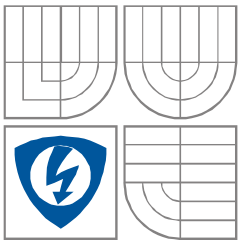
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



Mathematics 3
(Numerical Methods: Exercise Book)

Autor textu:
RNDr. Michal Novák, Ph.D.

Verze 1.1

Komplexní inovace studijních programů a zvyšování kvality výuky na FEKT VUT v Brně
OP VK CZ.1.07/2.2.00/28.0193

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Department of Mathematics FEEC BUT, 2014

http://www.umat.feec.vutbr.cz

novakm@feec.vutbr.cz

Graphics in METAPOST and METAFONT: Jaromír Kuben
Other graphics: Michal Novák

This text is supplemented with *maplets*, GUI applications of Maple. Links to maplets are higlighted using color or word *maplet*. Maplets do not need Maple installation on the client computer. However, they are Java based applications and as such require installation of Java and proper configuration of both Java and the web browser (including setting sufficient security level). Based on these settings, various dialogs or warnings may pop up when the maplet link is activated. Allow running the software and do not block it.

This text is supplemented with executable files programmed with MATLAB. Prior to executing these files, MATLAB Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on MATLAB Compiler Runtime see help on Mathworks website.

Contents

Introduction **2**

How to use this text **4**

1 Systems of linear equations **6**

1.1 Wording the problem . . . 6

1.2 A brief summary of the subject matter . . . 6

1.2.1 Methods of calculation . . . 6

1.2.2 Prior to starting the algorithms . . . 7

1.2.3 Stopping the algorithms . . . 8

1.3 Exercises . . . 9

1.3.1 Prior to starting the algorithms . . . 9

1.3.2 Exercising the algorithms . . . 9

1.4 Related issues . . . 14

1.5 Solving linear systems using MATLAB . . . 15

1.6 Exam tasks . . . 15

1.7 Supplementary electronic materials . . . 17

2 One nonlinear equation **18**

2.1 Wording the problem . . . 18

2.2 A brief summary of the subject matter . . . 18

2.2.1 Methods of calculation – general discussion . . . 18

2.2.2 Root isolation (or localizing the solution) . . . 18

2.2.3 Methods of calculation – formulas . . . 19

2.2.4 Methods of calculation – convergence . . . 19

2.2.5 Stopping the algorithms . . . 20
2.3 Exercises . . . 21
2.3.1 Prior to starting the algorithms . . . 21
2.3.2 Exercising the algorithms: bisection method and regula falsi . . . 22
2.3.3 Exercising the algorithms: Newton's method and simple fixed–point iteration . . . 26
2.4 Related issues . . . 32
2.5 Solving nonlinear equations using MATLAB . . . 33
2.6 Exam tasks . . . 35
2.7 Supplementary electronic materials . . . 36

3 Systems of nonlinear equations 37
3.1 Wording the problem . . . 37
3.2 A brief summary of the subject matter . . . 37
3.2.1 Methods of calculation . . . 37
3.2.2 Prior to starting the algorithms . . . 38
3.2.3 Stopping the algorithms . . . 39
3.3 Exercises . . . 39
3.3.1 Prior to starting the algorithms . . . 39
3.3.2 Exercising the algorithms: the Newton's method . . . 39
3.4 Related issues . . . 45
3.5 Solving systems of nonlinear equations using MATLAB . . . 46
3.6 Exam tasks . . . 48
3.7 Supplementary electronic materials . . . 49

4 Aproximations of functions: interpolations 50
4.1 Wording the problem . . . 50
4.2 A brief summary of the subject matter . . . 50
4.2.1 Possible approaches . . . 50
4.2.2 Interpolation polynomial . . . 51
4.2.3 Spline . . . 51
4.3 Exercises . . . 53
4.3.1 Prior to starting calculations . . . 53

4.3.2 Exercising the algorithms: interpolation polynomial and natural cubic splines . . . 54
4.4 Related issues . . . 59
4.5 Interpolation using MATLAB . . . 60
4.6 Exam tasks . . . 60
4.7 Supplementary electronic materials . . . 61

5 Least squares method **63**
5.1 Wording the problem . . . 63
5.2 A brief summary of the subject matter . . . 63
5.2.1 Idea of the method and its use . . . 63
5.2.2 Calculation of coefficients . . . 64
5.3 Exercises . . . 65
5.3.1 Prior to starting the calculation . . . 65
5.3.2 Calculation of coefficients . . . 65
5.4 Related issues . . . 70
5.5 The least squares method using MATLAB . . . 70
5.6 Exam tasks . . . 71
5.7 Supplementary electronic materials . . . 72

6 Numerical integration **74**
6.1 Wording the problem . . . 74
6.2 A brief summary of the subject matter . . . 74
6.2.1 The idea of numerical integration and its advantages . . . 74
6.2.2 Methods of calculation . . . 74
6.2.3 Precision and stopping the algorithms . . . 75
6.3 Exercises . . . 76
6.3.1 Prior to starting the algorithms . . . 76
6.3.2 Composite trapezoidal and Simpson's rules . . . 76
6.4 Related issues . . . 79
6.5 Numerical integration using MATLAB . . . 79
6.6 Exam tasks . . . 80
6.7 Supplementary electronic materials . . . 80

7 Ordinary differential equations **82**

7.1 Wording the problem . . . 82

7.2 A brief summary of the subject matter . . . 82

7.2.1 Existence and uniqueness of solution, stability . . . 83

7.3 Exercises . . . 84

7.3.1 Prior to starting the algorithms . . . 84

7.3.2 Exercising the algorithms . . . 85

7.4 Related issues . . . 87

7.5 Solution using MATLAB . . . 88

7.6 Exam tasks . . . 88

7.7 Supplementary electronic materials . . . 89

8 Mathematical dictionary **91**

8.1 Czech – English . . . 92

8.2 English – Czech . . . 100

Bibliography **106**

Introduction

This text is a supplementary material for *Mathematics 3*, a subject taught at Faculty of Electrical Engineering and Communication, Brno University of Technology. It aims at two target groups: *incoming Erasmus+ students of our faculty or students who study their courses in English*, as well as on a much more numerous group of *students of Czech classes*.

Preparing this exercise book was motivated by the needs of both of these groups. Motivation for students studying in English is obvious – so far there has not been an exercise book for the subject written in English. As far as students of Czech classes are concerned, we believe that a graduate of our faculty should (at least when using English for professional purposes) be bilingual. One of the obvious reasons – which is not likely to change in future – is the simple fact that no truly professional mathematical software has been localized into Czech so far. From this point of view, exercise books written in Czech and English should be treated as interchangeable equivalents – by both teachers and students.

This text is an integral part of a series of materials prepared for *Mathematics 3* by its current lecturers Irena Hlavičková and Michal Novák. In spring 2014 electronic teaching texts [1] were published in both Czech and English. In autumn / winter 2014 these were supplemented by lecture presentations (in Czech only) and commented exam tasks [4] (in Czech only). Finally, an update of the exercise book for the other half of *Mathematics 3*, probability and statistics, is to be published at the turn of 2014/5.

This text is not meant to be a substitute of the main teaching text [1]. The summaries of theory are too brief to be studied alone. They do not give more than a general view of philosophy of numerical solutions of the respective tasks and an overview of formulas actually used in the exercises included in the text. Instead of being copies of [1] they are retold and adapted versions of texts included in [1]. This is because they serve different purposes.

This exercise book is not the very first text of its kind for *Mathematics 3* as there exists [3] written in Czech. However, exercises in chapters 1 to 5 of this text differ from exercises included in [3]. Thus students of *Mathematics 3* have in fact access to almost twice as many exercises.

For practical purposes, a mathematical dictionary is included in this text. With both target groups of this exercise book in our minds we include both Czech – English and English – Czech dictionary. Its entries are restricted to the ones used in *Mathematics 3*.

Since the English version of [1] is a third party translation while this text was written by the author himself, there may be some discrepancies in terminology or in wording of definitions or theorems. The mathematical dictionary included as chapter 8 makes use of the corpus of [2] prepared in 2006. However, it was proofread, updated, corrected and expanded. Moreover, since export of [2] is not possible into LaTeX, it had to be physically rewritten and newly formatted.

December 2014 *author*

How to use this text

This text is an exercise book for one half of *Mathematics 3*, the introduction to numerical methods. Its division into chapters and the choice of its topics follow [1]. Each chapter is organized as follows.

- First, we word the problem.
- Then we give a brief summary of the subject matter.
- After this we include exercises – first tasks related to issues such as testing convergence or some prelimanary calculations or considerations, then tasks exercising the respective algorithms themselves.
- We include some related issues and discuss problems which are not obvious when one concentrates just on practising the algorithms.
- We always include a brief section discussing the way MATLAB, a leading professional numerically oriented mathematical software (the one you are going to use in further subjects of your studies), approaches the problem.
- Since the commented exam tasks [4] exist in Czech only, we include a few examples of authentic exam tasks.
- We finish each chapter with a list of electronic materials which might be of some help in further calculations or considerations or in students' own uncontrolled practice.

Results of all exercises in this text were computed in MATLAB using tailored `*.m` files which all have a form of algorithms modelling techniques taught in the subject. Thus all auxiliary calculations in all steps of every method were done using the built-in precision of MATLAB. All results in this exercise book are rounded to 4 decimal digits, i.e. as displayed by MATLAB using the `format short` command. Therefore, the reader should always do the exercise from the very beginning – starting from other than the initial approximation or from other than initial values will in a general case **not** lead to the results displayed for the next step.

All the exercises are *mathematical* ones (as opposed to the *engineering* exercises). They lack any relation to any specialized engineering subjects. Therefore, they must be solved without any additional information which is usually provided by the specific engineering

application. Thus we have to always carefully perform convergence tests, root isolations or properly discuss all possibilities. The coefficients or data we work with have no special meaning, i.e. if we regard e.g. a system of linear equations, the coefficients of the matrix and the vector are arbitrary real numbers. Naturally, in various specific engineering contexts they may express time (thus be positive), fall within a given interval, or be constructed using specific algorithms – as is e.g. the case of linear system (4.4) in chapter 4. However, since mathematics is a common ground of all engineering applications, any special cases must be ignored here.

1 Systems of linear equations

1.1 Wording the problem

We are going to solve systems of n linear equations of n unknowns. The system will be denoted as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned} \tag{1.1}$$

or

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{1.2}$$

where $\mathbf{A} = (a_{ij}), i, j = 1, \ldots, n$, $\mathbf{b} = (b_1, b_2, \ldots, b_n)^T$ and $\mathbf{x}$ is the vector of unknowns. We assume that $\mathbf{A}$ is a real matrix and $\mathbf{b}$ is a vector of real numbers.

1.2 A brief summary of the subject matter

1.2.1 Methods of calculation

You have learned two types of strategies which may be used to solve systems (1.1): *direct methods* (in *Mathematics 1*) or *iteration methods* (in this subject).

Iteration methods are applications of the *Banach fixed-point theorem*. We modify (1.2) to

$$\mathbf{x} = \mathbf{C}\mathbf{x} \tag{1.3}$$

and look for the fixed point (in our case vector) $\mathbf{x}$. Since the *Banach fixed-point theorem* is valid in some contexts only, we have to test its applicability, i.e. test convergence of the iteration method we choose.

The equations in (1.3) may have two forms:

$$\begin{aligned} x_1^{(r+1)} &= \frac{1}{a_{11}}\Big(b_1 - a_{12}\,x_2^{(r)} - a_{13}\,x_3^{(r)} - \cdots - a_{1n}\,x_n^{(r)}\Big) \\ x_2^{(r+1)} &= \frac{1}{a_{22}}\Big(b_2 - a_{21}\,x_1^{(r)} - a_{23}\,x_3^{(r)} - \cdots - a_{2n}\,x_n^{(r)}\Big) \\ &\vdots \\ x_n^{(r+1)} &= \frac{1}{a_{nn}}\Big(b_n - a_{n1}\,x_1^{(r)} - a_{n2}\,x_2^{(r)} - \cdots - a_{n\,n-1}\,x_{n-1}^{(r)}\Big), \end{aligned} \tag{1.4}$$

or

$$\begin{aligned} x_1^{(r+1)} &= \frac{1}{a_{11}}\Big(b_1 - a_{12}\,x_2^{(r)} - a_{13}\,x_3^{(r)} - \cdots - a_{1n}\,x_n^{(r)}\Big) \\ x_2^{(r+1)} &= \frac{1}{a_{22}}\Big(b_2 - a_{21}\,x_1^{(r+1)} - a_{23}\,x_3^{(r)} - \cdots - a_{2n}\,x_n^{(r)}\Big) \\ x_3^{(r+1)} &= \frac{1}{a_{33}}\Big(b_3 - a_{31}\,x_1^{(r+1)} - a_{32}\,x_2^{(r+1)} - \cdots - a_{3n}\,x_n^{(r)}\Big) \\ &\vdots \\ x_n^{(r+1)} &= \frac{1}{a_{nn}}\Big(b_n - a_{n1}\,x_1^{(r+1)} - a_{n2}\,x_2^{(r+1)} - \cdots - a_{n\,n-1}\,x_{n-1}^{(r+1)}\Big), \end{aligned} \tag{1.5}$$

where $x_i^{(r)}$, $i = 1, \ldots, n$, is the r–th approximation of the unknown x_i. We denote $\mathbf{x}^{(r)} = = (x_1^{(r)}, \ldots x_n^{(r)})$.

If we use (1.4) for the calculation, we speak about the **Jacobi method** while if we use (1.5), we speak about the **Gauss-Seidel method**.

1.2.2 Prior to starting the algorithms

Neither the Jacobi nor the Gauss-Seidel method shall be used without further considerations as some systems converge to the exact solution while others do not.

There are two types of convergence test: equivalencies and implications. Since *Mathematics 3* provides only an introduction to numerical methods, we focus on convergence tests in the form of sufficient conditions, i.e. implications.

In the tests we work with the concepts of *strict row diagonal dominance* and *strict column diagonal dominance*.

