem (Fig. 16.15). The matter is that the function odesolve is designed for solving a problem with initial conditions (Cauchy problem) but not for a boundary value problem when initial conditions are set at the endpoints of the segment.[2] Furthermore, we will solve the problem by the shooting method replacing the differential equation for the difference equation (i.e. we will use the difference **h** instead of the differential **dR**)—Fig. 16.16.

The equation in Fig. 16.16 is solved by means of Mathcad symbolic mathematics in relation to the variable $\mathbf{T}_{i+1}$. This equation has two roots one of those (lower) was transferred to the calculation and generated the recurrence formula. This formula can be used as follows: we know the value $\mathbf{T}_{i-1}$ and $\mathbf{T}_i$ and by the developed formula we find the value $\mathbf{T}_{i+1}$. This way we can fill the vector **T** and solve the problem by the

[2]The function `odesolve` can solve some boundary value problems but by no means all. There are special tools for boundary value problems in the Mathcad environment.

$$\frac{1}{R_i}\cdot\frac{1}{h}\cdot\left(\frac{R_{i+1}+R_i}{2}\cdot\frac{\lambda_0\cdot(1+\beta\cdot T_{i+1})+\lambda_0\cdot(1+\beta\cdot T_i)}{2}\cdot\frac{T_{i+1}-T_i}{h}-\frac{R_i+R_{i-1}}{2}\cdot\frac{\lambda_0\cdot(1+\beta\cdot T_i)+\lambda_0\cdot(1+\beta\cdot T_{i-1})}{2}\cdot\frac{T_i-T_{i-1}}{h}\right)=-q_v$$

$$\xrightarrow{solve,T_{i+1}}\left|\begin{matrix}-\dfrac{4\cdot h^2\cdot\left(\dfrac{2\cdot\lambda_0\cdot R_i+2\cdot\lambda_0\cdot R_{i+1}}{8\cdot h^2\cdot R_i}+\dfrac{\sqrt{\lambda_0\cdot(R_{i+1}+R_i)\cdot(\lambda_0\cdot R_i+\lambda_0\cdot R_{i+1}-2\cdot\beta\cdot\lambda_0\cdot R_{i-1}\cdot T_{i-1}+2\cdot\beta^2\cdot\lambda_0\cdot R_i\cdot T_i^2+}}{\cdots}\right.}{\beta\cdot\lambda_0\cdot R_{i+1}+\beta\cdot\lambda_0\cdot R_i}\\ -\dfrac{4\cdot h^2\cdot\left(\dfrac{2\cdot\lambda_0\cdot R_i+2\cdot\lambda_0\cdot R_{i+1}}{8\cdot h^2\cdot R_i}-\dfrac{\sqrt{\lambda_0\cdot(R_{i+1}+R_i)\cdot(\lambda_0\cdot R_i+\lambda_0\cdot R_{i+1}-2\cdot\beta\cdot\lambda_0\cdot R_{i-1}\cdot T_{i-1}+2\cdot\beta^2\cdot\lambda_0\cdot R_i\cdot T_i^2+}}{\cdots}\right.}{\beta\cdot\lambda_0\cdot R_{i+1}+\beta\cdot\lambda_0\cdot R_i}\end{matrix}\right.$$

Fig. 16.16 Difference equation of heat conduction in the cylindrical wall and its solving by means of Mathcad symbolic mathematics

$$n:=10000 \qquad h:=\frac{R_2-R_1}{n}=30\ \mu m \qquad i:=1,2..n \qquad R_i:=R_1+h\cdot i$$

$$i:=2,3..n-1 \qquad T_1:=T_{c1}=100\ ^\circ C$$

We change T_2 in order to have $T_n = T_{c2}$ $\quad T_2:=T_1-0.011436\ K$

$$T_{i+1}:=-\frac{4\cdot h^2\cdot\left(\dfrac{2\cdot\lambda_0\cdot R_i+2\cdot\lambda_0\cdot R_{i+1}}{8\cdot h^2\cdot R_i}-\dfrac{\sqrt{\lambda_0\cdot(R_{i+1}+R_i)\cdot(\lambda_0\cdot R_i+\lambda_0\cdot R_{i+1}-2\cdot\beta\cdot}}{\cdots}\right.}{\beta\cdot\lambda_0\cdot R_{i+1}+\beta\cdot\lambda_0\cdot R_i}$$

$$T_{c2}=20\ ^\circ C \qquad T_n=20.00444\ ^\circ C$$

Fig. 16.17 Solving the problem of the temperature profile by the shooting method

step-by-step approach (shooting method) i.e. build the temperature profile inside of the cylindrical wall. This operation is shown in Fig. 16.17.

Figure 16.17 shows the given variable n i.e. a number of cylindrical sections into which our wall is divided. The variable h is thickness of a section. Then we create the vector R storing the radii of wall sections. We have the value of the first element of the vector T_1 equal to the wall temperature on its left (inner) edge $\mathbf{T_{c1}}$ but we have no value of the second element of this vector $\mathbf{T_2}$ for the recurrence formula. So we create it artificially by means of the operator $\mathbf{T_2: = T_1 - 0.01\ K}$ e.g. and see what the value of the last element of the vector **T** ($\mathbf{T_n}$) is. If it is not equal (approximately) to the wall temperature value on the right (outer) edge $\mathbf{T_{c2}}$, then we change the value $\mathbf{T_2}$ and make the following shoot i.e. filling the elements of the vector **T:** $\mathbf{T_{i+1}: = f(T_{i-1}, T_i)}$. In addition we keep control of the diagram in Fig. 16.18 showing three moments of trial with different values of $\mathbf{T_2}$. This figure also shows the function linterp built in Mathcad making possible to create the function **T(r)** by two vectors **R** and **T** by the

$$\mathrm{T}(r) := \mathrm{linterp}(R, T, r) \qquad R := R_1, R_1 + \frac{R_2 - R_1}{n} .. R_2$$

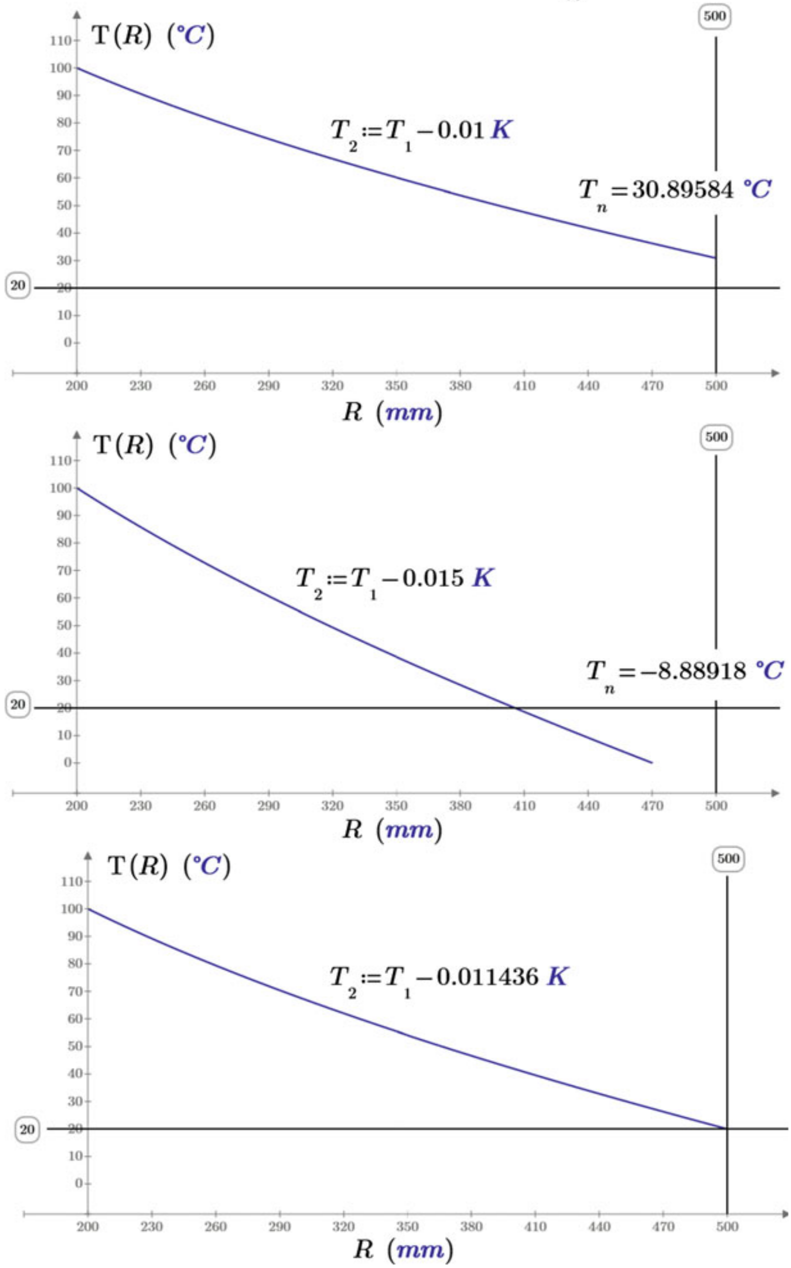


Fig. 16.18 Result of solving the problem of the temperature profile by the shooting method

piecewise linear interpolation method, the above function is used to build the desired temperature profile in the cylindrical wall.

Note. Figure 16.17 does not show the formulae for $\mathbf{T}_{i+1}$ in full. This is not only to save the space in the book but due to the fact that these formulae may be of no interest for study and/or manual entry in the calculation. One of the two formulae to be transferred to the calculation can be determined by the trial-and-error method. It is also possible to attempt to simplify the initial difference equation manually by substituting e.g. $\mathbf{R}_{i+1} + \mathbf{R}_i$ for $\mathbf{2R_i + h}$, and $\mathbf{R}_i + \mathbf{R}_{i-1}$ for $\mathbf{2R_i - h}$ etc. but it's no good. Or rather put it that way: Simplifications and substitutions are worth trying to do if the Mathcad fails to solve the initial equation. But it is necessary to remember that manual substitutions and simplifications are fraught with mistakes e.g. loss of brackets etc. True, the mechanism of measurement units used in the calculation allows discovering many of such mistakes preventing from e.g. adding temperature to length.

The shooting method (method of successive approximations) can be automated by means of Mathcad programming by implementing e.g. the half-interval method (shooting method). Such operation is implemented e.g. in Fig. 18.11 in Chap. 18. But the manual problem solving is good so much that it can be done in the environment of Mathcad Prime free version i.e. Mathcad Express.

If we increase the values $\mathbf{R}_1$ and $\mathbf{R}_2$ leaving the cylindrical wall thin, then it will tend to a flat wall in shape. Figure 16.19 shows the solving of the problem of calculation of the temperature profile in the flat endless wall with internal sources of heat for the case when wall material heat conduction does not depend on temperature. In this case the constant λ is taken out of the differential sign and the differential equation takes on the standard for Mathcad form. Here the function odesolve built in Mathcad is successfully used for the problem with initial conditions and the above described shooting method but when establishing not the second value of the temperature vector (Fig. 16.17) but temperature gradient on the left edge of the wall. Figure 16.19 shows the shooting method very well when at the right end of the diagram one can clearly see "undershoot" or "overshoot" if the temperature gradient T' is established incorrectly from the "artillery position" from the left wall edge.

But let's go back to the name of the study. The formula of the loss of head in the tubing displays the fundamental physical law connecting the change of kinetic energy of liquid flow due to friction on the walls and conversion of this energy to less valuable i.e. to heat energy. In the horizontal tubing (our simplified problem) the potential energy does not change but the kinetic energy "mv squared divided in half" changes. So the "two" in our formula of the loss of head is rather appropriate: it is difficult to understand the "physics" of the problem without it.

In connection with our above reasoning about gravity and "two" in the formula of the loss of head one is put in mind of an interesting theoretical issue regarding rivers i.e. "water-supply lines" providing the mankind, among other thing, with electric energy. For which purpose is hydroelectric dams built? The overwhelming majority of rather competent engineers and specialists can answer approximately as follows. A hydraulic dam is built on the river to increase the water level in the river

δ	λ	q_v	T_1	T_2
(mm)	$\left(\frac{W}{m \cdot K}\right)$	$\left(\frac{kW}{m^3}\right)$	$(°C)$	$(°C)$
200	1.5	50	20	20

Solve

$$T(0\ m) = T_1 \qquad T'(0\ m) = 3.333335\ \frac{K}{mm} \qquad \lambda \cdot \frac{d}{dx}\left(\frac{d}{dx} T(x)\right) = -q_v$$

$$T := \text{odesolve}(T(x), \delta)$$

$$x := 0\ m, \frac{\delta}{700} .. \delta \qquad T(\delta) = 20.0003333\ °C$$

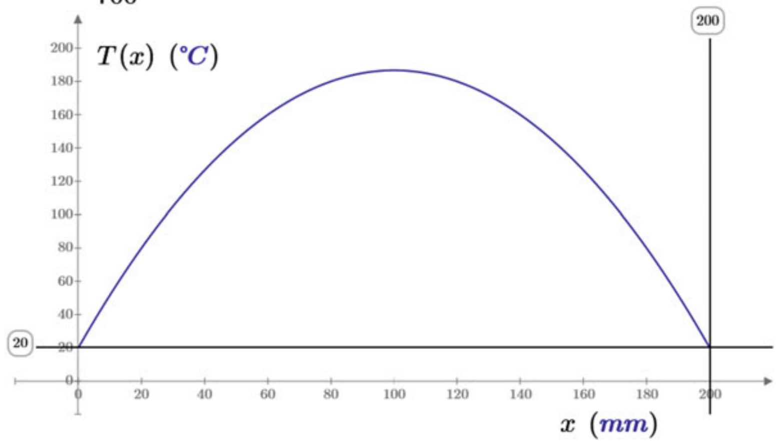


Fig. 16.19 Calculation of the temperature profile in the flat wall

and create water head before the hydraulic turbine generating electric power. But a dam is a purely passive structure and it is not a pump boosting the head! Our simple problem under consideration in this book will help us to answer this question correctly. A hydroelectric dam is built in order… to reduce the average water velocity in the river due to the increase of the river bed cross-section. On plains rivers low but long dams are built while on mountains rivers dams are high and narrow. Due to this the average water velocity in the river reduces sharply and therefore, losses due to water friction on the banks and bottom reduce. And this is the source of energy for the turbines of a hydro power plant generating electric power.

There is another way of generating electric power from the river which is now used in increasing frequency, particularly on mountains rivers and brooks without construction of any expensive and dangerous dams in seismic zones. A tube is laid parallel to the river bed with a hydro turbine installed at the lower end of the tube.

Water or a part of it goes down not the rough river bed but by the smooth tube. Due to this the potential energy of water (it should be noted once more that this energy does not change after the damming) determined by elevation difference does not dissipate so intensively and converts to electric power sufficient to provide a small settlement with lighting. Such mini- and micro hydro power plants are widely spread e.g. in Norway. Incidentally, such “power plants” are sometimes “built” at the ends of different tubes as well. The specific example: a town is supplied with natural gas by a high pressure gas pipeline. While household users need the gas under very low overpressure otherwise problems of gas leakages indoors will occur. Excess pressure can be released by gas throttling by means of pressure reducing valves with loss of such gas flow energy. And there is another energy efficient solution. It is possible to install a turbine generator at the end of such gas pipeline to convert gas overpressure to electric power. Some gas turbines at large power plants are equipped with such mini-power plants. The author has long ago seen an interesting hydrotechnical facility in Lithuania. In one ravine water went down with the elevation difference 2–3 m. These elevations were connected with a tube having a piston pump with a piston drive at the lower end of the tube. That pump delivered water to a house located much higher than the upper end of the tube. Such hydrotechnical facilities (i.e. hydraulically activated pumps) equipped with two pistons different in diameters located on the beams with different shoulders have been widely used in due time for water supply in Switzerland rich in rivers and brooks with large elevation difference.

Mention may be made of one more exotic but quite implementable method of energy generation from a river i.e. a stream of fresh water flowing into salt seas and oceans. There are many methods of production of fresh water from salt water e.g. distillation, freezing, ion exchange, reverse osmosis, electrodialysis… Sea (salt) water and electric power come to electrodialysis plants producing fresh water (demineralized water) and concentrate. But such plants are reversible. If salt water and fresh water is fed to such plants separately, such plant will generate electric power due to mixing of waters of different salinity. Such electrodialysis power stations are rather appropriate in the beds of large rivers flowing into seas and oceans. At present the similar pilot and semi-commercial plants exist and the issue of construction of such actual power plants is addressed.

A reverse osmosis plant is also a system to which salt water and electric power are fed and from which two water steams go out: demineralized water (permeate) and concentrate. But the principles of action of electrodialysis and reverse osmosis plants are different. In the electrodialysis plant electric current is directly fed to special electrodes forcing ions (cations and anions of salt admixtures in water) to move, separate by means of special membranes and finally to be removed from water. And in the reverse osmosis plants electric power is used for driving the pumps to boost the pressure of salt water to the value (osmotic pressure and higher) sufficient for water molecules to go (squeeze) through the finest pores of the special membranes. As this takes place, aquated cations and anions remain in salt water and are removed from the plant with blowdown concentrate. In principle, a reverse osmosis plant can also be forced to generate power by feeding two water streams of different salinity into it. But here there will be no direct power generation as in the electrolysis plant. In the reverse osmosis power plant i.e. in simply osmotic power plant one has to convert the energy of water head (osmotic pressure) to electric power by means e.g. of the above mentioned hydro turbines. Due to this no construction of such power plants is now under consideration.

Now water treatment systems for heat and nuclear power stations in Russia are subject to upgrading. Particularly, clarification units for calcification and coagulation and ion exchange units are being replaced for ultrafiltration and reverse osmosis membrane units. This results in sharp reduction of consumption of chemicals (lime, coagulating agents, acids, alkali) for water treatment and amount of waste water. But it also results in increase of power consumption for driving the pumps pushing water through the membranes. In this connection in order to save power it is reasonable to consider the issue regarding the use of a small steam turbine instead of electric motor to drive such pumps. This is the principle of operation of the pump delivering feed water to the boilers of the heat power plant.

Finally, one newspaper published the information that scientists proved that a square section tube has less hydraulic resistance than a round tube. Such square sections can under otherwise equal conditions deliver more water, gas, oil to customers… Besides that such tubes do not roll asunder when stacking and it is easier for repair robots to move inside of them… And this was April fool's joke. Nevertheless one tube-rolling mill wrote a formal letter to the newspaper editorial office signed by chief engineer informing that they are ready to manufacture such innovation tubes. That's what can happen if one does not know the elements of fluid dynamics set forth in this chapter. A tube is round because a circle as opposed to other plane figures (square, rhomb, triangle etc.) has the least perimeter length at the fixed surface.

Chapter 17
Cogeneration (CHP), Trigeneration (CCHP) and Quadrogeneration (CCHPI), or How Much of Mathematics Is Contained in Thermal Engineering

Valery Ochkov

Energy efficiency in Russia is air inlet to the conditioned room which is in its turn overheated by radiator.

www.anekdot.ru.

Abstract This chapter describes the methods of calculation of grids—hydraulic and electric. Emphasis on the mathematical basis of solving such problems. Described methods of improving the efficiency of power supply due to the simultaneous production of heat, electricity and cold.

Calculation of trigeneration processes (composite sequential and/or parallel generation of heat, electricity, and cold by one power plant) requires knowledge of the properties of working fluids, coolants and structural materials involved in these processes and taken from various directories and databases.

Currently, a designer using conventional technologies, when calculating the trigeneration process or individual processes for generating heat, electricity and/or cold, has to refer to paper or electronic sources of information on properties of the working fluids, coolants and structural materials. In some cases, a designer can work with highly specialized analysis software (CAD), which contains "hardwired" modules with the desired properties of the substances. Description of these programs can be found at www.rosteplo.ru.

The site of the chapter: https://www.ptcusercommunity.com/message/422965.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_17

Reference to the paper handbooks or electronic directories slows down a calculation process, increases the risk of errors because data is copied and transferred manually, because of missteps in typing numbers on the keyboard or because of a slip in the units: e.g. the directory specifies calories, and the computational program requires joules, or vice versa, etc. The specialized programs often have "hardwired" obsolete data with significant constraints for the input parameters, such as a set of working fluids and coolants (refrigerants). Very often such programs, instead of functional dependencies, have "hardwired" simple constants, e.g. water density is defined as 1000 kg/m^3 excluding the fact that this thermodynamic parameter of the most common working fluid and coolant in power engineering is a function of temperature and pressure. Further, the temperature in the older reference books may be fixed according to the outdated (ITS-68) and not to the current (ITS-90) scale (visit website http://twt.mpei.ac.ru/MCS/Worksheets/Thermal/T90-T68.xmcd to see interactive temperature rescaling for the above scales—also see Fig. 1.17 in Chap. 1).

The technology of using "cloud" functions for calculations avoids many of these shortcomings. This chapter illustrates this technology by a simple example of calculation of a pipeline through which fluid (coolant, refrigerant) used for conditioning of facilities flows. The calculation is carried out in the Mathcad [12].

The functions relating to the properties of working fluids, coolants and structural materials used in trigeneration are posted in the "clouds"—on an external server to which a Mathcad-calculation has "cloud" links. The Internet "clouds" are usually used to store photos. This is convenient and reliable, and requires practically no cost or servicing. The main thing is that your computer should have an Internet connection (wired or wireless), and you can view pictures from anywhere, from any computer, yours or someone else's. Often, the "clouds" store not only personal, but also service information, as well as programs for its collection, hosting, processing and delivery. Now this innovative technology ("cloud" office) is more and more used by many engineering and design offices, publishing houses and other organizations whose work nowadays is unthinkable without computers. Staffs of such "cloud" organizations do not occupy a separate room equipped with workstations, local area network and a server connected to the Internet but work from home from time to time going to a coffee shop, for example, for personal communication, and real, not virtual meetings. All this significantly reduces the cost of work and time on its execution, without affecting the quality.

So, once again recall what is a "cloud" function?

A simple example. It is necessary to perform a hydraulic calculation of a network used for heat or cold supply to consumers. Coolant is an aqueous solution of sodium chloride, which is in contrast to the pure water does not freeze at sub-zero temperatures. The author was led to this problem by his business trip to Baku, where he met with experts of Azerbaidzhan district heating grid (Azeristiliktedzhhizad), held a seminar and talked about simulators for the staff of the district heating grid [61, 62] and modern computational technologies in the field of power engineering. So, the chief engineer of the organization complained that in Baku that time they were building many modern high-rise residential buildings, but they all, with rare exceptions, were not connected to the existing municipal heating systems but adapted to the individual “boiler” heat supply. Baku is a city in the south, but they have quite strong and cold winds in the winter blowing from the Caspian Sea. Use the household electric heaters for heating the apartments during this season is rather expensive, and indeed electric grids would hardly sustain such a peak, and natural gas is relatively cheap and available. However near the newly built houses there are no gas boiler houses to supply heat to a house or an entire neighborhood, as is done, for example, in our new suburbs in the vicinities of Moscow. In Baku the new occupants have to supply their own heat. They have to install small gas boilers in their new houses (Fig. 17.1).

These small boilers (in Baku they are called “combi”), intended for individual cottages, are installed on the balconies of apartment buildings, where they can be easily serviced, repaired or replaced. Figure 17.1 also represents a compressor-condenser module of household air-conditioners (split units).

Of course, more rational and more “aesthetic” heating method is the construction of a centralized gas boiler facility to be located near the house, in its basement or on the roof. These boilers, by the way, are used to equip some of the new buildings in Moscow and several other Russian cities. But in Baku this is not done for several reasons, the main of which is that the builders are trying to minimize the costs for the construction of a new house, providing for the future tenants the opportunity (and obligation) to finish the construction of their houses by themselves. In Baku, it’s not just “wallpaper, tiles and sanitary ware” (as in Moscow), but also heating. By the way, to the “south of Baku”—in the Middle East, Central and South Asia, and Africa, they often refuse from centralized power supply, and provide for electric generators with internal combustion engines, which may be added to a house as an annex or placed on a balcony of an apartment building.

Fig. 17.1 Small boilers on the balcony of an apartment building to heat individual apartments

This, "technical and economical nonsense" and to put it mildly, irrationality, can be also seen in Russian cities where they build nice modern apartment buildings of business class, which after tenants move in of are spotted with air conditioners. Although it was possible to provide at once in these houses a centralized installation of industrial air conditioners or, more rational systems for energy efficient year-round hot water supply, heating in winter and cooling the apartments during the hot season.

One can certainly argue about the economic feasibility of the centralized heating and cooling system for modern apartment buildings. In Baku, for example, the cold time with winds does not last long, and many apartments on "emergency" basis are heated by electricity, rather than by hot water. By the way, that is done in Moscow during the cold weather period when the "heating radiator is still not on." But the fact that all of these air-conditioners, small boilers (especially electric generators) spoil the look of the house and degrade the environment is indisputable. Air conditioners are generating additional noise and dripping water on passers-by. Now in some Russian cities it is not allowed to mount air conditioners on the facades of houses. The new designs provide for special niche on the balcony for room air conditioners. In some European cities, open or "boiler" fuel combustion is not allowed, not only for heating but also for cooking purposes. A person cannot even have a barbecue on the coals in the garden of his single-family house. All this is done only by means of safe and "green" electricity, without paying special attention to the economic side of the issue. Use URL http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vol-13.xmcd to find an open interactive design comparison of systems for separate generation and cogeneration of electricity and heat. We can also discuss quadrogeneration. Some IT-organizations have powerful servers with a large self-contained power supply facilities equipped with gas turbines or reciprocating engines of electric generators. Such organization can generate electricity, heat, cooling and … its main product—information (combined cooling, heat, power and information) (CCHPI).

A fact which is currently playing against the classical heat supply scheme (CHP + peak hot water boilers + district heat networks) is as follows. At the time when district heat networks were designed and constructed there was no such an intense car traffic in the cities. Now, excavation and repair of main and local district heating networks lead to huge traffic jams, where standing, or rather, "smoking" machines "devour" the "equivalent fuel savings" that we receive due to district heating.

Ideally, the solution should be the following (it has been already implemented in many "advanced" countries caring about energy saving, ecology and aesthetics of buildings): a combined cooling and heating plant connected to the mains intended for the apartments is installed on the roof of a house, near a house or in its basement. The electricity is generated far from the house—at the modern multiwatt power plants—thermal ones (running on fossil or nuclear fuel), hydraulic, wind and other power plants. Also this house-installed cooling and heating plant is additionally equipped with modern energy saving systems—heat pumps, heat/cold accumulators (tanks with water, salt solution, or earth beds under the house), solar collectors, etc. The apartments in such a house are supplied with electricity through the wires, and, for example, an aqueous solution of salt—hot in the winter and cool (with sub-zero temperatures) in the summer. Here one could still add a centralized vacuum cleaner and a centralized removal of garbage sorted by tenants, but we dream too much.

On the site of the book one can find interactive, open calculations created in Mathcad and posted on the Internet, that shows the benefits of co-production of different types of energy. You can see two examples in Figs. 17.2 and 17.3, the title (see the Figure) of calculation and its main "result" operator.

Back to the aqueous solution of salt. Here, of course, one can consider other coolants (freon coolants, information about their properties is available on the described computation server), but salt is available, relatively inexpensive and, most importantly, safe (non-toxic). Moscow, for example, is standing on a huge underground salt lake, and the solution (brine) has long been used by Moscow power industry to regenerate the filters used for softening the make-up water to district heating systems.

So, to calculate the network of pipes through which the aqueous solution of sodium chloride flows, it is necessary to know the physical properties of the coolant, in particular its density and viscosity.

To obtain this information, the designer can run the online Google search engine, for example, and make it search using key Density NaCl solution, adding to it the word Mathcad. Without this word the search engine would reference to websites where both traditional and quite unnecessary tables are published representing dependency of density from temperature or constant. And what we need is a function for calculating and such that it is available ("visible" as the programmers say) in our Mathcad-calculation.

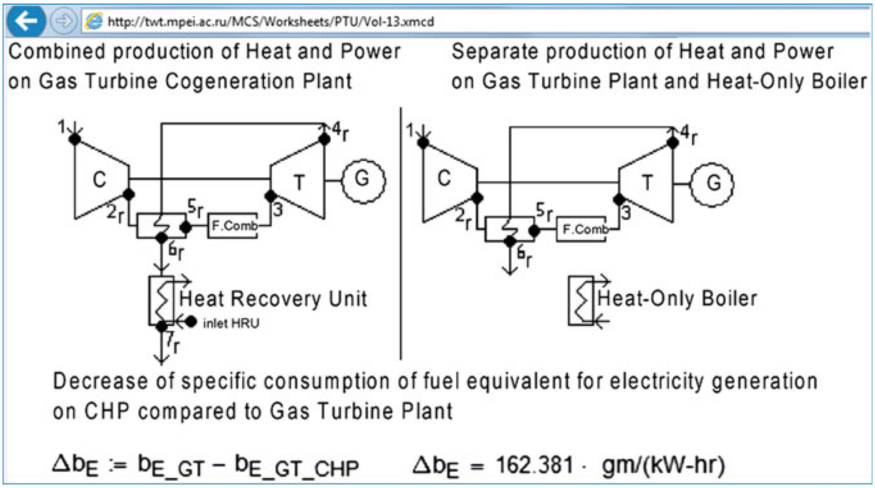


Fig. 17.2 Calculation of fuel economy on a CHP gas turbine compared to separate power generation

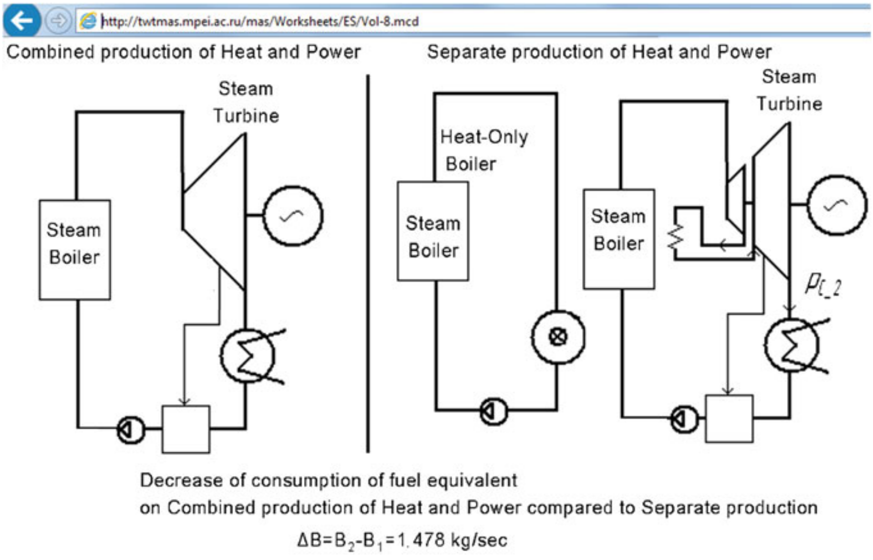


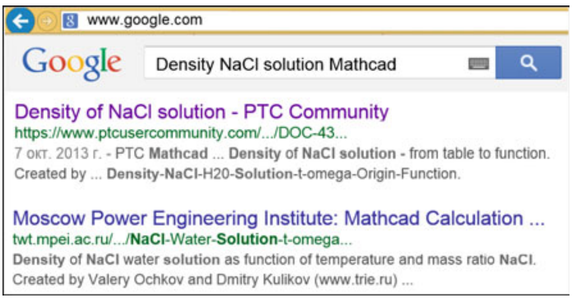
Fig. 17.3 Determination of equivalent fuel economy for district heating with the turbine type T (district heating) compared to separated production of the same amount of electricity and heat

Figure 17.4 shows that the search on the server by a key **Density NaCl solution Mathcad** issued links the first of which leads to the calculation site shown in Fig. 17.5.

On the site shown in Fig. 17.4 one can enter new values for temperature and concentration of NaCl aqueous solution and obtain a new computed value of the density of this solution. A visitor to the site can download a Mathcad file with a function that returns the density of the salt solution depending on temperature and concentration. But it is fashionable to do otherwise—to provide a link to the Mathcad-file. To know its address—see Fig. 17.6.

If one right-clicks on the link shown in Fig. 17.5, a list will drop whose last item stores information about the address of the desired file, see Fig. 17.6. This address

Fig. 17.4 Search in Google



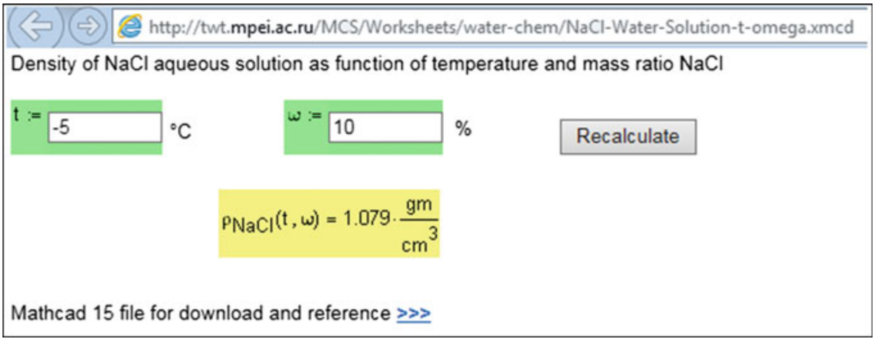


Fig. 17.5 Interactive computation of NaCl water solution density

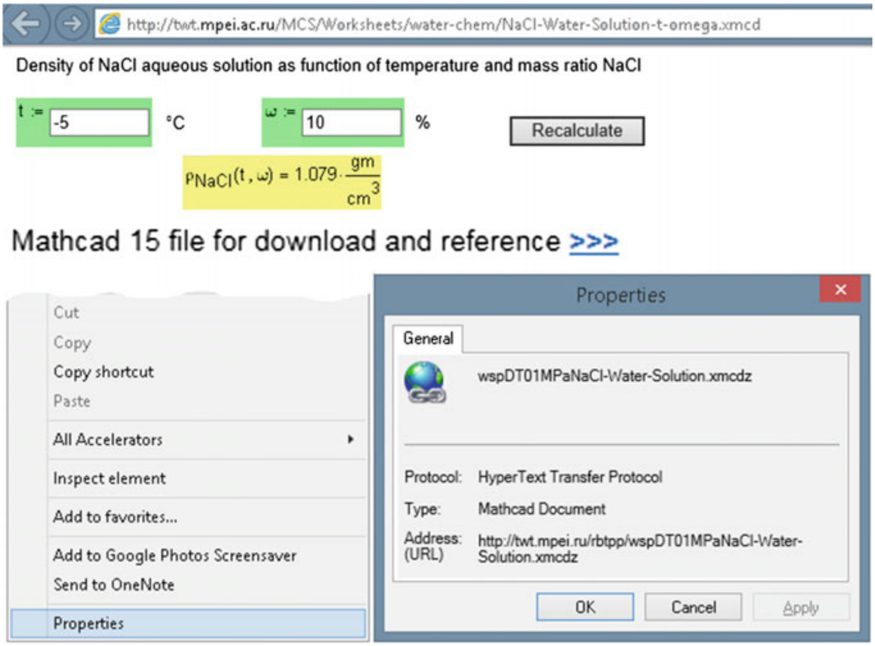


Fig. 17.6 Access to information about the file URL

will need to be copied and moved to the **Insert Reference** window in **Insert** menu, Mathcad 15, see Fig. 17.7.

Figure 17.7 displays command Insert|Reference in Mathcad 15 environment, which enables to make a link to a cloud function called **wspDT01MPaNaCl-Water-Solution.xmcdz** (**D**—density of NaCl water solution as function of temperature **T** at pressure **0.1 MPa**; suffix **wsp** is explained above).

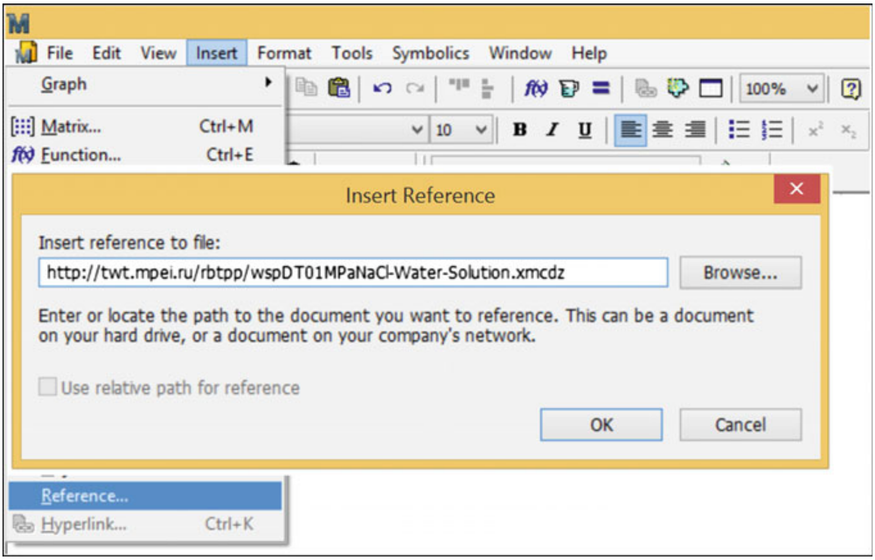


Fig. 17.7 Reference in Mathcad 15 environment

By the way, if one will make a search in the Russian version of Google by the keyword "online computations", the first item in the list of links displayed will be a link to the site, one of the pages of which is displayed in Fig. 17.4 here. This demonstrates the high popularity of and demand for the MPEI calculation server.

As we have repeatedly noted, Mathcad's environment has a very convenient tool—a link from the current calculation to another one, to another file. After such a link all the user's variables and functions defined in the document to which the link is provided, get visible in the working document.

These links, as we have said, can be done to files stored not only on your work station or in the local network, but on the Internet, in the "clouds", and we are going to use that when solving the problem of pipeline with heat and cool carrier—NaCl aqueous solution.

Figure 17.9 shows the calculation in Mathcad 15 environment for a simple piping system—determination of NaCl solution flow in its individual sections for a specified geometry of pipes and specified pump head. After entering the input data (geometric parameters of pipe sections: the length **L**, inner diameter **d**, roughness, temperature and concentration of NaCl aqueous solution) give a link to the two cloud functions returning the density and dynamic viscosity of the coolant and stored in the "clouds" on the sites shown in Figs. 17.5 and 17.8. After such links one can call the functions that returned the density (1140.175 kg/m^3) and viscosity (5.085 mPa·s) of NaCl solution with concentration 21 % and temperature –15 °C. It should be emphasized that Mathcad can work with physical quantities and their measuring units [8], which distinguishes it from other computer calculation programs—from Excel spreadsheets and programming languages (see Chap. 2).

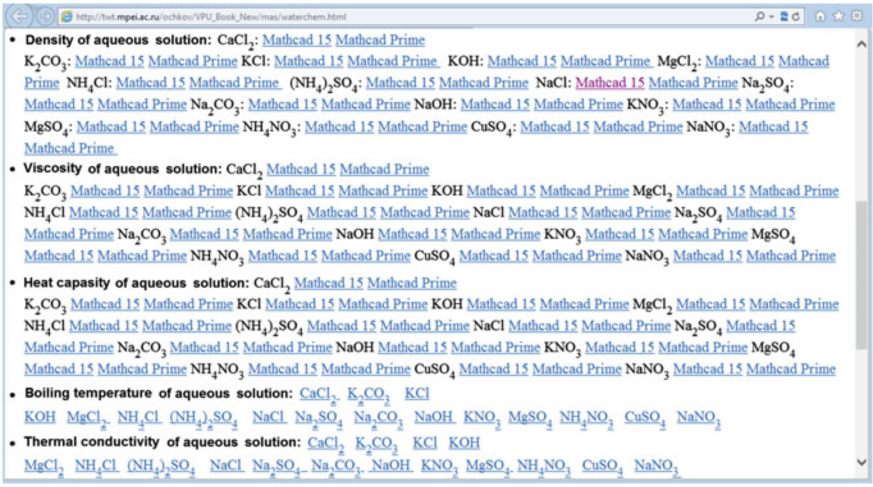
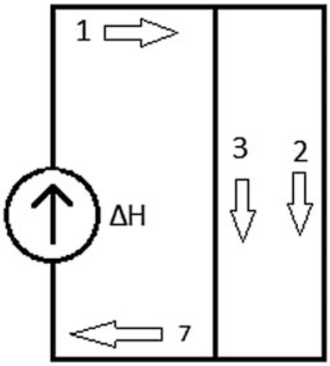


Fig. 17.8 Website page displaying physical properties of aqueous solutions of salts

In the calculation shown in Fig. 17.9, there are two hidden areas. They are open and shown in Figs. 17.10 and 17.11.

The first hidden area (Fig. 17.10) stores a function named λ_{TP} and the table itself in the form of inserted picture for which this function has been created. This function returns the fluid friction resistance coefficient as a function of Reynolds number Re and the relative roughness **Δ** of the pipe inner surface [63]. The table notes that in the range of Re = 2300–4000 (transient from laminar to turbulent mode) a friction coefficient cannot be determined. However, a corresponding function is given 0.03. This is done to perform the iterative calculation using this function, and the "little trick" will be mentioned below. Incidentally, the described site contains an interactive calculation (http://twt.mpei.ac.ru/MCS/Worksheets/Hydro/La-De-Re.xmcd) of the friction coefficient in the transient zone ("The Nikuradze spoon").

The second hidden area of the calculation, open and shown in Fig. 17.11, stores a function called Δh, which is used to calculate the differential (loss of) pressure of a pipeline depending of its length L, inner diameter d and fluid volumetric rate q. In this function, the program lines are a matrix with two columns and one line ("horizontal" vector), whose first element is a comment, and the second element is a formula. If the pipes had different roughnesses (failure of one of the assumptions), the function **Δh** should be supplemented by another argument **Δ**. Function **Δh** uses a function called $\boldsymbol{\lambda}_{\text{friction}}$, shown in Fig. 17.10. These are standard hydrodynamic formulas, which can be found in the handbooks or on the Internet. We have already used them in the previous chapter. The table shown in Fig. 17.10, was indeed found on the Internet, copied and transferred as a picture into our calculation as a comment. In this table, the fluid flow transient value $\boldsymbol{\lambda}_{\text{friction}}$ is not defined. But in the function $\boldsymbol{\lambda}_{\text{friction}}$ it equals to 0.03. We repeat, that this is done in order to be able to



Input data:

$L_1 := 50\text{m}$ $d_1 := 12\text{mm}$

$L_2 := 30\text{m}$ $d_2 := 9\text{mm}$

$L_3 := 10\text{m}$ $d_3 := 10\text{mm}$

$L_7 := 55\text{m}$ $d_7 := d_1$

$\Delta H := 100\text{m}$

NaCl solution temperature: $t := -15\ °\text{C}$

NaCl solution concentration: $\omega := 21\%$

The relative roughness of the inner surface of the pipe: $\Delta := 0.0005$

NaCl solution density: Reference:http://twt.mpei.ru/rbtpp/wspDT01MPaNaCl-Water-Solution.xmcdz

$$\rho := \rho_{NaCl}(t, \omega) = 1140.175\,\frac{\text{kg}}{\text{m}^3}$$

NaCl solution dynamic viscosity: Reference:http://twt.mpei.ru/rbtpp/wspViscT01MPaNaCl-Water-Solution.xmcdz

$$\mu := \mu_{NaCl}(t, \omega) = 5.085 \times 10^{-3} \cdot \text{Pa} \cdot \text{s}$$

The drag coefficient of friction - function of the relative roughness Δ and Re

Pressure drop dependence on the length portion of the pipe diameter, and fluid flow

Given Given constraints (equations) and the guess values of the solution

$$q_1 = q_2 + q_3 \qquad q_1 := 500\,\frac{\text{L}}{\text{hr}} \qquad q_2 := 300\,\frac{\text{L}}{\text{hr}} \qquad q_3 := q_1 - q_2 = 200\,\frac{\text{L}}{\text{hr}}$$

$$\Delta H = \Delta h(L_1, d_1, q_1) + \Delta h(L_2, d_2, q_2) + \Delta h(L_7, d_7, q_1)$$

$$\Delta H = \Delta h(L_1, d_1, q_1) + \Delta h(L_3, d_3, q_3) + \Delta h(L_7, d_7, q_1)$$

$$\Delta h(L_2, d_2, q_2) = \Delta h(L_3, d_3, q_3)$$

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} := \text{Find}(q_1, q_2, q_3) = \begin{pmatrix} 972.39 \\ 321.93 \\ 650.46 \end{pmatrix} \frac{\text{L}}{\text{hr}}$$

Verify

$$q_1 - q_2 - q_3 = -8.944667923005412 \times 10^{-18} \cdot \frac{\text{m}^3}{\text{s}}$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_2, d_2, q_2) + \Delta h(L_7, d_7, q_1) - \Delta H = 0\,\text{m}$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_3, d_3, q_3) + \Delta h(L_7, d_7, q_1) - \Delta H = 0\,\text{m}$$

$$\Delta h(L_2, d_2, q_2) - \Delta h(L_3, d_3, q_3) = -7.105 \times 10^{-15}\,\text{m}$$

Fig. 17.9 Example of hydraulic calculations in Mathcad 15

The drag coefficient of friction - function of the relative roughness Δ and Re

$\lambda_{friction}(\Delta, Re) :=$ "The drag coefficient of friction - function of the relative roughness Δ and Re"

return $\frac{64}{Re}$ if $Re \le 2300$

return 0.03 if $2300 < Re \le 4000$

return $\frac{0.3164}{Re^{0.25}}$ if $4000 < Re \le \frac{10}{\Delta}$

return $0.11 \cdot \left(\Delta + \frac{68}{Re}\right)^{0.25}$ if $\frac{10}{\Delta} < Re < \frac{560}{\Delta}$

$0.11 \cdot \Delta^{0.25}$ otherwise

Regime of fluid flow		Reynolds number	λ determination
Laminar		$Re < 2300$	$\lambda = \frac{64}{Re}$
Transitional		$2300 < Re < 4000$	Piping is not recommended
Turbulent	1st area	$4000 < Re < 10\frac{d}{\Delta_{э}}$	$\lambda_T = \frac{0.3164}{Re^{0.25}}$ (Blasius formula)
	2nd area	$10\frac{d}{\Delta_{э}} < Re < 560\frac{d}{\Delta_{э}}$	$\lambda_T = 0{,}11\left(\frac{\Delta_{э}}{d} + \frac{68}{Re}\right)^{0{,}25}$ (Altshul)
	3rd area	$Re > 560\frac{d}{\Delta_{э}}$	$\lambda_T = 0{,}11\left(\frac{\Delta_{э}}{d}\right)^{0{,}25}$ (Altshul)

The drag coefficient of friction - function of the relative roughness Δ and Re

Fig. 17.10 Calculation of the resistance to flow of fluid in the pipe

Pressure drop dependence on the length portion of the pipe diameter, and fluid flow

$\Delta h(L, d, q) :=$

("Cross-sectional area of the pipe" $F \leftarrow \pi \frac{d^2}{4}$)

("The rate of fluid flow in a pipe" $v \leftarrow \frac{q}{F}$)

("Kynematic viscosity of solution" $\nu \leftarrow \frac{\mu}{\rho}$)

("Reynolds number" $Re \leftarrow \frac{v \cdot d}{\nu}$)

$\lambda_{friction}(\Delta, Re) \cdot \frac{L}{d} \cdot \frac{v^2}{2g}$

Pressure drop dependence on the length portion of the pipe diameter, and fluid flow

Fig. 17.11 Calculation of pressure difference in the pipe sections conduct an iterative calculation for solving a system of algebraic equations (see Fig. 17.9). In the reference books and on the Internet one can find several named and nameless contradictory formulas for calculating the coefficient of fluid friction depending on the mode of its flow, and the pipe surface roughness. But this discrepancy to some extent is offset by the uncertainty of the concept “roughness”.

The function named **Δh** (Fig. 17.11) returns the fluid head loss in the pipeline section depending on the pipe length L, it inner diameter d and the fluid volumetric flow q. That is to say, the “internal” arguments of function **Δh**. “External” arguments of function **Δh**, not included in the list of its arguments are the roughness of pipe inner surface **Δ**, fluid density ρ and its dynamic viscosity **μ**. Function **Δh** has two constants: the ratio of the circumference to its diameter π and gravitational acceleration g. The second constant may be considered a constant conditionally, since it is insignificant, but it depends on the latitude and the altitude above sea level. The author has on his site an online calculation of gravitational acceleration depending on the above parameters of the area. But we should also not forget that the pipe system can be mounted not only on the Earth, but on the space station or even on the Moon or Mars, where g is significantly different from the value of 9.81 m/s^2. This we have noted in Chap. 16. Hence the function **Δh** should in principle have the following argument list: **L**, **d**, **q**, **Δ**, **ρ**, **μ**, **g**. But we have reduced it by deleting from it the variables that are common to all the sections of pipes.

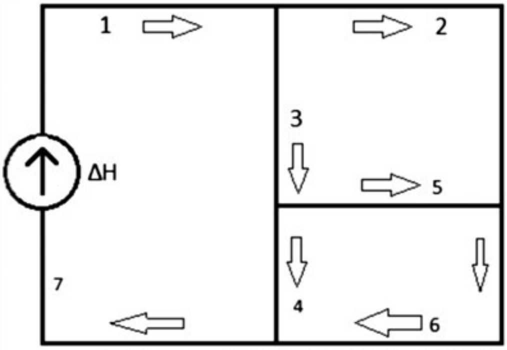
After determining the desired four functions, i.e. two cloud functions $\boldsymbol{\rho}_{\mathbf{NaCl}}$ and $\boldsymbol{\mu}_{\mathbf{NaCl}}$ (see Fig. 17.9) and two user’s functions $\boldsymbol{\lambda}_{\text{friction}}$ (see Fig. 17.10) and **ΔH** (see Fig. 17.11), it is easy to solve the problem by using the tool for solving systems of nonlinear algebraic equations built into Mathcad: key word **Given** and function **Find**—see Fig. 17.9. Remember, in solving the nonlinear algebraic equations in Mathcad’s environment a first approximation to the desired values is required. In the problem, as shown in Fig. 17.12, the value is corresponding to the pump head pressure and volumetric flow rate of fluid in some areas of the pipeline.

The system of equations is a balance of flow rates in tees ($\mathbf{q_1 = q_2 + q_3}$) and the pressure difference balance in certain closed sections of the network. Local resistance of the tees, pipe bends, stop valves and control valves in our calculation is not taken into account (another assumption). In addition, another significant assumption is taken: our pipes lie on a horizontal plane and elevation changes are not considered. But these restrictions are easy to remove.

Mathcad found such variables that when they are put into the equation with given values of the pump head and mass flow of fluid, the right and left sides of the equations differ only slightly or not at all differ. This is confirmed by checking, as shown in the bottom of Fig. 17.9. We emphasize in particular that the problem was solved not by analytical but by numerical (approximate) methods built into the function Find.

The direct use of “cloud” features via links to them, as shown in Fig. 17.9, has one significant limitation. The fact is that some organizations limit or completely block their employees’ direct access to the Internet from the workstations as we have already noted in Chap. 1. This is done for security purposes, and to ensure that employees are not distracted from their main job, going to the news and

Calculation of hydraulic network with NaCl aqueous solution



Input data

$L_1 := 50\ m \quad d_1 := 12\ mm$

$L_2 := 30\ m \quad d_2 := 7\ mm$

$L_3 := 10\ m \quad d_3 := 10\ mm$

$L_4 := 45\ m \quad d_4 := 5\ mm$

$L_5 := 5\ m \quad d_5 := 3\ mm$

$L_6 := 17\ m \quad d_6 := 7\ mm$

$L_7 := 55\ m \quad d_7 := d_1$

$q_1 := 1000\ L/hr$

The relative roughness of the inner surface of the pipe: $\Delta := 0.0005$

NaCl solution temperature: $t := -15\ °C$ NaCl solution concentration: $\omega := 21\%$

$$\rho := \rho_{NaCl}(t, \omega) = 1140.175\ \frac{kg}{m^3}$$

$$\mu := \mu_{NaCl}(t, \omega) = 5.085\ 10^{-3}\ Pa \cdot s$$

$\Delta H := 800\ m \qquad q_2 := 500\ L/hr \qquad q_3 := q_1 - q_2$

$q_4 := 200\ L/hr \qquad q_5 := q_3 - q_4 \qquad q_6 := 700\ L/hr$

$$q_1 = q_2 + q_3 \qquad q_1 = q_4 + q_6 \qquad q_3 = q_4 + q_5 \qquad q_6 = q_2 + q_5$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_3, d_3, q_3) + \Delta h(L_4, d_4, q_4) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_2, d_2, q_2) + \Delta h(L_6, d_6, q_6) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_3, d_3, q_3) + \Delta h(L_5, d_5, q_5) + \Delta h(L_6, d_6, q_6) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_2, d_2, q_2) - \Delta h(L_5, d_5, q_5) + \Delta h(L_4, d_4, q_4) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_2, d_2, q_2) - \Delta h(L_5, d_5, q_5) - \Delta h(L_3, d_3, q_3) = 0\ m$$

$$\Delta h(L_6, d_6, q_6) - \Delta h(L_4, d_4, q_4) + \Delta h(L_5, d_5, q_5) = 0\ m$$

$$\Delta h(L_2, d_2, q_2) + \Delta h(L_6, d_6, q_6) - \Delta h(L_4, d_4, q_4) - \Delta h(L_3, d_3, q_3) = 0\ m$$

$$\begin{bmatrix} \Delta H \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} := \text{Find}(\Delta H, q_2, q_3, q_4, q_5, q_6) \qquad \Delta H = 345.2\ m$$

$$\Delta p := \rho \cdot g \cdot \Delta H = 38.1\ atm$$

Fig. 17.12 Example of hydraulic calculations in Mathcad prime entertainment sites, communicating with their friends in social networks, etc. On a computer disconnected from the Internet the calculation shown in Fig. 17.9 will not work, and links included in it, will turn red (see Fig. 4.3 in Chap. 4), which would indicate an emergency situation. In this case, we can recommend the following.

The administrator of a local computer network ("sysadmin") in an organization with disabled Internet access from workstations of ordinary employees may, at the request of a designer, download the required function, check it for viruses and "Trojan horses" and place in a local network or on certain work stations. One can see in Fig. 17.7 the command **Save Target As**… that let you do this. One could give a link to a function downloaded, tested and placed in the organization, and can just built-in the function into a calculation, that in contrast to the relative links will not result in failure when moving the calculation from one computer to another. (Links, shown in Fig. 17.9, are the absolute links, not relative.)

Figure 17.12 shows the calculation of more complicated piping system in a newer version of Mathcad—Mathcad Prime environment. The initial value of the calculation is the flow of fluid through the pump $\mathbf{q_1}$, and the sought value is the pump's head $\mathbf{\Delta H}$, expressed in meters, and (after an appropriate translation) in atmospheres. Another difference between calculation in Fig. 17.12 and calculation in Fig. 17.9 is that Mathcad Prime environment uses a new tool—block **Solve**, but not a tandem of keyword **Given** and function **Find**, as Mathcad 15 environment does. Figure 17.13 shows the solution check.

One can make the piping system shown in Fig. 17.12 even more complicated by adding in the calculation some new equations that show the balance of flows and sums of pressure drops for individual closed annular sections. But soon it turns out that Mathcad cannot solve a system of equations. Also specialized packages aimed

Verify

$$q_1-q_2-q_3=-1.923\cdot10^{-13}\ \frac{L}{s} \qquad q_3-q_4-q_5=-1.399\cdot10^{-13}\ \frac{L}{s}$$

$$q_6-q_5-q_2=-1.286\cdot10^{-13}\ \frac{L}{s} \qquad q_1-q_4-q_6=-2.035\cdot10^{-13}\ \frac{L}{s}$$

$$\Delta h(L_1,d_1,q_1)+\Delta h(L_3,d_3,q_3)+\Delta h(L_4,d_4,q_4)+\Delta h(L_7,d_7,q_1)-\Delta H=\left(5.684\cdot10^{-14}\right)\ m$$

$$\Delta h(L_1,d_1,q_1)+\Delta h(L_2,d_2,q_2)+\Delta h(L_6,d_6,q_6)+\Delta h(L_7,d_7,q_1)-\Delta H=0.000\ m$$

$$\Delta h(L_1,d_1,q_1)+\Delta h(L_3,d_3,q_3)+\Delta h(L_5,d_5,q_5)+\Delta h(L_6,d_6,q_6)+\Delta h(L_7,d_7,q_1)-\Delta H=0.000\ m$$

$$\Delta h(L_1,d_1,q_1)+\Delta h(L_2,d_2,q_2)-\Delta h(L_5,d_5,q_5)+\Delta h(L_4,d_4,q_4)+\Delta h(L_7,d_7,q_1)-\Delta H=0\ m$$

$$\Delta h(L_2,d_2,q_2)-\Delta h(L_5,d_5,q_5)-\Delta h(L_3,d_3,q_3)=-1.91\cdot10^{-14}\ m$$

$$\left(\Delta h(L_6,d_6,q_6)-\Delta h(L_4,d_4,q_4)\right)+\Delta h(L_5,d_5,q_5)=-2.842\cdot10^{-14}\ m$$

Fig. 17.13 Verifying the hydraulic calculation in Mathcad prime environment at solving such systems of equations will be unable to solve them. One will need to use special "smart" techniques for solving such problems and highly specialized software, created specifically for the calculation of pipelines. But these programs will need information on the properties of liquids, available through a "cloud" technique, as described above in this chapter and in other places in the book.

Blocks for solving algebraic equations in Mathcad's environment can return not only specific numerical values, but also be used to create user-defined functions, which can be plotted. Now we use that feature, and construct a graph of a flow of liquid in our hydraulic system (see Fig. 17.12) as a function of a pump head. For this purpose, it is enough to make in the calculation, shown in this figure, small changes, displayed in Fig. 17.14.

First, remove from the calculation the initial approximation operator for the solution for the pump head. This operator is crossed out in Fig. 17.15. Second, function **Find** should be assigned not to a vector, as shown in Fig. 17.15, but to a user's function named q. This function is not simple, but a vector-function. By selecting its first element (in this calculation the system variable **ORIGIN** is equal to one), one can plot the flow of fluid through the pump as function of its pressure. Gaps in the line indicate that in these areas Mathcad could not solve a system of nonlinear algebraic equations.

The Mathcad has the tools for solving not only the systems of nonlinear algebraic equations (see Figs. 17.7 and 17.12), but also for solving systems of linear equations.

Our problem of the pipeline network has a direct equivalent in electrical engineering: the pump is a source of electrical current (generator, battery, etc.) with a certain electromotive force (**EMF**), expressed in volts, and the sections of pipelines, valves, tees, angles, etc. appear to be electrical resistance, who's value (ohms), unlike the "hydraulic resistance," depends little on the current. Therefore, solving a typical problem of electrical engineering—finding the current values in some parts of the circuit depending on the specified values of local resistances and **EMF** is reduced to solving a system of linear algebraic equations. Figure 17.15 represents a "remake" of our hydraulic problem, as shown in Fig. 17.12, into electrical problem.

Figure 17.15 shows how an electrical problem is solved with **Solver** block: enter the initial approximations into the solution: one ampere is assigned to all the six desired current values. But it is possible to assign other values. The selection of these values is an important stage in the solution. Mathcad sometimes may not be able to solve the system and asks to change the initial approximations. Engineer solving on a computer his problem is to know about what decision he needs, and provide a first approximation near this point. Next, in calculation in Fig. 17.15 they record current balance equations (first Kirchhoff's law), and the balance of "energy" for individual closed branches of electrical network (second Kirchhoff's law). Function Find returned a value of six currents providing an approximate solution of the original system. We have a problem with eight equations and six unknowns.

~~$\Delta H := 800 \cdot L/hr$~~ $\quad q_2 := 500\ L/hr \quad q_3 := q_1 - q_2$

$q_4 := 200\ L/hr \quad q_5 := q_3 - q_4 \quad q_6 := 700\ L/hr$

$$q_1 = q_2 + q_3 \quad q_1 = q_4 + q_6 \quad q_3 = q_4 + q_5 \quad q_6 = q_2 + q_5$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_3, d_3, q_3) + \Delta h(L_4, d_4, q_4) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_2, d_2, q_2) + \Delta h(L_6, d_6, q_6) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_3, d_3, q_3) + \Delta h(L_5, d_5, q_5) + \Delta h(L_6, d_6, q_6) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_1, d_1, q_1) + \Delta h(L_2, d_2, q_2) - \Delta h(L_5, d_5, q_5) + \Delta h(L_4, d_4, q_4) + \Delta h(L_7, d_7, q_1) = \Delta H$$

$$\Delta h(L_2, d_2, q_2) - \Delta h(L_5, d_5, q_5) - \Delta h(L_3, d_3, q_3) = 0\ m$$

$$\Delta h(L_6, d_6, q_6) - \Delta h(L_4, d_4, q_4) + \Delta h(L_5, d_5, q_5) = 0\ m$$

$$\Delta h(L_2, d_2, q_2) + \Delta h(L_6, d_6, q_6) - \Delta h(L_4, d_4, q_4) - \Delta h(L_3, d_3, q_3) = 0\ m$$

$$q(\Delta H) := \text{Find}(q_1, q_2, q_3, q_4, q_5, q_6)$$

$$q_1(\Delta H) := q(\Delta H)_1 \qquad \Delta H := 0\ m, 5\ m..1000\ m$$

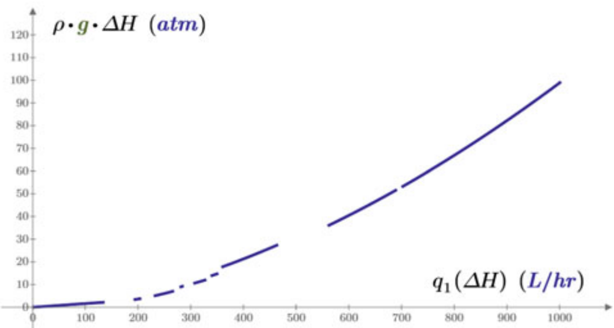
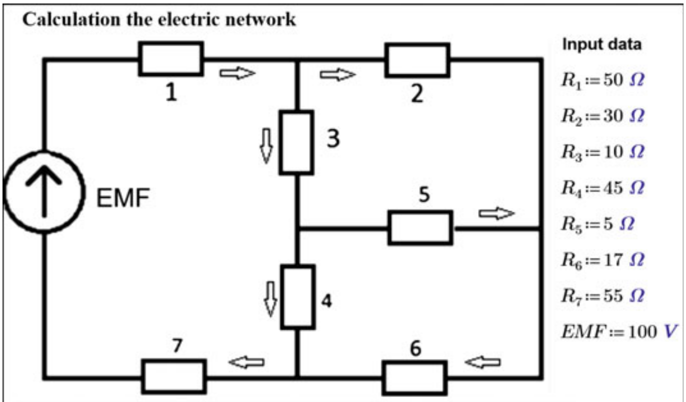


Fig. 17.14 Creating a characteristic curve of fluid flow through the pump versus its head

The problem, therefore, is overdetermined,[1] since the number of equations exceeds the number of the unknowns. Basically, there are two extra equations here. But it is

[1] *Mathematician's comment*. Not every problem in which the number of equations is greater than the number of unknowns, is overdetermined. For example, if the equations are linearly dependent, the described system can be defined (in fact this is shown in an example of calculation), and uncertain. The ratio of the number of equations and the number of unknowns is not a criterion of overdetermination. Probably, you should not generally pay as much attention to this delicate issue. It might be better to say that anything can happen with the systems, but the package itself can (or cannot!) figure it out.



$$[I_1\ I_2\ I_3\ I_4\ I_5\ I_6] := [1\ 2\ 3\ 4\ 5\ 0]\cdot A$$

$$1\ \Omega\cdot I_1 - 1\ \Omega\cdot I_2 - 1\ \Omega\cdot I_3 + 0\ \Omega\cdot I_4 + 0\ \Omega\cdot I_5 + 0\ \Omega\cdot I_6 = 0\ V$$

$$0\ \Omega\cdot I_1 + 0\ \Omega\cdot I_2 + 1\ \Omega\cdot I_3 - 1\ \Omega\cdot I_4 - 1\ \Omega\cdot I_5 + 0\ \Omega\cdot I_6 = 0\ V$$

$$0\ \Omega\cdot I_1 - 1\ \Omega\cdot I_2 + 0\ \Omega\cdot I_3 + 0\ \Omega\cdot I_4 - 1\ \Omega\cdot I_5 + 1\ \Omega\cdot I_6 = 0\ V$$

$$1\ \Omega\cdot I_1 + 0\ \Omega\cdot I_2 + 0\ \Omega\cdot I_3 - 1\ \Omega\cdot I_4 + 0\ \Omega\cdot I_5 - 1\ \Omega\cdot I_6 = 0\ V$$

$$(R_1 + R_7)\cdot I_1 + 0\ \Omega\cdot I_2 + R_3\cdot I_3 + R_4\cdot I_4 + 0\ \Omega\cdot I_5 + 0\ \Omega\cdot I_6 = EMF$$

$$(R_1 + R_7)\cdot I_1 + R_2\cdot I_2 + 0\ \Omega\cdot I_3 + 0\ \Omega\cdot I_4 + 0\ \Omega\cdot I_5 + R_6\cdot I_6 = EMF$$

$$R_2\cdot I_2 - R_5\cdot I_5 - R_3\cdot I_3 = 0\ V$$

$$R_5\cdot I_5 + R_6\cdot I_6 - R_4\cdot I_4 = 0\ V$$

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} := \mathrm{Find}(I_1, I_2, I_3, I_4, I_5, I_6) = \begin{bmatrix} 0.795 \\ 0.238 \\ 0.557 \\ 0.243 \\ 0.314 \\ 0.552 \end{bmatrix} A$$

Fig. 17.15 Example of hydraulic calculation in Mathcad prime environment

possible to add another pair of equations,[2] by applying Kirchhoff's second rule to closed circuit branches **EMF-R$_1$-R$_3$-R$_5$-R$_6$-R$_7$** and **EMF-R$_1$-R$_2$-R$_5$-R$_4$-R$_7$**. The problem in this case is even more overdetermined, but the decision in this case will be the same. The solution shown in Fig. 17.15 may be quite acceptable for a user if he is sure of the uniqueness of the solutions, i.e. the ability to seek the solution by

[2] *Mathematician's comment*. If the problem is overdetermined, then why add another equation? Or why they cannot be added if they are physically necessary? Something should be justified. Probably, to refer to the theory of calculating circuits, or to something else. In any case, before considering the numerical solution of the system one should substantiate the adequacy of a chosen mathematical model (having the form of a system of linear algebraic equations) of a studied physical model of electrical circuit.

$$1\ \Omega\cdot I_1-1\ \Omega\cdot I_2-1\ \Omega\cdot I_3+0\ \Omega\cdot I_4+0\ \Omega\cdot I_5+0\ \Omega\cdot I_6=0\ V$$

$$0\ \Omega\cdot I_1+0\ \Omega\cdot I_2+1\ \Omega\cdot I_3-1\ \Omega\cdot I_4-1\ \Omega\cdot I_5+0\ \Omega\cdot I_6=0\ V$$

$$0\ \Omega\cdot I_1-1\ \Omega\cdot I_2+0\ \Omega\cdot I_3+0\ \Omega\cdot I_4-1\ \Omega\cdot I_5+1\ \Omega\cdot I_6=0\ V$$

$$1\ \Omega\cdot I_1+0\ \Omega\cdot I_2+0\ \Omega\cdot I_3-1\ \Omega\cdot I_4+0\ \Omega\cdot I_5-1\ \Omega\cdot I_6=0\ V$$

$$(R_1+R_7)\cdot I_1+0\ \Omega\cdot I_2+R_3\cdot I_3+R_4\cdot I_4+0\ \Omega\cdot I_5+0\ \Omega\cdot I_6=EMF$$

$$(R_1+R_7)\cdot I_1+R_2\cdot I_2+0\ \Omega\cdot I_3+0\ \Omega\cdot I_4+0\ \Omega\cdot I_5+R_6\cdot I_6=EMF$$

$$R_2\cdot I_2-R_5\cdot I_5-R_3\cdot I_3=0\ V$$

$$R_5\cdot I_5+R_6\cdot I_6-R_4\cdot I_4=0\ V$$

$$M:=\begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & -1 \\ R_1+R_7 & 0\ \Omega & R_3 & R_4 & 0\ \Omega & 0\ \Omega \\ R_1+R_7 & R_2 & 0\ \Omega & 0\ \Omega & 0\ \Omega & R_6 \\ 0\ \Omega & R_2 & -R_3 & 0\ \Omega & -R_5 & 0\ \Omega \\ 0\ \Omega & 0\ \Omega & 0\ \Omega & -R_4 & R_5 & R_6 \end{bmatrix} \quad v:=\begin{bmatrix} 0\ A \\ 0\ A \\ 0\ A \\ 0\ A \\ EMF \\ EMF \\ 0\ V \\ 0\ V \end{bmatrix} \quad \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} := \mathrm{lsolve}(M,v)=\begin{bmatrix} 0.795 \\ 0.238 \\ 0.557 \\ 0.243 \\ 0.314 \\ 0.552 \end{bmatrix} A$$

Fig. 17.16 The solution of an overdetermined system of linear algebraic equations in Mathcad Prime environment

any methods. But it requires us to define the uniqueness of what solution is in question—for solving the electrical problem or a particular system of linear algebraic equations, which, in a general case, can poorly describe the physical model because it has options. In addition, it is confusing that it is necessary to set the initial approximations for the solution.[3] This is usually done when the system has multiple solutions, but it is required to find only one. The user should at least roughly know how many decisions a posed actual problem would have, and what tactics in solving a particular problem should be used. Nonlinear systems would have several solutions, and our system, we repeat, is linear. Such a system usually has only one solution, but there are cases where there are no solutions or infinitely many. Now we will show how it is possible to determine these three cases in a linear system of algebraic equations.