Definition 1.1. The matrix A is called **strictly row diagonally dominant** if and only if

$$|\,a_{ii}| > \sum_{j=1, j\neq i}^{n} |\,a_{ij}| \quad \text{for } i = 1, \ldots, n \tag{1.6}$$

(or if in every row of the matrix the absolute value of the element on the matrix diagonal is greater than the sum of absolute values of all other elements in that row) and **strictly column diagonally dominant** if and only if

$$|a_{jj}| > \sum_{i=1, i\neq j}^{n} |a_{ij}| \quad \text{for } j = 1, \ldots, n \tag{1.7}$$

(or if in every column of the matrix the absolute value of the element on the matrix diagonal is greater than the sum of absolute values of all other elements in that column).

Obviously, strict row or column diagonal dominance may be achieved (or disrupted) by re-ordering the system.

Theorem 1.2. *If the matrix* $\mathbf{A}$ *of* (1.2)*, i.e. of* $\mathbf{Ax} = \mathbf{b}$*, is strictly row or strictly column diagonally dominant, then the Jacobi and the Gauss-Seidel method converge to the exact solution of the system and that for any choice of the initial approximation.*

If (1.2) cannot be used, we can use the following theorem.

Theorem 1.3. *If the matrix* $\mathbf{A}$ *of* (1.2)*, i.e. of* $\mathbf{Ax} = \mathbf{b}$*, is regular, then the system*

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

is such that the Gauss-Seidel method will converge to the exact solution of the system and that for any choice of the initial approximation.

Notice that $\mathbf{A}^T$ is the transpose of matrix $\mathbf{A}$. Finally, the following theorem might help in deciding whether the iteration solution of a system will converge to the exact solution.

Theorem 1.4. *If* $\mathbf{A}$ *is a strictly row or strictly column diagonal matrix, then it is regular.*

1.2.3 Stopping the algorithms

In our introductory course we stop the algorithm when

$$\| \mathbf{x}^{(r)} - \mathbf{x}^{(r-1)} \| < \varepsilon,$$

i.e. when the increments of all unknowns are smaller than the precision we require. However, it is to be noted that this does not mean that we get the solution with the required precision ε. Notice that $\| \cdot \|$ in fact denotes a uniform vector norm, the special case of which is the absolute value of differences of the respective components, i.e.

$$|x_i^{(r)} - x_i^{(r-1)}| < \varepsilon, \tag{1.8}$$

for all $i = 1, \ldots, n$.

1.3 Exercises

1.3.1 Prior to starting the algorithms

1. Give an example of a system of 3 linear equations of 3 variables which has no solution.

2. Give an example of a system of 3 linear equations of 3 variables which has infinitely many solutions.

3. Give an example of a system of 3 linear equations of 3 variables which has exactly 2 solutions (i.e. exactly 2 vectors are its solution).

4. Give the definition of a *regular matrix*.

5. Give an example of a matrix $(3, 3)$ which is regular.

6. Give an example of a matrix $(5, 5)$ which is regular.

7. Give an example of a matrix $(5, 4)$ which is regular.

8. Give an example of a matrix which is row strictly diagonally dominant yet not column strictly diagonally dominant.

9. Give an example of a matrix which is column strictly diagonally dominant yet not row strictly diagonally dominant.

10. Give an example of a matrix $\mathbf{A}$ which is regular yet the system $\mathbf{Ax} = \mathbf{b}$ is not solvable by the Gauss-Seidel method.

Test yourself in matrix multiplication and in calculating determinants using the following maplets: Multiplication of matrices and Calculating the determinant.

1.3.2 Exercising the algorithms

In Exercises 1–5 perform 2 steps of the Jacobi's method. As a new exercise always perform 2 steps of the Gauss–Seidel method. In both cases choose $\mathbf{x}^{(0)} = (0, 0, 0)$ *for the initial approximation. Then compare your results and the exact solution obtained using either Cramer's rule or the Gaussian elimination method. Prior to starting the iteration algorithms always check convergence of the chosen method. If convergence is not secured by the form of the linear system, try to secure it.*

Results of all exercises are always included as decimal numbers rounded to four digits after the decimal point.

1.

$$\begin{aligned} x_1 + 6x_2 - 8x_3 &= -6.6 \\ 10x_1 - 3x_2 + 6x_3 &= 8.6 \\ x_1 + 15x_2 - 9x_3 &= 0.8 \end{aligned}$$

Solution: The system is not in the form securing convergence of either Jacobi or Gauss–Seidel method as suggested by Theorem 1.2. However, if we rearrange the system to order of equations (2), (3), (1), both methods will converge. Alternatively, for the Gauss–Seidel method we could multiply the system as suggested by Theorem 1.3.
The Jacobi method gives the following results: $\mathbf{x}^{(1)} = (0.8600; 0.05330; 0.8250)$, $\mathbf{x}^{(2)} = (0.3810; 0.4910; 0.9725)$, while the Gauss-Seidel methods (after rearranging the system) gives $\mathbf{x}^{(1)} = (0.8600; -0.0040; 0.9295)$ and $\mathbf{x}^{(2)} = (0.3011; 0.5910; 1.3059)$.
The exact solution of the system is $\mathbf{x} = (0,5; 1; 1,5)$.

2.

$$\begin{aligned} 3x_1 + 10x_2 - 4x_3 &= 9 \\ 20x_1 + 2x_2 + 3x_3 &= 25 \\ 2x_1 - x_2 + 5x_3 &= 6 \end{aligned}$$

Solution: The system is not in the form securing convergence of either Jacobi or Gauss–Seidel method as suggested by Theorem 1.2. However, if we rearrange the system to order of equations (2), (1), (3), both methods will converge. Alternatively, for the Gauss–Seidel method we could multiply the system as suggested by Theorem 1.3.
The Jacobi method gives the following results: $\mathbf{x}^{(1)} = (1.2500; 0.9000; 1.2000)$, $\mathbf{x}^{(2)} = (0.9800; 1.0050; 0.8800)$, while the Gauss-Seidel methods (after rearranging the system) gives $\mathbf{x}^{(1)} = (1.2500; 0.5250; 0.8050)$ and $\mathbf{x}^{(2)} = (1.0768; 0.8990; 0.9491)$.
The exact solution of the system is $\mathbf{x} = (1; 1; 1)$.

3.

$$\begin{aligned} 3x_1 + 4x_2 - 5x_3 &= 2 \\ 4x_1 + 5x_2 - 3x_3 &= 6.75 \\ 5x_1 - 4x_2 + 3x_3 &= 1.5 \end{aligned}$$

Solution: The system is not in the form securing convergence of either Jacobi or Gauss–Seidel method as suggested by Theorem 1.2. Nor can it be rearranged so that this convergence test is passed. However, we can multiply the system as suggested by Theorem 1.3. To be more specific, if we denote the system as

$\mathbf{Ax} = \mathbf{b}$, then the Gauss–Seidel method will converge for the system $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$, i.e.

$$\begin{aligned} 50x_1 + 12x_2 - 12x_3 &= 40.5 \\ 12x_1 + 57x_2 - 47x_3 &= 35.75 \\ -12x_1 - 47x_2 + 43x_3 &= -25.75 \end{aligned}$$

We get $\mathbf{x}^{(1)} = (0.8100; 0.4567; 0.1264)$ and $\mathbf{x}^{(2)} = (0.7307; 0.5775; 0.2364)$. The exact solution of the system is $\mathbf{x} = (0.75; 1.5; 1.25)$.

4.

$$\begin{aligned} 2.95x_1 + 3.84x_2 + 2.54x_3 &= 8.75 \\ 4.56x_1 + 3.95x_2 + 2.83x_3 &= 5.57 \\ 7.78x_1 + 4.17x_2 + 3.41x_3 &= -0.79 \end{aligned}$$

Solution: The system is not in the form securing convergence of either Jacobi or Gauss–Seidel method as suggested by Theorem 1.2. Nor can it be rearranged so that this convergence test is passed. However, we can multiply the system as suggested by Theorem 1.3. To be more specific, if we denote the system as $\mathbf{Ax} = \mathbf{b}$, then the Gauss–Seidel method will converge for the system $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$, i.e.

$$\begin{aligned} 90.0245x_1 + 61.7826x_2 + 46.9276x_3 &= 45.0655 \\ 61.7826x_1 + 47.7370x_2 + 35.1518x_3 &= 52.3072 \\ 46.9276x_1 + 35.1518x_2 + 26.0886x_3 &= 35.2942 \end{aligned}$$

We get $\mathbf{x}^{(1)} = (0.5006; 0.4479; -0.1510)$ a $\mathbf{x}^{(2)} = (0.2720; 0.8550; -0.2883)$.
However, there is a crucial problem with the system – it has infinitely many solutions! The third equation was constructed as $(-1) \cdot (2\cdot$ first equation $-3\cdot$ second equation).
Also, notice the most problematic behaviour of both Maple and MATLAB attempting to solve this particular system. When applying the Gauss–Jordan elimination, i.e. forward and backward Gaussian elimination, to the original system, we get that the solution of the system is $\mathbf{x} = (-6.5211; -12.5554; 30)$. However, when applying the Gauss–Jordan elimination to $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$, we get that the solution of the system is $\mathbf{x} = (-2.2489; 4.0063; 0)$. This is also a result given by MATLAB. When computing the determinant of $\mathbf{A}$, we get 0., i.e. "numerical zero", in Maple and `-1.9511e-015` in MATLAB. When attempting to solve the system in MATLAB using the inverse matrix, i.e. we get a warning that

```
Matrix is close to singular or badly scaled.
        Results may be inaccurate.
```

Maple does not give any warning for its `gaussjord` command. Yet when using `gausselim`, i.e. the first half of `gaussjord`, we may notice that something is strange, as the last line of the matrix, i.e. line from which $x_3 = 30$ was obtained, reads $0 \quad 0 \quad -0.1\ 10^{-9} \quad -0.3\ 10^{-8}$.
Regard this as an example of neccessity to verify assumptions of every theorem. In our case, the statement of Theorem 1.3 is valid for regular matrices only – yet our matrix is singular!

5.

$$\begin{aligned} 10x_1 + 2x_2 - 3x_3 &= -0.5 \\ 7x_1 - 14x_2 - 3x_3 &= -5.6 \\ x_1 + 4x_2 + 33x_3 &= 24.4 \end{aligned}$$

Solution: Thanks to Theorem 1.2 we may use both Jacobi and Gauss–Seidel method.
The Jacobi method gives the following results: $\mathbf{x}^{(1)} = (-0.0500; 0.4000; 0.7394)$, $\mathbf{x}^{(2)} = (0.0918; 0.2166; 0.6924)$, while the Gauss-Seidel methods gives $\mathbf{x}^{(1)} = (-$ $-0.0500; 0.3750; 0.6955)$ and $\mathbf{x}^{(2)} = (0.0836; 0.2928; 0.7014)$.
The exact solution of the system is $\mathbf{x} = (0.1; 0.3; 0.7)$.

In the remaining exercises choose a suitable iteration method and find the solution with precision $\varepsilon = 0.01$. The meaning of "finding the solution with precision ε" is described in section 1.2.3.

As suggested in Theorem 1.2, you may use an arbitrary initial approximation. The results below were obtained for $\mathbf{x}^{(0)} = (0; 0; 0)$. If you use a different initial approximation, your results (and especially the number of steps) might vary.

6.

$$\begin{aligned} 50x_1 + 2x_2 + 4x_3 &= 39 \\ 5x_1 + 50x_2 + 8x_3 &= 46 \\ 3x_1 + 2x_2 + 40x_3 &= 18 \end{aligned}$$

Solution: In the 4^{th} step of Jacobi's method we get $\mathbf{x}^{(4)} = (0.72; 0.79; 0.36)$.

7.

$$\begin{aligned} 2x_1 + 4x_2 + 5x_3 &= 20 \\ 5x_1 + 2x_2 + 4x_3 &= 30 \\ 4x_1 + 5x_2 + 2x_3 &= 40 \end{aligned}$$

Solution: Only Gauss–Seidel method may be used to solve the system (application of Theorem 1.3 is neccessary.) In the 20^{th} step of the method we get

$\mathbf{x}^{(4)} = (5.58; 4.18; -1.58)$. *However, if we proceed further with the calculations, we get slightly different results. Similarly, when applying the* `gaussjord` *command in Maple, we get the solution rounded off to two decimal points as* $\mathbf{x} \doteq (5.59; 4.16; -1.56)$.

8.

$$\begin{aligned} 3.5x_1 + 2.9x_2 + 5.8x_3 &= 6.4 \\ 2.7x_1 + 3.1x_2 + 1.8x_3 &= 5.8 \\ 4.3x_1 - 3.9x_2 - 8.2x_3 &= 0.4 \end{aligned}$$

Solution: The exact solution of the system is $\mathbf{x} = (1, 1, 0)$. Since assumptions of Theorem 1.2 are not (and cannot be) fulfilled, only the Gauss–Seidel method may be used to find it. After we apply Theorem 1.3, we get

$$\begin{aligned} 38.03x_1 + 1.75x_2 - 10.10x_3 &= 39.78 \\ 1.75x_1 + 33.23x_2 + 54.38x_3 &= 34.98 \\ -10.10x_1 + 54.38x_2 + 104.12x_3 &= 44.28 \end{aligned}$$

which will give us the solution already in the 3^{rd} step of the method. *In this exercise consider multiplying each equation of the orginal system by* 10 *for easier calculations.*

9.

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 2.58 \\ 2x_1 - 3x_2 - 4x_3 &= 7.49 \\ 3_1 - x_2 + 2x_3 &= 5.63 \end{aligned}$$

Solution: Since assumptions of Theorem 1.2 are not (and cannot be) fulfilled, only the Gauss–Seidel method may be used to find it. After we apply Theorem 1.3, we get

$$\begin{aligned} 14x_1 - 7x_2 + 1x_3 &= 34.45 \\ -7x_1 + 14x_2 + 16x_3 &= -22.94 \\ x_1 + 16x_2 + 29x_3 &= -10.96 \end{aligned}$$

In the 38^{th} step of the Gauss–Seidel method we get $\mathbf{x}^{(38)} = (3.39; 1.66; -1.41)$ *However, the exact solution of the system is (rounded off to 4 decimal digits)* $\mathbf{x} = (3.4571; 1.7814; -1.48)$.

10.

$$\begin{aligned} 7.35x_1 - 2.875x_2 - 1.25x_3 &= -1.075 \\ 1.25x_1 + 4.875x_2 + 2.9x_3 &= 9.85 \\ 8.6x_1 + 2x_2 + 1.65x_3 &= 3.29 \end{aligned}$$

Solution: The system has no solution.

1.4 Related issues

In all the above exercises the size of the systems is relatively small. In most cases there is $n = 3$ and it never exceeds 5. However, in real-life situations we work with systems of hundreds or thousands of equations. This has two important implications:

1. We choose from a wide variety of methods, not only Jacobi or Gauss-Seidel. Often, we adjust the two above mentioned methods so that they could converge faster – we use e.g. succesive over-relaxation.

2. If we stick to using the Jacobi or Gauss-Seidel methods (as is relevant for *Mathematics 3*), we soon realize that the practical performing of the above convergence tests or of the algorithm itself has a number of problematic places. This includes:

   (a) deciding whether the system has exactly one solution,

   (b) deciding whether matrix $\mathbf{A}$ is strictly diagonally dominant or rearranging the system in such a way that it is strictly diagonally dominant,

   (c) choosing a suitable initial approximation,

   (d) precision problems resulting from slow convergence,

   (e) the issue of stopping the algorithm after required precision has been reached.

In fact, in *Mathematics 1* you learned that deciding whether the system has exactly one solution is based upon comparing ranks of $\mathbf{A}$ and $\mathbf{A}|\mathbf{b}$. Yet how is rank computed? Or rather, once we compute the ranks, is it necessary to apply iteration methods?

Also, suppose we have a large system of linear equations. If it is an arbitrary system, what is the chance that the matrix $\mathbf{A}$ is row or column strictly digonally dominant? What is the chance that the system will be rearrangeable so that it is such as we need? For large and arbitrary systems it is obviously very small. However, Theorem 1.4 states that regularity of a matrix results from its strict diagonal dominance. Or, by definition, regularity of a matrix means that the matrix has a non-zero determinant. Yet how does one compute it for large systems?

Furhtermore, given an arbitrary system, how can we estimate a reasonable initial condition? Does the choice of the initial condition influence the speed of convergence? Is the speed of convergence important? Notice that we have inaccuracies in every step of every method, which means that we naturally want to perform as small number of steps as possible.

Notice also that stopping the algorithm by

$$\| \mathbf{x}^{(r)} - \mathbf{x}^{(r-1)} \| < \varepsilon,$$

does not mean that we have reached the required precision. All we get when this condition holds, is that the increments of approximations are sufficiently small. However, they can be sufficiently small for time long enough for their sum to become great enough.

Finally, we must be aware of ill-conditioned systems, i.e. systems in which small changes of input data (in our case coefficients of the system) result in great changes of output data (in our case values of $\mathbf{x}$).

1.5 Solving linear systems using Matlab

This section of the text goes beyond the subject matter discussed in this chapter as when using professional mathematical sofwtare methods far more advanced then the basic versions of Jacobi or Gauss-Seidel methods are used. However, it is included for the reader to have at least a basic insight in how the topic is treated in MATLAB.[1]

Using MATLAB solution of a system of linear equations may be found in a number of ways. As a starting point choose the help entry (depending on MATLAB version yet typically called) `User Guide` > `Mathematics` > `Linear Algebra` > `Systems of Linear Equations`, resp. `Functions` > `Mathematics` > `Linear Algebra` > `Linear equations`. Here, `linslove`, the parameters of which are matrix $\mathbf{A}$ and vector $\mathbf{b}$ if the task is set as $\mathbf{Ax} = \mathbf{b}$, is the most often used command. You may also use commands such as `mldivide` or commands for computing the matrix inverse such as `inv` or `pinv`. However, it is always necessary to know the exact documentation of the respective command (so that you know what exactly the command performs). One must also verify validity of all specific conditions and properties of both the task and the matrix $\mathbf{A}$ and vector $\mathbf{b}$.

When using the Gauss-Seidel method transposing matrices may be necessary. This is done using the `transpose` command.

Sparse matrices are a special case. For these, see the help entry (depending on MATLAB version yet typically) `User Guide` > `Mathematics` > `Linear Algebra` > `Sparse Matrices`. The overview of iterative methods for sparse matrices is included in the help entry typically called `Functions` > `Mathematics` > `Sparse Matrices` > `Linear Equations (Iterative Methods)`

Regardless of the type of matrix and / or vector, the data input must have correct format. Therefore, study how to enter matrices and vectors in MATLAB.

1.6 Exam tasks

Below you can find some authentic exam tasks.

1. We have the following system of linear equations

$$x + y + z = 7$$

[1] We ask the reader to consider this in every corresponding section in the remaining chapters.

$$\begin{aligned} 2x + 3y - z &= 4 \\ ax + 4y &= 11 \end{aligned}$$

Out of options $a = 3$, $a = 5$ choose the one for which the system has exactly one solution. Then verify and / or secure convergence of an iteration method of your choice. Perform 2 steps of the method for initial approximation $\mathbf{x}^{(0)} = (0, 0, 0)$.

2. We have the following system of linear equations

$$\begin{aligned} x + 20y + 2z &= 5 \\ 5x + 2y - 20z &= 9 \\ 10x - 2y + 3z &= 8 \end{aligned}$$

Verify and / or secure convergence of an iteration method of your choice. Perform 2 steps of the method for initial approximation $\mathbf{x}^{(0)} = (0, 0, 0)$.

3. We have the following system of linear equations

$$\begin{aligned} 5x + 2y + 2z &= 5 \\ x + 5y - 3z &= 9 \\ x - 2y + 8z &= 9 \end{aligned}$$

Verify and / or secure convergence of an iteration method of your choice. Perform 2 steps of the method for initial approximation $\mathbf{x}^{(0)} = (0, 0, 0)$. Then compare one of the values $x^{(2)}$, $y^{(2)}$, $z^{(2)}$ with its exact value obtained using a direct method of your choice.

4. We have vectors $\vec{u} = (1, 2, 3)$, $\vec{v} = (3, 2, 1)$, $\vec{w} = (2, 1, 3)$. Find the first approximation of coefficients a, b, c such that

$$a\vec{u} + b\vec{v} + c\vec{w} = \vec{k},$$

where $\vec{k} = (3, 5, 8)$. Choose $(0, 0, 0)$ for the initial approximation.

5. We have the following system of linear equations

$$\begin{aligned} 5x + 2y + 2z &= 5 \\ x + 5y - 3z &= 6 \\ x - 2y + 8z &= 7 \end{aligned}$$

Verify and / or secure convergence of the Gauss–Seidel method. Perform 3 steps of the method for initial approximation $\mathbf{x}^{(0)} = (0, 0, 0)$.

1.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, MATLAB Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on MATLAB Compiler Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that MATLAB starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of MATLAB output) as far as indexing is concerned.

1. Numerical solution of the system of linear equations

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Calculating the determinant
2. Analytical solution of the system of linear equations
3. Multiplication of matrices

2 One nonlinear equation

2.1 Wording the problem

We are looking for all points $\zeta \in \mathbb{R}$ such that $f(\zeta) = 0$, where f is a real function of one real variable x other than $f(x) = ax + b$.

2.2 A brief summary of the subject matter

2.2.1 Methods of calculation – general discussion

The requirement of finding solutions of nonlinear equations is not a new one as you have already solved such tasks at the secondary school level. However, you have always dealt with *specific* tasks: trigonometric equations, logarithmic equations, cubic equations, etc. The techniques of solution you have used were analytic ones (you obtained precise results) and were tailored to the type of equation.

In *Mathematics 3* we are going to use methods applicable (in general) on any type of equation without need of any further classification. However, the result we obtain will be (in general) inaccurate.

There will be a crucial difference between the analytic secondary school level approach and the new numerical one. With the analytic approach we start with – often neglected – discussion of domain of f (which very often equals $\mathbb{R}$). Then we solve the equation using a type-tailored method. Finally, we check whether the solution belongs to the domain of f. However, in the numerical approach we first of all have to roughly localize the solution and then either narrow the respective interval or approach the solution using iteration.

The methods of calculation we use in *Mathematics 3* divide into two groups: *narrowing the interval* or *iteration from a given estimate*.

2.2.2 Root isolation (or localizing the solution)

Various strategies may be used to localize the solution of $f(x) = 0$. In *Mathematics 3* we suggest two simple ones.

Theorem 2.1. *If the function f is continuous on the interval $I = \langle a, b\rangle$ and if $f(a)$ and $f(b)$ have opposite signs, i.e. if*

$$f(a) \cdot f(b) < 0, \tag{2.1}$$

then the equation $f(x) = 0$ has at least one solution in I.

Or we can transform the equation $f(x) = 0$ to $f_1(x) = f_2(x)$ which transforms the task of finding the intersection of $f(x)$ and the x–axis to finding the intersection of $f_1(x)$ and $f_2(x)$. In many cases plotting or sketching $f_1(x)$, $f_2(x)$ might be easier or more illustrating than plotting $f(x)$.

2.2.3 Methods of calculation – formulas

The **bisection method** takes an interval, divides it in halves and chooses the half with the solution to be halved again. The k–th interval is denoted as $\langle a_k, b_k\rangle$, its midpoint is denoted as x_k. Obviously,

$$x_k = \frac{a_k + b_k}{2}. \tag{2.2}$$

The **regula falsi method** is similar to bisection, yet intervals are not split in halves. For the k–th interval $\langle a_k, b_k\rangle$ we regard the splitting point to be

$$x_k = b_k - \frac{b_k - a_k}{f(b_k) - f(a_k)}\, f(b_k). \tag{2.3}$$

The **Newton's method** does not divide the interval in which a solution has been localized. Instead it chooses a starting point (initial approximation x_0) and using the formula

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{2.4}$$

aims at converging to the exact solution.

The **simple fixed–point iteration** does not divide the interval either. We transform the equation $f(x) = 0$ to $x = g(x)$ and apply the *Banach fixed-point theorem* because in a favourable context the limit of the sequence

$$x_{k+1} = g(x_k) \tag{2.5}$$

is the exact solution of $f(x) = 0$.

2.2.4 Methods of calculation – convergence

Furhter on, for all methods, we suppose that $I = \langle a, b\rangle$ is such an interval that $f(x) = 0$ has *exactly one* solution on it.

Theorem 2.2. *If $f(x)$ is continuous on I, then the bisection method always converges to the exact solution.*

Theorem 2.3. *If $f(x)$ is continuous on I, then the regula falsi method always converges to the exact solution.*

The issue of convergence of the Newton's method is rather complex. In *Mathematics 3* we use the following condition to test it.

Theorem 2.4. *If simultaneously*

- *$f(x)$ has continuous first derivative $f'(x)$ and continuous second derivative $f''(x)$ on I,*
- *$f'(x) \neq 0$ for all $x \in I$,*
- *the sign of $f''(x)$ does not change on I,*

and we choose initial approximation $x_0 \in I$ such that

- *$f(x_0) \cdot f''(x_0) > 0$,*

then the Newton's method will converge to the exact solution.

When testing convergence of the simple iteration method we should verify assumptions of the *Banach fixed-point theorem*. Since this is often not easy, we use the following equivalent convergence test for the equation transformed into $x = g(x)$.

Theorem 2.5. *Suppose that $g(x)$ maps the interval I onto iteself and that it has a derivative there. If there exists $\alpha \in \langle 0, 1)$ such that*

$$|\, g'(x)| \leq \alpha \quad \forall x \in \langle a, b\rangle \ , \tag{2.6}$$

then on I there exists a fixed point ξ of $g(x)$ and the sequence of approximations (2.5) converges to it for an arbitrary initial approximation $x_0 \in \langle a, b\rangle$.

2.2.5 Stopping the algorithms

Suppose we want to work with precision ε. It is obvious that when narrowing an interval shorter than ε using the bisection or regular falsi method, we always get the solution with the required precision. Naturally, provided we narrow such an interval that $f(x)$ is continuous on it.

However, the issue is more complicated with the other two methods. Just as was the case of systems of linear equations in section 1.2.3, we – for reasons of simplicity in this introductory course – stop the algorithm when the increments of approximations are small enough, i.e. when $|x_k - x_{k-1}| < \varepsilon$ for some $k \in \mathbb{N}$. However, this does not necessarily mean that x_k is the solution with the required precision. In real-life situations we may also set the number of approximations before starting the calculation. When this number is exceeded for a certain initial approximation, we may try another one.

2.3 Exercises

2.3.1 Prior to starting the algorithms

1. Give an example of a function and an interval such that the function has more than one solution on it. Then shift the interval (do not change its length) so that the function has no solution on it. Then shift it again so that the function has exactly one solution on it. Try for various types of elementary functions and their linear combinations and compositions.

2. Give an example of a function which has no solution. Try for various types of elementary functions and their linear combinations and compositions.

3. Give an example of a function and an interval for which the test in Theorem 2.1 may not be used.

4. Give an example of a function and an interval for which the test in Theorem 2.1 can be used.

5. Give an example of a function and an interval for which the test after Theorem 2.1 is useful for a person having access to pencil and paper only.

6. Give an example of a function and an interval for which the test in Theorem 2.1 is of no use for a person having access to pencil and paper only.

7. Give an example of a function and an interval for which both the bisection method and the regula falsi method fail.

8. Give an example of a function and an interval for which the bisection method fails but the regula falsi method does not.

9. Give an example of a function and an interval for which the Newton's method fails.

10. Give an example of a function $f(x)$ and an interval such that after transforming $f(x) = 0$ to $x = g(x)$ we get exactly two different possible functions $g(x)$ such that exactly one of them will provide convergence of the simple iteration method. Then repeat this for three possible functions $g(x)$ where at least one will provide convergence.