Figure 17.16 shows the solution to our "hydraulic network" problem using one of Mathcad's tools intended for solving systems of linear algebraic equations with the function **lsolve** (l—linear).

In the upper part of Fig. 17.16 one can see our equations arising from the first and second Kirchhoff laws (rules), supplemented with "zero" terms and coefficients so as to obtain a system of linear equations in the classical writing: the coefficients

[3]*Mathematician's comment*. Why is that? Is it always? Which system? Who or what is required to specify them in this case? We need to know the specifics of the numerical method, hardwired into the package.

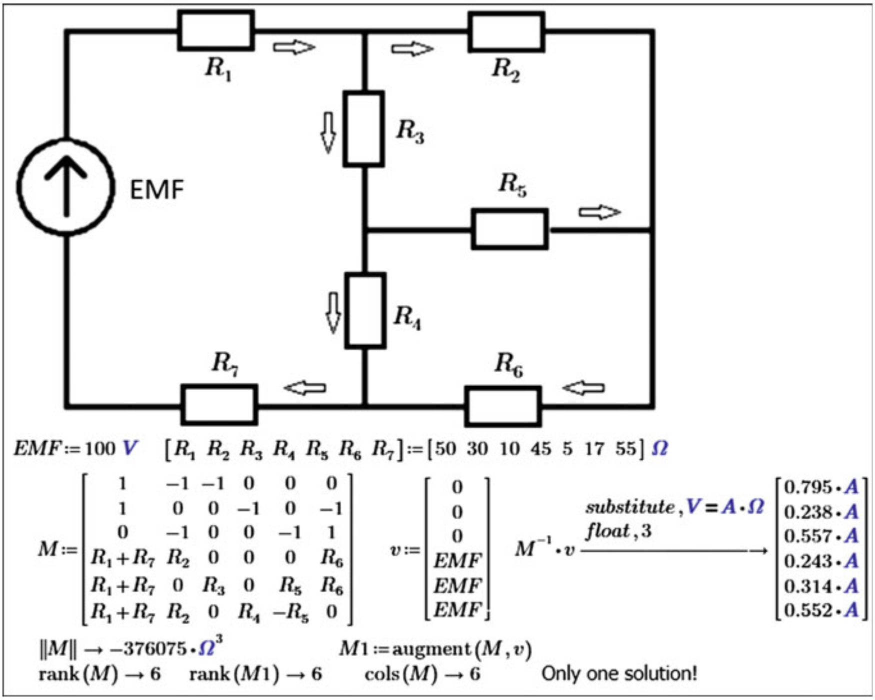


Fig. 17.17 Analysis and solution of the "classical" system of linear algebraic equations in Mathcad prime environment involving the basics of linear algebra

of the unknowns, and the unknowns and the right-side parts of the equations. Then one can create a matrix of coefficients of the unknowns **M** and an absolute term vector **v**. Function lsolve returned us a solution which is a repetition of the solution in Fig. 17.15, but without any "initial approximations" that confused us.

From our system, we can remove the two extra equations,[4] as we have noted above. After that matrix **M** will become square, and it will be possible to calculate the determinant and invert it (find the inverse matrix). The inverse matrix can be multiplied by vector of free members and … we get a solution. That's just where the knowledge of linear algebra, which we acquired at the University, would be useful for us! In Fig. 17.17 is solved a "classical" system of six linear algebraic equations with six unknowns using a "classical" method—vector multiplication of inverted (reverse) matrix of coefficients of the unknowns by free member vectors. The function of rank that returns the rank of a square matrix **M** and an expanded matrix

[4] *Mathematician's comment*. Why are they superfluous, why these are superfluous and not others? From the point of view of the theory of systems of linear algebraic equations, the extra equations are found, and not in the only way, only after finding the rank of the augmented matrix. Here it is probably more appropriate, without going into the mathematical details, to provide physical or technical justification for allocating extra equations.

$$M := \begin{bmatrix} 0 & -1 & 0 & 0 & -1 & 1 \\ 0 & 1 & 1 & -1 & 0 & -1 \\ 0 & R_1+R_7 & R_1+R_7 & R_3+R_4 & R_3 & 0 \\ 0 & R_1+R_2+R_7 & R_1+R_7 & 0 & 0 & R_4 \\ 0 & R_2 & 0 & -R_3 & -(R_3+R_5) & 0 \\ 0 & 0 & 0 & -R_4 & R_5 & R_6 \end{bmatrix} \quad v := \begin{bmatrix} 0 \\ 0 \\ EMF \\ EMF \\ 0 \\ 0 \end{bmatrix}$$

$$M1 := \text{augment}(M, v)$$

$$\text{rank}(M) \rightarrow 5 \qquad \text{rank}(M1) \rightarrow 5 \qquad \text{cols}(M) = 6$$

Fig. 17.18 Analysis of wrongly composed linear algebraic equations

M1 helps to make sure that we have a unique solution. Rouché–Capelli theorem says that "the system of linear algebraic equations is consistent if and only if the rank of its basic matrix equals the rank of its augmented matrix, and the system has a unique solution if the rank is equal to the number of the unknowns, and an infinite number of solutions, if the rank is less than the number of the unknowns."

Our system of six linear algebraic equations with six unknowns (see Fig. 17.17) contains three first Kirchhoff rules and three second Kirchhoff rules recorded. If one disrupts this proportion, e.g. writes two first Kirchhoff rules for two "tees" and four second Kirchhoff rules for four closed loops, then he can get an incorrect solution.[5] And the most unpleasant thing is that this solution will be very similar to the true one.[6] If we change our model (write two instead of three first Kirchhoff rules and four, not three, second Kirchhoff rules), the ranks of the new primary and a new extended matrices thus becomes equal to 5 (see Fig. 17.18), but not 6 (see Fig. 17.17), which will signal an error in the solution, since our system has 6 unknowns (see above the Rouché–Capelli theorem). The conclusion is the following for the problem in question, a mathematical model, adequate to physical model, contains exactly six linearly independent equations which can be properly separated from a greater (excessive) number of consistent equations using either a mathematical approach (calculate the rank of the matrix and the allocation basis minor) or physical approach (determine physical significance of each of the equations). The first approach requires a good knowledge of the general theory of solving systems of linear algebraic equations, the second approach, in addition to the knowledge of the theory of circuits, requires a well-developed intuition and skill

[5] *Mathematician's comment-question*. And you cannot get it? How is it determined?

[6] *Mathematician's comment-question*. What does it mean "similar"? If the solution is unique and approximative (numerically) coincides with the desired, then for the technical calculation this is not a nuisance, but the norm. If the mathematical theory proves the absence of a solution, which is yet present (i.e. calculated), or the existence of an infinite number of solutions (unknowns are more in number than the rank of the system), but only one is calculated, it means that the used model is incorrect. And this is a very big trouble.

in solving such problems. But if you do not aim at reducing the number of equations, the solution of overdetermined algebraic system could be simply charged to a computer.

Analysis of linear equation systems in our "non-linear" Study is not accidental. The original hydraulic system is non-linear, but can be temporarily linearized, i.e. take the rough assumption that the resistance of the pipeline does not depend on the fluid flow rate. After that it will be possible to solve such a linear equation system using the function lsolve (see Fig. 17.16) or multiplying the inverse matrix of coefficients of the unknowns by a free members vector (see Fig. 17.17). For completeness, here you can also add a Gauss method, that, by the way, in theory easily and unambiguously copes with any number of equations and unknowns. The resulting solution is then to be used as a first approximation for the function Find, applied to a nonlinear original problem of a network of pipelines (see Fig. 17.12). By the way, our problem on the electrical circuit (see Figs. 17.15, 17.16 and 17.17) is also nonlinear. With the passage of electrical current the resistance gets heated. And the value of the electric resistance, same as the value of thermal resistance see Chap. 20), strongly depends on the temperature. If electric current is small (with a slight heat resistance), these changes can be ignored, but in case of thermal electrical installations one cannot do this. One can use a ohmmeter to measure the resistance of an incandescent bulb, divide the square of voltage of the grid by the measured resistance of the tungsten filament, and … do not get the value of the bulb's wattage, stamped on it. The electrical resistance of tungsten greatly increases when it is heated. Otherwise, a tungsten filament of a light bulb would immediately burn out when it is turned on. This is an interesting problem at the interface of heat and electrical engineering: to calculate variation in time of a bulb filament temperature when the bulb is switched on or off, as well as at power surges. This problem belongs to the class of problems for differential equations, which we shall consider in next chapter. An analogue of this non-standard heat exchanger problem is shown in Fig. 18.10 in Chap. 18.

Let's go back to the second title of the chapter. Immanuel Kant is credited with the saying: "Every natural science contains as much truth as much mathematics it contains." In the Moscow Power Engineering Institute, where the author teaches a course "Information Technology", freshmen in parallel are taught a course of higher mathematics, containing sections related to linear algebra (see above) and calculus (mathematical analysis). In subsequent semesters they are also given a course of numerical methods in mathematics. Without this triad it is impossible to proceed with thermal engineering—thermodynamics and heat-mass exchange.

Foundations of linear algebra, more precisely, matrix calculus, with the original and inverted matrices, matrix ranks, etc. were illustrated here by solving a problem, as shown in Fig. 17.17. Function **Find** (see Figs. 17.12, 17.13 and 17.14) includes various methods of a numerical solution of systems of linear and non-linear algebraic equations. (One could see the list, in Mathcad 15 environment right click on the function **Find**.) Now we will look at the basics of calculus again using "heat exchange" problems. However, the first of the problems discussed further is not a calculus, since it does not contain a differential or integral calculus, any rows or vector analysis. It is a stereometric problem of elementary mathematics. It shows that the systems of algebraic equations can be not only overdetermined (see above), but also underdetermined. This happens, for example, when the number of equations is less than the number of unknowns. Such a problem can be seen in the description of one more “hydraulic” device—in the geometry of the submarine “Nautilus”—a “character” of Jules Verne’s novel “Twenty Thousand Leagues Under the Sea” and “The Mysterious Island”. Here’s what you can learn from a conversation with Captain Nemo Aronnax on the size of the submarine, “Here, Professor Aronnax, the drawings of the vessel in which you are. The vessel is a highly elongated cylinder with conical ends. <…> Its area equals one thousand eleven and forty-five hundredths square meters, the volume is equal to one thousand five hundred and two tenth cubic meters; in short, the ship fully immersed in water displaces fifteen hundred and two-tenths of cubic meters or tons of water.”

Figure 17.19 shows the solution of the problem of the size of the submarine “Nautilus” in Mathcad’s environment using the symbolic mathematics operator **solve**. This operator returns in this problem not specific numerical values but forms expressions, which are then displayed on the diagram. The fact is that the

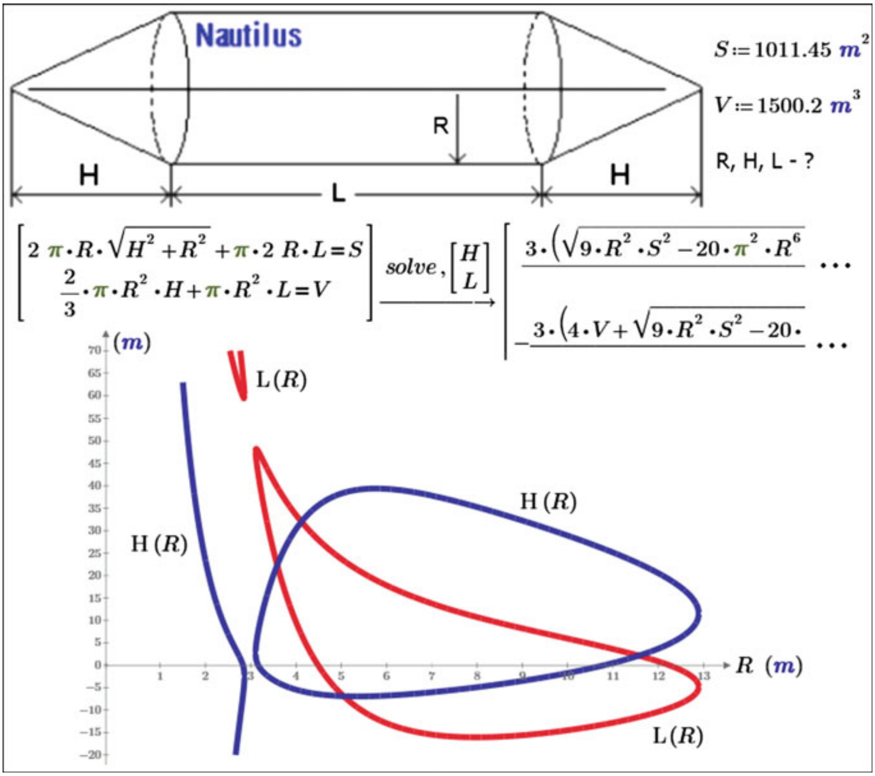


Fig. 17.19 Graphic analytic solution of underdetermined system of nonlinear algebraic equations

underdetermined systems of algebraic equations can have multiple solutions. In this case, the sub dimensions (length **H** and **L**) for a given geometry of the submarine, its known volume and surface area depend on the radius of the cross section of the cylindrical part **R**. In addition, the system of our two nonlinear equation has two solutions. Therefore, the operator solve returned us not a vector but a matrix consisting of two lines (length **H** and **L**) and two columns (two solutions). This matrix is shown in Fig. 17.19 only partially. Here these dependencies are displayed graphically in the range of R from 0 to 13 m. Assuming that the **R**—sectional radius of the cylindrical part of the submarine is, for example, 3.13 m, the length of its conical parts (bow and stern) H equals to 5.627 or 0.823 m and the length of the cylindrical central portion **L** is 44.991 or 48.194 m. The first case (H = 5.627 m and L = 44.991 m) is most likely, since in the second solution (H = 0.823 m and L = 48.194 m) the bow would have been too blunt, that would increase the water resistance when the boat was moving[7] and would not correspond to the image of "Nautilus" in the illustrations for the books by Jules Verne.

Our discussion of the under- and overdetermined systems of algebraic equations from a mathematician's point of view is somewhat flimsy. Just the earlier versions of Mathcad did not have a built-in function **lsolve**, and linear equations systems had to be solved only by multiplying the inverted matrix of coefficients of the unknowns by the free terms vector. The matrix of coefficients of the unknowns in this case could only be square. Announcing of function lsolve in the new Mathcad version emphasizes that now it is possible to solve the under- or overdetermined systems of algebraic equations.

Now we will look at the basics of the application of calculus to the problem which is also related to our hydraulic (thermal) networks.

In the heating network, the loss of water coolant which is stored in a make-up water tank has to be replenished. The tank normally has a form of a right circular cylinder. And what are the proportions? If the cylindrical tank with an upper lid has a preset volume (e.g. 30 m^3), it is possible to find the values of the radius r of the cylinder bottom and its height h, for which the total surface area of the tank is minimal. And therefore, the consumption of metal in the manufacture of the tank and its insulation material will be minimal.[8] This problem is solved in Fig. 17.20 via Mathcad's "numerical" function minimize using again the first approximations—3 m each for the radius and height of the cylinder. Why 3 m? (See our above remarks about the choice of the point of the first approximations to the solution.)

But here we can recall the calculus and solve the problem of the cylinder not for its particular specified volume (an engineering approach to the problem), but for its general form (a mathematical approach to the problem), and to prove that any of a right circular cylinder with a minimum total surface area has its base diameter equal to its height. The solution of this general problem by Mathcad's symbolical

[7]The problem of the dynamics of motion of a body in a viscous medium, we consider in the next chapter.

[8]Engineers have to solve not purely technical but technical and economic problems.

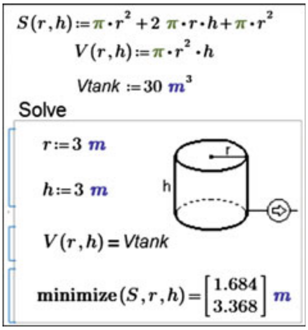


Fig. 17.20 Numerical solution of the problem of the optimal size of the cylinder

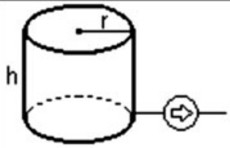
mathematics means is shown in Fig. 17.21 through argument search by value at which the derivative is zero. The forum https://www.ptcusercommunity.com/message/196860 shows numerical and analytical solutions of the optimization problem for the vessels (geometric objects) and other forms—cylinders without a top lid, the upper cone with or without a lid, a cone with hemisphere at the bottom ("ice cream cone") and others.

We have previously mentioned the engineering and mathematical approaches to solving problems. In this regard, we can recall the story. The balloon broke free of the clouds. The people flying saw a man on the ground and shouted, "Where are we?" "You are in the basket of the balloon"—was the answer. It was the answer of a mathematician. Only from mathematician one can hear quite accurate and completely useless answers.[9] Conventionally, it is believed that the use of Mathcad's numerical mathematics is an engineering approach to the problem, and the use of symbolic mathematics—a mathematical approach. A problem (and not only heat engineering problem), of course, should be solved starting from its general statement and one should search for its general solution. But very often it happens that a general solution is "absolutely accurate and absolutely useless." Symbolic mathematics often produces either all the answers (see Fig. 17.21), of which one should also be able to select a desired one, or generates no answer at all. With numerical mathematics it's as if you push the computer to the right solution when you specify the right approximation.[10] It is good to remember or prove for yourself once again (see Fig. 17.21), that the right circular cylinder has the minimum total surface area if its base diameter is equal to its height.

When solving a particular problem about designing a water storage vessel with a specified volume, it is easier to go directly to numerical methods and to minimize

[9]Present-day continuation of this "joke with the beard" is below. People on the balloon shout in response that the batteries got low in their navigator. The man on the ground looks at his smartphone with navigation and shouts back, "You are at the point of 0.9677 rad north latitude and 0.646 rad east longitude. Only mathematicians measure angles in radians, not degrees.

[10]Caution! The computer can be pushed to the wrong decision.



$$V=\pi\cdot r^2\cdot h \xrightarrow{solve,h} \frac{V}{\pi\cdot r^2} \qquad h=\frac{V}{\pi\cdot r^2}$$

$$S=2\,\pi\cdot r\cdot h+\pi\cdot r^2+\pi\cdot r^2 \xrightarrow{substitute,h=\frac{V}{\pi\cdot r^2}} S=\frac{2\cdot(\pi\cdot r^3+V)}{r}$$

$$\frac{d}{dr}\frac{2\cdot(\pi\cdot r^3+V)}{r}\rightarrow 6\cdot\pi\cdot r-\frac{2\cdot\pi\cdot r^3+2\cdot V}{r^2}$$

$$6\cdot\pi\cdot r-\frac{2\cdot\pi\cdot r^3+2\cdot V}{r^2}=0 \xrightarrow{solve,r} \begin{bmatrix} \left(\frac{V}{2\cdot\pi}\right)^{\frac{1}{3}} \\ -\frac{\left(\frac{V}{2\cdot\pi}\right)^{\frac{1}{3}}}{2}+\frac{\sqrt{3}\cdot\left(\frac{V}{2\cdot\pi}\right)^{\frac{1}{3}}\cdot 1i}{2} \\ -\frac{\left(\frac{V}{2\cdot\pi}\right)^{\frac{1}{3}}}{2}-\frac{\sqrt{3}\cdot\left(\frac{V}{2\cdot\pi}\right)^{\frac{1}{3}}\cdot 1i}{2} \end{bmatrix}$$

$$r_{opt}:=\left(\frac{V}{2\cdot\pi}\right)^{\frac{1}{3}} \qquad h_{opt}:=\frac{V}{\pi\cdot r_{opt}^{\;2}} \qquad \frac{h_{opt}}{r_{opt}}\xrightarrow{simplify}2$$

Fig. 17.21 Analytical solution of the problem of the optimal size of the cylinder

the objective function with two unknowns r and h (see Fig. 17.20). However, you can think a little with your computer and figure out that this problem has one unknown quantity—the radius of the cylinder, for example. The minimum of a one argument function can be found in the diagram (Fig. 17.22). But we can limit ourselves to a solution with two unknown quantities[11] (see Fig. 17.18) using the KISS-principle: keep it simple, stupid! The main thing here is a skillful combination of analytical and numerical methods for solving problems and … participation of mathematicians in their critical analysis. Let me give a concrete example. Author of the book sent this chapter to one good mathematician, who corrected the text and

[11]In the transportation problem shown in Figs. 7.4, 7.5 and 7.6, we were guided by the KISS-principle and introduced four unknowns, although it was possible to have only one unknown.

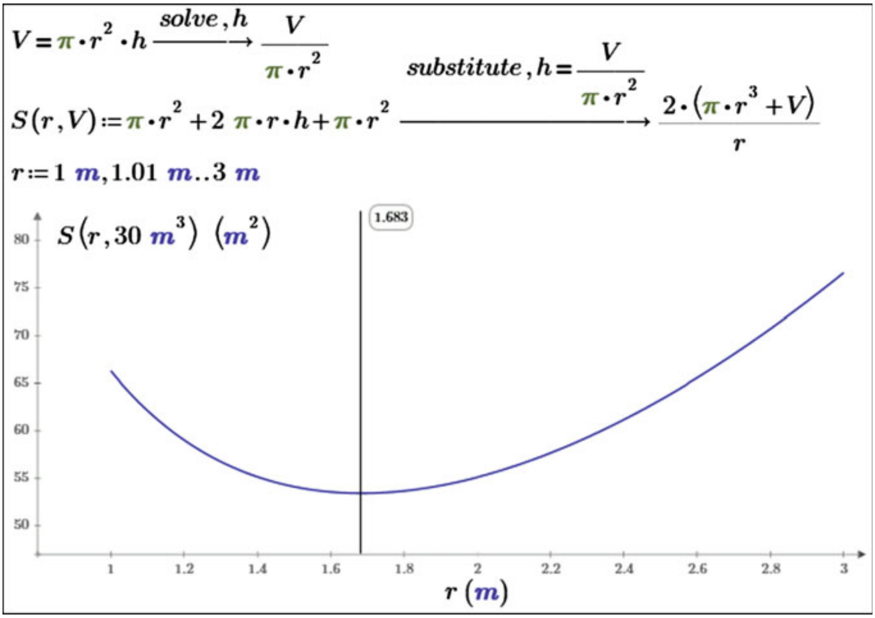


Fig. 17.22 The combination of analytical and graphical methods in solving the problem of the optimum size of the cylinder

made valuable comments, as shown in the footnotes. Then the mathematician corrected the other studies and became a co-author. Well, the mathematics itself needs to be learned, even with the help of Mathcad [64, 65].

The problem of the size of the cylinder can be easily solved in MS Excel environment using **Solver** (see Fig. 7.7 Chap. 7).

Let's go back to the first title of the chapter—to quadrogeneration (CCHPI). Additional generation of information at a thermal power plant, as was mentioned above, is, of course, pure exotica. Although it is possible to give an example, when a Moscow CHP let into the building of its former compressor station one large IT-company that has installed there its powerful servers and piston generators for backup, an uninterrupted power supply for computers. These servers require cooling, and the room itself, where they are installed, requires air conditioning. Here you have an example of quadrogeneration (CCHPI)—one industrial facility simultaneously generates both heat and electricity (all CHP do it), and also cold, and… information.

But power plants, producing heat and electricity, can be useful in other areas. Some thermal power plants in Western Europe, for example, burn the garbage. And not just the one that is formed by the current city life and which cannot be disposed by other means, but also the "old" waste stored on the so-called "landfills" that represents a real environmental threat, e.g. to groundwater. In addition, these landfills have a "bad smell" in the literal and figurative sense. Another benefit from a power plant may be as follows. Large cities in Central and Northern Russia, the Urals and Siberia in the winter have significant problems with the disposal of snow removed from the roads. This snow is not simple water, but is highly contaminated with icing reagents, which are sprayed over the road after each snowfall. Our municipal CHP in the winter period could store the snow on its territory and in summer melt the snow separating icing reagents from it for their subsequent reuse in the coming winter. The purified water can be used to make-up the power units and cooling systems of turbine condensers. Efficiency of such a power plant, by the way, would rise by reducing the temperature of cooling water flowing to the condenser [66, 67]. Power plants have chemical shops where salt and other impurities are removed from the natural water. They can be used to clean "last year's snow" melting water.

In many northern countries (Scandinavia, Canada), they never spray icing reagents over the fallen snow on the country roads, the snow is simply raked from the road or tamped straight on the road and sometimes dusted with granite chips, which in the spring is collected, cleaned of the mud and used again. But the use of granite chips on city streets is problematic, also due to the fact that they will clog the sewers where the snow is dumped from snow melting plants, for example, or in a natural way in spring or in a thaw.

For snow-melting in Moscow they started to apply not only sewage wastewater, but also warm water from CHP heated in the turbine condensers with straight-through cooling systems where water is taken directly from the river and discharged back into the river. With this in mind, it should be recognized that the scheme of dumping snow into CHP drains is similar to the old rejected scheme of direct dumping of snow into the rivers. One "cosmetic" difference—the platforms which are frightening the environmentalists and used to dump the snow from the trucks are no longer built on embankments—now the snow is dumped into a river remotely (by the way, this method have been used all the time at CHP when dealing with snow on the territory of the power plant). But the CHP in Moscow may be more actively and in a more civilized way involved in solving the problem of snow in the city and addressing some of their own problems.

We ourselves need to clearly understand, that the snow containing various icing regents cannot be discharged into the sewerage system. This snow, we repeat, should be segregated from the regular snow when collected, then it should be processed and separated from the icing reagents which may be reused, if the city must spray roads with icing reagents. This should be done, not only considering the sewage farm's environment, but simply by a common cultural principle, "it is necessary to clean up after yourself!". Moscow CHP can be very useful, especially those that are located closer to the center of Moscow.

Imagine such an ideal scheme for processing contaminated snow. Snow, more precisely, the brown muddy mush, taken from the Moscow roads sprayed with icing reagents, is brought into special bins at Moscow CHP sites. Maybe it is not even worth melting the snow in winter, especially spending the fuel, and better to wait till warm weather. In summer, the snow will be melting by itself and also due to the CHP waste heat, increasing, by the way, the efficiency of the power plant due to the lower temperature of steam condensation in condensers. CHP chemical shops will prepare make-up water for boilers not from natural water, which, by the way, they need to buy, but from this melt water and they will extract icing reagents to be reused next winter. At the same time you can find some kind of balance: how much reagents is spent to fight ice, how much these reagents is extracted from the “snow mush” and how much of them, alas, remains in the city, including those on the lawns, in the parks and public gardens. All this, of course, requires more detailed technical and economic calculations and elaborations. But this should be given consideration right now.

A CHP in the center of a city can survive if in addition to heat and power generation it does something else for the benefit of the city. Who will in the near future endure the smoking pipes and steaming cooling towers in the city center, and not only in the center of Moscow, if alternative, more environmentally friendly and more energy-efficient methods of electricity and heat supply are emerging and have been already implemented somewhere? Moscow CHP can survive only if they radically change the technology of its production … and even changed its name. What is CHP now?! This is an energy company in the city, which generates heat in the form of hot water or steam and simultaneously produces electricity. If a CHP wants to remain “in the city”, it should start solving the first line pressing urban problems: disposal of snow (both fresh and last year snow), recycling waste, etc., while simultaneously producing heat and electricity.

Chapter 18
Differential Equations in Thermal Engineering

Valery Ochkov

Abstract This chapter describes the common types of differential equations which you will work when solving the problems of thermal engineering tasks. Described the numerical methods for solving such problems. Continued analysis of finite element methods applied to heat conduction problems.

In the previous chapter, we reviewed the possibilities of Mathcad for numerical solution of the system of linear and nonlinear algebraic equations. The tools described there are used to derive values of variables that convert equations into identities. Specifically, into nearly identities, the right and left sides of equations are slightly different from each other, they are the same within specified numerical tolerance. Such "slightness" (it is equal to 0.001 by default) is stored by the system variable **CTOL** (**c**onvergence **tol**erance). This value can be changed, while maintaining the balance, the ratio of mathematical precision and real physical error in real physical units. In practice, the error is often so great that calculations with mathematical precision of 0.001 are meaningless. Chapters 6 and 17 given in the book describe the symbolic (analytical) solution of the system of algebraic equations, where solutions represent formulas rather than numbers.

The site of the chapter: https://www.ptcusercommunity.com/message/423039.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_18

Mathcad can numerically solve differential equations and systems of differential equations, where solutions would represent functions rather than numbers. If no additional initial or boundary conditions are given, then the number of functions being the solution of a differential equation (system of equations) is infinite. In the analytic solution, substitution of the obtained function in the differential equation transforms this equation into an identity. If an equation or system meets certain conditions (theorem of unique existence), then a unique solution exists satisfying given initial conditions. In the numerical solution, a sequence of points is sought that lie in the arbitrary neighborhood of the function graph with the preset parameters of a numerical method being a precise solution. Points of such sequence connected in a certain order by straight or curved sections (interpolation), are usually called a graph of approximate (numerical) solution of the corresponding differential equation or system. Numerical solution of the problem can only be sought if there is confidence in the existence and uniqueness of the analytic solution.

Description of Mathcad tools intended to solve differential equations and their systems, will be given by means of an example that supplements heat engineering, rather hydrotechnical examples given in Chaps. 16 and 17.