Test yourself in plotting graphs of elementary functions using the following maplets: Graphs of some elementary functions, Graphs of functions.

2.3.2 Exercising the algorithms: bisection method and regula falsi

Below you will find 10 equations $f(x) = 0$. In this section each of them is solved using the bisection method and regula falsi. In section 2.3.3 the same equations (under the same numbers) are solved using the Newton's method and the simple fixed–point iteration.

Using the bisection and regula falsi algorithms find the solution of $f(x) = 0$ *on interval* $I = \langle a, b\rangle$ *with precision* ε. *Use Theorem 2.2 and Theorem 2.3 to test convergence of the methods. In the case of bisection stop the algorithm when bisecting an interval shorter than* 2ε *– in this case denote its midpoint to be the approximate solution of the equation. In the case of regula falsi stop the algorithm when there is* $|x_k - x_{k-1}| < \varepsilon$ *for some* $k \in \mathbb{N}$.

1. $f(x) = \mathrm{e}^x - \cos x - 2$, $a = 0$, $b = 1$, $\varepsilon = 0.01$

   Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. If we plot e.g. $f_1(x) = \mathrm{e}^x$ and $f_2(x) = \cos x + 2$, we see that there is exactly one solution of the equation $f(x) = 0$ on I. Therefore both bisection and regula falsi methods will find the solution.

   Solution using the bisection method: The approximations are: $x_0 = 0.5$, $x_1 = = 0.75$, $x_2 = 0.8750$, $x_3 = 0.9375$, $x_4 = 0.9688$, $x_5 = 0.9531$, $x_6 = 0.9453$.

   Solution using regula falsi: The approximations are: $x_0 = 0.9183$, $x_1 = 0.9481$, $x_2 = 0.9488$

2. $f(x) = \mathrm{e}^{\frac{2}{3}x} - x^2 - 1$, $a = 0.5$, $b = 1.5$, $\varepsilon = 0.01$

   Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. If we plot e.g. $f_1(x) = \mathrm{e}^{\frac{2}{3}x}$ and $f_2(x) = x^2 + 1$, we see that there is exactly one solution of the equation $f(x) = 0$ on I. Therefore both bisection and regula falsi methods will find the solution. Naturally, it is rather difficult to sketch $f_1(x) = \mathrm{e}^{\frac{2}{3}x}$ in pencil yet notice that the sketch need not be accurate to suggest that there is only one intersection of the parabola and exponential on I.

   Solution using the bisection method: The approximations are: $x_0 = 1$, $x_1 = = 0.75$, $x_2 = 0.8750$, $x_3 = 0.9375$, $x_4 = 0.9063$, $x_5 = 0.9219$, $x_6 = 0.9141$.

   Solution using regula falsi: The approximations are: $x_0 = 0.7150$, $x_1 = 0.8387$, $x_2 = 0.8911$, $x_3 = 0.9103$, $x_4 = 0.9169$

   Question: Can we work on an interval with an endpoint $x = 0$ such as e.g. $\langle 0, 1\rangle$?

   Answer: No, because $x = 0$ is a solution of the equation and the method would stop with finding $x = 0$ as a solution.

   Modification: Change the interval to $I_2 = \langle 0.5; 5\rangle$.

Convergence: There are two solutions of the equation on the interval. The bisection method converges to $x \doteq 0.91$ while regula falsi results diverges. First, we divide $\langle 0.5, 5\rangle$ and then $\langle 0.1526; 5\rangle$. However, for this interval we get the division point $x = -0.0559$, i.e. out the interval. Then the method fails. The function is plotted on $\langle 0.5, 5\rangle$ in Fig. 2.1. Project the geometrical meaning of regula falsi onto the plot to see why this happens.

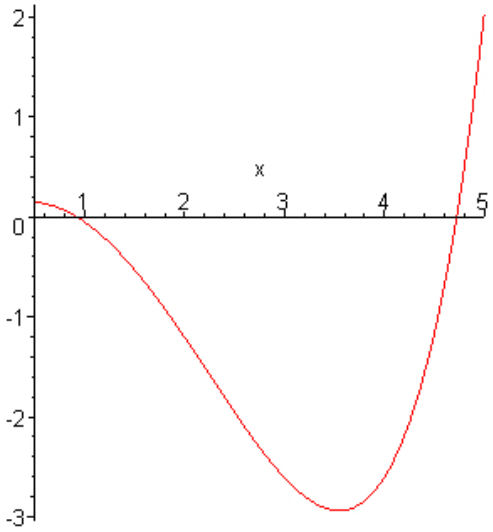


Fig. 2.1: Plot for Exercise 2: Even regula falsi may fail

3. $f(x) = \sin x - x^3 + 1$, $a = 0.1$, $b = 1.5$, $\varepsilon = 0.01$

   Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. If we plot e.g. $f_1(x) = \sin x + 1$ and $f_2(x) = x^3$, we see that there is exactly one solution of the equation $f(x) = 0$ on I. Therefore both bisection and regula falsi methods will find the solution.

   Solution using the bisection method: The approximations are: $x_0 = 0.8$, $x_1 = = 1.15$, $x_2 = 1.325$, $x_3 = 1.2375$, $x_4 = 1.2813$, $x_5 = 1.2594$, $x_6 = 1.2484$, $x_7 = 1.2539$.

   Solution using regula falsi: The approximations are: $x_0 = 0.7212$, $x_1 = 1.0971$, $x_2 = 1.2149$, $x_3 = 1.2419$, $x_4 = 1.2476$.

4. $f(x) = \mathrm{e}^x \cos x + 1$, $a = 0$, $b = 2$, $\varepsilon = 0.01$

   Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. However, sketching graphs of two functions in pencil is impossible here due to the product $\mathrm{e}^x \cos x$. Thus without a computer plot of the function and with the knowledge from

Mathematics 3 only we cannot tell whether there is *exactly one* solution of the equation on I (or whether there are more of them).

Solution using the bisection method: The approximations are: $x_0 = 1$, $x_1 = = 1.5$, $x_2 = 1.75$, $x_3 = 1.625$, $x_4 = 1.6875$, $x_5 = 1.7188$, $x_6 = 1.7344$, $x_7 = = 1.7422$.

Solution using regula falsi: The approximations are: $x_0 = 0.9816$, $x_1 = 1.5364$, $x_2 = 1.7026$, $x_3 = 1.7378$, $x_4 = 1.7446$.

5. $f(x) = \mathrm{e}^x \sin x - 2$, $a = 0$, $b = 1$, $\varepsilon = 0.01$

Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. However, sketching graphs of two functions in pencil is impossible here due to the product $\mathrm{e}^x \sin x$. Thus without a computer plot of the function and with the knowledge from *Mathematics 3* only we cannot tell whether there is *exactly one* solution of the equation on I (or whether there are more of them).

Solution using the bisection method: The approximations are: $x_0 = 0.5$, $x_1 = = 0.75$, $x_2 = 0.875$, $x_3 = 0.9375$, $x_4 = 0.9063$, $x_5 = 0.9219$, $x_6 = 0.9141$.

Solution using regula falsi: The approximations are: $x_0 = 0.8744$, $x_1 = 0.9195$, $x_2 = 0.9210$.

6. $f(x) = e^{x^2} - x - \frac{5}{6}$, $a = 0.5$, $b = 1.5$, $\varepsilon = 0.01$

Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. When sketching the graphs of e.g. $f_1(x) = \mathrm{e}^{x^2}$ and $f_2(x) = x + \frac{5}{6}$ it is rather difficult to decide whether the exponential and the line intersect. However, it is obvious that if they do intersect, then the intersection is on $\langle 0, 1 \rangle$.

Solution using the bisection method: The approximations are: $x_0 = 1$, $x_1 = = 0.75$, $x_2 = 0.625$, $x_3 = 0.5625$, $x_4 = 0.5938$, $x_5 = 0.6094$, $x_6 = 0.6016$.

Solution using regula falsi: The approximations are: $x_0 = 0.5068$, $x_1 = 0.5134$.

Problem: How can such a big difference be explained?

Answer: We must always be careful about the choice of the method. We get the explanation after projecting the geometrical interpretation of the regula falsi method onto the graph of $f(x)$ on I – see Fig. 2.2. In plain words, the function is too steep which causes the increments of approximations to be "too small for too long" with regula falsi. The problem cannot occur with the bisection method, though.

7. $f(x) = \tan x + x - 2$, $a = 0$, $b = 2$, $\varepsilon = 0.01$

Convergence: The function is *not* continuous on I. Therefore, neither Theorem 2.2 nor Theorem 2.3 may be used. There is $f(0) < 0$ as well as $f(2) < 0$ which – if

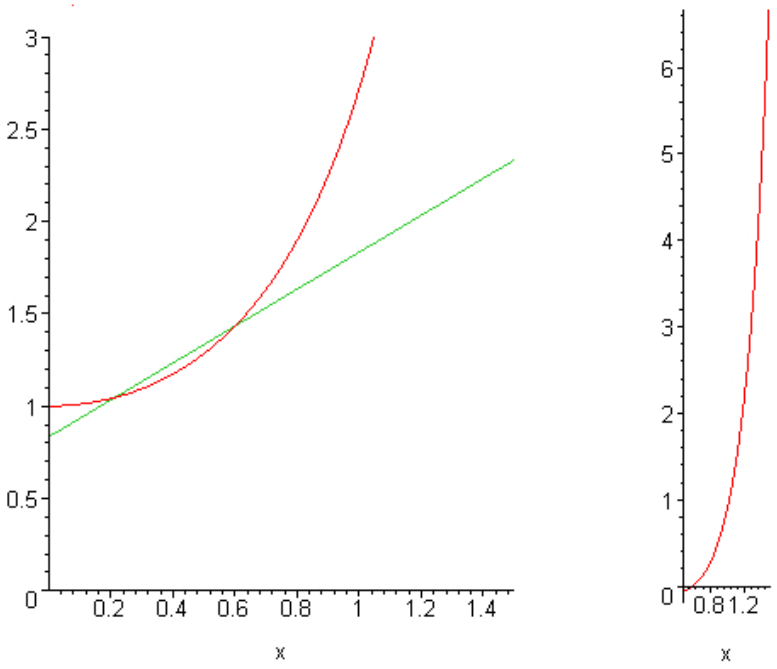


Fig. 2.2: Plots for Exercise 6: Root isolation (left) and $f(x)$ on $I = \langle 0.5; 1.5\rangle$ (right)

the function was continuous on $\langle 0; 2\rangle$ (which is not true!) – would not suggest anything about possible solutions of $f(x) = 0$ on I. However, when we plot the trigonometric function and the line, i.e. $f_1(x) = \tan x$, $f_2(x) = 2 - x$, it is obvious that the equation has infinitely many solutions. One of them is located on $\langle 0; 2\rangle$.

Solution using the bisection method: *Notice that this is risky because convergence is not secured!* The approximations are: $x_0 = 1$, $x_1 = 0.5$, $x_2 = 0.75$, $x_3 = 0.875$, $x_4 = 0.8125$, $x_5 = 0.8438$, $x_6 = 0.8594$, $x_7 = 0.8516$.

Solution using regula falsi: *Notice that this is risky because convergence is not secured!* The approximations are: $x_0 = -21.6170$, $x_1 = 4.4528$, $x_2 = -1.0525$, $x_3 = 1.3477$, $x_4 = 0.2948$, $x_5 = 0.5810$, $x_6 = 0.7104$, $x_7 = 0.7758$, $x_8 = 0.8106$, $x_9 = 0.8296$, $x_10 = 0.8401$, $x_11 = 0.8460$.

Important: With regula falsi we finally "somehow" come very close to the exact solution. However, notice that first few steps are rather "wild" as we have stepped out of the interval I. This is naturally something that is not permitted by convergence tests.

8. $f(x) = \ln x + x - 2$, $a = 0.5$, $b = 2$, $\varepsilon = 0.01$

Convergence: The function is continuous on I. There is $f(a) \cdot f(b) < 0$, i.e. the equation $f(x) = 0$ has at least one solution on I. If we plot e.g. $f_1(x) = \ln x$ and $f_2(x) = 2 - x$, we see that there is exactly one solution of the equation

$f(x) = 0$ on I. Therefore both bisection and regula falsi methods will find the solution.

Solution using the bisection method: The approximations are: $x_0 = 1.5$, $x_1 = = 1.625$, $x_2 = 1.4375$, $x_3 = 1.5313$, $x_4 = 1.5781$, $x_5 = 1.5547$, $x_6 = 1.5664$, $x_7 = 1.5605$.

Solution using regula falsi: The approximations are: $x_0 = 1.6398$, $x_1 = 1.5740$, $x_2 = 1.5606$, $x_3 = 1.5579$.

9. $f(x) = x^5 + 2x^4 - 12x^3 + 2x - 10$, $\varepsilon = 0.01$, find the smallest positive solution

   Convergence: The function is a polynomial, i.e. it is continuous on $\mathbb{R}$. First of all we have to identify the interval with the smallest positive solution. Obviously $f(0) = -10$ which means that we may want to look for the smallest number s such that $f(s) > 0$ in order to be able to apply Theorem 2.1. It is to be noted that this strategy need not be good as we have infinitely many points to test. Yet using it, we may narrow the interval to $\langle 0; 3\rangle$ or even to $\langle 2; 3\rangle$. The results below are valid for $I = \langle 2; 3\rangle$.

   Solution using the bisection method: The approximations are: $x_0 = 2.5$, $x_1 = = 2.75$, $x_2 = 2.625$, $x_3 = 2.6875$, $x_4 = 2.6563$, $x_5 = 2.6406$, $x_6 = 2.6484$.

   Solution using regula falsi: The approximations are: $x_0 = 2.3304$, $x_1 = 2.5159$, $x_2 = 2.5954$, $x_3 = 2.6250$, $x_4 = 2.6355$, $x_5 = 2.6390$.