At the heat engineering facilities, the main task is often heat removal rather than heat input. We can generate electrical energy at the thermal power plant (TPP) only when part of heat produced from combustion of organic or nuclear fuel is discharged to the environment. This follows from the basic laws of thermodynamics, and this is determined by the thermal efficiency of the power plant, which we calculated in Chap. 4 and optimized (maximized) in Chap. 6. At the power station, such heat is quite often discharged into the atmosphere through water spray in cooling towers or special spray cooling ponds. The turbine exhaust steam enters the condenser through the tubes of which cooling water flows. Water steam condenses on the outer surface of condenser tubes and is converted into water (condensate). Condensate is recycled to the steam boiler (see Chap. 10) through regenerative heaters (see Chap. 6) by the feed pump (see Chap. 7). Thus, cooling water is heated. Then it is either discharged to the water pond, where from it is taken (river, lake, sea, power plant cooling pond, etc.), or cooled down in spray cooling ponds or cooling towers, and then recycled to the steam turbine condenser (so-called closed loop of process water supply to the power station). Figure 18.1 shows one of the spray cooling ponds intended for water cooling.

The figure also shows a piping system similar to that calculated in Chap. 17. The pipes are fitted with nozzles which spray and cool down water, mainly, by means of partial evaporation of moisture and convective heat exchange with the ambient air. The cooling tower is different from the spray cooling pond in that the entire piping system with nozzles is enclosed in the concrete, wooden, or metal shell in the form of a wide pipe tapered upwards that creates air draught thereby intensifying heat

Fig. 18.1 Spray cooling pond

and mass transfer.[1] Water, by the way of spraying, may be enriched with oxygen. Such spray systems are sometimes installed in dying reservoirs for water aeration, cooling, and, consequently, slowing certain "harmful" biological processes therein. The famous 140-m fountain on the Geneva Lake in Switzerland (see Fig. 18.7) is not only a tourist attraction and advertising for the company that supplied the pump for the fountain, but also a device for feeding the lake water with oxygen and cooling it to some extent.

Spray cooling ponds are also widely used in industrial air-conditioning and refrigeration systems (see Chap. 19).

Let's fit the pipe end shown in Fig. 16.6 with several nozzles at different angles to the horizon from which water will flow at the speed of v_0, and see (calculate) how water will flow out breaking into droplets. We'll simplify the task—we'll simulate the flight of a single spherical drop of water under the force of inertia, gravitational forces, and air resistance. Rather, we should say as follows: no force of inertia exists, but there is the Newton's law, according to which the sum of forces acting on the body (on our droplet)—friction force and gravitational force—is the product of body mass by its acceleration. In daily use, this product is sometimes referred to as the force of inertia.

The problem will feature two pairs of required functions depending on time t: droplet position in the space, or rather, in the xy plane (x axis is a horizontal line, and y axis is a vertical: we solve the planar rather than spatial problem) and droplet speed v using these two coordinates.

Speed is the first path derivative with respect to time, and acceleration is the second path derivative with respect to time, or the first speed derivative with respect

[1]There are also so-called dry cooling towers (Heller cooling towers) in which water and air are segregated by a metal (usually aluminum) wall. Water cooling in such cooling towers is not as inefficient, but there is no loss of cooling water due to evaporation and droplet entrainment. In "wet" cooling towers and spray cooling ponds salt concentration in water is increased due to evaporation, which might lead to deposition of scale in the tubes of the steam turbine condenser, increase in steam condensation temperature and reduction in efficiency of the power unit. The network calculation placed at http://twt.mpei.ac.ru/MCS/Worksheets/Water/acidation.xmcd, allows us to calculate measures aimed at prevention of scale formation in the turbine condensers.

to time. Hence, the differential equation is obtained—the equation of balance of forces applied to the droplet. Such forces will be laid along x and y axes followed by occurrence of pairs of equations. The frictional force acting on the droplet depends on a lot of parameters. In aerodynamics, it is conditionally assumed that it is proportional to the cross sectional area of the droplet S, air density ρ_{air} and droplet velocity squared v. The droplet in flight itself would remain spherical—a two-dimensional material point with the properties of a spherical droplet (assumption). The proportionality factor f in the resistance calculation formula depends on the dimensionless Reynolds Re number (criterion) only (product of the droplet speed v and its diameter D, divided by air kinematic viscosity $\mathbf{v_{air}}$). Here, as in the problem about water flow in the pipe (see Chaps. 16 and 17), observed are laminar, transitional and turbulent modes of air flow-around by the drop of water. In the upper part of Fig. 18.2, dependence is graphically recorded and displayed, "picked up" from the Internet (see https://www.ptcusercommunity.com/message/223872). With the Reynolds number value greater than 576, the factor **f** becomes constant (0.5—see graph in Fig. 18.2).

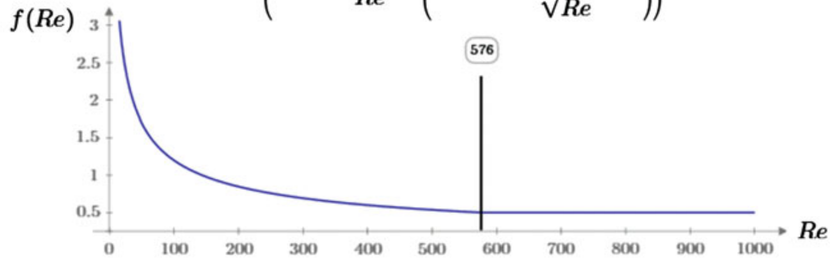
Figure 18.2, following the graph and table with input data[2] (wind speed is additionally introduced[3] $\mathbf{v_w}$[4]), shows calculation of the drop mass value mass, and its cross section **S**. Further, the Solve unit is inserted in the calculation with the field named Restrictions stores the description of our problem: initial conditions (droplet position and velocity at the initial time with t = 0 s), four differential equations describing velocity as the path derivative (horizontally and vertically) in terms of time and balance of forces acting on the droplet in two directions. The Solver field of the Solve unit contains a odesolve function built into Mathcad designed for the numerical solution of differential equations and their systems. The custom function odesolve returned four user-defined functions named **x**, **y**, $\mathbf{v_x}$ and $\mathbf{v_y}$. These are not quite the usual functions, since they cannot be just sent to the "printer" by the operator x(t) → , i.e. by the formula composed of certain elementary functions (**sin**, **cos**, etc.) and elementary operators (addition, subtraction, exponentiation, etc.). Dependencies for **x**, **y**, $\mathbf{v_x}$ and $\mathbf{v_y}$ were obtained by interpolating the desired functions based on their tabulated values generated during the numerical solution of differential equations. However, functions **x**, **y**, $\mathbf{v_x}$ and $\mathbf{v_y}$ might be used for calculations (for example, to find out the droplet coordinates and/or its speed in a second of flight) and plotting. This is exactly what we have done (Figs. 18.3, 18.4 and 18.5).

[2]Temperature and pressure values used to determine the values of such thermal and physical parameters, can be entered rather than values of water and air densities and air kinematic viscosity value. We did so in other studies of the book, based on the "cloud" features and templates.

[3]Wind affects spray cooling ponds operation. The above mentioned fountain on the Geneva Lake is automatically deactivated at high wind speed, elsewise water from the fountain would irrigate the lakeshore.

[4]The wind blows from right to left. But the wind speed component could be taken into account in the vertical direction. It takes place in the above mentioned cooling towers with the vertical air draught. We can also switch to the spatial task, i.e. enter the wind speed component in the direction perpendicular to *x* and *y* axes.

$$f(Re) := \mathbf{if}\left(Re < 4, \frac{24}{Re}, \mathbf{if}\left(Re < 576, \frac{12}{\sqrt{Re}}, 0.5\right)\right)$$



D	ρ_{air}	ρ_w	ν_{air}	v_0	α	v_w	t_{end}
(mm)	$\left(\frac{kg}{m^3}\right)$	$\left(\frac{kg}{m^3}\right)$	$\left(\frac{m^2}{s}\right)$	$\left(\frac{m}{s}\right)$	(deg)	$\left(\frac{m}{s}\right)$	(s)
1	1.205	998.2	$1.5 \cdot 10^{-5}$	15	60	1	2.5

$$mass := \pi \cdot \frac{4}{3} \cdot \left(\frac{D}{2}\right)^3 \cdot \rho_w = 0.523\ mg \qquad S := \pi \cdot \left(\frac{D}{2}\right)^2 = 0.785\ mm^2$$

Solve

$$x(0\ s) = 0\ m \qquad y(0\ s) = 0\ m \qquad v_x(0\ s) = v_0 \cdot cos(\alpha) \qquad v_y(0\ s) = v_0 \cdot sin(\alpha)$$

$$v_x(t) = \frac{d}{dt} x(t) \qquad v_y(t) = \frac{d}{dt} y(t)$$

$$mass \cdot \frac{d}{dt} v_x(t) = -f\left(\frac{(v_x(t) + v_w) \cdot D}{\nu_{air}}\right) \cdot \frac{(v_x(t) + v_w)^2}{2} \cdot S \cdot \rho_{air}$$

$$mass \cdot \frac{d}{dt} v_y(t) = -f\left(\frac{v_y(t) \cdot D}{\nu_{air}}\right) \cdot \frac{v_y(t)^2}{2} \cdot S \cdot \rho_{air} - g \cdot mass$$

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} := \mathbf{odesolve}\left(\begin{bmatrix} x(t) \\ y(t) \\ v_x(t) \\ v_y(t) \end{bmatrix}, t_{end}\right)$$

Fig. 18.2 Simulation of free flight of spherical drop

It would be useful to generate functions being the solution of the system of differential equations with additional parameters. In our problem of a flying droplet, the main argument of unknown functions **x**, **y**, $\mathbf{v_x}$ and $\mathbf{v_y}$ is time **t**. The additional parameter in solution of this problem might be, for example, the nozzle inclination angle. This allows to plot curves not only using one jet (see Figs. 18.3, 18.4 and 18.5), but also using several jets. This could be done with the **odesolve** function by means of

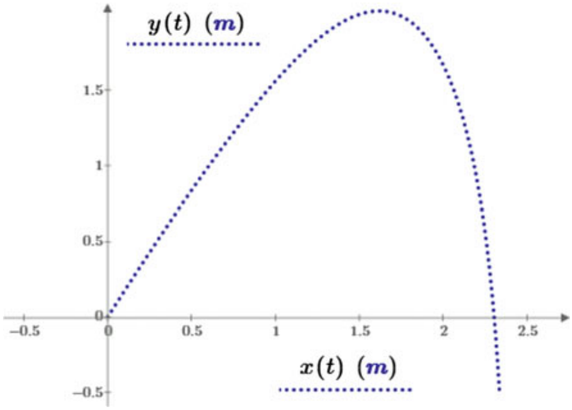


Fig. 18.3 Flight of water drop in the absence of wind

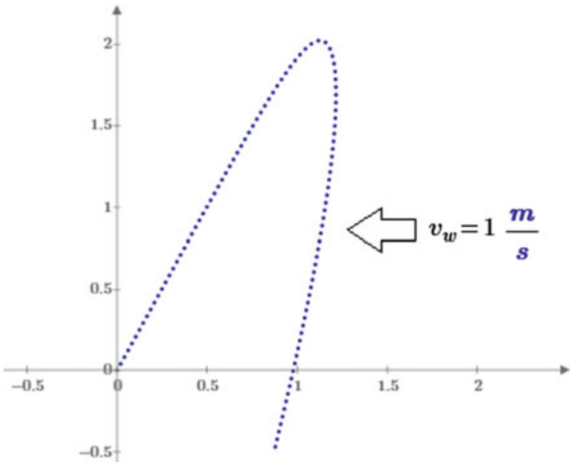


Fig. 18.4 Flight of water drop in weak headwind

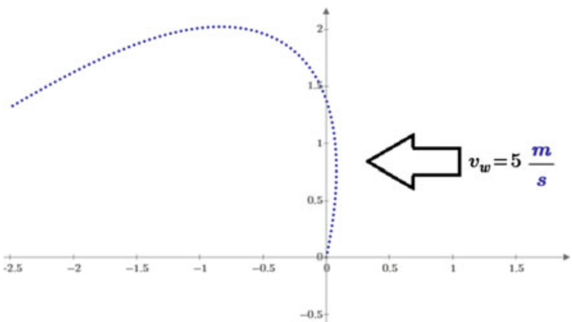


Fig. 18.5 Flight of water drop in strong headwind

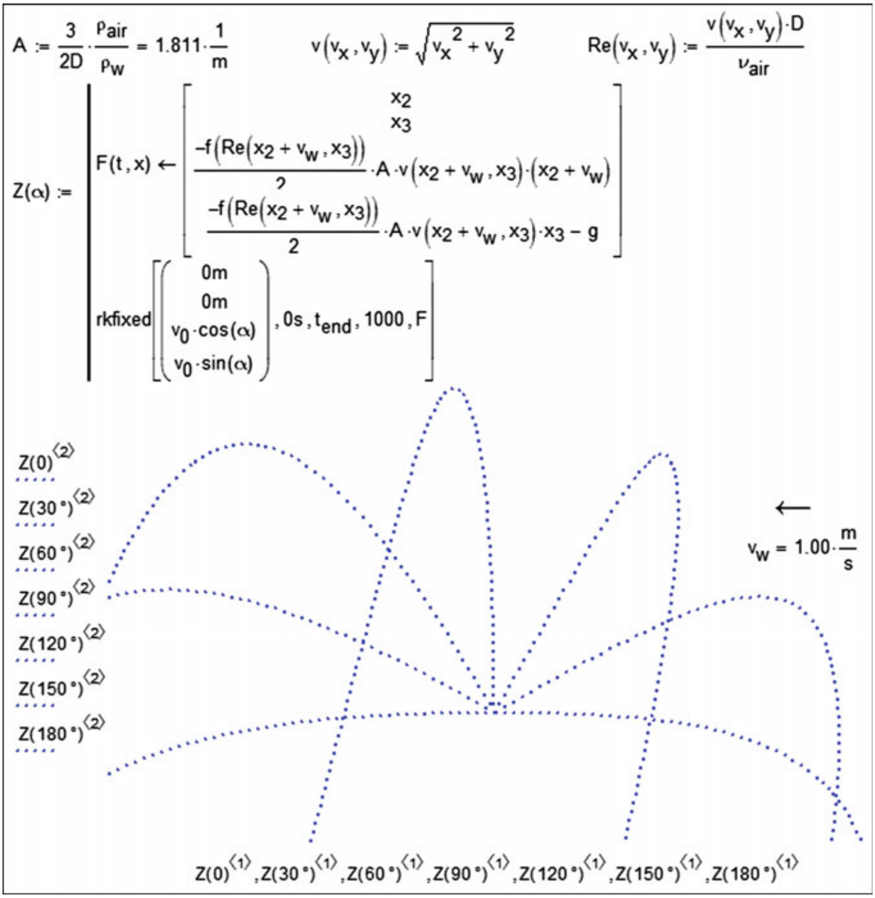


Fig. 18.6 Solution of differential equations with additional parameter

an undocumented technique described at the forum https://www.ptcusercommunity.com/message/228509. The documented technique in solving this problem (using a local function) is shown in Fig. 18.6.

The **rkfixed** function built into Mathcad involved in the solution shown in Fig. 18.6 returns the function-matrix that stores discrete values of **t**, **x**, **y**, $\mathbf{v_x}$ и $\mathbf{v_y}$ (columns of matrix). The last four values (**x**, **y**—droplet coordinates and $\mathbf{v_x}$, $\mathbf{v_y}$—its speed horizontally and vertically) in Fig. 18.6 are designated as the vector elements (columns of matrix) **x**—$\mathbf{x_1}$, $\mathbf{x_2}$, $\mathbf{x_2}$ и $\mathbf{x_4}$. The zero column of our function-matrix contains discrete values of time **t**. Our function-matrix contains 1000 rows by the fourth argument of the **rkfixed** function.

By varying the wind speed and keeping other initial values constant, animation can be created—the live picture by modifying the path of the water droplet flight at different nozzle inclination angles: 0, 30, 60, 90, 120, 150, and 180 angular degrees. One frame of such animation at the wind speed of 1 m/s is shown in Fig. 18.6.

The technology of creating such animation is described in Chap. 12. The animation itself can be viewed here: https://www.ptcusercommunity.com/videos/4530.

The system of equations shown in Fig. 18.2 should be, of course, much more complicated. It should, for example, more accurately take into account the velocity components along horizontal and vertical directions and absolute value of water droplet velocity, as is done in the solution shown in Fig. 18.6. Such nuances are discussed at forum https://www.ptcusercommunity.com/message/228797. But it is possible to get confined to a simplified solution shown in Fig. 18.2, taking into account the fact that our spherical water droplet in the actual flight will deform and break up into smaller droplets. Our simplified solution gives a quite believable animation. And this is sufficient for initial (quality) problem statement. The problem can be further simplified: assume, for example, the air friction coefficient of the droplet equal to 0.5—see the first formula and graph in Fig. 18.2.

The correctness of our model of the droplet flight can be verified qualitatively rather than quantitatively: for example, compare two images—the flight curve of a virtual droplet and type of real fountain of Geneva, mentioned above (Fig. 18.7).

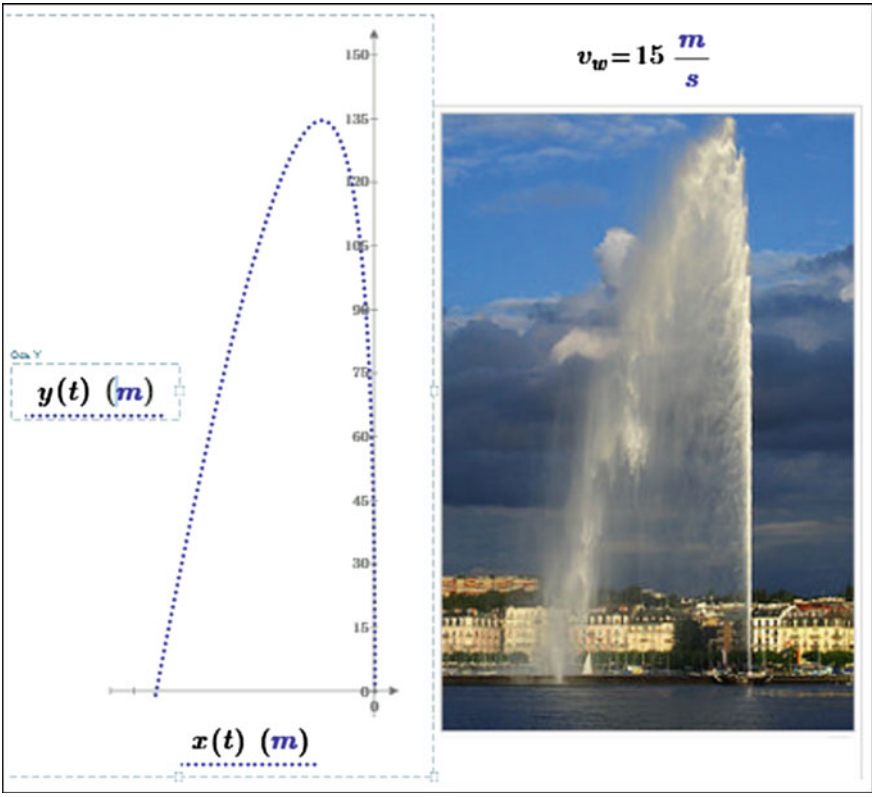


Fig. 18.7 Virtual and real fountains

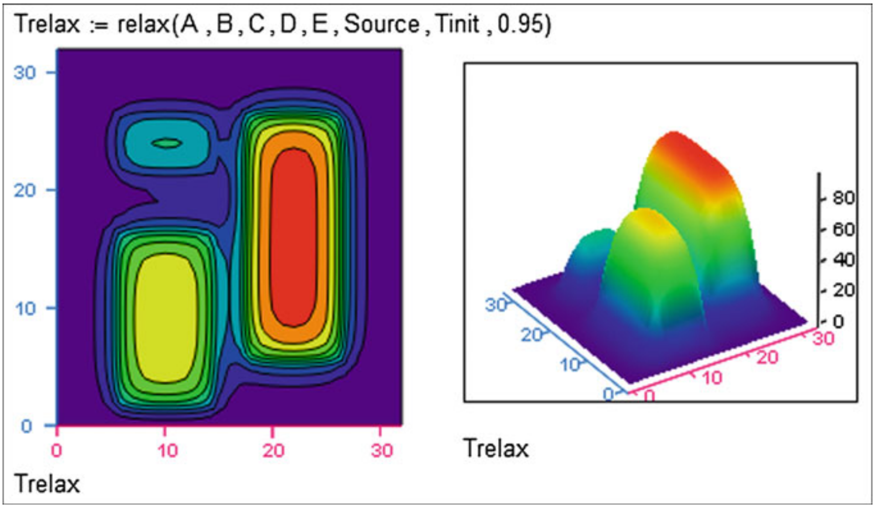


Fig. 18.8 Simulation of temperature field of computer board

Figures 18.3, 18.4, 18.5 and 18.6 showed statement and numerical solution of the initial problem for the system or problem with the initial conditions of so-called ordinary differential equations—see letter o in the title of the odesolve function. The solution of such system is a function of one argument.

But in Mathcad 15 environment, tools are available that generate a function of two arguments while solving differential equations. One such problem was just at the author's hand who wrote and corrected the text the tablet PC, the back wall of which was warm in several places. This resulted in cooling down the tablet chips without using noisy and energy consuming fans built into laptops and desktops.[5] Figure 18.8 shows the final part of the Mathcad-document that calculates the temperature field of three chips on the computer board with two dimensions of x and y.

The problem displayed in Fig. 18.8, uses the relax function built into Mathcad, which returns the temperature values matrix (chip temperature in our problem) depending on the input data—the chip coordinates, their specific heat evolution and heat transfer conditions (arguments of the relax function). This problem is well described in detail in the electronic manual in terms of heat and mass transfer by Solodov [14]—http://twt.mpei.ac.ru/solodov. On this site, the reader will find other useful materials related to the topic of this book, use of Mathcad and Internet for thermotechnical calculations.

Based on the function of two variables rather than one variable, two partial derivatives can be assumed—for the first argument and for the second argument. The **pdesolve** function built into Mathcad 15 (**p**—partial) is designed for numerical

[5]Elements of supercomputers are sometimes provided with water cooling. Water in such systems can also be cooled down in spray cooling ponds, which were simulated in this chapter.

solution of the differential equation in partial derivatives. Sometimes, such equations are, by mistake, called the equation of mathematical physics. Equations in partial derivatives and equations of mathematical physics—two different, though overlapping, mathematics branches. The equation in partial derivatives is not required to describe the actual physical process, and the mathematical physics equation need not be differential: it might well be integrated (see Fig. 20.1) or integral-differential. The pdesolve function work is shown in Fig. 18.9

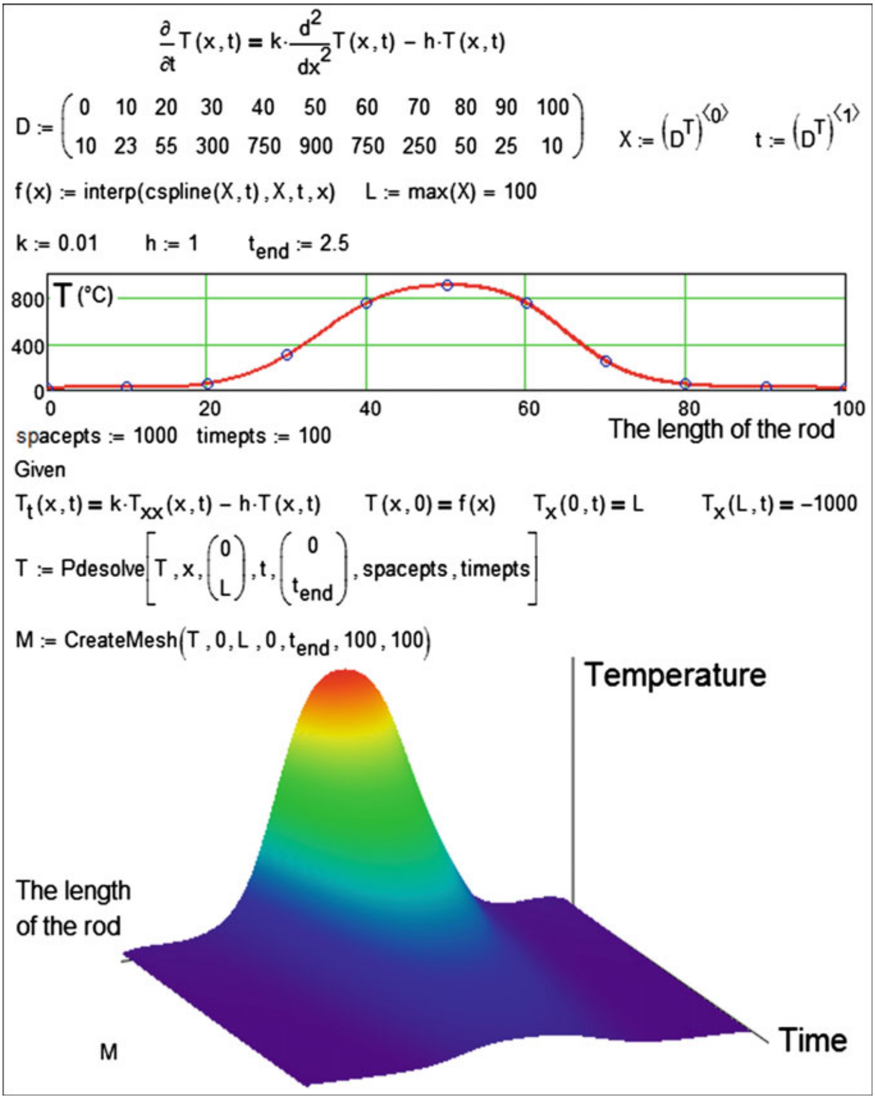


Fig. 18.9 Numerical solution of non-stationary heat transfer problem

examplified by solving the problem of non-stationary heat conductivity. The problem shown in Fig. 18.8, as well as in Chap. 20—those were stationary heat conductivity problems, when thermal characteristics of the facilities does not change with time.

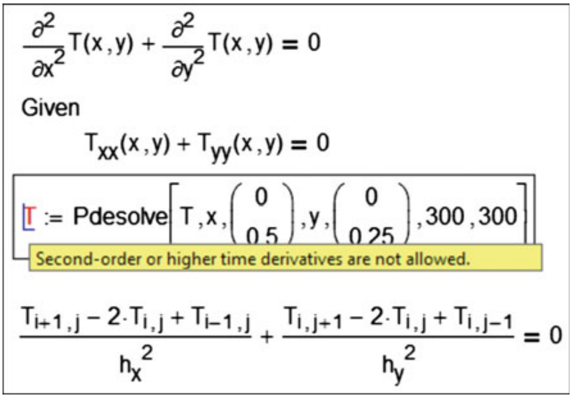
The essence of the problem: take a thin metal rod with the central part being heated. Then the rod cools down due to heat exchange with the environment. It is required to find temperature values of the rod along its length (the first argument of the unknown **T** function) and upon a certain time period (the second argument of the **T** function). Heat of the rod, as was mentioned above, is lost from the side surface of the rod into the environment whose temperature is 0 °C. The rod ends are heat insulated. Heat exchange in such a system is defined by two constants **k** and **h**, that establish intensity of heat transfer along the rod length and into the environment. Problem solution: temperature in the rod as a function of two arguments **x** (point position on the rod) and **t** (time) is sought through the **pdesolve** function (see Fig. 18.9), where the desired **T** function is displayed by the surface. It is assumed that in such a setting the three-dimensional problem (the rod is a section rather than a body), heat transfer in the space is equivalent to that of the one-dimensional problem. The initial temperature value along the rod was set by eleven points, based on which spline-interpolation was made (see Chap. 1). From the book site, animation can be downloaded showing the rod cooling down in time.

Of course, to solve more complex problems of stationary and non-stationary heat transfer, more powerful computing resources should be uses (e.g. COMSOL program—www.comsol.com or ANSYS—www.ansys.com) solving problems by the finite element method. Some of such "finite" elements (parallelepiped-chip and thin cylindrical rod) have been calculated—see Figs. 18.8 and 18.9. A more sophisticated thermotechnical facility is divided into millions of similar elementary finite facilities connected to each other. Example—in Chap. 16, where we, to calculate thermal engineering parameters of the pipe, divided it into separate short sections (see Fig. 16.10). In the pre-computer era, it was impossible even to think about solving such a complex computational problem. In the numerical solution of problems by the finite element method in the Mathcad environment, programming tools will need to be used.

The **pdesolve** function built into Mathcad 15 has very elegantly solved the problem of non-stationary temperature field in the metal rod (Fig. 18.9)—restored the **T(x, t)** function which describes the rod temperature depending on its geometrical point (**x**) and time (**t**). But one should not be greatly reliant on the **pdesolve** function. The **pdesolve** function, for example, has not coped with the problem of finding the temperature field for the two-dimensional stationary problem without internal heat sources, which is reduced to solution of a differential equation in partial derivatives, as shown in Fig. 18.10.

Due to this and other reasons, the **pdesolve** function was deleted from the list of those built into Mathcad Prime. But this problem is not difficult to solve using programming tools in the environment and Mathcad 15 and Mathcad Prime by substitution of the differential for the difference that was already done in Chap. 16 (see Figs. 16.15 and 16.16). In Fig. 18.10, under an error message (it is

Fig. 18.10 Unsuccessful attempt to call **pdesolve** function



not quite true—the second derivative is based on the second measurement rather than on the time), the finite-difference approximation in the grid nodes is shown into which a flat plate is divided [60]. Using such approximation, one can make a system of difference equations and solve it by the iterative method.

In Fig. 18.11, the Mathcad Prime 3 software is shown to solve this heat transfer problem: given is a rectangular plate with the pre-set temperature $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$, and $\mathbf{T}_4$ in four angles of the plate. Along the plate edges, the temperature varies linearly from one angle to another.[6] Find temperature distribution in the plate—generate the function **T(x, y)**.

In Fig. 18.11, in addition to parameters of the plate itself, the following values have been further defined: $\mathbf{n_x}$ and $\mathbf{n_y}$—the number of grid divisions on the plate in the horizontal (**x**) and vertical (**y**) directions, disparity of the solution ($\boldsymbol{\varepsilon}_\mathbf{T}$) and maximum number of iterations (**n**). In the program shown in Fig. 18.11, operators are intended for the following:

- setting the **Temperature($\mathbf{T_0}$, $\mathbf{T_1}$, x)** function for linear temperature distribution between two points;
- temperature calculation at the grid nodes at the plate edges: $T_{\mathbf{i,1}}$, $\mathbf{T_{i,ny+1}}$, $T_{\mathbf{1,j}}$ и $T_{\mathbf{nx+1,j}}$; thus, the remaining (internal) matrix elements become zero;
- calculation of the horizontal ($\mathbf{h_x}$) and vertical ($\mathbf{h_y}$) grid spacing, as well as auxiliary value of **R**[7]; moreover, the **T** matrix is copied into the **TT** matrix; the auxiliary **TT** matrix will store temperature values at the grid nodes at the previous calculation; the first calculation does not contain the previous calculation,

[6]These are the boundary conditions of the first kind. Under boundary conditions of the second kind, heat flows at the body boundaries, and under boundary conditions of the third kind—heat exchange conditions (heat exchange coefficient values).