10. $f(x) = \mathrm{e}^{x+1} - 2\cos 3x - 2$, $\varepsilon = 0.01$, find the real solution with the smallest absolute value

    Convergence: The function is continuous on $\mathbb{R}$. Identifying the interval without computer plotting requires accuracy in sketching the graphs of e.g. $f_1(x) = \mathrm{e}^{x+1}$ and $f_2(x) = 2\cos 3x + 2$ as the equation has one solution in $\langle -1; 0\rangle$ and one solution in $\langle 0; 1\rangle$. With a good deal of accuracy we find out that we have to restrict our computations to $I = \langle 0; 1\rangle$.

    Solution using the bisection method: The approximations are: $x_0 = 0.5$, $x_1 = = 0.25$, $x_2 = 0.125$, $x_3 = 0.1875$, $x_4 = 0.2188$, $x_5 = 0.2344$, $x_6 = 0.2422$.

    Solution using regula falsi: The approximations are: $x_0 = 0.1482$, $x_1 = 0.2175$, $x_2 = 0.2392$, $x_3 = 0.2447$.

2.3.3 Exercising the algorithms: Newton's method and simple fixed–point iteration

In this section we are looking for solutions of the same equations $f(x) = 0$ on the same intervals $I = \langle a; b\rangle$ as discussed in section 2.3.2. The numbering of exercises corresponds so that the reader could easily compare the solutions and / or numbers of steps neccessary to achieve the required precision and / or obstacles connected with applying respective methods on the given equations. Required precision in all tasks remains at $\varepsilon = 0.01$;

the algorithms will stop when there is $|x_k - x_{k-1}| < \varepsilon$ for some $k \in \mathbb{N}$. Convergence is tested using Theorem 2.4 for the Newton's method or using Theorem 2.5 in case of simple fixed–point iteration. From section 2.3.2 we already know that there is exactly one solution of $f(x)$ on I.

1. $f(x) = \mathrm{e}^x - \cos x - 2$, $a = 0$, $b = 1$, $\varepsilon = 0.01$

   Convergence of the Newton's method: We have that $f'(x) = \mathrm{e}^x + \sin x$, $f''(x) = \mathrm{e}^x + \cos x$. Obviously, both of these are continuous on I. It is easy to verify that $y_1(x) = \mathrm{e}^x$ and $y_2(x) = \sin x$ do not intersect in I, i.e. the condition $f'(x) \neq 0$ holds on I. Also obviously, $f''(x)$ does not change its sign on I. Testing condition $f(x_0) \cdot f''(x_0) > 0$ is rather problematic on I yet it holds e.g. for $x_0 = 1$. Thus, use this as the initial approximation of the Newton's method.

   Solution using the Newton's method: The approximations are $x_0 = 1$, $x_1 = = 0.9500$, $x_2 = 0.9488$.

   Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. Either $\mathrm{e}^x = 2 + \cos x$, which results in $x = \ln(2 + \cos x)$ or $\cos x = \mathrm{e}^x - 2$, i.e. $x = \arccos(\mathrm{e}^x - 2)$. In the former case the derivative is $g_1'(x) = -\frac{\sin x}{2+\cos x}$. Obviously, the condition stated in 2.5 holds on I. In the latter case we have that $g_2'(x) = -\frac{\mathrm{e}^x}{\sqrt{1-(\mathrm{e}^x-2)^2}}$ and the condition stated in Theorem 2.5 does not hold anywhere on I. Use the same initial approximation as for the Newton's method, i.e. $x_0 = 1$.

   Solution using the simple fixed–point iteration: The approximations are $x_0 = 1$, $x_1 = 0.9540$, $x_2 = 0.9472$.

2. $f(x) = \mathrm{e}^{\frac{2}{3}x} - x^2 - 1$, $a = 0.5$, $b = 1.5$, $\varepsilon = 0,01$

   Convergence of the Newton's method: We have that $f'(x) = \frac{2}{3}\mathrm{e}^{\frac{2}{3}x} - 2x$, $f''(x) = \frac{4}{9}\mathrm{e}^{\frac{2}{3}x} - 2$. Obviously, both of these are continuous on I. Functions $y_1(x) = \frac{2}{3}\mathrm{e}^{\frac{2}{3}x}$ and $y_2(x) = 2x$ have one positive intersection – once we realize this it is not difficult to state that $f'(x) \neq 0$ holds on I. Similarly we find out that $f''(x)$ does not change its sign on I. Condition $f(x_0) \cdot f''(x_0) > 0$ holds e.g. for $x_0 = 1.5$. Thus, use this as the initial approximation of the Newton's method.

   Solution using the Newton's method: The approximations are $x_0 = 1.5$, $x_1 = = 1.0524$, $x_2 = 0.9332$, $x_3 = 0.9204$, $x_5 = 0.9203$.

   Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. Either $\mathrm{e}^{\frac{2}{3}x} = 1 + x^2$, which results in $x = = \frac{3}{2}\ln(1 + x^2)$ or the form which results in $x = \sqrt{\mathrm{e}^{\frac{2}{3}x} - 1}$. In the former case the derivative is $g_1'(x) = \frac{3x}{1+x^2}$ and the condition stated in 2.5 holds nowhere

on I. In the latter case we have that

$$g_2'(x) = \frac{\mathrm{e}^{\frac{2}{3}x}}{3\sqrt{\mathrm{e}^{\frac{2}{3}x} - 1}}$$

and the condition stated in Theorem 2.5 holds on the whole I. Use the same initial approximation as for the Newton's method, i.e. $x_0 = 1$.

Solution using the simple fixed–point iteration: The approximations are $x_0 = 1$, $x_1 = 1.3108$, $x_2 = 1.1816$, $x_3 = 1.0947$, $x_4 = 1.0367$, $x_5 = 0.9980$, $x_6 = 0.9722$, $x_7 = 0.9549$, $x_8 = 0.9435$, $x_9 = 0.9358$.

3. $f(x) = \sin x - x^3 + 1$, $a = 0.1$, $b = 1.5$, $\varepsilon = 0.01$

Convergence of the Newton's method: We have that $f'(x) = \cos x - 3x^2$, $f''(x) = -\sin x - 6x$. Obviously, both of these are continuous on I. Functions $y_1(x) = \cos x$ and $y_2(x) = 3x^2$ have one positive intersection on I. Even with rough sketching in pencil it may be located somewhere in $\langle 0; 1\rangle$ (towards its midpoint). Therefore we adjust I to e.g. $\langle 1; 1.5\rangle$. Functions $y_3 = -\sin x$ and $y_4 = 6x$ have one intersection only – obviously $x = 0$, which is not in I. Thus $f''(x)$ does not change sign on adjusted $I = \langle 1; 1.5\rangle$. Condition $f(x_0) \cdot f''(x_0) > 0$ holds e.g. for $x_0 = 1.5$. Thus, use this as the initial approximation of the Newton's method.

Solution using the Newton's method: The approximations are $x_0 = 1.5$, $x_1 = = 1.2938$, $x_2 = 1.2509$, $x_3 = 1.2491$.

Problem: The interval I had to be adjusted because of $f'(x)$. When ignoring this we for e.g. $x_0 = 0.2$ get $x_{12} = 1.2491$ yet the way to it is rather "wild" as $x_0 = = 0.2$, $x_1 = -1.1844$, $x_3 = -0.7315$, $x_4 = 0.1085$, $x_5 = -1.0461$, $x_6 = -0.5863$, $x_7 = 2.6867$, $x_8 = 1.8906$, $x_9 = 1.4550$, $x_{10} = 1.2807$, $x_{11} = 1.2500$. In other words, we jump out of the interval and back in it again almost at random and reach the solution almost by accident. Notice that a well programmed algorithm will not allow jumping out of the interval specified as an input of the method (or at least will comment on this).

Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. One results in $x = \sqrt[3]{1 + \sin(x)}$, the other one in $x = \arcsin(1 - x^3)$. However, we work on $I = \langle 0.1; 1.5\rangle$ which exceeds the domain of definition of $\arcsin x$. Derivative in the former case is

$$g'(x) = \frac{\cos x}{3\sqrt[3]{(1 + \sin x)^2}},$$

for which the condition stated in Theorem 2.5 holds everywhere on I. Use the same initial approximation as for the Newton's method, i.e. $x_0 = 1.5$.

Solution using the simple fixed–point iteration: The approximations are $x_0 = 1.5$, $x_1 = 1.2594$, $x_2 = 1.2497$.

4. $f(x) = \mathrm{e}^x \cos x + 1$, $a = 0$, $b = 2$, $\varepsilon = 0.01$

   Convergence of the Newton's method: We have that $f'(x) = \mathrm{e}^x(\cos x - \sin x)^2$, $f''(x) = -2\mathrm{e}^x \sin x$. Obviously, both of these are continuous on I. When testing whether $f'(x) = 0$ for some $x_p \in I$ we easily find out that $x_p = \frac{\pi}{4}$. Thus we have to adjust I to e.g. $\langle 0.8; 2\rangle$. Obviously, $f''(x)$ does not change sign on I. Condition $f(x_0) \cdot f''(x_0) > 0$ holds e.g. for $x_0 = 2$. Thus, use this as the initial approximation of the Newton's method.

   Solution using the Newton's method: The approximations are $x_0 = 2$, $x_1 = = 1.7881$, $x_3 = 1.7476$, $x_4 = 1.7461$.

   Problem: In Exercise 3 the adjustment of the interval I resulted in "wild approximations", yet the solution was eventually found. Notice that situation is different with this equation as choosing e.g. $x_0 = 0$ results in $x_1 = -2$, $x_2 = -16.1395$, $x_3 = 7701251.963$, etc.

   Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. One results in $x = \ln\left(-\frac{1}{\cos x}\right)$, the other one in $x = \arccos\left(-\frac{1}{\mathrm{e}^x}\right)$. However, I is not included in the real part of domain of definition of $g_1(x) = \ln\left(-\frac{1}{\cos x}\right)$ as the argument of the logarithm is negative on I. Moreover, there is $g_1'(x) = \tan x$ which is not continuous on I yet – with the exception of the discontinuity – defined on I again. Anyway, assumptions of Theorem 2.5 are not fulfilled for $g_1(x)$. Do the discussion concerning the other iteration form yourself. *You will have to restrict the interval.* Use the same initial approximation as for the Newton's method, i.e. $x_0 = 2$.

   Solution using the simple fixed–point iteration: The approximations are $x_0 = 2$, $x_1 = 1.7065$, $x_2 = 1.7533$, $x_3 = 1.7449$.

5. $f(x) = \mathrm{e}^x \sin x - 2$, $a = 0$, $b = 1$, $\varepsilon = 0.01$

   Convergence of the Newton's method: This exercise is similar to Exercise 4. The interval need not be adjusted, though, as when testing on I the condition $f'(x) \neq 0$, we get that $x_p \notin I$. In this exercise e.g. $x_0 = 1$ may be chosen for the initial approximation.

   Solution using the Newton's method: The approximations are $x_0 = 1$, $x_1 = = 0.9235$, $x_3 = 0.9210$.

   Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. One results in $x = \ln\left(\frac{2}{\sin x}\right)$, the other one in $x = \arcsin\left(\frac{2}{\mathrm{e}^x}\right)$. However, both are problematic so that restricting the interval is needed. In the former form we easily find out that we have to restrict the interval so that validity of $\cos x > \sin x$ is provided. In the latter case, finding the adjusted interval is much more complicated. In both case, though, too little remains of the original interval I.

6. $f(x) = e^{x^2} - x - \frac{5}{6}$, $a = 0.5$, $b = 1.5$, $\varepsilon = 0.01$

Convergence of the Newton's method: Thanks to the steepness of $f(x)$ (depicted in Fig. 2.2 in section 2.3.2) and to the fact that $f'(x) = 2xe^{x^2} - 1$, $f''(x) = 2e^{x^2} + 4x^2e^{x^2}$, we may choose an arbitrary x_0 greater than the root as the initial approximation of the Newton's method. Use e.g. $x_0 = 1$.

Solution using the Newton's method: The approximations are $x_0 = 1$, $x_1 = = 0.8005$, $x_2 = 0.6709$, $x_3 = 0.6127$, $x_5 = 0.6005$, $x_6 = 0.6000$.

Remark: However, when choosing e.g. $x_0 = 5$, we obtain the solution with the requested precision in 31 steps. When choosing $x_0 = 10$ (which even though permitted by the convergence test is obvious nonsence!) we get that e.g $x_{99} = = 1.4620$ and $x_{100} = 1.2021$.

Convergence of the simple fixed–point iteration: It is rather obvious that the simple fixed–point iteration is not suitable for solving this particular equation.

7. $f(x) = \tan x + x - 2$, $a = 0$, $b = 2$, $\varepsilon = 0.01$

Convergence of the Newton's method: We have that $f'(x) = \tan^2 x + 2$ and $f''(x) = 2(1 + \tan^2 x)\tan x$. Obviously $f'(x)$ is always positive on I and $f''(x)$ is also positive on I (with the exception of $f''(0) = 0$). From section 2.3.2 we already know that we must be aware of the fact that $f(x)$ is not continuous on I. Yet when regarding all aspects of the problem, we may narrow I to e.g. $\langle 0; 1.5\rangle$ and choose e.g. $x_0 = 1$ for the initial approximation.

Solution using the Newton's method: The approximations are $x_0 = 1$, $x_1 = = 0.8740$, $x_2 = 0.8539$, $x_3 = 0.8535$.

Modification: For $x_0 = 1.5$ we get $x_1 = 1.4323$, $x_3 = 1.3087$, $x_4 = 1.1177$, $x_5 = = 0.9293$, $x_6 = 0.8586$, $x_7 = 0.8536$.

Remark: When ignoring the discontinuity of $f(x)$ on I and choosing e.g. $x_0 = 2$, we get – if lucky – a different solution. For $x_0 = 2$ we get $x_4 = 2.6$ which is a solution of $f(x) = 0$ (yet not the one we wanted to find).

Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. One results in $x = 2 - \tan x$, the other one in $x = \arctan(2 - x)$. Yet the derivative of the former one obviously does not meet the condition stated in Theorem 2.5. The derivative of $g_2(x) = \arctan(2 - x)$ is $g_2'(x) = \frac{1}{1+(2-x)^2}$, for which the condition in Theorem 2.5 holds on the whole of I. Use the same initial approximation as for the Newton's method, i.e. $x_0 = 1.5$.

Solution using the simple fixed–point iteration: The approximations are $x_0 = 1.5$, $x_1 = 0.4636$, $x_2 = 0.9938$, $x_3 = 0.7885$, $x_4 = 0.8807$, $x_5 = 0.8416$, $x_6 = 0.8587$.

8. $f(x) = \ln x + x - 2$, $a = 0.5$, $b = 2$, $\varepsilon = 0.01$

Convergence of the Newton's method: We have that $f'(x) = \frac{1}{x} + 1$ and $f''(x) = -\frac{1}{x^2}$. Sketching $f(x), f'(x), f''(x)$ is rather easy here – see Fig. 2.3. We see that we may choose e.g. $x_0 = 0.5$.

Solution using the Newton's method: The approximations are $x_0 = 0.5$, $x_1 = = 1.2310$, $x_2 = 1.5406$, $x_3 = 1.5571$, $x_4 = 1.5571$.

Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. One results in $x = 2 - \ln x$, the other one in $x = \mathrm{e}^{(2-x)}$. While the latter one may obviously not be used in the former one we have $g_1'(x) = -\frac{1}{x}$ which means that we have to restrict I to $(1; 2\rangle$. Use e.g. $x_0 = 2$ as the inital approximation.

Solution using the simple fixed–point iteration: The approximations are $x_0 = 2$, $x_1 = 1.3069$, $x_2 = 1.7324$, $x_3 = 1.4505$, $\ldots x_9 = 1.5621$, $x_{10} = 1.5540$.

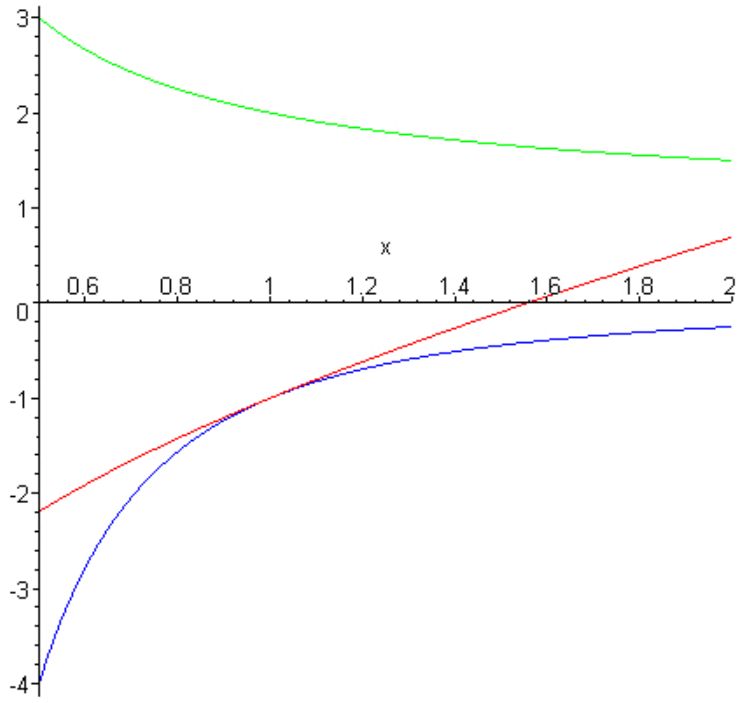


Fig. 2.3: Plots for Exercise 8: $f(x)$ (red), $f'(x)$ (green), $f''(x)$ (blue)

9. $f(x) = x^5 + 2x^4 - 12x^3 + 2x - 10$, $a = 2$, $b = 3$, $\varepsilon = 0.01$

Convergence of the Newton's method: Computing the derivatives is trivial. It is also trivial to find out that both of them are continuous and positive on I. Since e.g. $f(3)$ is also positive, we may choose $x_0 = 3$ as the initial approximation.

Solution using the Newton's method: The approximations are $x_0 = 3$, $x_1 = = 2.7425$, $x_3 = 2.6518$, $x_4 = 2.6410$, $x_5 = 2.6409$.

Convergence of the simple fixed–point iteration: There are too many options for transforming the equation to the form $x = g(x)$. We do not include the discussion here.

10. $f(x) = \mathrm{e}^{x+1} - 2\cos 3x - 2$, $a = 0$, $b = 1$, $\varepsilon = 0.01$

Convergence of the Newton's method: Already in section 2.3.2 we have seen that solving this exercises without computer plotting is problematic. The same holds for the Newton's method since $f'(x) = \mathrm{e}^{x+1} + 6\sin 3x$ and $f''(x) = \mathrm{e}^{x+1} + +18\cos 3x$. All the three functions are depicted in Fig. 2.4. We see that a good choice of the initial approximation is e.g. $x_0 = 0.5$.

Solution using the Newton's method: The approximations are $x_0 = 0.5$, $x_1 = = 0.2764$, $x_3 = 0.2473$, $x_4 = 0.2464$.

Convergence of the simple fixed–point iteration: When adjusting $f(x) = 0$ to $x = g(x)$ we have two options. One results in $x = -\frac{6\sin 3x}{2+2\cos 3x}$. This does not meet the condition of Theorem 2.5. The other one results in $x = \frac{1}{3}\arccos(\mathrm{e}^{x+1} - 1)$. This is problematic due to the domain of definition of $y = \arccos x$.

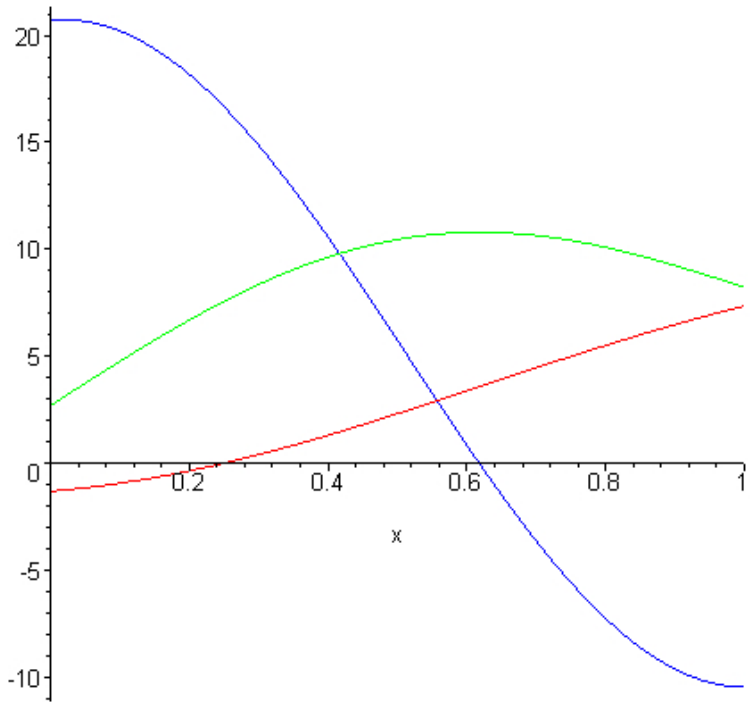


Fig. 2.4: Plots for Exercise 10: $f(x)$ (red), $f'(x)$ (green), $f''(x)$ (blue)

2.4 Related issues

In this section we should first of all make clear that *Mathematics 3* is an *introductory course* only. Therefore, all issues mentioned below will be set within the framework of the subject matter included in other teaching materials.

When facing the problem of finding a solution of a nonlinear equation, we first of all have to localize the solution we are interested in. Yet how exactly is this done? Regard an arbitrary function – "invent" your own one as an arbitrary combination of e.g. four elementary functions including operations such as composition of functions – and try to localize some of its solutions. You will soon realize that in a general case you know neither anything about the solutions nor – which is even worse – can you seek help with the computer. In most real-life situations the localization of the solution follows from the nature of the engineering task, i.e. from areas beyond mathematics.

In the convergence tests we have used we must decide whether a function is continuous on an interval. Obviously, this is not easy for an arbitrary function. Knowledge of elementary functions will help immensely. It must be poited out that testing continuity of a function on an interval is impossible in a pointwise manner. Yet it is pointwise manner that is likely to be performed by computer software. Naturally, the user will be prompted and warned but it will be the user that will have to decide.

There is an issue even more fundamental than contuity of a function – its domain of definition. How do we establish it? Can we – without further considerations – rely on the computer output and take it for granted? In Fig. 2.5 we consider three simple non-linear equations: $|\sqrt{1-x}|-1=0$, $\left(\sqrt{2-x}\right)^2-1=0$ and $|\ln(1-x)|-1=0$ and let Maple 12 plot the respective functions on $\langle -2,5;2{,}5\rangle$. Are the plots relevant or irrelevant? Correct or incorrect? Notice that the explanation is very simple here. However, will you notice for a more complicated function?

Finally, notice that the convergence tests we use are *examples* of convergence tests only. Moreover, they are implications only which means that the fact that a certain function does not meet their assumptions does not imply that the method diverges and cannot be used. All we know in such cases is that the behaviour of the method is unpredictable. In Theorem 2.4 we test existence and continuity of certain functions – see above for possible issues. In Theorem 2.5 we test values of a certain function on an interval. How exactly is this performed? And finally, all tests included in section 2.2.4 assume that we work on an interval where there is *exactly one* solution. Keep in mind that Theorem 2.1 speaks about *at least one* solution, which is not the same.

2.5 Solving nonlinear equations using Matlab

In a general case of $f(x)$ we use mainly the `fzero` command, the input of which is a user defined function. The solution is searched using an algorithm which combines the bisection and secant methods on interval $\langle a,b\rangle$ which is assumed to be such that $f(x)$ is contiuous on it. The command is included in a specialized `Optimization Toolbox`.

In a special case when $f(x)$ is a polynomial, the command `roots` may be used. The polynomial roots are calculated using eigenvalues of a certain special matrix (again using numerical methods).

Function $f(x)$ may be entered using a process described under the help entry typically

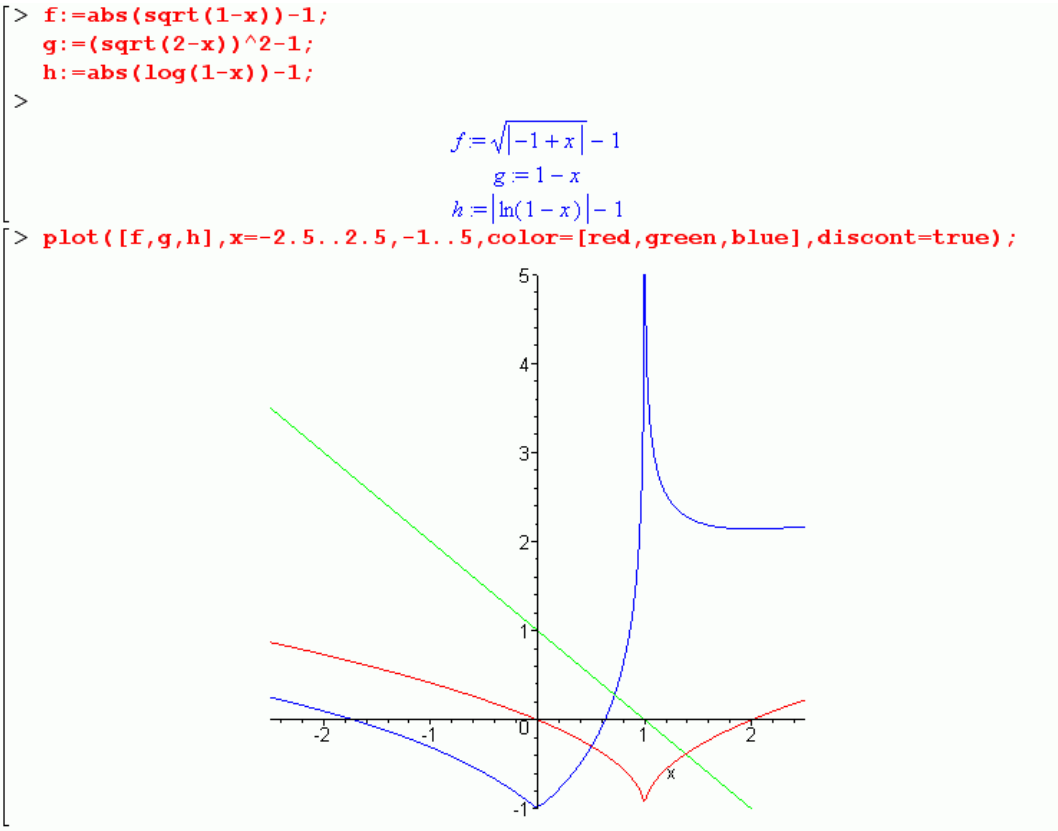


Fig. 2.5: Caution must be exercised with computer plots. Check whether these Maple plots are correct or incorrect (or under what conditions they are correct).

called `User Guide` > `Mathematics` > `Function Handles` (depending on the version of MATLAB). Symbolic variables included in `Symbolic Math` Toolbox (more precisely, the `sym` command) may be used as well.

In case we want to plot the functions, `plot` command or techniques described in `Getting Started` > `Graphics`, resp. `User Guide` > `Graphics`, resp. `Functions` > `Graphics` help entries (precise location and naming depends on the version of MATLAB) may be used. However, using the built-in `optimtool` is a better option.

Should you use differentiation (using the `diff` command), caution must be exercised because depending on the command input either a derivative or a difference may be obtained.

2.6 Exam tasks

Below you can find some authentic exam tasks.

1. We have two functions: $f(x) = \cos x + x^2 - 3$, $g(x) = \ln(x-1) - x^2 + 4$. They both have at least one positive zero point smaller than $x = 2$. Find the smallest of these using one of the methods that have been discussed in the subject. Work with precision $\varepsilon = 0.1$. If you choose a method which might converge as well as diverge on the interval you work with, secure convergence of the method by verifying all necessary criteria.