[7]This value should not be entered into calculation, and directly used in the iterative formula to make it clearer to understand. But from the cycle, only that can be taken out which can be calculated outside the loop. This speeds up calculation. And it might take us a lot of time, because it involves nested loops with a lot of iterations.

a (m)	b (m)	T_1 $(°C)$	T_2 $(°C)$	T_3 $(°C)$	T_4 $(°C)$	n_x	n_y	ε_T (K)	n
0.5	0.25	100	50	200	150	100	50	0.0001	10000

```
T := | Temperature(T0, T1, x) ← T0 + (T1 − T0)·x
     | for i ∈ 1..nx+1
     |   [ T[i,1] ← Temperature(T1, T2, (i−1)/nx)   T[i,ny+1] ← Temperature(T4, T3, (i−1)/nx) ]
     | for j ∈ 1..ny+1
     |   [ T[1,j] ← Temperature(T1, T4, (j−1)/ny)   T[nx+1,j] ← Temperature(T2, T3, (j−1)/ny) ]
     | [ hx ← a/nx   hy ← b/ny   R ← (hx/hy)^2   TT ← T ]
     | for s ∈ 1..n
     |   | IsεT ← 1
     |   | for i ∈ 2..nx
     |   |   | for j ∈ 2..ny
     |   |   |   | T[i,j] ← ((T[i−1,j] + T[i+1,j]) + R·(T[i,j−1] + T[i,j+1])) / (2·(1+R))
     |   |   |   | if |T[i,j] − TT[i,j]| > εT
     |   |   |   |   | IsεT ← 0
     |   | if IsεT = 1
     |   |   | return T
     |   | TT ← T
     | "No solution"
```

Fig. 18.11 Solution of problem about temperature field in the plate

thus, this matrix is introduced artificially here (these four operators are written in one line in the form of a matrix with one line and four columns (undocumented technique in the Mathcad environment) in the program for its compactness);

- in the loop with the **s** parameter, the **Is**$\boldsymbol{\varepsilon_T}$ feature is introduced, which is assigned a single value; if this feature remains equal to unity (see below), this means that solution has been found in the next approximation, since the temperature value in each grid node is approximately equal to such value in the previous approximation, rather, it differs in terms of modulus by no more than the predetermined $\boldsymbol{\varepsilon_T}$ value;
- further, in the loop with s parameter, approximations are made to solving the problem: two nested loops with **i** and **j** parameters fill the **T** matrix—further approximation to the solution, and then are verified if the value of the next element of such matrix is close to the corresponding value of the **TT** matrix

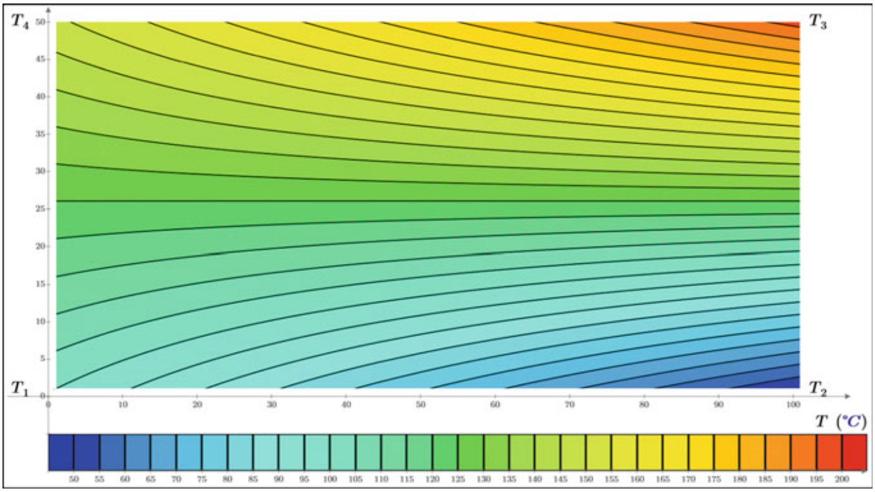


Fig. 18.12 Temperature field in flat plate

element (previous approximation). If this is performed for all elements of such two matrices, then the calculation is finished, and **T** matrix returned, if not, approximations continue, but not indefinitely long and until when the value of the **s** variable exceeds the **n** variable value. In this case, "No solution" message is issued.

Figure 18.12 shows solution of the problem about temperature field in the plate through plotting the level lines (Counter Prot), where the isotherms are drawn from 60 to 190 °C with the pitch of 5 K. Isotherms T = 50 °C and T = 200 °C turned to points (right angles of the plate). Below the graph, the temperature color scale is given.

Afterword. This terrible word "difeqs"…

Here's what you can read about difeqs in one online dictionary of youth slang:

"Meaning: differential equations, a system of differential equations. The training course on differential equations, systems of differential equations, or differential calculations in general. Besides, the appropriate examination, lecture, course of lectures, assignments, etc. Examples: And yet this problem can be solved through difeqs. People, and who deals with difeqs? Tomorrow I have to pass difeqs, and I have not yet prepared cribs."

If the teacher sees in this course a formal scholastic component only, he/she quickly turns difeqs into "torture instruments" for poor students. Analytical solution of even most simple differential equations requires knowledge of more than a dozen specific techniques and good integration skills. If the teacher has deep understanding of the semantic essence of differential equations, feels the balance between analytical and numerical solution methods, he/she can easily involve students into this most "physical" branch of mathematics.

Difeqs are time-tested methods and tools of research of surrounding real physical processes. For example, body cooling or heating (see above), current passage through electrical circuit or radioactive decay, body falling down (of our drop of water) under the action of gravity, pendulum oscillation, etc.

Let's talk about the recent past of students, postgraduate students, and engineers—about school years. After all, it was the first time when familiarization with the elements of differential equations theory begins. But this fact is persistently hidden from students. Teachers of physics believe that it is more useful for a students to learn the calculation formula than to understand where this formula is derived from. Teachers of mathematics believe that it is enough to make a student learn how to define a derivative using formal rules and tables than justify the need for study of this mathematical concept.

Earlier, there were excuses—inaccessibility of computer tools intended for solution of difeqs. Now, such excuses seem to be unsound. Almost all physical and mathematical phenomena studied at school or university can be easily simulated using differential equations solved by means of computer software with good graphics and animation.

Once one of the authors of this book gave classes on informatics in a very progressive Moscow Lyceum (high shcool). In terms of lessons, solution methods were reviewed in the environment of mathematical software Mathcad for differential equations. The director of the Lyceum (incidentally, a physicist by education, and part-time—Professor of the Physics Platform at the prestigious Moscow university) said that the words "differential equations" namely "difeqs" are imposed a strict secret ban at high school [68]. The students should at least understand simple algebraic equations, but here they have to deal with the differential ones… But when this director was shown what equations will be addressed at the lessons and how they will be solved on the computer, the director changed his mind, and expressed confidence that it will be interesting, understandable, and, what is most important, useful for the students.

The Mathcad solves differential equations numerically, but not analytically. This means that we obtain a set of numbers—table of arguments and values of the desired function, rather than the formula-solution. The desired function can be represented graphically and even animated. By changing the initial conditions, it is possible to follow how the graphics will change. For example, solve the same problem, but with a different initial condition. Moreover, such initial conditions can be empirically selected which provide the desired behavior of the solution.

The main thing you need to understand that notorious difeqs are necessary and useful simulation tool for a lot of physical and technical processes, rather than a school or university "horror story" and "torture instrument". That's what we tried to show in this chapter.

Chapter 19
Refrigeration Cycles

Volodymyr Voloshchuk

Abstract In the chapter a description is given for thermodynamic cycles calculations approaches used in refrigeration, air-conditioning facilities, heat pumps. Given the characteristics of cloud-based functions that support these calculations.

According to one statement of the Second Law of Thermodynamics if heat is to be transferred from low to high temperature external work is required. To transfer heat from a low-temperature reservoir to a high-temperature reservoir takes a device known as a refrigerator or a heat pump.

Such heat engines use working fluids with low boiling temperatures near atmospheric pressure—refrigerants: carbon dioxide, ammonia, freons, etc.

A typical example of a refrigeration device is the vapor-compression cycle shown in Fig. 19.1. There are four main components to the cycle: the evaporator, refrigerant compressor, condenser, and the expansion (or 'metering') valve. The liquid refrigerant evaporates in the evaporator at a temperature that is less than that of the surroundings where heat $\mathbf{q_L}$ is removed naturally from the surroundings. Then the refrigerant enters the compressor as a low pressure vapor and is compressed to a high pressure by the work of the compressor $\mathbf{l_C}$. From the compressor the refrigerant flows to the condenser as a superheated vapor at high pressure and temperature. The condenser is a heat exchanger where heat $\mathbf{q_H}$ is rejected from the refrigerant so that it is converted to a high pressure liquid. The condensation

The site of the chapter: https://www.ptcusercommunity.com/message/423040.

V. Voloshchuk (✉)
National University of Water Management and Natural Resources, Rivne, Ukraine
e-mail: Vl.volodya@gmail.com

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_19

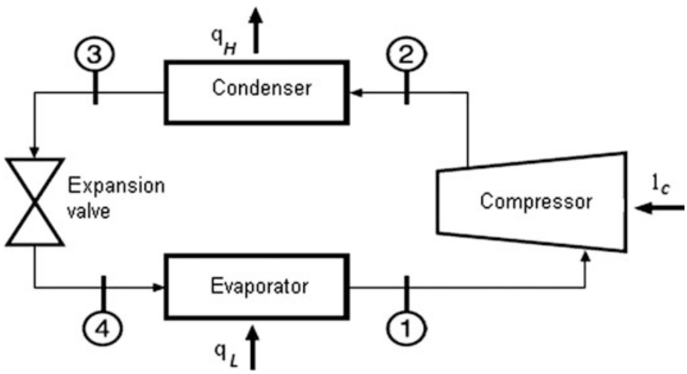


Fig. 19.1 Schematic of vapor-compression refrigeration cycle

temperature is higher than that of the high temperature surroundings, and heat is rejected naturally from the condensing refrigerant to the surroundings. The refrigerant liquid then passes to an expansion valve where the flow is throttled from high to low pressure. The cycle is closed.

The efficiency with which a refrigerator or heat pump utilizes electricity to produce either cooling or heating is known as the coefficient of performance. The cooling coefficient of performance, COP_R, is the cooling effect at the evaporator divided by the work done at the compressor. The heating coefficient of performance, COP_{HP}, is the heating effect at the condenser divided by the work done at the compressor [69–71]:

$$\mathrm{COP}_R = \frac{q_L}{l_C}.$$

$$\mathrm{COP}_{HP} = \frac{q_H}{l_C}.$$

Calculations of refrigerators are impossible without values of thermodynamic properties of working fluids (refrigerants) which are used in the cycles. In the days of manual calculations, it was convenient to use "paper" property tables or diagrams for thermodynamic values of refrigerants in saturation and single regions. Computer calculations require special software enabling to determine properties of working substances depending on parameters of refrigeration cycle—temperature, pressure, density, etc.

One of the most widely used and convenient computer programs for the properties of working fluids and heat carriers/coolants used in power engineering (including industry scale installations and small installations using organic working media) is **WaterSteamPro** [23]. After downloading this program from the website www.wsp.ru and installing it on a computer,[1] more than 300 functions not only on the properties of working fluids, but also on some processes in thermodynamic cycles become available for thermal engineering calculations.

Downloading functions from the Internet (or installing them on a computer from a disk or other carrier if the workstation does not have access to the Internet) has one essential drawback.

Computer programs, in particular, those for calculating the thermophysical properties of individual substances and their mixtures are constantly upgraded and improved. This is mainly due to the fact that new formulations (sets of formulas with their description)[2] emerge, that determine the procedures used to calculate the concrete properties of certain substances.[3] In addition, errors and inaccuracies in the existing computer programs are corrected, their application domains are extended, and their performance is improved (speed of operation, volume of computer memory, etc.). Such programs are also continuously updated due to changes made in the hardware and operating system software. Users of computer programs on the properties of substances frequently fail to notice these changes and work with outdated versions. Also, users face additional difficulties if they change their computer and/or operating system: old computer programs cannot be installed and started any longer.

In view of what was said above, and taking into account that at present almost all computers on which engineering calculations are carried out (in particular, thermal engineering ones) have constant highspeed connection to the Internet, we have suggested a new technology for working with functions on the thermophysical properties of working fluids, heat carriers/coolants, and materials used in thermal power engineering that is based not on downloading computer programs but on making reference to the functions stored on websites, or in "clouds".

Figure 19.2 shows a part of the "cloud" calculation server of the Moscow Power Engineering University's (MPEI) National Research University (www.mpei.ru) and Ltd Trieru (www.trie.ru) containing a list of links to a wide range of web-pages with functions for calculating thermophysical properties of working substances (including refrigerants). The links are located in the interactive handbook "Power &

[1]Almost all software can be used for carrying out these calculations, such as the Excel spreadsheet processor, Mathcad engineering calculator, Maple symbolic computation language, high level programming languages C, BASIC, Pascal, and FORTRAN. In this paper, we limited ourselves to considering only the Mathcad.

[2]As regards water and steam (including solutions, e.g. seawater), which are used as the main working fluids in power engineering, such formulations are developed and approved by the International Association for the Properties of Water and Steam, IAPWS (see the website www.iapws.org).

[3]Here is a concrete example. In September 2011, the IAPWS approved a new formulation for calculating the thermal conductivity of water and/or steam. Users who downloaded the computer programs earlier will continue to work with their old versions for a long period of time, whereas those who employ the new information technology described in this paper will immediately begin to work with the software implementation of the new formulation on the thermal conductivity of water and/or steam.

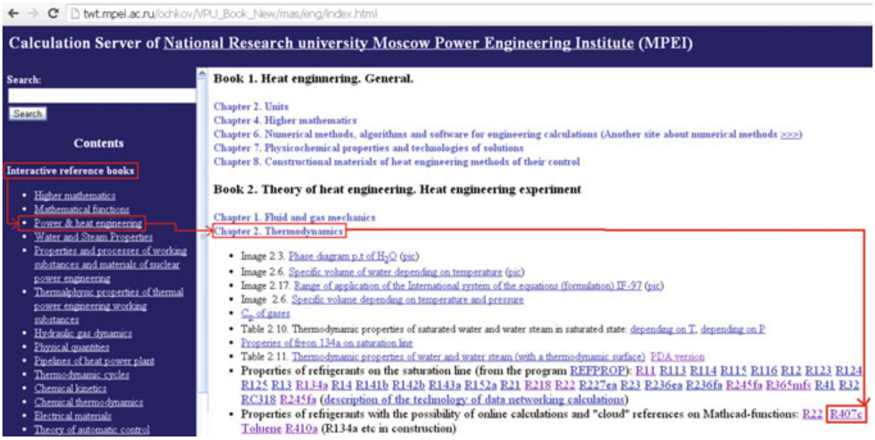


Fig. 19.2 The web-page on properties of refrigerants

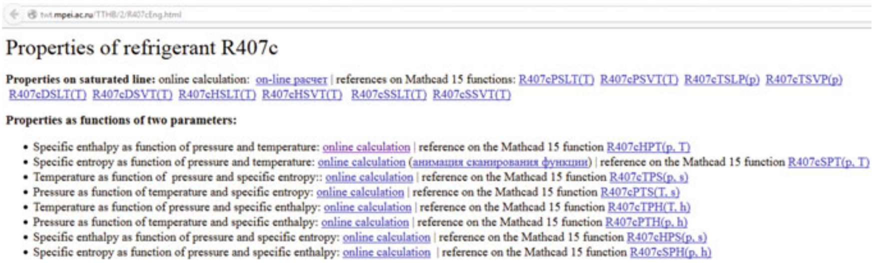


Fig. 19.3 The web-page on "cloud" functions of thermophysical properties of refrigerant R407c with options of online computing, download and reference

heat engineering", "Book 2. Theory of heat engineering. Heat engineering experiment", "Chap. 2. Thermodynamics", which can be found in the table of contents "Interactive reference books".

Figure 19.3 shows the web-page http://twt.mpei.ac.ru/TTHB/2/R407cEng.html, which is chosen from the list presented on the previous figure—"Properties of refrigerants with the possibility of online calculations and "cloud" references on Mathcad-functions". This page contains "cloud" functions for on-line computing, references and downloading Mathcad-functions on thermophysical properties of refrigerant R407c. Among the "cloud" functions there are also some for computing properties at the saturation state depending on one argument. Besides the web-page provides functions of thermophysical properties, which depend on two parameters as initial data: pressure and temperature, pressure and specific enthalpy, pressure and specific entropy, temperature and specific entropy.

If a user of the web-page needs to determine, for example, specific enthalpy of refrigerant R407c as a function of pressure and temperature then he has to navigate

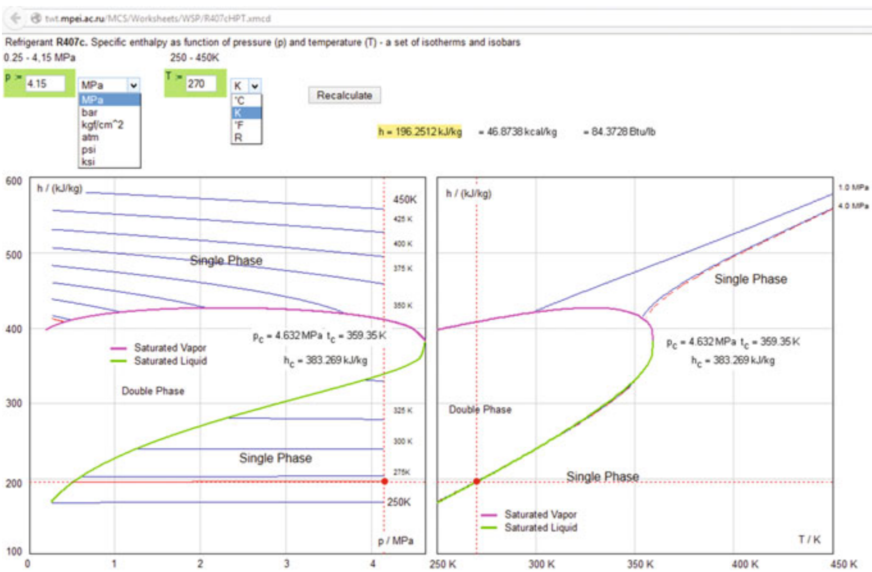


Fig. 19.4 A copy of the internet-page for online calculation of specific enthalpy as a function of pressure and temperature. Illustration of the determined point in *h*, *p*- and *h*, *T*-phase diagrams

with the link on-line calculation (see Fig. 19.3) to the calculation page shown on the Fig. 19.4.

The user is able to change initial data (in this case pressure and temperature) and to choose required dimensions. After clicking the button **Recalculate** he obtains the results and graphical illustrations of the calculated point in *h*, *p*- and *h*, *T*-phase diagrams.

The mathematical software Mathcad (the web-page, shown in Fig. 19.4, was created with it—see Chap. 11) enables us to solve a wide range of scientific, engineering and technical problems without using traditional programming. Formulas are written in the Mathcad in the language of mathematics, which is naturally readable and does not demand deep knowledge of programming language. Mathcad gives us the possibility to see results clarified by plots and diagrams—a feature that favorably distinguishes it from traditional programming languages and spreadsheets. The computer application Mathcad also gives the possibility to use units for checking the correctness of calculations and for presenting results in a more convenient way, especially in engineering calculations. Mathcad has an option to exchange computational applications with other computer programs (such as MathSoft's Axum and S-PLUS, Microsoft Excel, MATLAB). All that makes Mathcad calculations more efficient.

The Mathcad has a convenient tool: references to another Mathcad document the variables and functions of which become accessible or, as programmers say, visible in the Mathcad document from which the corresponding reference is made. The Mathcad user need not open another calculation document and insert it in his or her calculation document: it is sufficient to make a reference to the appropriate file. After that, the user can use the functions programmed in the external document as if they have already been created in his or her document. Such reference can be made not only to Mathcad documents (files with the extensions *.mcd, *.mcdz, *.xmcd, *.xmcdz, *.mcdx, and *.mcdxz) stored on a workstation or in a local area network (LAN), but also to documents available on websites. Owing to this circumstance, wide possibilities are opened for implementing a new technology of engineering calculations.

The usage of "cloud" functions on the properties of refrigerant is illustrated in Figs. 19.5, 19.6 and 19.7 by a calculation in the Mathcad of the simplest ideal refrigeration thermodynamic cycle. This example is located on the web-site http://twt.mpei.ac.ru/MCS/Worksheets/PTU/RefrMachine-eng.xmcd for open interactive calculations without installing Mathcad on personal computer (see Chapter "Heat pumps and refrigerators" in interactive reference book "Thermodynamic cycles" on the website www.vpu.ru/mas). If Mathcad software is installed on user's computer it is possible to download the corresponding Mathcad file from the same chapter of the book "Thermodynamic cycles" located on http://twt.mpei.ac.ru/TTHB/2/RefrMachine-eng.xmcdz.

Usually the calculation begins from interactive procedure of data input. For the introduced refrigeration device (its scheme is shown in Fig. 19.5)—they are temperature of the saturated vapor refrigerant in the evaporator and in the condenser, superheating in the evaporator and subcooling in the condenser. If a user enters zero values of superheating/subcooling, calculations of parameter (specific enthalpy and entropy) of refrigerant in a single region do not work and after clicking with the lefthand mouse button on the functions which determine these parameters information window appears with statement "point is in saturation region". This means that calculation with functions on working fluid properties for a single region is impossible.

Pressure losses along the length of each component of the refrigeration device are neglected for simplicity. But it is not difficult to take into account these and other "**non-idealities**". It is also easy to take into consideration that in actual device entropy of the refrigerant after compression increases ($\mathbf{s_2 > s_1}$) and enthalpy of the refrigerant after thermal expansion valve slightly decreases ($\mathbf{h_4 < h_3}$). On a website http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-27-eng.xmcd (the same Chapter "Heat pumps and refrigerators" in interactive reference book "Thermodynamic cycles" on the website www.vpu.ru/mas) an open interactive calculation recourse for determining coefficient of performance of vapor-compression refrigeration cycle is located where this "non-ideality" is taken into account.

In order to calculate pressure of the refrigerant in evaporator and condenser it is required to have an expression of pressure as a function of temperature on the saturation line. As a rule in this case, engineers refer to tables, where this expression is "printed" with discrete values of pressure and temperature, and, if necessary, interpolate data and enter the received result (pressure) into calculation. In another case engineers do calculation with the help of some formula which expresses dependence of temperature on pressure on the saturation line. In this situation

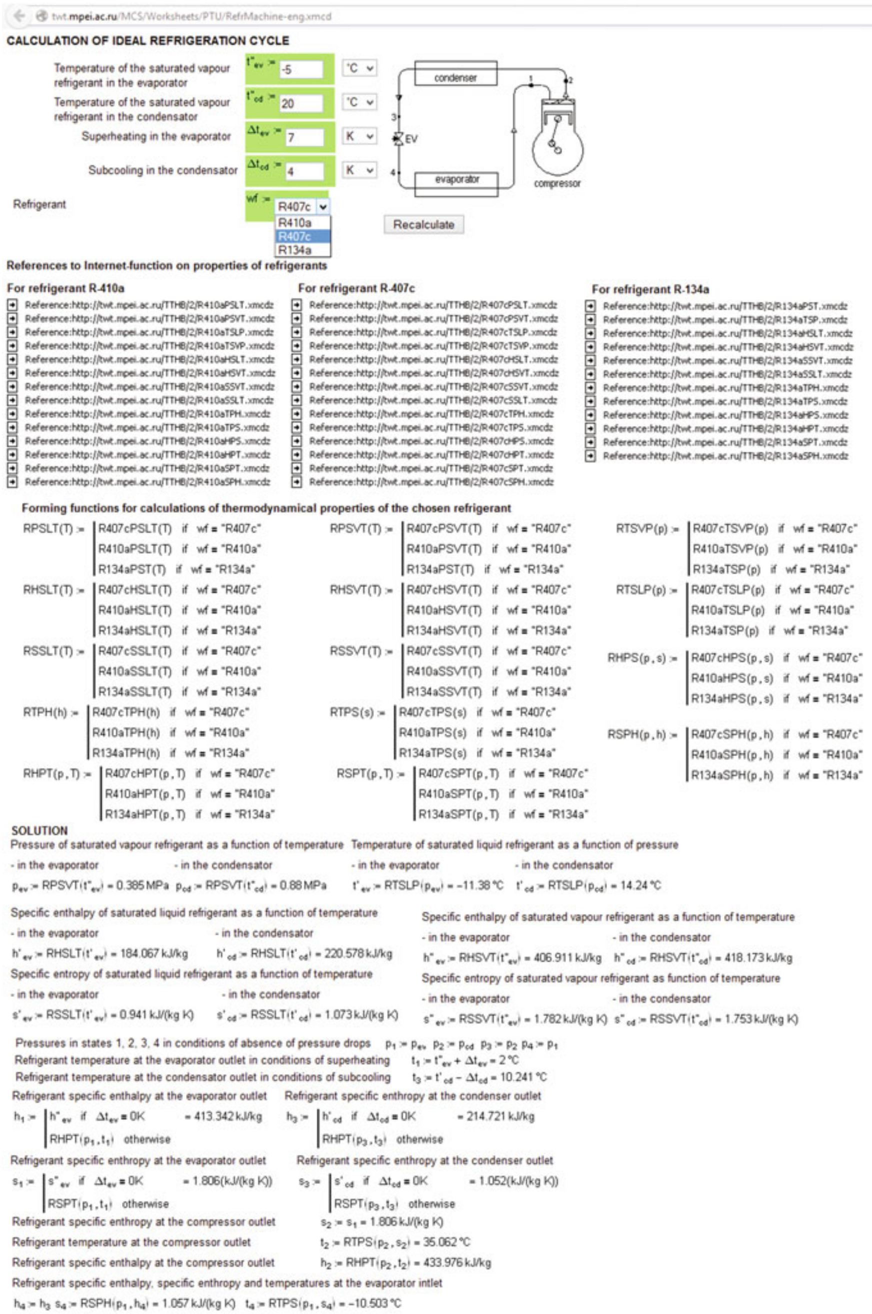


Fig. 19.5 Schematic and calculation of ideal vapor-compression refrigeration cycle

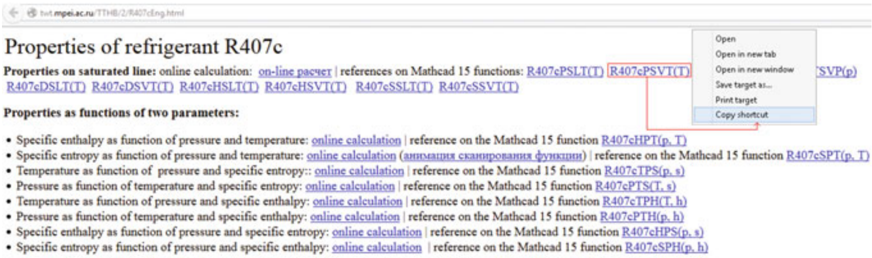


Fig. 19.6 Link to the "cloud" function

engineers or scientists need to spend additional time for manual calculation and input necessary properties into calculation or for search and integration into calculation function on thermophysical properties of refrigerants. The resources of the website www.vpu.ru/mas are proposed to help engineer to solve his working tasks.

If to analyze Fig. 19.3 a link **R407cPSVT(T)** can be observed, clicking on which Mathcad-function titled as **R407cPSVT(T)** can be downloaded. The function **R407cPSVT(T)** determines pressure of vapor R407c on the saturation line as a function of temperature. It is possible not to download and place this function into user's worksheet but just to do Internet-reference to the function.

If to move the pointer of the mouse to the link **R407cPSVT(T)**, see Fig. 19.3, and to click its right button a dialog window will appear (see Fig. 19.6) where the corresponding URL address of the necessary "cloud" function can be seen and copied.

To make this function activated in calculations of refrigerant cycles it is necessary to make a reference to it in the MathCad worksheet. This procedure is shown in Fig. 19.7: in Mathcad choose the menu items **Insert** then **Reference** and insert the previously copied address shown in Fig. 19.6 in the appeared window. After such procedure (see Fig. 19.5) it is possible to calculate the necessary pressures $\mathbf{p_{ev}}$ and $\mathbf{p_{cd}}$ of refrigerant R407c and to make the result visible: $\mathbf{p_{ev}}$: = **R407cPSVT** $(\mathbf{t''_{ev}})$ = … (*comment*: pressure is shown in pascals but a user can fix other units, for example megapascals (MPa) as demonstrated in Fig. 19.5). Pressures $\mathbf{p_{ev}}$ and $\mathbf{p_{cd}}$ are calculated at the specified temperatures on base of function relationships on

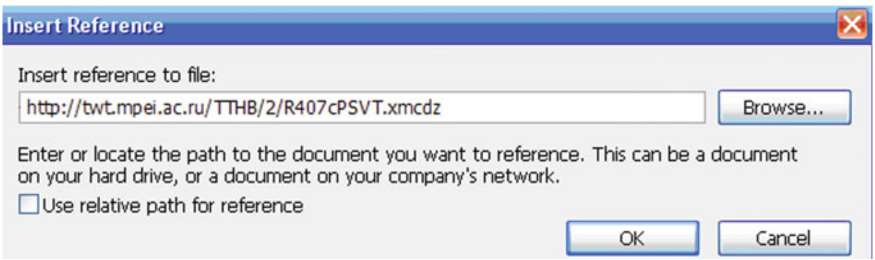


Fig. 19.7 "Cloud" function referenced into Mathcad-worksheet

properties of vapor on the saturation line. Refrigerant R407c—zeotropic mixture—mechanical mixture of refrigerants R32/R125/R134a (mass fractions respectively are 23/25/52 %) with different saturation temperatures at the same pressure. Temperature "glide" for the given working fluid is rather high and is approximately 5–7 K. So, calculation of the refrigerant cycle on the base of this refrigerant should take into account this peculiarity. With the help of functions on properties of liquid refrigerant on the saturation line temperatures $\mathbf{t'_{ev}}$ and $\mathbf{t'_{cd}}$ are determined at pressures $\mathbf{p_{ev}}$ and $\mathbf{p_{cd}}$ calculated earlier. Then for determining parameters of the refrigerant in specified points of a cycle appropriate functions on properties of working fluids in single phases are used.

So, due to internet-references it is possible to activate all necessary functions on thermodynamic properties of refrigerant R407c.

After all necessary functions are activated it is not difficult to do thermodynamic calculation of refrigerant cycle (see Fig. 19.5). Among functions which are shown in Fig. 19.3 there are also backward functions, for example **R407cTPS(p, s)**—a function for computing temperature of the refrigerant R407c depending on pressure and specific entropy.

When all necessary functions on properties of refrigerant are at hand, it is easy to plot *Ts*- and *ph*-diagrams of the refrigeration cycle (see Fig. 19.8). The methodology of plotting such diagrams is presented in [34].

According to the introduced technology it is possible to make all desired functions applied to calculation of a refrigerator. The proposed technology will work on any computer with installed Mathcad and Internet connection.

If necessary, the user can click with the left-hand mouse button on any reference shown in Fig. 19.5, download, and open the chosen Mathcad document storing the desired function for independent working with it. This document can be saved on a workstation (on the own computer) or in the LAN of the user's company and make a reference to it already in the new place of its storage: not in the Internet (in "clouds"), but in a local ("grounded") place. This is advisable if the connection with the Internet is insufficiently reliable or has limitations. But in that case it is best to install all functions on the properties of working substances used in thermal power engineering on the user's computer from a disk received from the developers or simply by visiting the website WaterSteamPro, which was mentioned above.

The technologies of references and downloading described in this book have their advantages and drawbacks. A compromise (intermediate) technology consists in installing the WaterSteamPro software on the user's computer and regularly updating it. If thermal engineering calculations are carried out on computers having reliable connection with the Internet, the technology of references described in this paper can be used. This technology allows users to access a rich set of other functions useful for specialists in thermal power engineering, which are placed on the MPEI computation server.

Functions on thermophysical properties of refrigerants are created on the base of spline-interpolation of tabulated data generated by the program NIST (www.nist.gov) **REFPROP** (http://www.nist.gov/srd/nist23.htm). This technology is developed and explained in the Chap. 11.

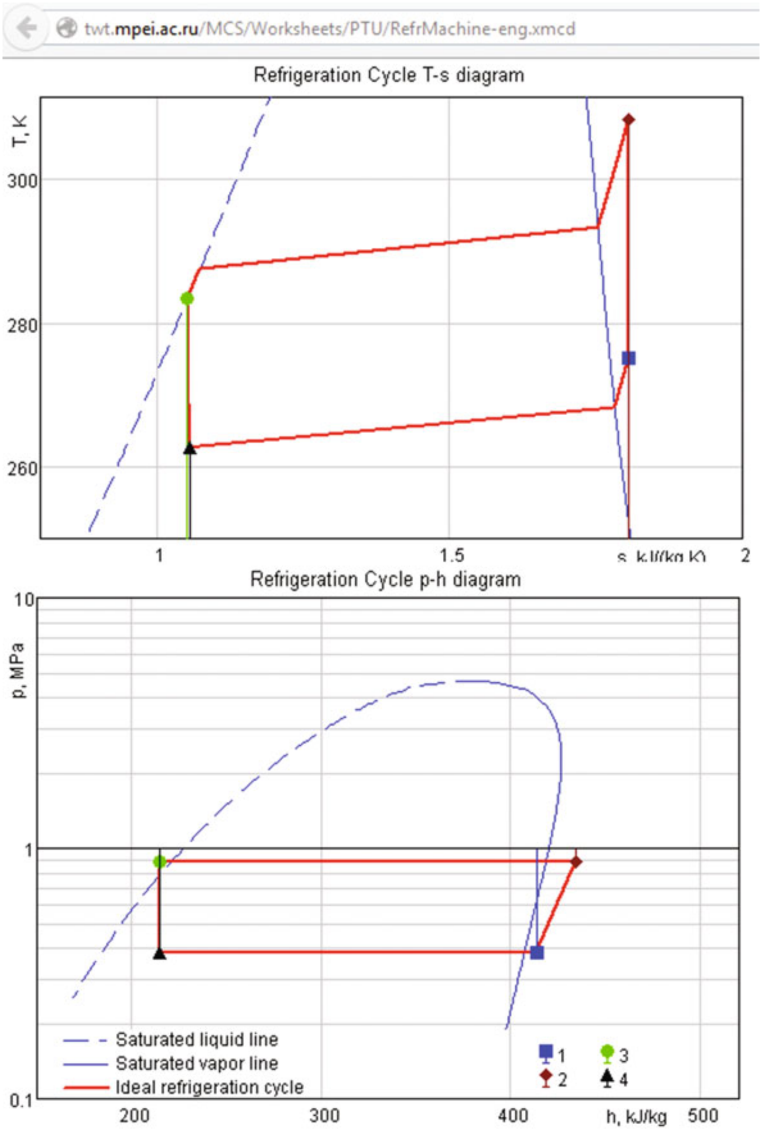


Fig. 19.8 *Ts*- and *ph*-diagrams for a simple vapor-compression refrigeration cycle

An example of input data block for interactive thermodynamic analysis of simple vapor-compression heat pump cycle, which is prepared on the technology of Mathcad Calculation Server is shown in Fig. 19.9. The introduced calculation is located on the server www.vpu.ru/mas at http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-27-eng.xmcd.

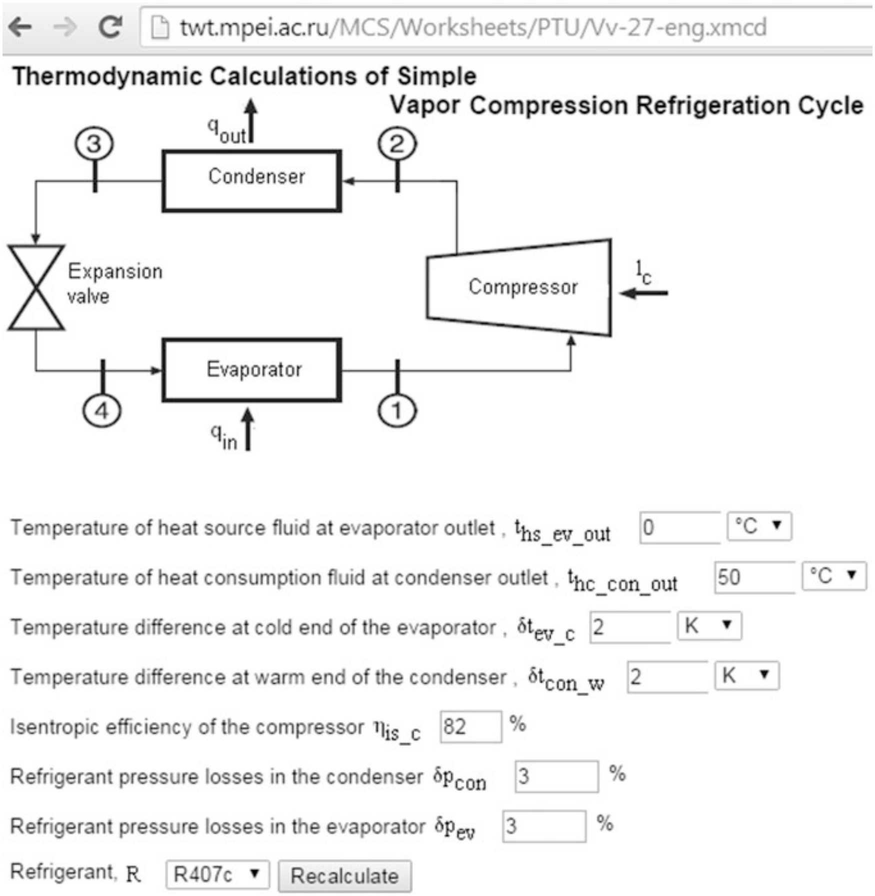


Fig. 19.9 Input data block and schematic of simple vapor-compression refrigeration cycle prepared on technology of Mathcad calculation server

The web-page on this calculation can be found on the server in the interactive reference handbook "Thermodynamic cycles", Chap. "Heat pumps and refrigerators". As input data it is proposed: temperature of heat source fluid at evaporator outlet, temperature of heat consumption fluid at condenser outlet, temperature difference at cold end of the evaporator, temperature difference at warm end of the condenser, isentropic efficiency of the compressor, refrigerant pressure losses in the condenser, refrigerant pressure losses in the evaporator and a type of refrigerant. For determining of thermodynamic properties of refrigerant references on "cloud" functions were used.

Ts-, *ph*- and *hs* *p*roperty diagrams for the presented simple vapor-compression cycle of a heat pump, plotted in Mathcad Calculation Server are shown in Fig. 19.10.

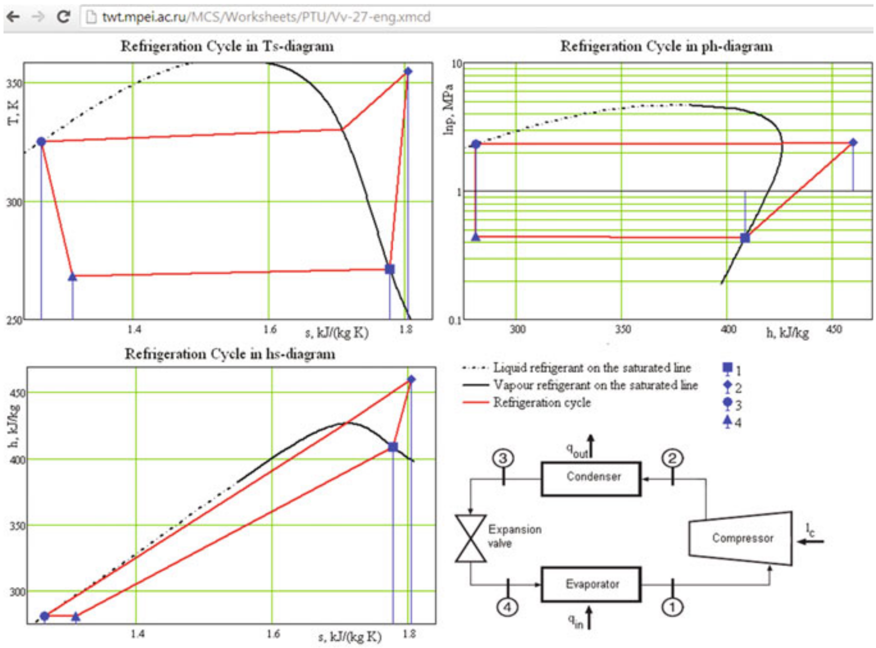


Fig. 19.10 *Ts*-, *ph*- and *hs* property diagrams for a simple vapor-compression refrigeration cycle, plotted in Mathcad calculation server

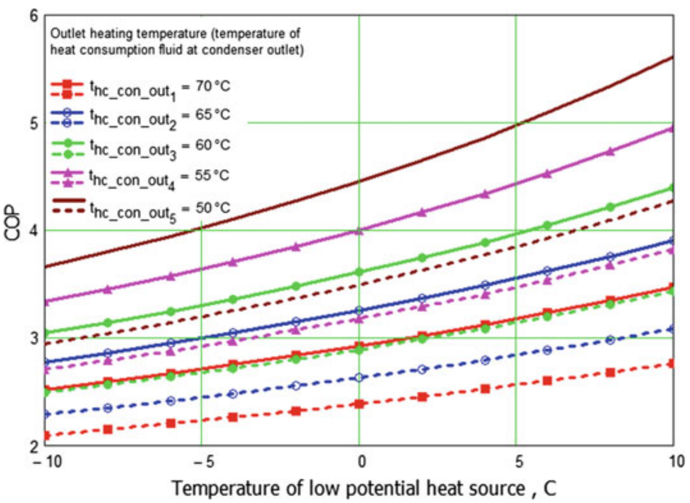


Fig. 19.11 Results of on-line calculations on determining influence of temperatures of the source and heat consumption on **COP** of a heat pump cycle: *dashed lines*—actual cycle; *solid lines*—ideal cycle; working fluid—refrigerant R407c

In the Fig. 19.11 results of on-line calculations made with the help of web-page http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-27-eng.xmcd (see interactive reference handbook "Thermodynamic cycles", Chap. "Heat pumps and refrigerators", "Thermodynamic calculations of Simple Vapor-Compression Refrigeration Cycle") are introduced. The results demonstrate influence of the temperature of low potential heat source and outlet heating temperature on the coefficient of performance of simple vapor-compression ideal and actual cycles of a heat pump. R407c is chosen as a refrigerant.

Such dependences are widely spread in literature. But with the help of the introduced recourses it is possible to investigate a quite wide variety of input conditions (temperature of low potential heat source, outlet heating temperature, compressor efficiency, pressure losses etc.) and, for example, to determine numerically the most irreversibilities taking place in a heat pump cycles and so on.

In actual refrigeration cycles irreversible processes take place which decreases their efficiency. Energy losses for friction due to moving of working fluid, heat losses, pressure drops, temperature differences can be the reasons of these irreversibilities.

In the calculations of the actual cycle contrary to ideal one 3 % pressure drops in evaporator and condenser, minimum temperature differences in evaporator and condenser $\Delta \mathbf{t}$ = 2 °C and 82 % isentropic efficiency of compression is taken into account.

The Fig. 19.11 shows that for the chosen refrigerant, outlet heating temperature **t** = 50 °C and increasing temperature of low potential heat source from −10 to 10 °C coefficient of performance of ideal cycle of a heat pump increases from 3.7 to 5.6. For actual cycle in the same temperature conditions **COP** of a heat pump increases from 2.9 to 4.3 which is by 22…26 % lower in comparison with ideal cycle. Besides **COP** increasing can be observed when temperature of heat consumption fluid is decreased. For example, for temperature of low potential heat source **t** = 0 °C and decreasing temperature of heat consumption fluid from +70 to +50 °C coefficient of performance of actual cycle of a heat pump increases from 2.4 to 3.5. It should be noted that compressor efficiencies are a strong function of pressure ratio. In the introduced calculations efficiency of compression was set as constant for different temperatures of the source and heat consumer and as a consequence for different pressure ratios. In the proposed web-resource it is possible to change efficiency of compression.

Figure 19.12 presents results of on-line calculations made with the help of the same web-page http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-27-eng.xmcd for the purpose of determining influence of working fluid type on **COP** at different temperatures of heat source and outlet heating temperatures.

As it can be seen that for temperature changes of low potential heat source from –10 to 10 °C and heating consumer from 50 to 70 °C replacing refrigerant R407c on R134a results increasing **COP** of a heat pump by 0.3…0.6 units (Fig. 19.12).

There are a variety of ways that the refrigeration cycle can be tailored to suit an application in a better way (not always necessarily resulting in a higher **COP**) than the simple (basic) vapor compression cycle.

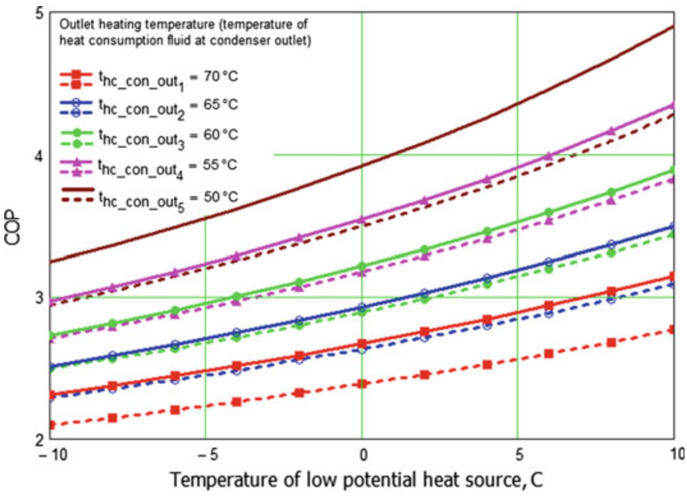


Fig. 19.12 Results of on-line calculations on determining influence of temperatures of the source and heat consumption on **COP** of an actual heat pump cycle *dashed lines*—refrigerant R407c; *solid lines*—refrigerant R134a

Consider the addition of a single heat exchanger to the basic vapor compression cycle, exchanging heat between the fluid leaving the evaporator and the fluid leaving the condenser, shown in the schematic of Fig. 19.13.

Figure 19.14 shows a heat pump cycle in *Ts*-, *ph*- и *hs*- *p*roperty diagrams plotted in Mathcad Calculation Server in accordance with input data introduced in the Fig. 19.13. This web-page is located at http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-28.xmcd (see interactive reference handbook "Thermodynamic cycles", Chap. "Heat pumps and refrigerators", "Thermodynamic calculations of Simple Vapor-Compression Refrigeration Cycle with Regeneration"). From *ph*- *p*roperty diagram it can be observed that due to addition of a regenerator (heat exchanger) the specific enthalpy (and temperature) of the preexpansion state are reduced and conversely the specific enthalpy (and temperature) of the pre-compression state are increased. These differences point out two intended benefits of this cycle modification.

First, since the specific enthalpy remains constant during expansion, a reduction of the specific enthalpy prior to expansion results in a reduction of specific enthalpy prior to evaporation. Therefore the unit will have more evaporative heat transfer to provide more evaporator cooling capacity. Second, the state prior to compression is further away from the saturated vapor line. For most compressors, it is imperative that the state of the refrigerant prior to compression does not have any liquid in the form of droplets or mist, since liquid entrained in a vapor undergoing compression tends to damage the fast moving parts of a compressor, seriously degrading the performance and working life span of the compressor. For this reason, it is usually desirable for the refrigerant to enter the compressor as a superheated vapor, several

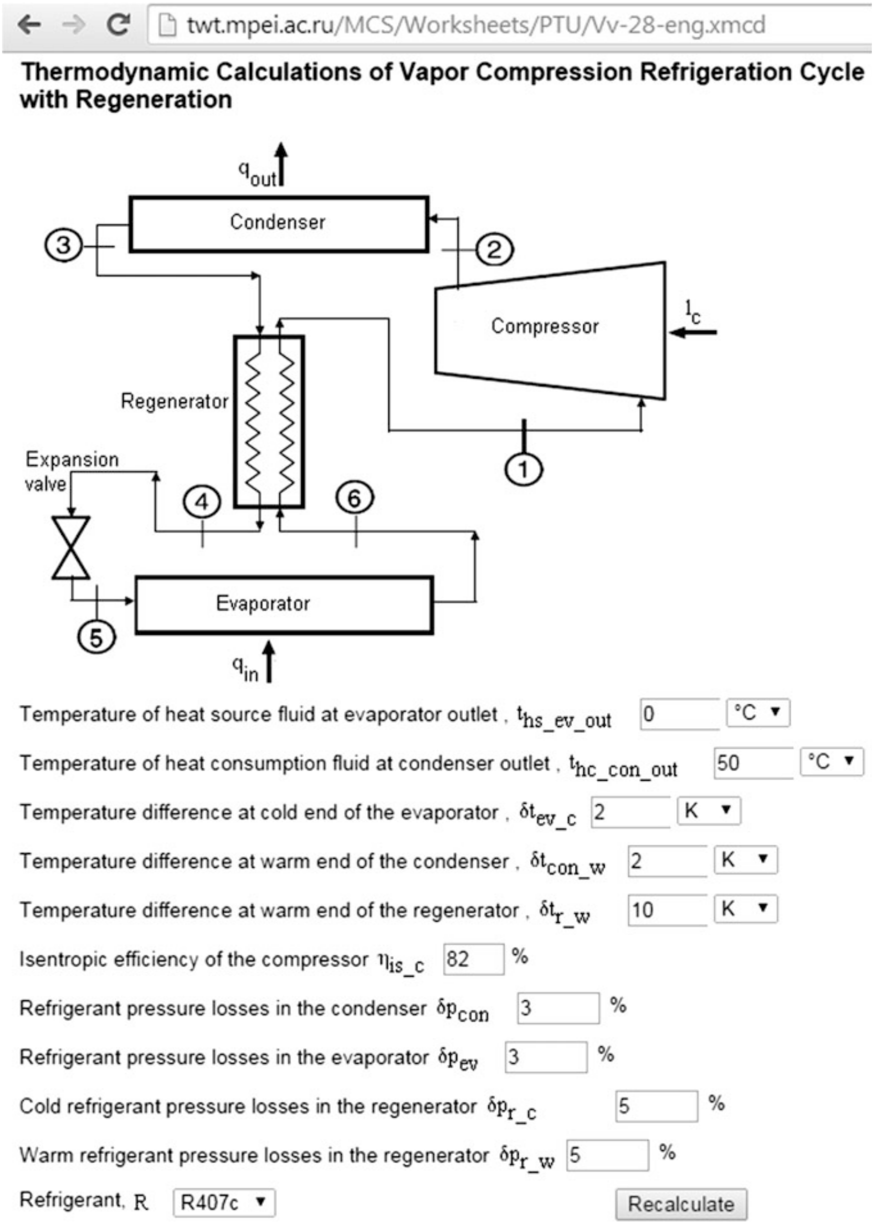


Fig. 19.13 Input data block and schematic of vapor-compression refrigeration cycle with regeneration prepared on technology of Mathcad calculation server

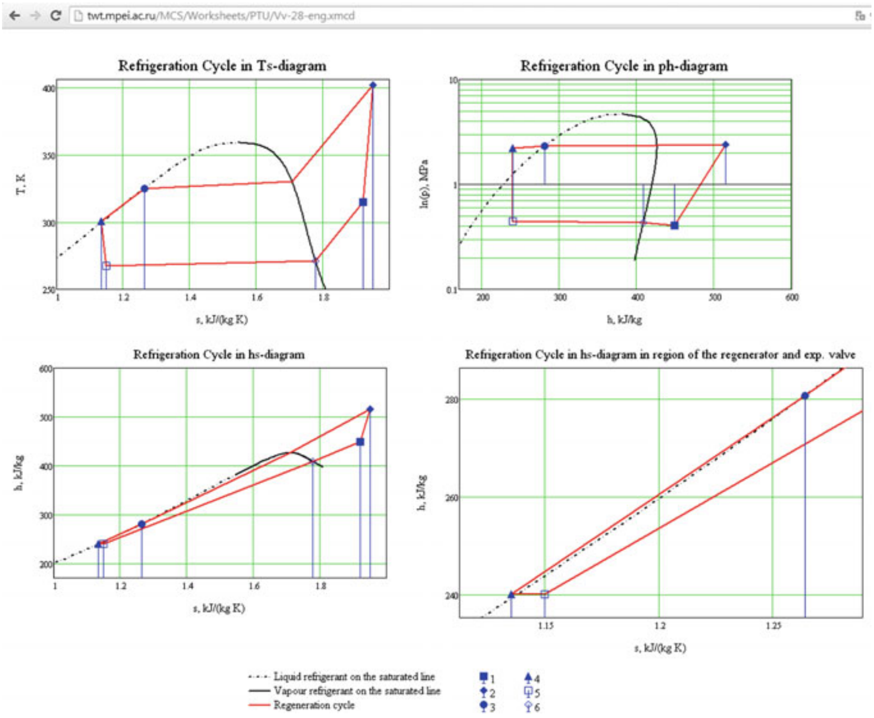


Fig. 19.14 *Ts*-, *ph*- and *hs*- property diagrams for a vapor-compression refrigeration cycle with internal heat exchange, plotted in Mathcad calculation server

degrees above the saturation temperature at the pre-compression pressure. The internal heat exchanger, by increasing the enthalpy and temperature of the pre-compression refrigerant, assists in ensuring that a superheated vapor with no liquid droplets enters the compressor.

In some cases, the internal heat exchanger will also serve to increase the **COP** of the cycle, but this depends on the refrigerant used. For example, according to calculations made in the web-page http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-28.xmcd with input data shown in Fig. 19.13, coefficient of performance of a heat pump using R407c is 3.54. At the same time, according to the calculation results shown in Fig. 19.11 under the same input data, for the simple (basic) heat pump cycle **COP** is 3.49 which is lower by 0.4 units. It should be noted that this increase can be higher due to better compressor efficiency in a cycle with an internal regenerator.

In the event that a high **COP** is of greater importance compared to other factors, it is possible to significantly increase the **COP** of a basic cycle through the use of a multistage vapor compression cycle. This is especially true when the pressure ratio between the heat rejection and heat absorption pressures is large (5 or more). Multistaging involves one or more intermediate pressures between the heat rejection and heat absorption pressures, and a series of compressors operating between successive pressure intervals. The primary benefit of a multistage compression cycle is the reduction in compressor work compared to that of a single stage compressor. Typically, the isentropic efficiency of most compressors used in refrigeration applications tends to decrease with increasing pressure ratio beyond a value of 3 (it should be noted that the isentropic efficiency tends to decrease sharply with decreasing pressure ratio below a value of 2). So, for a high pressure ratio, the average isentropic efficiency for two successive compressors could be significantly higher than that of one compressor, reducing the total amount of work required.

There are several forms of multiple compressor cycles. The first is the cascade cycle, which is simply a chain of single stage vapor compression cycles operating in series, such that the condenser of a lower temperature cycle provides the heat input to the evaporator of the higher temperature cycle. For each pressure circuit the most favorable refrigerant can be adjusted to satisfy reliable and efficient modes of work. In this case evaporation in each stage should take place at pressure slightly higher atmospheric in order to prevent air leak into the system and a critical temperature of a refrigerant should be much higher condensation temperature to decrease irreversibilities associated with compression and expansion.

Another type of multi-compressor vapor compression cycle is the flash chamber/regenerative intercooling multistage vapor compression cycle. There are actually two sub-types—each is described here. Both cycles involve compression of the same refrigerant in two or more stages; only two will be shown here for demonstration purposes. At the intermediate stage there is a flash chamber, into which the refrigerant expands from the condenser, and the saturated liquid and vapor at the intermediate pressure can be separated. The saturated liquid expands from the intermediate pressure to the evaporation pressure. These aspects are common to both subtypes.

One of the subtypes also includes a mixing chamber (the flash chamber) where saturated vapor from the flash chamber mixes with the vapor leaving the low pressure stage compressor. This vapor-mixing chamber acts as a regenerative intercooler since it cools the superheated vapor leaving the low-pressure stage compressor using lower temperature saturated refrigerant, mixing the two prior to the next stage of compression.

An example of input data block for thermodynamic analysis of two-stage vapor-compression heat pump cycle with flash chamber and separate vapor mixing intercooler, which is prepared on the technology of Mathcad Calculation Server is shown in Fig. 19.15. The calculation is located on the server www.vpu.ru/mas at http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-30-eng.xmcd. The web-page on this calculation can be found on the server www.vpu.ru/mas in the interactive reference handbook “Thermodynamic cycles”, Chap. “Heat pumps and refrigerators”, “Thermodynamic calculations of Two-Stage Vapor Compression Refrigeration Cycle with Flash Chamber and Separate Vapor Mixing Intercooler”.

Figure 19.16 shows graphical illustration of two-stage vapor compression cycle with flash chamber and separate vapor mixing intercooler on *Ts*-, *ph*- and

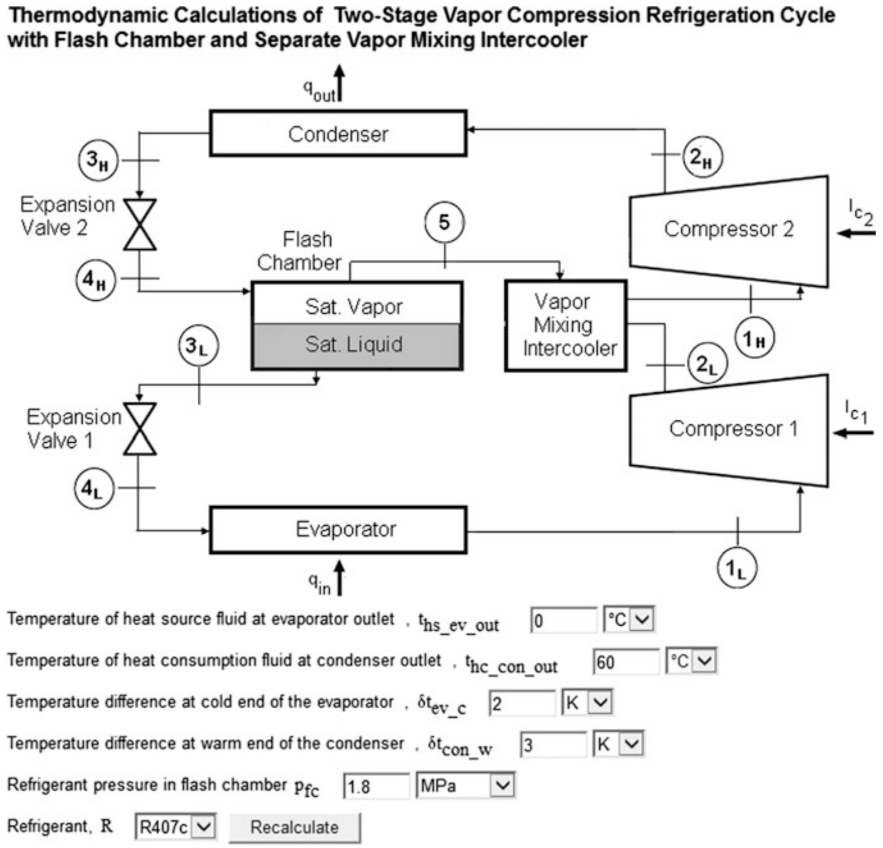


Fig. 19.15 Input data block and schematic of two-stage vapor compression refrigeration cycle with flash chamber and separate vapor mixing intercooler prepared on technology of Mathcad calculation server

hs-diagrams which were plotted with the help of interactive calculation in Mathcad Calculation Server in accordance with initial data presented in Fig. 19.15.

One of the optimization task of thermodynamic analysis of two-stage vapor compression refrigeration cycle is a task of determining intermediate pressure in separate vapor mixing intercooler for which coefficient of performance has the highest value.

For solving this task the same web-page http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-30-eng.xmcd of the server www.vpu.ru/mas can be used. By changing pressure in flash chamber p_{fc} it is possible to calculate coefficient of performance **COP** and to plot functional relationship **COP = f(**$\mathbf{p_{fc}}$**)**.

Figure 19.17 illustrates such graphical relationship plotted in MathCad. From this graph, it can be seen that there is an optimum value of pressure $\mathbf{p_{fc\_opt}}$ for which **COP** is the highest. In the analyzed case according to data presented in Fig. 19.15,

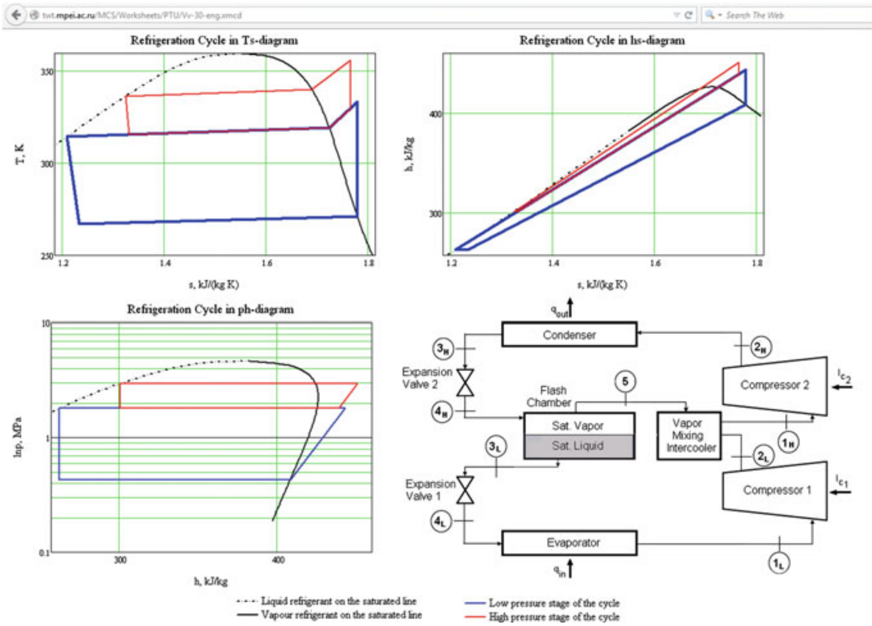


Fig. 19.16 *Ts*-, *ph*- and *hs* property diagrams for two-stage vapor compression refrigeration cycle with flash chamber and separate vapor mixing intercooler, plotted in Mathcad calculation server

this pressure is equal $\mathbf{p_{fc\_opt}}$ = 1.5007 MPa. For determining $\mathbf{p_{fc\_opt}}$ instrument of tracing is used and demonstrated in Fig. 19.17 which is possible in Mathcad.

Optimal value of intermediate pressure in separate vapor mixing intercooler $\mathbf{p_{fc\_opt}}$ can also be determined with the help of other resources of Mathcad. For example, using programming operators it is possible to create functional relationship between coefficient of performance and pressure $\mathbf{p_{fc}}$. In this case a developer of this relationship can use the algorithm which is introduced at http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-30-eng.xmcd. Besides, for calculation of thermodynamic properties of refrigerants in programming block of Mathcad-worksheet it is proposed to do reference on "cloud" functions. This procedure is described above.

A fragment of developing functional relationship **COP = f($\mathbf{p_{fc}}$)** in Mathcad-worksheet with the help of programming block is shown in Fig. 19.18.

Having function **COP = f($\mathbf{p_{fc}}$)** it is possible to determine its extreme using Mathcad operators the choice of which depend on peculiarity of function. In the analyzed case it is proposed to use operator **Maximize(COP, p_{fc})**—determining local maximum. From Fig. 19.18 it can be seen that, on contrary of graphical solution presented in Fig. 19.17, such numerical solution of the proposed task gives more exact result: $\mathbf{p_{fc\_opt}}$ = 1.481 MPa.

In the other flash chamber/regenerative intercooling multistage subtype, there is no separate mixing chamber.

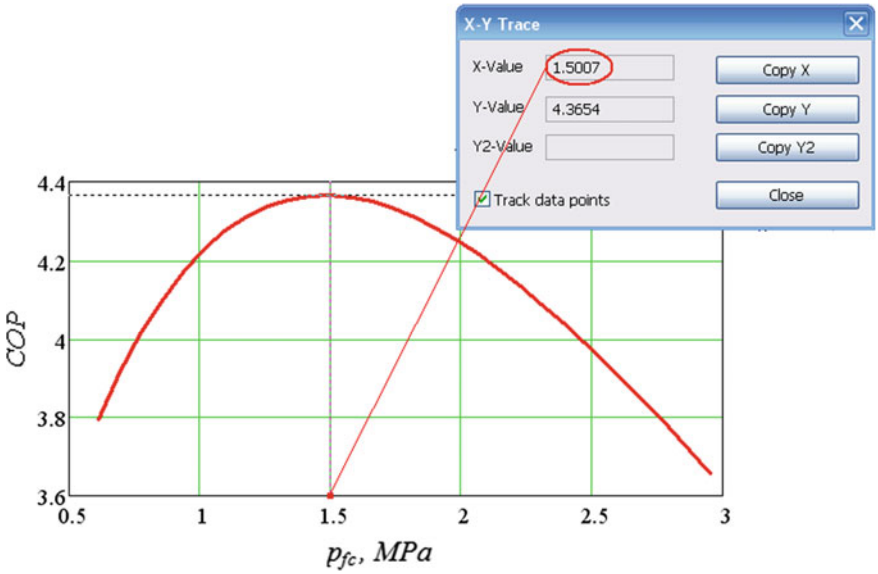


Fig. 19.17 Graphical relationship **COP = f(p$_{fc}$)** in two-stage vapor compression refrigeration cycle with flash chamber and separate vapor mixing intercooler determined with the help of web-page http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-30-eng.xmcd and plotted in Mathcad

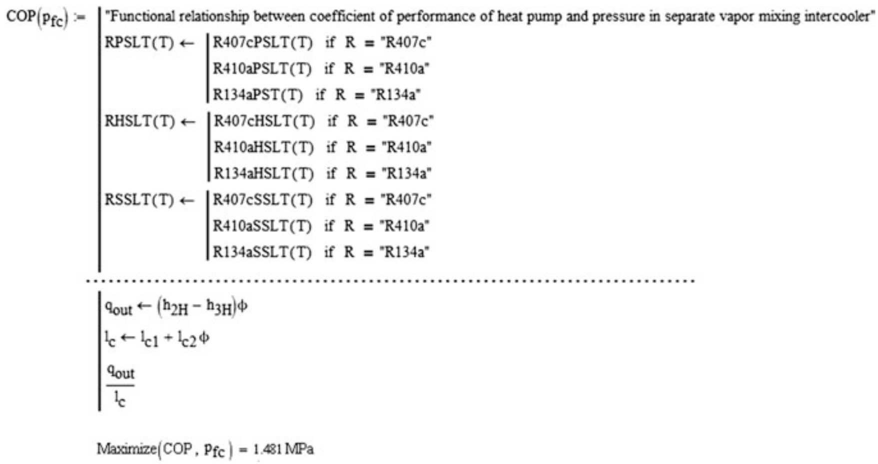


Fig. 19.18 Fragment of creation of functional relationship **COP = f(p$_{fc}$)** in Mathcad-worksheet with the help of programming block and determining optimal value of pressure **p$_{fc_opt}$** with the operator **Maximize**

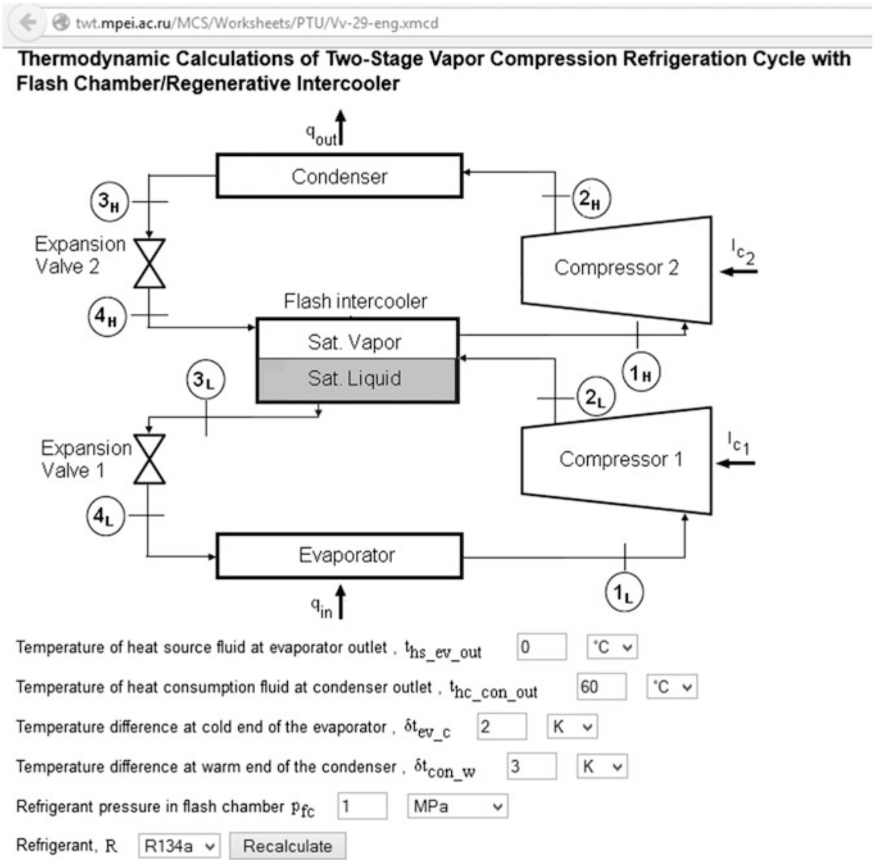


Fig. 19.19 Input data block and schematic of two-stage vapor compression refrigeration cycle with combination flash chamber/regenerative intercooler prepared on technology of Mathcad calculation server

An example of input data block for thermodynamic analysis of ideal two-stage vapor-compression heat pump cycle with combination flash chamber/regenerative intercooler, which is prepared on the technology of Mathcad Calculation Server is shown in Fig. 19.19. The calculation is located on the server www.vpu.ru/mas at http://twt.mpei.ac.ru/MCS/Worksheets/PTU/Vv-29-eng.xmcd. The web-page on this calculation can be found on the server www.vpu.ru/mas in the same interactive reference handbook "Thermodynamic cycles", Chap. "Heat pumps and refrigerators", "Thermodynamic calculations of Two-Stage Vapor Compression Refrigeration Cycle with Flash Chamber/Regenerative Intercooler".