2. Solve $10\mathrm{e}^{x-2} + \sin 3x - 3 = 0$ using a method of your choice on the interval $I = \langle 0, 1\rangle$. If you choose a method which might converge as well as diverge on the given interval, secure convergence of the method by verifying all necessary criteria.

3. Let a be the number of letters of your first name, b the number of letters of your surname and c the number of the current month. Construct the polynomial which has exactly three roots a, b, c. Prove (other than by making a reference to the above request) that either a or b or c is the zero point of the polynomial. Then perform two steps of the Newton's method for such an initial approximation that will successfully lead to finding the zero point of your choice.

4. Regard the equation $\mathrm{e}^x + ax + b = 0$. Choose a, b so that the equation has exactly one negative solution. Then find this solution with precision $\varepsilon = 0.1$ using either the Newton's method or the simple fixed–point iteration. Secure convergence of the method by verifying all necessary criteria.

5. Find an arbitrary solution of $\ln(x-3) + (x-2)^2 - 3 = 0$ using one of the methods that have been discussed in the subject. Work with precision $\varepsilon = 0.1$. If you choose a method which might converge as well as diverge on the interval you work with, secure convergence of the method by verifying all necessary criteria.

2.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, Matlab Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on Matlab Compiler Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that Matlab starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of Matlab output) as far as indexing is concerned.

1. Numerical methods of solution of one nonlinear equation

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Calculating function values
2. Graphs of some elementary functions
3. Graphs of functions

3 Systems of nonlinear equations

3.1 Wording the problem

We are going to solve systems of n nonlinear equations of n unknowns $x_1, x_2, \ldots, x_n$, where by "nonlinear equations" we mean equations studied in chapter 2. The system will be denoted as

$$\begin{aligned} f_1(x_1, \ldots x_n) &= 0 \\ f_2(x_1, \ldots x_n) &= 0 \\ &\ldots \\ f_n(x_1, \ldots x_n) &= 0 \end{aligned} \tag{3.1}$$

or using the vector notation $\mathbf{F}(\mathbf{x}) = \mathbf{o}$ where $\mathbf{F} = (f_1, \ldots, f_n)^T$, $\mathbf{x} = (x_1, \ldots, x_n)^T$ is a vector of unknowns and $\mathbf{o}$ is a zero vector.

3.2 A brief summary of the subject matter

3.2.1 Methods of calculation

Calculating the solution of systems (3.1) is rather easy in cases when the solution of the system may be interpreted as the intersection of a line and a conic section or a surface. However, this is not true in a general case.

In *Mathematics 3* we discuss two algorithms, both of which are used for solving one nonlinear equation and adjusted for solving systems of nonlinear equations: the **Newton's method** and the **simple fixed-point iteration**.

When using the **Newton's method** we first compute partial derivatives of each function f_i, $i = 1, \ldots, n$, with respect to each variable x_i, $i = 1, \ldots, n$. These partial derivatives

form matrix

$$\mathbf{F}' = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ & & \ddots & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \tag{3.2}$$

We choose initial approximation $\mathbf{x}^{(0)} = \left(x_1^{(0)}, \dots, x_n^{(0)}\right)^T$ and in every step of the method solve the linear system

$$\mathbf{F}'(\mathbf{x}^{(k)}) \cdot \boldsymbol{\delta}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)}) \ , \tag{3.3}$$

where $\boldsymbol{\delta}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = (\delta_1^{(k)}, \dots, \delta_n^{(k)})^T$ and the matrix on the left-hand side of (3.3) is matrix (3.2) into which the initial approximation has been substituted and the matrix on the right-side of (3.3) is formed by the equations in (3.1) into which the initial approximation has been substituted.

When using the **simple fixed–point iteration**, we rewrite the system (3.1) into

$$\begin{aligned} x_1 &= g_1(x_1, x_2, \dots, x_n) \\ x_2 &= g_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ x_n &= g_n(x_1, x_2, \dots, x_n) \end{aligned} \tag{3.4}$$

and look for the fixed point of this system. Respecting the convergence criteria we choose an initial approximation $\mathbf{x}^{(0)}$ and compute further approximations using

$$\mathbf{x}^{(k+1)} = \mathbf{G}(\mathbf{x}^{(k)}). \tag{3.5}$$

3.2.2 Prior to starting the algorithms

Generally speaking, the convergence tests for both Newton's method and the simple fixed-point iteration are so complicated that they are beyond capacities of *Mathematics 3*.

However, there still remains the issue of localizng the solution. If possible, plotting the functions helps. However, we must be aware of the geometrical interpretation of the solution. In the case of two equations we look for the intersection of two planar curves. In the case of three equations we look for the intersection of three surfaces. Yet such a geometrical interpretation fails for more equations.

Therefore, for reasons of simplicity, we distinguish two cases in this introductory course: the initial approximation is either given or students are asked to find it (for two equations) when plotting the planar curves is easy.

3.2.3 Stopping the algorithms

Similarly to the case of one nonlinear equation, we stop the algorithm when the increments of approximations are small enough for all unknowns, i.e. when

$$\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \| < \varepsilon \quad \text{, i.e.} \quad \| \boldsymbol{\delta}^{(k-1)} \| < \varepsilon$$

where $\| \cdot \|$ denotes a suitable vector norm. However, we often equal $\| \cdot \|$ with absolute value.

In real-life situations we may also set the number of approximations before starting the calculation. When the number is exceeded for a certain initial approximation, we may try another one.

3.3 Exercises

3.3.1 Prior to starting the algorithms

1. Recall the equations of the conic sections you know from your secondary school studies, i.e. ellipse, parabola, hyperbola and circle. Change parameters of the equations or shift the conic sections.

2. Find partial derivatives of the following functions with respect to all variables:

   (a) $z(x, y) = e^{2x+3y} + \sin(4x + 5y) - 1$

   (b) $z(x, y) = \tan x^2 + y^2 + xy - 2$

   (c) $z(x, y) = \ln \frac{x+3}{y-4} - 3$

   (d) $z(x, y) = \frac{x^3 + \sin x^2}{\cos(y^5 - 4)}$

   (e) $z(x, y) = (x - 1)^2 + (y - 5)^3$

   (f) $z(x, y) = 2^{x+y} - 3x^2y^4 - 6x$

   (g) $z(x, y) = \ln(\sin x^4 + y) - y \cos^2 x + 7$

   Verify your results using the Partial derivatives maplet.

3. Recall the plotting procedure in your mathematical computer software.

3.3.2 Exercising the algorithms: the Newton's method

Below you will find 10 exercises representing systems of nonlinear equations. The results of the exercises are given for the Newton's method only since we believe that the issue of finding iteration forms $x = g(x)$ has been discussed in detail in section 2.3.3 and here we

would in fact only repeat it. For examples of systems of nonlinear equation solved using simple fixed–point iteration see the full version of the teaching material [1].

Since finding solutions of the systems with a given precision ε would be too tiring (and in fact, pointless), perform 2 steps of the Newton's method only. Notice that not all of the exercises have to be solved numerically. As a further exercise, identify them and find their exact solutions using secondary school techniques.

1. Choose the initial approximation as $\mathbf{x}^{(0)} = (-1; -2)$.

$$\begin{aligned}(x-1)^2 + (y+2)^2 &= 9\\ (x+2)^2 + (y-1)^2 &= 9\end{aligned}$$

Solution using the Newton's method: After we substitute the initial approximation we get the system

$$\begin{aligned}-4\delta_1 &= 5\\ 2\delta_1 - 6\delta_2 &= -1,\end{aligned}$$

which means that $\delta_1 = -1.25, \delta_2 = -0.25$. Thus $\mathbf{x}^{(1)} = (-2.25; -2.25)$. After we substitute the first approximation we get the system

$$\begin{aligned}-6.5\delta_1 + 0.5\delta_2 &= -1.625\\ -0.5\delta_1 + 6.5\delta_2 &= -1.625,\end{aligned}$$

which means that $\delta_1 = 0.23214, \delta_2 = 0.23214$. Thus $\mathbf{x}^{(2)} = (-2.0179; -2.0179)$.

2. Choose the initial approximation as $\mathbf{x}^{(0)} = (1; 1)$.

$$\begin{aligned}\sin x + 2\cos y &= 1\\ 3\cos x - 5\sin y &= -2\end{aligned}$$

Solution using the Newton's method: After we substitute the initial approximation we get the system

$$\begin{aligned}0.5403\delta_1 - 1.6829\delta_2 &= -0.9221\\ -2.5244\delta_1 - 2.7015\delta_2 &= -0.4136\end{aligned}$$

which means that $\delta_1 = -0.3145, \delta_2 = 0.4469$. Thus $\mathbf{x}^{(1)} = (0.6855; 1.4469)$. After we substitute the first approximation we get the system

$$\begin{aligned}0.7741\delta_1 - 1,9847\delta_2 &= 0.1198\\ -1.8993\delta_1 - 0.6177\delta_2 &= -0.3606,\end{aligned}$$

which means that $\delta_1 = 0.1859, \delta_2 = 0.0121$. Thus $\mathbf{x}^{(2)} = (0.8714; 1.4591)$. The functions are depicted on Fig. 3.1.

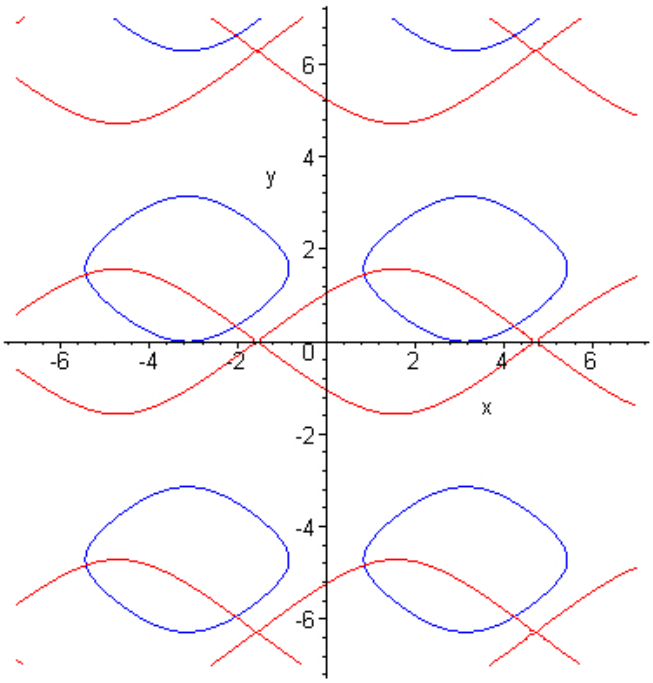


Fig. 3.1: Plots for Exercise 2 – the first function in red, the second function in blue

3. Choose the initial approximation as $\mathbf{x}^{(0)} = (-1; 1)$.

$$\begin{aligned} \sin(x+y) + \mathrm{e}^x &= 1 \\ \cos(x+y) - \ln y &= 1 \end{aligned}$$

Solution using the Newton's method: After we substitute the initial approximation we get the system

$$\begin{aligned} 1.3679\delta_1 + \delta_2 &= 0.6321 \\ -\delta_2 &= 0 \end{aligned}$$

which means that $\delta_1 = 0.4621, \delta_2 = 0$. Thus $\mathbf{x}^{(1)} = (-0.5379; 1)$. After we substitute the first approximation we get the system

$$\begin{aligned} 1.4791\delta_1 + 0.8951\delta_2 &= -0.0298 \\ -0.4458\delta_1 - 1.4458\delta_2 &= 0.1049, \end{aligned}$$

which means that $\delta_1 = 0.0292, \delta_2 = -0.0815$. Thus $\mathbf{x}^{(2)} = (-0.5087; 0.9185)$.

4. Choose the initial approximation as $\mathbf{x}^{(0)} = (4; -2)$.

$$\begin{aligned} (x-2)^2 + y^2 &= 9 \\ (x-3)^2 + y &= 1 \end{aligned}$$

Solution using the Newton's method: After we substitute the initial approximation we get the system

$$\begin{aligned} 4\delta_1 - 4\delta_2 &= 1 \\ 2\delta_1 + \delta_2 &= 2 \end{aligned}$$

which means that $\delta_1 = 0.75, \delta_2 = 0.5$. Thus $\mathbf{x}^{(1)} = (4.75; -1.5)$. After we substitute the first approximation we get the system

$$\begin{aligned} 5.5\delta_1 - 3\delta_2 &= -0.8125 \\ 3.5\delta_1 + \delta_2 &= -0.5625, \end{aligned}$$

which means that $\delta_1 = -0.1563, \delta_2 = -0.0156$. Thus $\mathbf{x}^{(2)} = (4.5938; -1.5156)$.

5. Choose the initial approximation as $\mathbf{x}^{(0)} = (1; -4)$. *Notice that this exercises differs from the previous one in the power of* $(x-2)$ *in the first equation as instead of* $(x-2)^2$ *we now have* $(x-2)^3$.

$$\begin{aligned} (x-2)^3 + y^2 &= 9 \\ (x-3)^2 + y &= 1 \end{aligned}$$

Solution using the Newton's method: After we substitute the initial approximation we get the system

$$\begin{aligned} 3\delta_1 - 8\delta_2 &= -6 \\ -4\delta_1 + \delta_2 &= 1 \end{aligned}$$

which means that $\delta_1 = -0.0690, \delta_2 = 0.7241$. Thus $\mathbf{x}^{(1)} = (0.9310; -3.2759)$. After we substitute the first approximation we get the system

$$\begin{aligned} 3.4281\delta_1 - 6.5517\delta_2 &= -0.5098 \\ -4.1379\delta_1 + \delta_2 &= -0.0048, \end{aligned}$$

which means that $\delta_1 = 0.0228, \delta_2 = 0.0898$. Thus $\mathbf{x}^{(2)} = (0.9539; -3.1861)$.