Figure 19.20 proposes graphical illustration of two-stage vapor compression cycle with combination flash chamber/regenerative intercooler on *Ts*-, *ph*- and *hs*-diagrams which were plotted with the help of interactive calculation in Mathcad Calculation Server in accordance with initial data presented in Fig. 19.19.

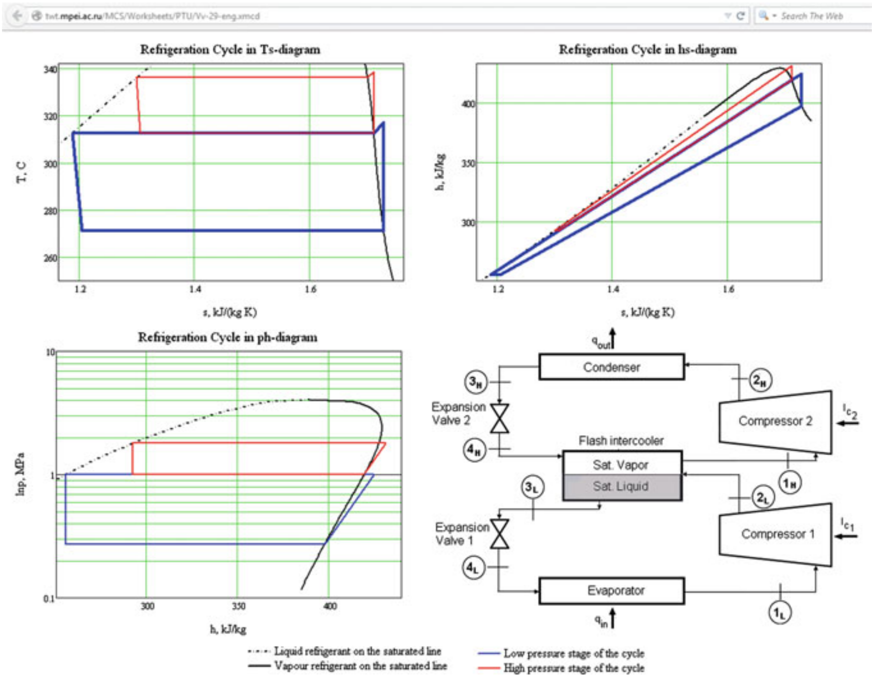


Fig. 19.20 *Ts*-, *ph*- and *hs* property diagrams for two-stage vapor compression refrigeration cycle with combination flash chamber/regenerative intercooler, plotted in Mathcad calculation server

If to compare diagrams in Figs. 19.16 and 19.20 it can be seen that in cycle with combination flash chamber/regenerative intercooler the state of vapor at compressor inlet is saturated, but in cycle with flash chamber and separate vapor mixing intercooler the state of vapor at compressor inlet is superheated. As a result, in the first cycle (Fig. 19.16) performance conditions are better and effectiveness of compression is higher.

Using resources of the server www.vpu.ru/mas located in the interactive reference handbook "Thermodynamic cycles", Chap. "Heat pumps and refrigerators" interactive calculations have been performed for determining coefficients of performance of the four introduced above ideal cycles of heat pumps. The initial data were the following: temperature of heat source at evaporator outlet **t** = 2 °C, temperature of heat consumption fluid at condenser outlet **t** = 55 °C, temperature difference at cold end of the evaporator **δ** = 2 °C, temperature difference at warm end of the condenser **δ** = 2 °C, type of the refrigerant—R407c. The results of calculations are presented in Fig. 19.21. It can be seen that for the same input conditions addition of a heat exchanger to the basic vapor compression cycle, exchanging heat between the fluid leaving the evaporator and the fluid leaving the

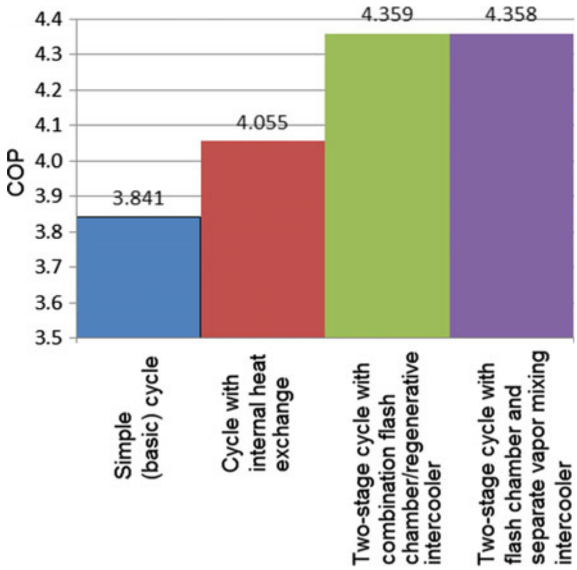


Fig. 19.21 Results of interactive calculations for determining influence of heat pump cycle type on the coefficient of performance which were performed in Mathcad calculation server (type of the refrigerant—R407c)

condenser, results increasing **COP** from 3.841 to 4.055. Additional changes to the basic heat pump cycle, e.g. two-stage compression, result increase of **COP** to 4.359 for cycle with combination flash chamber/regenerative intercooler and to 4.358 for cycle with flash chamber and separate vapor mixing intercooler.

Chapter 20
Three-Layer Thermal Engineering Cake or a Conclusion

Valery Ochkov

Abstract In the final chapter of the book described some methods for solving problems of stationary thermal conductivity. Arguments about the relationship of mathematics and thermal engineering, on the content of lectures on mathematics for students are provided.

Figure 20.1—This is not merely a picture, not one of those drawings that *just* illustrate the studies. The author has this figure printed in color, framed and hung on the wall in his office among other "works of art" created with Mathcad. The author brings his guests-mechanical (or thermal) engineers to the "picture" and asks them to solve the heat exchange problem hidden in it.

Figure 20.1 is, as follows from its title, a Mathcad-document with the solution of the problem of stationary thermal conductivity of a flat three-layer infinite wall.

Input data for calculation: the ambient temperature on both sides of the wall ($\mathbf{T}_{b1}$ and $\mathbf{T}_{b2}$), the heat transfer coefficients between the environment and the wall at both sides ($\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$), the wall area **F**, the thickness of the layers of our heat engineering cake $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$ and $\boldsymbol{\delta}_3$, and also thermal conductivity values of the material of the layers $\boldsymbol{\lambda}$. It is necessary to determine the temperature at the edges of the three-layer wall ($\mathbf{T}_1$ and $\mathbf{T}_4$) and at the boundaries of the layers ($\mathbf{T}_2$ and $\mathbf{T}_3$), and heat flow **Q** through the "three-layer cake". In this cake, sorry, calculation, there are three "flavors".

The first "flavor"—this is how Mathcad can delicately solve the specific heat flux balance equations **Q** in the form of one, so to speak, "five-layer" equation, where the unknowns $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$ and $\mathbf{T}_4$ are positioned in places corresponding to the limits of the definite integrals. And we obtained the integrals because in our

The site of the chapter: https://www.ptcusercommunity.com/message/423051.

V. Ochkov (✉)
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

V. Ochkov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_20

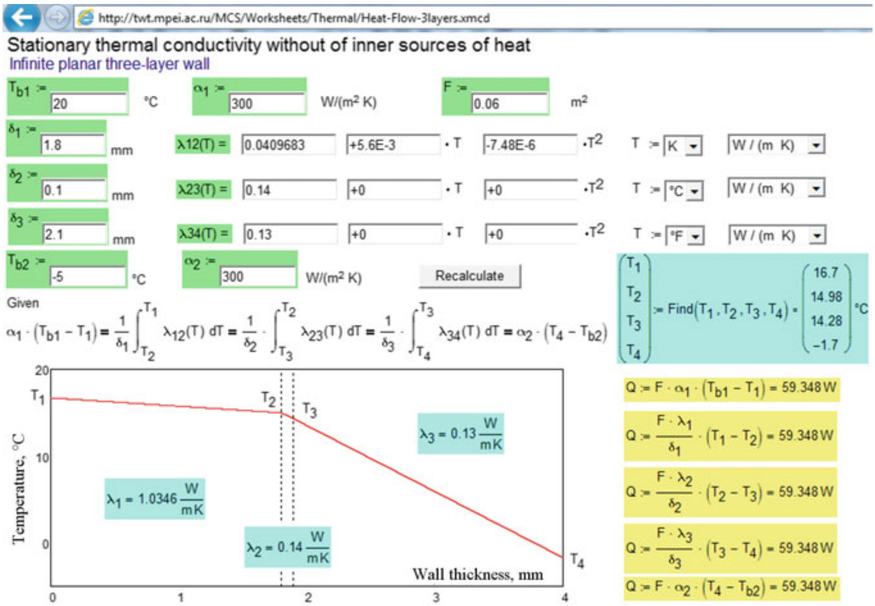


Fig. 20.1 Thermal conductivity of a three-layer wall

calculation thermal conductivity of the wall's material may be not only a constant but a function of temperature. In the references the temperature dependence of thermal conductivity of the construction, thermal insulation and other metals is usually presented in the form of a polynomial of first or second degree. The argument of this polynomial (a straight line or a parabola) may be any temperature in Celsius or Kelvin (absolute temperature). In American references one can even come across Fahrenheit or Rankine (Absolute) units. Temperature, but not as a relative scale, but as a temperature unit, is also present in a thermal conductivity unit. This all is very confusing for inexperienced mechanical and thermal engineers and often leads to mistakes in their calculations.

Figure 20.2 shows an Internet site where the operations related to defining thermal conductivity of insulating materials are automated.

A visitor to the web-site (see Fig. 20.2) selects from the drop-down list the necessary material, specifies the temperature in Celsius, clicks **Recalculate** and receives the answer: the thermal conductivity value of the selected material at a given temperature, its density, the maximum temperature at which the material can be used, and the formula used to calculate the thermal conductivity as function of the temperature. In particular, thermal conductivity of the mineral wool grade 75 (see Fig. 20.2) is described by a linear function, wherein the temperature should be Celsius. For other materials the formula may be in the form of a second order polynomial (a parabola).

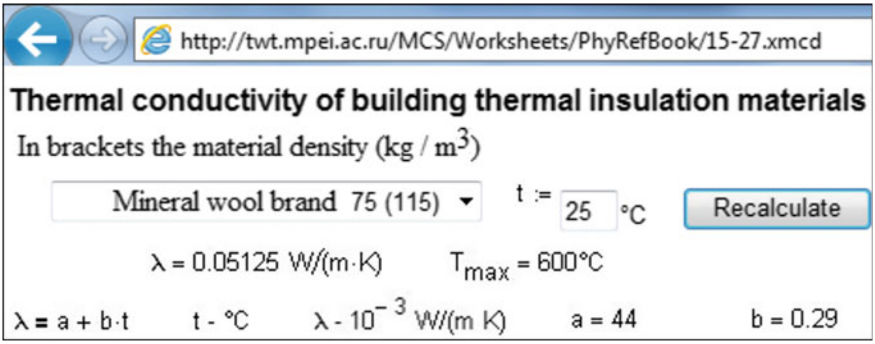


Fig. 20.2 Heat conductivity of heat insulation materials

If a visitor to the web-site in question wants to insert in a Mathcad-worksheet a thermal-conductivity-from-temperature-dependence formula, he will have to act so, as shown in Fig. 20.3: to deprive the temperature units leaving only the numerical values of the temperature in Celsius (t/K − 273.15), and multiply the result by a thermal conductivity unit.

The formula shown in Fig. 20.2 is a classical empirical formula linked not only to the physical quantities (temperature and thermal conductivity), but also to specific conductivity units (milliwatts divided by the meter and Kelvin) and temperature (Celsius). Working with empirical formulas is described in Chap. 2.

By the way, about thermal conductivity units. In the old reference books for the thermal conductivity unit (formerly known as—thermal conductivity coefficient) one can see not only degree Kelvin (or simply kelvins) but also degree Celsius which is equal to kelvin. Degrees Celsius in Mathcad is indicated with somewhat strange unit **Δ°C** (see Fig. 20.3). And one more point. The thermal conductivity unit is usually cut: do not write **W · m/(m^2 · K)**, but **W/(m · K)**, making it difficult to understand the physical nature of this parameter.[1] Printout of the variable value **λ** with **Δ°C** shown in Fig. 20.3 demonstrates that if we take a plate of our material of 1 m^2 in area and 1 cm thick and let a 5.125 W heat pass through that, the temperature difference at the edges of the wall will be equal to one degree Celsius: from one side, for instance, 18 °C, and from the other side—19 °C.

European reference books are gradually avoiding degrees Celsius in thermal conductivity units and replacing them with kelvins, at the same time taking into account the temperature scales of 1968 and 1990 (see Chap. 1). But in the USA, as noted in Chap. 1, one can still see Fahrenheit for thermal conductivity unit, although the specific entropy unit uses degrees Rankine. Engineers in the US subconsciously feel that thermal conductivity is related with a temperature gradient which is usually measured on the Celsius scale in Europe or on the Fahrenheit scale

[1] “Stupid” Mathcad when displaying data on the screen simplifies and adds to the value of specific energy or enthalpy the unit m^2/s^2 instead of the usual kJ/kg. True: simplicity is worse than robbery.

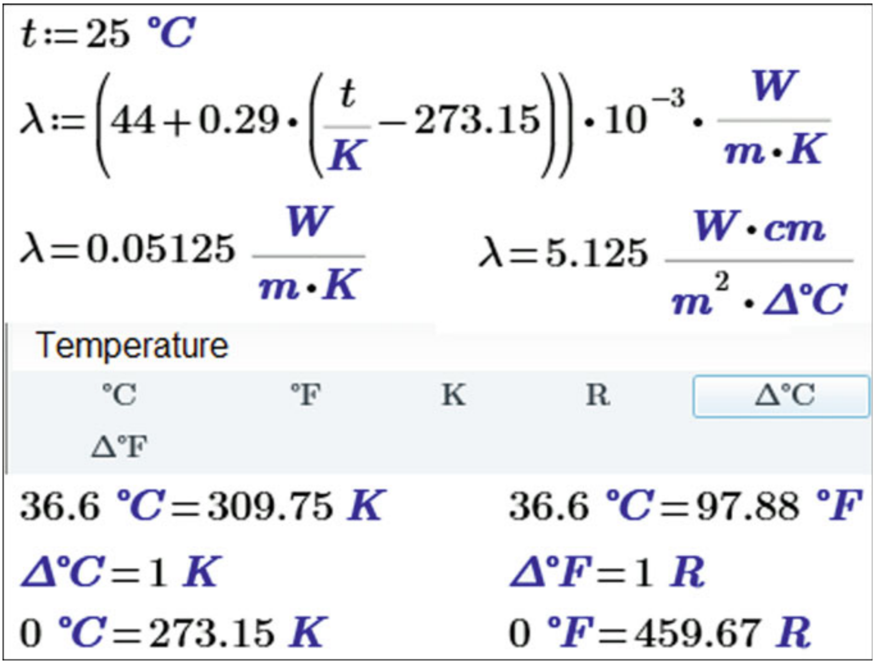


Fig. 20.3 A calculation of thermal conductivity dependence on temperature in Mathcad Prime

in the USA, and the concept of entropy relates with the absolute zero of temperature, which is associated with the kelvin temperature unit in Europe or with degrees Rankine in America. There is no way to break this duality tradition and no reason indeed! Everything will be clear to a competent engineer, and the illiterate may get entangled in any situation.

At the end of Fig. 20.3 one can see the operators translating values of temperatures to different scales and in different units. And whereas in the heads of our mechanical engineers "sits" a constant 273.15, for their American counterparts, it is 459.67.

Surely it is more correct to give empirical dependence of thermal conductivity and other thermophysical parameters on the temperature in the Kelvin, rather than in Celsius, as shown in Fig. 20.2. Figure 20.4 shows a reference web-site with data on the thermal conductivity of pure metals, where the visitor to the site can get not only the desired thermal conductivity value, but also generate a polynomial of n-th degree, describing the temperature dependence using kelvins, and not Celsius, as in Fig. 20.2.

The author's calculations server has reference pages for thermal conductivity of various substances and materials, where one can change the temperature differential range and obtain a simple (linear) dependence for the selected range—see, for

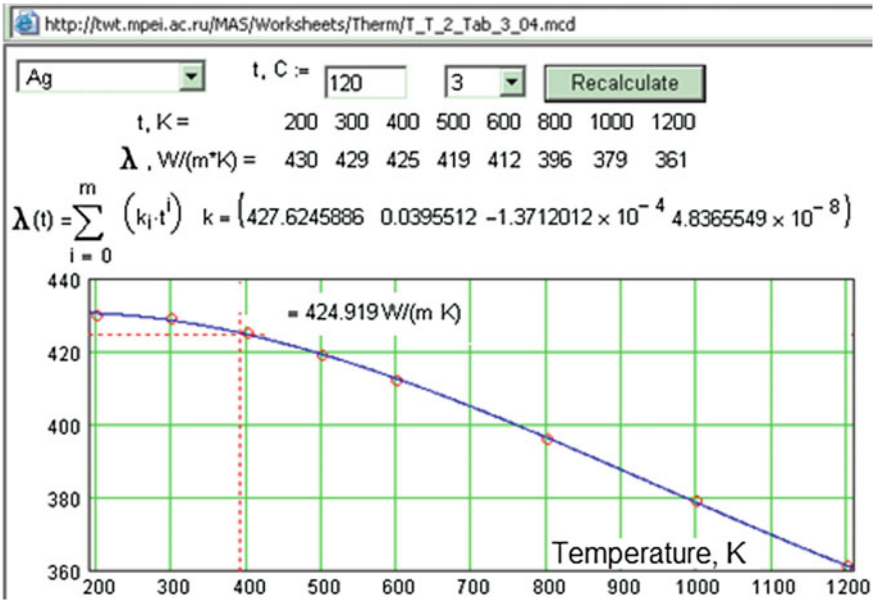


Fig. 20.4 Thermal conductivity of solid materials and substances

instance, Fig. 20.5, where one can see the entire "dependency" or a selected portion of it "under a magnifying glass."

Back to Fig. 20.1. A visitor to this site has freedom to choose the units in the formula: it can be a constant, linear function or a parabola with an argument either in Celsius, or to an absolute scale.

Working with the temperatures is the second "flavor" of our three-layer thermal engineering "pie": on the site, as shown in Fig. 20.1, one can enter the temperature dependence of thermal conductivity of the materials in the layers as constants or polynomials.

As shown in Fig. 1.8, Chap. 1, if some materials cover the entire temperature range it can lead to oscillations—"whipping" of the curve near the points of the original conductivity table.

The third "flavor" of the three-layer cake, as shown in Fig. 20.1, is as follows. This figure (paper sheet) is sandwiched between *the glass* and A4 *cardboard* frame. So, if the temperature near the glass is equal to 20 °C, and near the cardboard backing of the figure is minus 5 °C, the temperature in this "pie" would be distributed as shown in Fig. 20.1 with the figures and diagram: the thermal conductivity of glass is given by a polynomial of the second degree, which is taken from the site shown in Fig. 20.2, and the thermal conductivity of paper and cardboard—by constants found on the Internet. Heat flow through the picture's area of 0.06 m^2 is 60 W.

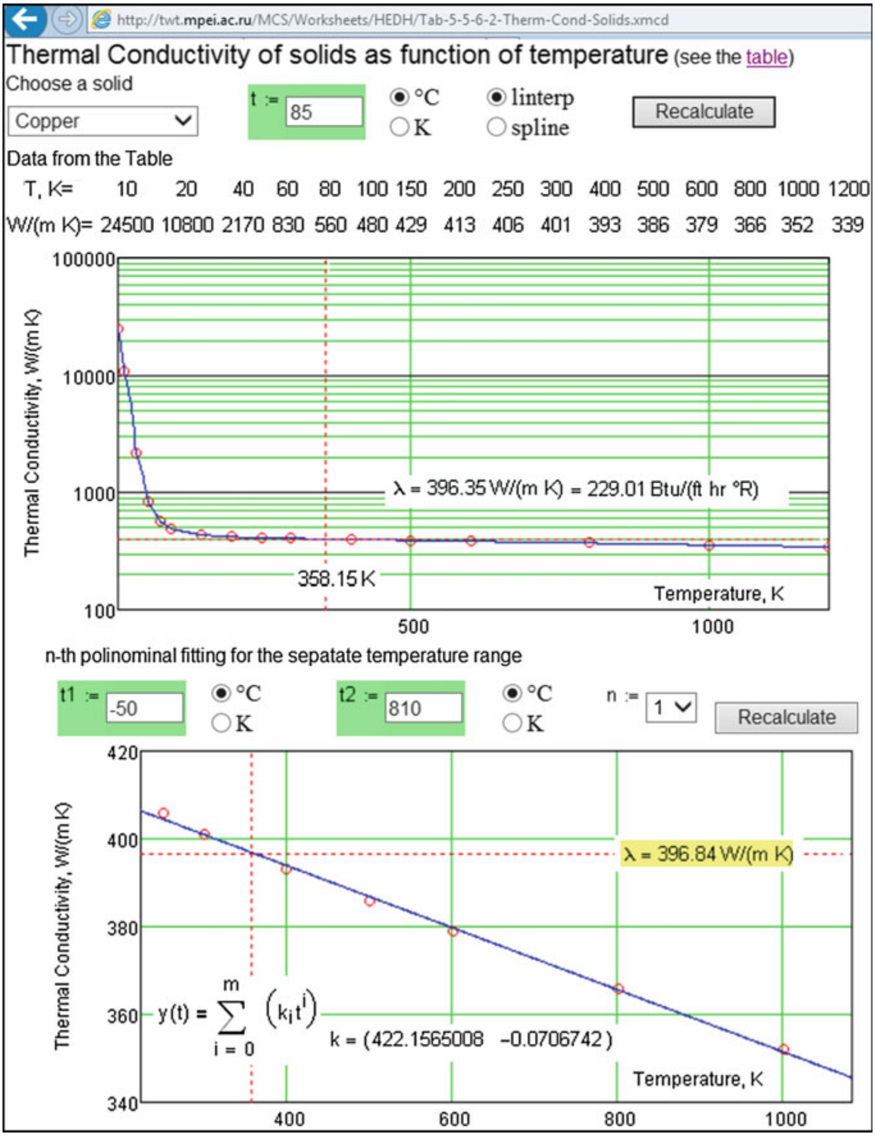


Fig. 20.5 Inertia of reference tabulated thermal conductivity data

These days much attention is paid to energy saving in heating buildings. We mentioned that in Chap. 17 when considering the optimum ways of power supply. Energy could be saved also through the insulation of old buildings, by putting new thermal insulation panels on their walls. The website, as shown in Fig. 20.1, will help us to calculate the effectiveness of such measures.

The author's website on energy efficiency and energy saving http://twt.mpei.ac.ru/ochkov/VPU_Book_New/ES/index.html has lots of such published interactive network calculations. Selecting the thickness of the insulation is a typical optimization problem. By increasing the thickness of the insulation, we reduce heat loss, but increase the cost of the insulation itself, its installation and maintenance. By reducing the thickness of the insulation, we increase heat loss. It is difficult to solve this optimization problem each time for each specific case. Therefore, special codes (SNIP) have been developed which can be used to simplify the calculations. An example of such a calculation of the pipeline through which hot water is running from a cogeneration plant to a heat load can be found at http://twt.mpei.ac.ru/MCS/Worksheets/Thermal/Izol-Trub-Teploset.xmcd.

One of the author's hobbies—creating drawings using a Mathcad, that he could then insert into a frame, put on the wall and show to his guests intriguing them with the "form and content" of this "work of art". An example of one of them see in Fig. 20.1. Other examples can be found on the author's website.

We'll speak separately about a "well-known to a narrow circle" of thermal engineers author's glass cube that has inside the laser traced thermodynamic water and steam surface with a fixed diagram of a simple steam turbine cycle within "temperature–enthalpy–entropy" coordinates (Fig. 20.6).

Virtual WaterSteamPro-cube is posted in the Internet at http://twt.mpei.ac.ru/TTHB/2/wspCube.html (see Fig. 20.7).

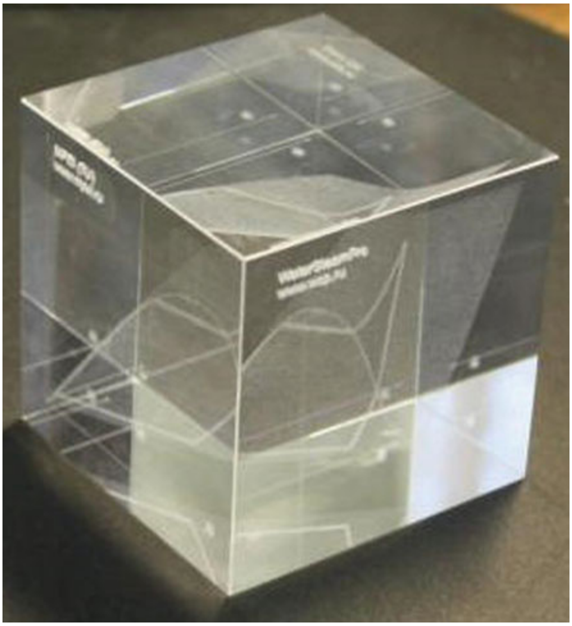


Fig. 20.6 WaterSteamPro-cube

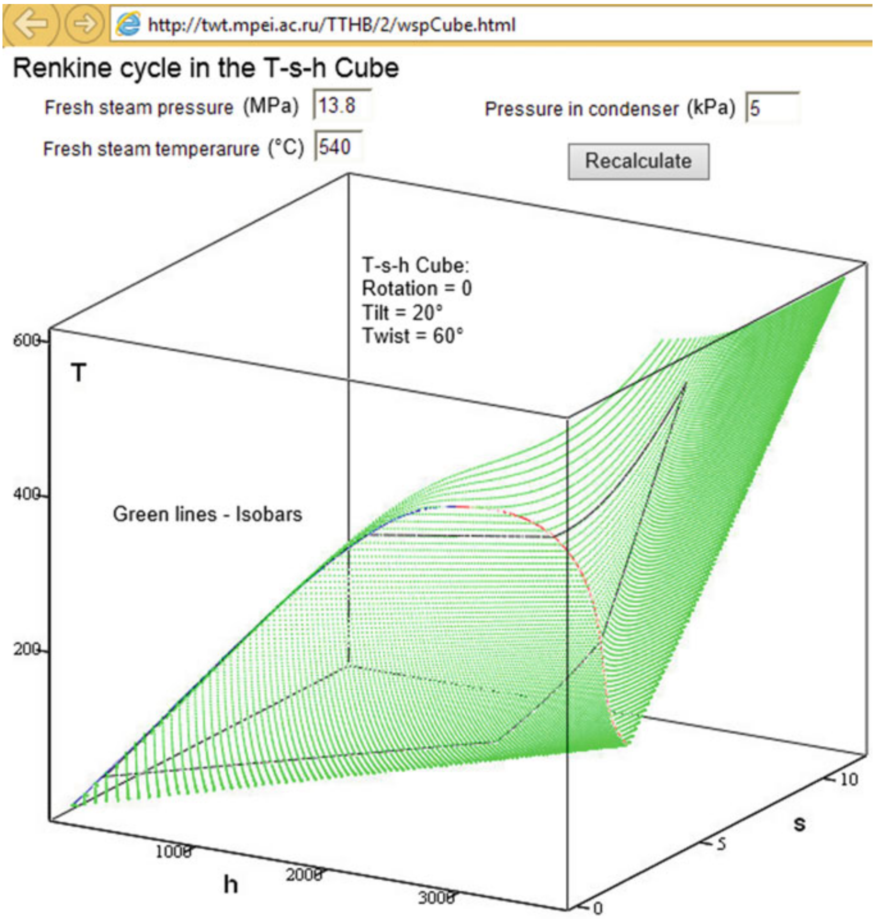


Fig. 20.7 Virtual WaterSteamPro-cube

If the cube turns with its one side to the viewer, one could see the *T*, *s*-diagram of a steam turbine cycle, by the second side—*h*, *s*-diagram and by the third side—*T*, *h*-diagram. The enlarged view of the lower left corner of *T*, *h*-diagram can be seen in Fig. 5.2 in Chap. 5.

This thermodynamic glass cube is a souvenir which the author gives as a gift to his colleagues and partners, in particular the users of the program WaterSteamPro (www.wsp.ru) and "Electronic Encyclopedia of Energy" (www.trie.ru). It is interesting to give this cube to the chief engineers of thermal power plants and energy companies.[2] By the reaction of the people to this gift one can immediately see from

[2]
- Mom, who built the tower?
- Engineer Eifel.

what "origin" this person has come up to the rank of top manager. Chief engineers of thermal power plants are promoted from thermal engineers or electricians. So, the chief engineer who is a thermal engineer immediately appreciates this gift, starts to turn the cube over in his hands and talk about his student years, when he was "fiddling" with the calculation of steam turbine cycle when writing a thesis project. In the eyes of the chief engineer who is an electrician by education one can see at once some confusion, the question—what is it? Here we have to explain that this is a diagram of a steam-turbine power unit installed on your power plant, traced on the thermodynamic surface of water and steam, and for you, electrical engineer, this is like a hysteresis loop, for example. In order to smooth out some confusion, we have to tell the old Soviet joke about the two Soviet violinists who were returning home after an international competition.[3] The first violinist took the second place and grieved about it. The second violinist, which took the last place, was quite happy, and he says to the first one to stop suffering, and that the second place is a very good result, there were, in fact, three hundred participants. The first violinist objected to that in the sense that the winner was given a Stradivarius violin as a gift. The second "violinist" asks, "What is that?"—"Well, it's like a Stalin gun for you"—this was the answer of the first violinist. So, a glass cube with the steam turbine cycle—this is the same cube, but with a hysteresis loop for an electrical engineer.

In addition to the flat (see Fig. 20.1) and volumetric (see Fig. 20.6) "heat engineering works of art" the author has a more practical souvenir—a wall clock with a dial shown in Fig. 20.8. On the clock, besides the numbers—typical Mathcad-operators—there are three Internet addresses, which, as the author hopes, will be "desktop" addresses for the readers of this book, will be bookmarked in your favorites of the Internet browser. If the virtual clock is animated with moving arrows attached to it, the clock can serve as a desktop picture on the computer screen. More about this clock: https://www.ptcusercommunity.com/message/430540.

Our "three-layer thermal engineering cake with flavors" can be seen as a "dessert" to this book, as a kind of conclusion. Conclusive remarks.

The author keeps repeating to his students that from the studies they need to try getting… fun in the first place and only after that knowledge and skills as well. Chapter is the work, very hard work, but without pleasure, without love for work, even an unburdensome, prestigious and highly paid job can turn into hard labor. Pleasure is unthinkable without jokes, without the light and ironic attitude toward the most serious things. And what could be more "serious" than mathematics, or the same heat engineering? Our secondary school with its cramming of answers for the Unified State Exam completely discourages many students from craving knowledge

(Footnote 2 continued)

- This is like our father—an engineer?
- No, son, our dad is chief engineer.

Chief engineer mainly carries out not creative but administrative functions. Therefore he is correctly renamed Technical Director.

[3] Another joke—see the previous footnote.

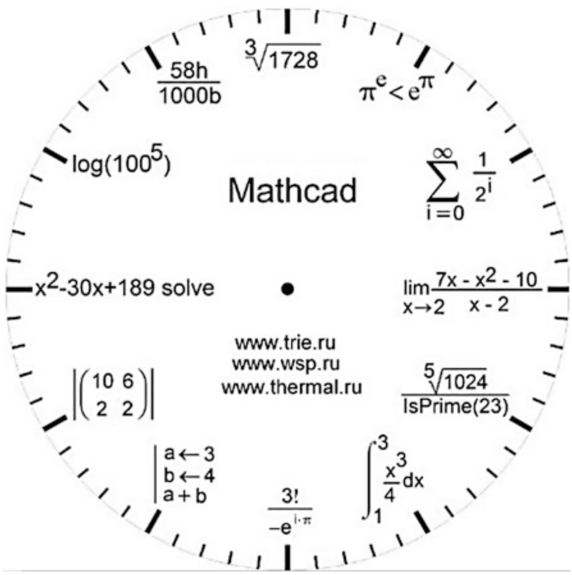


Fig. 20.8 Thermal engineering Mathcad clock

and "getting pleasure from the process of learning." This book can be seen as an attempt to return to the students interest to this most important of human abilities—to acquire knowledge. Many students aspire after a computer, but their future engineering profession is not computer technology (IT), but all the same heat engineering.

The authors hope that this book will help these students at least through the computer and the Internet to restore the ability to receive pleasure from studying at a technical college.

One of the authors of this and the Russian edition of the book Valery Ochkov reads "Information Technology" course to the freshmen in MPEI (Moscow Power Engineering Institute). In their second year they have a course in thermodynamics delivered by Aleksey Alexandrov, another author of the Russian book. We hope that this book will be a useful and, most importantly, a pleasant bridge between these academic disciplines. And the basis of this "bridge" is mathematics, the lectures in mathematics in MPEI are read by Elena Bogomolova, the third author of the Russian book… "Construction and servicing of the bridge" is maintained by the other authors of the Russian book.

More than half a century ago, S.M. Ulam, the "father" of the Monte Carlo method, wrote [72] that each year about two hundred thousand theorems are published in the mathematical journals. They are added to the already proven ones, and the amount of mathematical knowledge (yet the same way as knowledge in the other scientific disciplines) increases more and more. Although mathematics teachers do not even try to explain in their lectures something mathematically new, capable to modernize the engineering education, they barely have time even to teach students the skills and methods which are two centuries old. The result is usually disappointing, the students poorly know the "old" mathematical techniques and do not own the "new" one. Trying to answer the question "So what's the point of teaching mathematics to engineers?", one inevitably comes to the conclusion that the amount of mathematical knowledge received now at the Technical University should be seriously reduced and its nature drastically changed.

Of course, the idea that every engineer has to be a good mathematician continues to reign with us. Undoubtedly, as it was at the dawn of engineering sciences. Back in the thirties of the last century, the outstanding scientist Norbert Wiener [73] read in the University an annual course of combined lectures in mathematics and electrical engineering. Now no one would think to combine these two disciplines into one. Engineering and technological demands of society have changed dramatically.

The technical means used in solving problems has also changed. Foremost the engineers, in fact, are consumers, rather than creators, of the finished mathematical product, which is also "hidden" somewhere in the depths of computer engineering developments. This fact must determine the current trends in mathematics education of engineers.

In order to select the new priorities in teaching mathematics in a technical college and be able to sacrifice a part for the sake of development and prosperity of the whole, let's have a look at the basic processes that have occurred in the last half-century in the field of application of mathematics to engineering sciences. The main thing we will see is that the object of the application of mathematical knowledge has changed significantly.

The technologically weak computer of the twentieth century required that future engineers should have good skills in algebraic transformations, differentiation and integration, to facilitate the calculations carried out by means of a slide rule and old math functions tables, and later with the help of calculators, slow IBM-360 and low-power PC-286. In that time the development of all the world mathematics was aimed at finding analytical solutions and creating suitable numerical methods for their most accurate implementation.

Now with the advent of high-performance computing tools and environments the analytical engineering research has moved into the background. And it can be clearly seen from the examples in this book. For successful study of heat engineering the current students should have good command of mathematical tools relying on the potential of computing programs such as Mathcad features.

Many numerical methods of mathematics are implemented in the Mathcad, and their list is constantly updated. The methods themselves get rid of the programming errors that are inevitable in the beginning of work. For example, one of the first versions of Mathcad plotted hyperbole as the letter N, which surprised many students. Now diagrams created in Mathcad are flawless.

Referring students to software, it is important to consider that behind the ease of use of finished programs there may be big problems hiding related to borders and the peculiarities of their application. And if earlier each researcher, as a rule, thoroughly studied the appropriate numerical method, wrote by himself a computer program and knew its features, limitations and drawbacks, now, using the computer transformations, which have become standard, the engineer often has no idea of the wide range of their features. This is indicated by Mathcad users writing to the forums, and there are a lot of tips in this book on how to avoid an error or correct a mistake.

It is obvious that under present conditions of the developed applied technologies, the vast majority of students do not need the mathematical knowledge, and probably that will not be useful to them in their future work. And then what will they need? There are three global challenges put before a teacher of mathematics in a technical college. The first—to provide the student with a minimum mathematical tools technique required for the study of special disciplines. The second—to clarify as strictly as possible those mathematical foundations on which the research and computing processes are based. The third—to lay the fundamentals of a complex method of formulating and solving problems, to teach the student to check and analyze the result and provide a means of verification.

So, what is needed in this regard from mathematics by a mechanical engineer and what heat engineering in a frame of Mathcad can give to teachers of mathematics? We will be able to answer immediately to both questions by analyzing what mathematical tools technique was used in the solution the heat engineering problems discussed in this book.

To perform thermal design calculations using Mathcad means the user shall have at least basic skills of matrix and vector calculus. He should have a clear idea about matrix, vector and scalar and their properties. Transposition, addition, multiplication and inversion of matrices, calculation of a rank of matrix—these are operations a heating engineer should feel at home with. Of course, it is not required from him to have skills to "manually" invert, for example, a fourth-order square matrix. But he should have a good idea of the fact that if all the elements of a matrix are so small that the computer perceives them as zeros, the matrix cannot be inverted, though in theory the inverse matrix exists. In this regard, especially many questions arise when solving systems of linear algebraic equations, which are one of the main mathematical calculation tools for thermal problems. To select the most convenient and the most accurate method of numerical solution of systems of linear algebraic equations a mechanical engineer should know the strengths and weaknesses of both Gauss method, and Cramer's rule, and methods of solving matrix equations. Knowing the general theory of linear equations systems, the Mathcad user can either be confident of the correctness of the computational result, or envisage the appearance of a computational "incident", or even make discovery in engineering when analyzing the resulting contradiction between the expected theoretical and obtained computational result.

Another focal point of the study seems to be functions of one real variable. One theoretically needs to know everything about that. In the thermal calculations in Mathcad, symbolic operations on functions absolutely equally co-exist with approximate methods of calculation applicable to functions. There is virtually no chapter in this book, where actions relating to the functions of one variable were not used in solving thermal problems. This is calculation of values, interpolation and extrapolation, the search of roots and creating the inverse function, character or numerical differentiation and integration, the search for extrema and study of the behavior of the function. There is a nuance, the specificity of thermal problems is such that calculations are performed mainly with polynomials, power functions and logarithms, as well as with piecewise-defined functions, i.e. such functions that in the different parts of the domain are set with different explicit expressions. Special attention should be paid to the study of the properties of just these functions.

It is also useful to navigate freely in matters of change of variables—in fact it's not just a jump to the other measuring units.

Symbolic and numerical differentiation and integration deserve special attention. You need to know their characteristics and understand in which cases they may not be applicable. This is impossible to understand without knowing the mathematical foundations of these operations. No less complicated are procedures for solving simple differential equations and systems of such equations. For instance, to obtain a numerical solution one needs to set an initial approximation. But this can be done only when the researcher knows the areas where this solution exists, in which areas this solution is unique, and where he can set the starting point. This means that heating engineers just need to know the basics of a theory of differential equations, even if Mathcad exempts him from the need to seek solutions analytically.

Do not think that for the engineer it is sufficient to only study the functions of one variable. Many examples discussed in this book refer to conducting calculations and study of functions of two variables, and if we talk about the three-dimensional non-stationary heat exchange problem, then there we will have to deal with the function of four variables: the three corresponding to coordinates of space, while the fourth is time. Mathcad can do many useful things with the functions of several variables: create diagrams (family of curves), seek extremes (local and global), solve algebraic equations and partial differential equations. To review this range of issues, it is enough to re-read, for example, Chaps. 6, 7 and 18.

It has been said that one of the tasks of a teacher of mathematics is to teach students self-control and self-test. Mathcad also provides great help for this. Only a researcher who has confidence in the correctness of intermediate calculations can move on. Control is given much attention to in our book. This is control of dimensions, and control with labels and implicit theoretical mathematical control. Even when starting to solve a heat engineering problem using a specific mathematical method, a good engineer can predict on the basis of general theoretical knowledge what the result should be.

Particularly noteworthy is the error of calculation. The mathematician has to familiarize students with the general theory of errors. The programmer should tell in detail how calculation errors get accumulated due to the representation of numbers and rounding rules laid down in the machine operations. Mechanical engineer specialist should tell the student why measurement errors occur and how they affect the accuracy of the calculation formulas. Only in this case, you can expect that a method for solving a heat engineering problem will be chosen by a student consciously and usefully.

In the book we touched upon a "generation gap" arising from disputes about the benefits of computing. Let us discuss this in more detail.

Of course, the role of trigonometry in mathematical and engineering sciences is difficult to challenge. It is even more difficult when in the environment of the broad and widespread use by the students of a variety of computational tools one has to prove the feasibility of hours of exercises in differentiation and integration of complex functions containing the sine. Therefore, it's better, don't you think, without tormenting the students with jobs, in which they see no reason, to use the desire of the future engineers to learn something really useful? Leave the trigonometry, along with the imaginary unit, to electricians with their rotating generator rotors producing an alternating electrical current.

One of the most complex mathematical topics of the first semester is "Function study and plotting." A lot of resources available on the Internet, make it possible to create a curve without making any special effort. And it would be unnatural to the students not to use them in the performance of a home work or typical calculation assignment in mathematics. But evidently there is not any benefit to learning from only printing on a computer a formula and then clicking Build key.

In this situation, one could initially require from the students that their detailed "manually" done calculations of every limit and every derivative should be tested on computer each time. The students may be also allowed to provide a diagram created by a computer, rather than to give a sketch drawn by themselves. It is important to only put a condition, that all the extremal and inflection points, and intervals of tangents and asymptotes should be indicated on the prepared diagram. For these operations Mathcad could and should be used but certainly not in its full extent, and in a form adapted for first-year students. We just give them a reference to the "mathematical" page of MPEI calculations server. Of course, it is allowed to use other resources and programs as well.

Such work gives undoubted positive effects. A student who proceeds to a complex and unusual for him job is mentally relaxed: no need to "deceive the teacher," using the computer prompts; there is no fear that the task will not be done in time, or that the answer will be wrong. This allows you to focus on the essence of the work. The student learns that complex multistage calculations can and should be checked at every convenient stage in order with a smaller loss of time to solve the problem correctly. And finally, what is particularly valuable for the education of future engineers, in case of discrepancies in the results of their own research and computer checks the students should decide who is still wrong: human or computer?. In this case, students are looking for any error in their calculations and logical conclusions, or apply other mathematical software for check, or consult with a teacher, not being afraid to admit that the work has been done not by themselves. As a result of such a true research, the students not only get acquainted with various mathematical tools, generously scattered on the Internet, but also starting to realize that all these tools are created by specific people in specific scenarios using numerical methods with certain restrictions for the parameters and scope. One of the students handing-in his paper to a Math teacher said, "I failed with a computer check of plotting, because I found two vertical asymptotes, and the computer creates only one, and I do not know what to do, because I'm sure about my calculations." Obviously, the discussion of this "effect" during the lesson was useful for all the students in the group. To be fair we note that the students working with Mathcad have much less difficulty in solving math problems than the other students.

Thus, it is evident that the structure, sequence and technique of sharing mathematical knowledge requires changes [74]. It is necessary to review the entire teaching material on higher mathematics to eliminate or edit some math sections and include others, more relevant to the future profession. This choice of material should be conducted taking into account the practical importance of the studied sections. As for the numerical methods, it is not necessary to teach methods or their implementation (which is a task of computing software like Mathcad), but understanding of the nature and principles.

One of the teachers said, "If we do not know what to teach, let's go back to the meaning."

And the last thing. In almost all the Studies of the book there is water. And there is nothing surprising about that, water—is the source of life. No wonder the space probes are searching at distant planets for water as the first sign of possible life on them. In our book we discussed the technical principles for using water as the working fluid and coolant. But, as the poet said, (http://www.stihi-rus.ru/1/Sluckiy/40.htm), "For some reason physics are kept high, and lyricists are kept somewhat down." In [75] heat engineering, chemical, biological, economic, and other scientific and technical aspects of water are supplemented with components of philological, art history, psychology and even religious nature…

References

1. Ochkov VF, Utenkov VF, Orlov KA (2000) Teplotekhnicheskiye raschety v srede Mathcad (Thermal Engineering calculations with Mathcad). Teploenergetika J 2:73-78. http://twt.mpei.ac.ru/ochkov/WSPHB/TE-2-2000.pdf. Accessed Sept 2015
2. Yankov GG (2011) Tekhnicheskaya kniga v sovremennoy Rossii: shans na vyjivanie (Technical book in modern Russia: the chance of survival). University Book J 6:67–70. http://www.unkniga.ru/vuz/101-tehknoga.html. Accessed May 2011
3. Invernizzi CM (2013) Closed power cycles: thermodynamic fundamentals and applications. Springer, Berlin
4. Kirillin VA, Sychev V, Sheindlin AE (2008) Tekhnicheskaya termodinamika (Technical Thermodynamics). Publishing House MPEI, Moscow
5. Sychev VV (2009) Slozhnyye termodinamicheskiye sistemy (Complex thermodynamic systems). Publishing House MPEI, Moscow
6. Aleksandrov AA (2006) Termodinamicheskiye osnovy tsiklov teploenergeticheskikh Ustanovok (The thermodynamic basis of cycles of thermal power plants). Publishing House MPEI, Moscow
7. Andryushchenko A (1967) Osnovy tekhnicheskoy termodinamiki real'nykh protsessov (Fundamentals of engineering thermodynamics of real processes). Higher School Publishing House, Moscow
8. Ochkov VF (2002) Fizicheskiye i ekonomicheskiye velichiny v Mathcad i Maple (Physical and economic value of Mathcad and Maple). Finance and statistics Publishing House, Moscow
9. Korobov VI, Ochkov VF (2012) Chimichny rozrakhunki v sededovitschi Mathcad (Chemical calculations in Mathcad). Dnipropetrowsk National University Publishing House, Dnipropetrowsk
10. Ochkov VF (2009) Mathcad 14 dlia studentov i injenerov (Mathcad 14 for students and engineers). BHV-Petersburg, Saint Petersburg
11. Korobov V, Ochkov VF (2011) Chemical kinetics with Mathcad and Maple. Springer, Berlin
12. Ochkov VF (2007) Mathcad 14 dlia studentov, injenerov I dizaynerov (Mathcad 14 for students, engineers and designers). BHV-Petersburg, Saint Petersburg
13. Ochkov VF (2005) Mathcad 12 dlia studentov i injenerov (Mathcad 12 for students and engineers). BHV-Petersburg, Saint Petersburg
14. Solodov AP, Ochkov VF (2004) Differential Models: An Introduction with Mathcad. Springer
15. Vockoboynikov YE, Ochkov VF (2002) Programmirovaniye i resheniye zadach v pakete Mathcad (Programming and problem solving in the package Mathcad). Publisher House NSABU, Novosibirsk
16. Ochkov VF (2001) Sovety pol'zovatelyam Mathcad (Tips and tricks for Mathcad users). Publishing House MPEI, Moscow. http://twt.mpei.ac.ru/ochkov/Units/Forword_book.htm. Accessed Sept 2001

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9

17. Ochkov VF (1999) Resheniye inzhenerno-tekhnicheskikh zadach v srede Mathcad (The decision of engineering tasks in Mathcad). Thermal engineering and heat engineering reference, Book 1. Publishing House MPEI, Moscow
18. Ochkov VF (1999) Mathcad 8 Pro dlia studentov i injenerov (Mathcad 8 Pro for students and engineers). Computer Press, Moscow
19. Ochkov VF (1997) Mathcad 7 Pro dlia studentov i injenerov (Mathcad 7 Pro for students and engineers). Computer Press, Moscow
20. Ochkov VF (1996) Mathcad 6 PLUS dlia studentov i injenerov (Mathcad PLUS 6.0 for students and engineers). Computer Press, Moscow
21. Ochkov VF et al (2007) Internet-versiya spravochnika "Teploenergetika i teplotekhnika. Instrumental'nyye sredstva sozdaniya i razvitiya" (Internet-version of the reference book "Thermal engineering and heat engineering: tools for development"). Publishing house MPE, Moscow. http://twt.mpei.ac.ru/TTHB/5. Accessed Sept 2007
22. Klimenko AV, Zorin VM (2004) Teploenergetika i teplotekhnika. Spravochnik v chetyrekh knigakh (Thermal engineering and heat engineering reference in 4 volumes). Publishing house MPE, Moscow. http://twt.mpei.ac.ru/TTHB/tthb.html. Accessed June 2009
23. Aleksandrov AA, Orlov KA, Ochkov VF (2009) Teplofizicheskiye svoystva rabochikh veshchestv teploenergetiki: Internet-spravochnik (Thermal and physical properties of working fluids of heat engineering: Internet guide). Publishing house MPEI, Moscow. http://twt.mpei.ac.ru/rbtpp/eng. Accessed June 2009
24. Ochkov VF, Orlov K, Ochkov AV et al (2013) Oblachnyy servis po svoystvam rabochikh veshchestv kholodil'nykh ustanovok (Cloud services on the properties of working fluids of refrigeration units). Int Acad Refrig J 2:23–28. http://twt.mpei.ac.ru/ochkov/WSPHB/Web-Refr-Ochkov-R-407c.pdf. Accessed June 2013
25. Ochkov VF, Orlov KA, Frenkel ML et al (2012) Cloud service on the properties of working fluids for thermal engineering calculations. Therm Eng J 59(7):566–572. http://twt.mpei.ac.ru/ochkov/WSPHB/Web-function-Power-Eng.pdf. Accessed Sept 2015
26. Ochkov VF, Orlov KA, Znamenskiy VE (2011) Teplotekhnicheskiye raschety s oporoy na internet-funktsii po svoystvam rabochikh veshchestv teploenergetiki (Thermal engineering calculations based on internet functions on thermophysical properties of working fluids). New Russ Power Eng J 6:40–49. http://twt.mpei.ac.ru/ochkov/WSPHB/Ochkov-Znamensky-Web-Rankine.html. Accessed June 2011
27. Ochkov VF (2009) Publikatsiya v Internete teplofizicheskikh svoystv veshchestv: problemy i resheniya pri rabote s tablitsami (Publication in Internet of thermal properties of materials: problems and solutions when working with tables). Proceedings Academenergo J 2:13–32. http://twt.mpei.ac.ru/ochkov/TablSite. Accessed Sept 2011
28. Kunick M, Kretzschmar HJ, Gampe U (2012) Schnelle und flexible Berechnung thermodynamischer Stoffwerte mit Spline-Interpolation für die Modellierung instationärer Energieumwandlungsprozesse. In: Honekamp W, Schindler P, Tagungsband der 13. Nachwuchswissenschaftlerkonferenz mitteldeutscher Fachhochschulen Görlitz, pp 209–214, Re Di Roma-Verlag, Remscheid
29. www.nist.gov/srd/nist23.cfm. Accessed Sept 2005
30. http://trc.nist.gov/tde.html. Accessed Sept 2005
31. www.wsp.ru Accessed June 2015. Accessed July 2005
32. Aleksandrov AA, Dzhuraeva EV, Utenkov VF (2013) Thermal conductivity of sodium chloride aqueous solutions. Therm Eng J 60(3):190–194
33. Ochkov VF, Khusnullin A (2010) Yedinitsy zhestkosti vody i prochiye kontsentratsii (Units of water hardness and other concentration). Water Purif Treat Supply J 6:21–27. http://twt.mpei.ac.ru/ochkov/Chem_Units/Water-Conc.pdf. Accessed Sept 2010
34. Ochkov VF (2009) Postroyeniye diagramm termodinamicheskikh tsiklov: shag za shagom (Thermodynamic cycles: step-by-step calculation). Autom IT in Electr Eng J 2–3. http://twt.mpei.ac.ru/TTHB/2/ThermCycleMCS-Create.html. Accessed Sept 2009

35. Ochkov AV (2014) Komp'yuternaya programma po izucheniyu i avtomatizirovannoy proverke znaniya tekhnologicheskikh skhem (Computer program for the study and the automated validation of knowledge of technological schemes). Abstracts of the international scientific conference informatization engineering education Inforino-2014, April 15–16 2014. url http://inforino2014.mpei.ru. Accessed Apr 2014
36. Ebeling C (1988) GeminiII: a second generation layout validation program. In: Proceedings of ICCAD
37. Ochkov VF (2007) The development of an electronic encyclopedia of power engineering: an informational contribution to production and education processes. Therm Eng J 54(7):519–524
38. Aleksandrov AA, Ochkov VF, Orlov KA, Ochkov AV (2007) Teplofizicheskiye svoystva vody i vodyanogo para v Internete (Thermal and physical properties of water and water steam in the Internet). Promyshlennaya energetika J 2. url http://twt.mpei.ac.ru/ochkov/WspIn. Accessed Oct 2007
39. Aleksandrov AA, Ochkov VF, Orlov KA (2005) Uravneniya i programmy dlya rascheta svoystv gazov i produktov sgoraniya (Equations and software for calculation of gas and combustion product properties). Teploenergetika J 3
40. Ochkov VF, Orlov KA, Frenkel ML, Ochkov AV, Znamenskiy VE (2012) "Oblachnyy" servis po svoystvam rabochikh veshchestv dlya teplotekhnicheskikh raschetov (Cloud Service on the Properties of Working Fluids for Thermal Engineering Calculations). Teploenergetika J 7:79–86. url http://twt.mpei.ac.ru/ochkov/WSPHB/Web-function-Power.pdf. Accessed Sept 2012
41. Ochkov VF (2005) Teplotekhnicheskiy spravochnik v Internete (Thermotechnical guide in the Internet). New Russ Electr Power Ind J 5. url http://twt.mpei.ac.ru/ochkov/VPU_Book_New/mas/NRE_5_5. Accessed Oct 2005
42. Ochkov VF (2011) "Oblachnyy" servis po svoystvam rabochikh tel i materialov atomnoy energetiki (Cloud service on the properties of working fluids and materials of nuclear power). Autom IT Electr Eng J 3:4–8. url http://twt.mpei.ac.ru/TTHB/npp/CC.pdf. Accessed Sept 2011
43. Orlov KA, Aleksandrov AA, Ochkov VF et al (2002) Programmnyy kompleks "WaterSteamPro" dlya rascheta teplofizicheskikh svoystv vody i vodyanogo para (WaterSteamPro software system for water and steam thermal and physical properties calculations). In: X Russian conference on thermal and physical properties of substances: paper extracts—Kazan. Publishing house, Butlerovskie soobscheniya. url www.wsp.ru. Accessed Sept 2005
44. Tupov VB (2005) Snizheniye shuma ot energeticheskogo oborudovaniya (Decrease of power equipment noise). Publishing house MPEI, Moscow
45. Gurvich LV et al (1996) Thermodynamic properties of individual substances. Begell Publishing House
46. Ochkov VF, Ko CK, Gibadullin IA, Piskotin SA (2012) Internet-spravochniki: rabota s formulami (Internet Guides: Operations with formulas). Otkrytoe obrazovanie J 3. url http://twt.mpei.ac.ru/ochkov/Mathcad-15/InternetFormulasOchkov.pdf. Accessed Sept 2012
47. Martynova OI, Nikitin AV, Ochkov VF (1990) Water treatment: calculations on a personal computer (Water Treatment calculations on a personal computer). Energoatomizdat, Moscow. url http://twt.mpei.ac.ru/ochkov/VPU_PC/index.htm. Accessed Sept 2000
48. Ochkov VF (2009) Optimizatsiya "podarka teplotekhnikov plokhim khimikam-vodnikam" (Optimization of heat engineers gift to bad water chemists). Water Treat Conditioning Supply J 11(23):52–54. url http://twt.mpei.ac.ru/ochkov/Boiler/index.html. Accessed Sept 2009
49. Ochkov VF, Makushin AA, Ko CK (2010) "Zhivyye" raschety, funktsii i raschetnyye protsedury po svoystvam ionitov (Live calculations, functions, and calculation procedures for ionite properties//Water treatment, conditioning and supply, No. 11. url http://twt.mpei.ac.ru/ochkov/Opt-Filtr/R-H.html. Accessed Sept 2005

50. Ochkov VF (1997) Kak ya prodaval programmu (How I sold a program). ComputerPress J 7. url http://twt.mpei.ac.ru/ochkov/paint.htm. Accessed Sept 2000
51. Ochkov VF, Gavrilenko SS (2010) Programmirovannyy vybor fil'trov (Programed choose of filters). Water Treat Conditioning Supply J 7. url http://twt.mpei.ac.ru/ochkov/Opt-Filtr/Filter-2.pdf. Accessed Nov 2010
52. Ochkov VF, Gurke S, Znamensky VE (2014) Rabota na forumakh — vazhnyy aspekt inzhenernoy deyatel'nosti (Work on forums—an important aspect of engineering). Abstracts of International Scientific Conference Informatization of engineering education Inforino. url http://twt.mpei.ac.ru/ochkov/CloudUniOchkov.pdf. Accessed Sept 2014
53. Zorin VM (2008) Atomnyye elektrostantsii (Nuclear power plants. Main technological process). Moscow Power Engineering Institute Publishing House, Moscow
54. Ochkov VF, Orlov KA, Ivanov EN, Makushin AA (2013) Calculation and visual displaying of the water chemistry conditions in return cooling systems at thermal power stations. Therm Eng J 60:465–471. http://twt.mpei.ac.ru/ochkov/WSPHB/ThermEng1307007OchkovKOR.pdf. Accessed Sept 2013
55. Ochkov VF, Orlov KA, Ochkov AV, Znamenskiy VE (2013) Application of the cloud technologies to perform heat engineering calculations. Therm Eng J 60(9):679–685. url http://twt.mpei.ac.ru/ochkov/WSPHB/CloudFormulas.pdf. Accessed Sept 2013
56. Ochkov VF, Ustyuzhanin EE, Ko CK, Shishakov VV (2015) Thermophysical databases: from tables to interactive internet resources and cloud templates. High Temp J 53(4):515–520. url http://twt.mpei.ac.ru/ochkov/TBT-BD-Internet-eng.pdf. Accessed Nov 2015
57. Ochkov VF (2013) Zhivyye kinematicheskiye skhemy v Mathcad (Live kinematic diagrams in Mathcad). Open Educ J 3:27–33. http://twt.mpei.ac.ru/ochkov/Mathcad-15/kinematic.html. Accessed Sept 2013
58. Ochkov VF, Aleksandrov AA, Orlov KA (2007) Termodinamicheskiye tsikly: raschety v Internete (Thermodynamic cycles: Online calculations). Vestnik MPEI J 1:43–50. http://twt.mpei.ac.ru/ochkov/Therm_Cycle_Art/index.html. Accessed Sept 2015
59. Wang D, Zhi LH (2007) Symbolic-numeric computation. Springer, Berlin
60. Velichko VI, Solokhin VI (2014) Teploprovodnost': zadachi i uprazhneniya (chislennoye resheniye) (Heat conduction: problems and exercises (numerical solution). Publishing House MPEI, Moscow
61. Ochkov VF, Orlov KA, Kondakova GY, Ochkov AV (2009) Trenazher dispetchera teplovoy seti (Heat grid dispatcher training simulator). Energetic J 4:38–41. url http://twt.mpei.ac.ru/ochkov/trenager/ThermoNet/page38.png (39, 40 and 41.png). Accessed Sept 2009
62. Ochkov VF, Mescheryakov SV, Orlov KA et al (2007) Trenazher dlya podgotovki personala teplovykh setey (Training simulator for heat grid staff). New Russ Power Ind J 1:33–43. http://twt.mpei.ac.ru/ochkov/trenager/ThermoNet/Article_NRE.html Accessed Feb 2007
63. Ochkov VF, Orlov KA, Chzho Ko Koa, Anokhin DN (2012) Oblachnyye funktsii dlya inzhenernykh raschetov vodosnabzheniya (Clod functions for water supply engineering calculations). Water Supply Sewerage J 9–10:68–74. url http://twt.mpei.ac.ru/GDHB/CloudFunction.pdf. Accessed Sept 2012
64. Ochkov VF, Bogomolova EP (2015) Teaching mathematics with mathematical software. J Humanistic Math 5(1):265–285. url http://twt.mpei.ac.ru/ochkov/teaching_of_math.pdf. Accessed July 2015
65. Plis AI, Slivina NA (2003) Matematicheskiy praktikum dlya inzhenerov i ekonomistov (Mathcad. Mathematics workshop for engineers and economists. Education guidance). Finance and Statistics, Moscow
66. Ochkov VF (2005) Sneg v megapolise i energetika (Snow in a city and power industry). Power Supply Water Treatment J 1. url http://twt.mpei.ac.ru/ochkov/Snow/index.html. Accessed Sept 2005
67. OchkovVF (2010) Sneg v bol'shom gorode (Snow in a big city). Water Treat Conditioning Supply J 3(27):72–74. url http://twt.mpei.ac.ru/ochkov/Snow/index-vvv.html. Accessed Sept 2010

68. Ochkov VF, Bogomolova EP (2015). Eto strashnoye slovo difury (This Terrible Word "Difeqs"). Inf Sch J 1:55–58. url http://twt.mpei.ac.ru/ochkov/ODE.pdf. Accessed Sept 2015
69. Sokolov EY, Brodiansky VM (1981) Energeticheskiye osnovy transformatsii tepla i protsessov okhlazhdeniya (Energy transformation basics of heat and cooling processes). Energoatomizdat, Moscow
70. Cengel Y, Boles M (2006) Thermodynamics: an engineering approach, 5th edn. McGraw-Hill Higher Education, Boston
71. Moroziuk TV (2006) Teoriya kholodil'nykh mashin i teplovykh nasosov (Theory of chillers and heat pumps). Negoziant Publishing House, Odessa
72. Ulam S (2001) Priklyucheniya matematika (Mathematician's adventures). RHD Publishing House, Izhevsk
73. Wiener N (1964) I am a mathematician. MIT Press, Cambridge
74. Yartseva AD, Chernavsky AV (2012) O matematike: problemy prepodavaniya (About mathematics, problems in teaching). Publishing House Znak, Moscow
75. Ochkov VF (2009) Voda v nauchnom, kul'turologicheskom i religioznom aspektakh (Water in a scientific, cultural and religious aspects). Water Treat Condition Supply J 7:64–70. url http://twt.mpei.ac.ru/ochkov/WSCR/WaterSCR.pdf. Accessed Sept 2009
76. Miyagawa K, Hill PG (2001) Rapid and accurate calculation of water and steam properties using the tabular Taylor series expansion method. J. Eng. Gas Turbines Power 123(3):707–712. doi:10.1115/1.1367340
77. Kunick M, Kretzschmar HJ, Gampe U (2008) Fast calculation of thermodynamic properties of water and steam in process modelling using spline interpolation. In: Proceedings of the 15th international conference on the properties of water and steam, Berlin, 8-11 Sept 2008
78. Wagner W, Kretzschmar HJ (2008) International steam tables—properties of water and steam based on the industrial formulation IAPWS-IF97. Springer, Berlin
79. Parry W, Bellows J, Gallagner J, Harwvey A (2000) ASME steam tables for industrial use. ASME Press, New York
80. Tsanev SV, Burov VD, Remezov AN (2006) Gazoturbinnye i parogazovye ustanovki teplovyh elektrostancii (Gas turbine and combined cycle gas turbine thermal power plants). Publishing House MEI, Moscow
81. Andryushchenko AI (1998) O termodinamicheskoy effektinvosty slozhnyh tsiklov GTU v parogazovyh ustanovkah (The thermodynamic efficiency of complex cycles of gas-turbine units in combined-cycle installations). Teploenergetika J 3
82. Andryushchenko AI (1975) Osnovy termodinamiki tsiklov teploenergeticheskikh ustanovok (The basis of thermodynamics of cycles of thermal power plants). Higher School, Moskow