6. Choose the initial approximation as $\mathbf{x}^{(0)} = (4; 4)$.

$$\begin{aligned} (x-2)^2 + \ln y &= 9 \\ \sin x + (y-3)^2 &= 1 \end{aligned}$$

Solution using the Newton's method: After we substitute the initial approximation we get the system

$$\begin{aligned} 4\delta_1 + 0.25\delta_2 &= 3.6137 \\ -0.6536\delta_1 + 2\delta_2 &= 0.7568 \end{aligned}$$

which means that $\delta_1 = 0.8622, \delta_2 = 0.6602$. Thus $\mathbf{x}^{(1)} = (4.8622; 4.6602)$. After we substitute the first approximation we get the system

$$\begin{aligned} 5.7243\delta_1 + 0.2146\delta_2 &= -0.7310 \\ 0.1492\delta_1 + 3.3204\delta_2 &= -0.7674, \end{aligned}$$

which means that $\delta_1 = -0.1192, \delta_2 = -0.2258$. Thus $\mathbf{x}^{(2)} = (4.7429; 4.4344)$.

7. Choose the initial approximation as $\mathbf{x}^{(0)} = (-4; -2)$.

$$\begin{aligned} x\sin y - y\cos x + \mathrm{e}^{x+y} &= 2 \\ y\sin x - x\cos y + \ln(xy) &= 2 \end{aligned}$$

Solution using the Newton's method: Unlike in the previous exercises including the matrix of partial derivatives (3.2) may prove relevant here.

$$\mathbf{F}' = \begin{pmatrix} \sin y + y\sin x + \mathrm{e}^{x+y} & x\cos y - \cos x + \mathrm{e}^{x+y} \\ y\cos x - \cos y + \frac{1}{x} & \sin x + x\sin y + \frac{1}{y} \end{pmatrix}$$

After we substitute the initial approximation we get the system

$$\begin{aligned} -2.4204\delta_1 + 2.3207\delta_2 &= -0.3324 \\ 1.4734\delta_1 + 3.8940\delta_2 &= 3.0988 \end{aligned}$$

which means that $\delta_1 = 0.6606, \delta_2 = 0.5458$. Thus $\mathbf{x}^{(1)} = (-3.3394; -1.4542)$. After we substitute the first approximation we get the system

$$\begin{aligned} -1.2707\delta_1 + 0.6003\delta_2 &= 0.1009 \\ 1.0101\delta_1 + 2.8255\delta_2 &= 0.3170, \end{aligned}$$

which means that $\delta_1 = -0.0226, \delta_2 = 0.1203$. Thus $\mathbf{x}^{(2)} = (-3.3619; -1.3339)$.

Remark: Notice that the partial derivatives are not defined for all $x, y \in \mathbb{R}$. Thus choosing the initial approximation such that $x = 0$ or $y = 0$ is impossible. This is true not only for *choosing* the initial approximation – the method will fail also whenever $x = 0$ or $y = 0$ is *computed* as the solution of the linear system.

8. Choose the initial approximation as $\mathbf{x}^{(0)} = (-1; -1)$.

$$\begin{aligned} x^2\sin y &= -1 \\ \frac{y^2\sin 2x}{x} &= 1 \end{aligned}$$

Solution using the Newton's method: The matrix of partial derivatives (3.2), i.e. the matrix of functions out of which the numerical matrix for the system of linear equations is computed, is

$$\mathbf{F}' = \begin{pmatrix} 2x\sin y & x^2\cos y \\ \frac{y^2(2x\cos 2x - \sin 2x)}{x^2} & \frac{2y\sin 2x}{x} \end{pmatrix}$$

After we substitute the initial approximation we get the system

$$\begin{aligned} 1.6829\delta_1 + 0.5403\delta_2 &= -0.1585 \\ 1.7416\delta_1 - 1.8186\delta_2 &= 0.0907 \end{aligned}$$

which means that $\delta_1 = -0.0598, \delta_2 = -0.1071$. Thus $\mathbf{x}^{(1)} = (-1.0598; --1.1071)$. After we substitute the first approximation we get the system

$$\begin{aligned} 1.8958\delta_1 + 0.5023\delta_2 &= 0.0046 \\ 2.1378\delta_1 - 1.7825\delta_2 &= 0.0132, \end{aligned}$$

which means that $\delta_1 = 0.0033, \delta_2 = -0.0034$. Thus $\mathbf{x}^{(2)} = (-1.0565; -1.1106)$.

Remark: Notice that in this exercise we encounter the same problem as in the previous one as some partial derivatives are not defined for all $x, y \in \mathbb{R}$. To be more specific, $x = 0$ is problematic. However, the problem is deeper here. For some values – such as e.g. $y = 0$ – the partial derivatives are defined yet $\mathbf{F}'$ becomes singular which means that the linear system does not have exactly one solution.

9. Choose the initial approximation as $\mathbf{x}^{(0)} = (2; 2)$.

$$\begin{aligned} \ln(x+y) + \mathrm{e}^{x-y} - xy + 2 &= 0 \\ \mathrm{e}^{y-x} - \ln x - \frac{x}{y} + 2 &= 0 \end{aligned}$$

Solution using the Newton's method: The matrix of partial derivatives (3.2), i.e. the matrix of functions out of which the numerical matrix for the system of linear equations is computed, is

$$\mathbf{F}' = \begin{pmatrix} \frac{1}{x+y} + \mathrm{e}^{x-y} - y & \frac{1}{x+y} - \mathrm{e}^{x-y} - x \\ -\mathrm{e}^{y-x} - \frac{1}{x} - \frac{1}{y} & \mathrm{e}^{y-x} + \frac{x}{y^2} \end{pmatrix}$$

After we substitute the initial approximation we get the system

$$\begin{aligned} -0.75\delta_1 - 2.75\delta_2 &= -0.3863 \\ -2\delta_1 + 1.5\delta_2 &= -1.3069 \end{aligned}$$

which means that $\delta_1 = 0.6299, \delta_2 = -0.0313$. Thus $\mathbf{x}^{(1)} = (2.6299; 1.9687)$. After we substitute the first approximation we get the system

$$\begin{aligned} 0.1860\delta_1 - 4.3497\delta_2 &= -0.2855 \\ -1.4044\delta_1 + 1.1948\delta_2 &= -0.2134, \end{aligned}$$

which means that $\delta_1 = 0.2156, \delta_2 = 0.0749$. Thus $\mathbf{x}^{(2)} = (2.8455; 2.0435)$.

Remark: Notice that in this exercise we again encounter the problem with the domain of definition of some partial derivatives.

10. Find the solution which is more distant from the origin.

$$\begin{aligned} x^2 + y^2 - 6x - 2y + 1 &= 0 \\ 4x^2 + 9y^2 - 8x + 18y - 23 &= 0 \end{aligned}$$

Solution: In this exercise we first have to plot or sketch the graphs of two conic sections. However, sketching them is rather easy as all we apply is secondary school knowledge. The respective plots can be seen in Fig. 3.2. We can see that $\mathbf{x}^{(0)} = (4; -2)$ can be a good initial approximation. After we substitute it we get the system

$$\begin{aligned} 2\delta_1 - 6\delta_2 &= -1 \\ 24\delta_1 - 18\delta_2 &= -9 \end{aligned}$$

which means that $\delta_1 = -0.333, \delta_2 = 0.0556$. Thus $\mathbf{x}^{(1)} = (3.6667; -1.9444)$. After we substitute the first approximation we get the system

$$\begin{aligned} 1.3333\delta_1 - 5.8889\delta_2 &= -0.1142 \\ 21.3333\delta_1 - 17\delta_2 &= -0.4722, \end{aligned}$$

which means that $\delta_1 = -0.0082, \delta_2 = 0.0175$. Thus $\mathbf{x}^{(2)} = (3.6585; -1.9269)$.

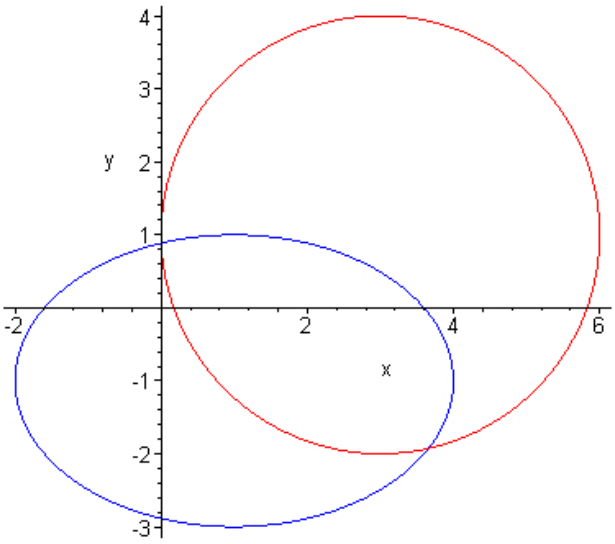


Fig. 3.2: Plots for Exercise 10 – the first conic section in red, the second one in blue

3.4 Related issues

Since we have in fact concentrated on the algorithms themselves (stating that other issues are beyond the scope of *Mathematics 3*), there is not much to consider in this section.

At this level of introduction to numerical methods of solution of systems of nonlinear equations we only have to be aware of a few important issues which may result from not paying attention to convergence tests.

Since in the Newton's method we in each step solve a system of linear equations, we must be aware of the fact that their solutions need not be unique. Also, we get very soon in troubles when exercising "blind" substitution, i.e. when not checking interim results. Regard e.g. system

$$\begin{aligned} \sin(x+y) + e^y &= 1 \\ \cos(x-y) - \ln y &= 1 \end{aligned} \tag{3.6}$$

with the initial approximation $\mathbf{x}^{(0)} = (3, 1)$ which is close to the solution, the second component of which has the smallest positive value among all solutions. We get that $\mathbf{x}^{(3)}$ is complex!

Related with this is the issue of accuracy when obtaining solutions of the linear system in each step. Notice that every inaccuracy in computing any δ_i, $i = 1, \ldots, n$, enters the computing of function values or values of partial derivatives which make up coefficients of the linear system used in the next step. Obviously, for systems of two or three equations direct methods should be used, yet these cannot be used for large systems.

Moreover, here again we have to be careful when localizing the solution by using plotting commands without proper understanding. In Fig. 3.3 and 3.4 compare two plots of (3.6) – one given by Maple using

```
implicitplot([sin(x+y)+exp(y)-1,cos(x-y)-log(y)-1],x=-5..5,y=-5..1,
        grid=[100,100],color=[red,green],scaling=constrained)
```

and the other by Matlab.

Finally, we must be cautious with domains of definition of all functions involved in the solution process. E.g. for (3.6) compare the domain of the respective functions and their partial derivatives. Does this have an influence on the choice of the initial approximation? Does it influence any of the furhter approximations? Also, consider (3.6) and

$$\begin{aligned} \sin(x+y) + e^y &= 1 \\ x - y - \arccos(1 + \ln y) &= 0 \end{aligned} \tag{3.7}$$

The way (3.7) was obtained from (3.6) is obvious. Is this correct? Are the systems equivalent?

3.5 Solving systems of nonlinear equations using Matlab

What has been mentioned for one equation in chapter 2 holds in fact for a system of nonlinear equations as well. Instead of `fzero` we use `fsolve` which looks for the solu-

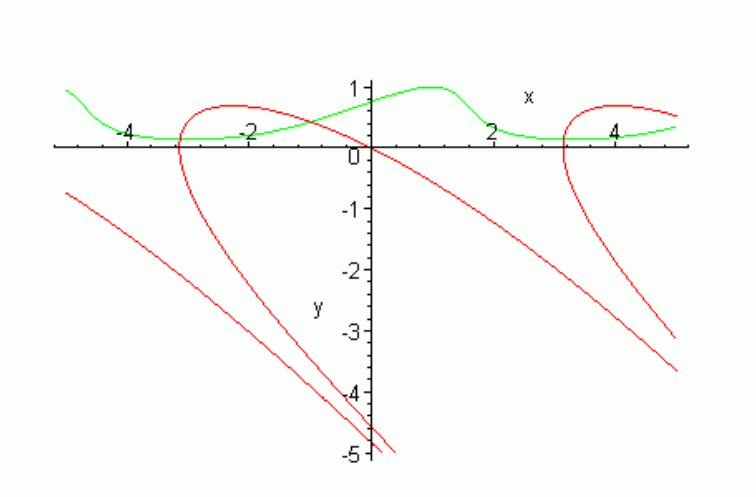


Fig. 3.3: Plot of (3.6) using Maple

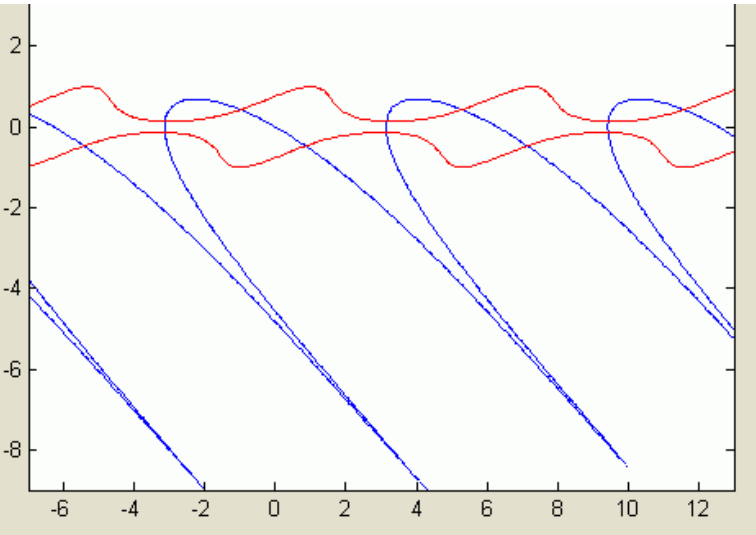


Fig. 3.4: Plot of (3.6) using MATLAB

tion of a system $\mathbf{F}(\mathbf{x}) = \mathbf{o}$ using the given initial approximation. The optional parameters of this command include the choice of the method of calculation or setting parameters of the chosen method. Like the `fzero` command `fsolve` is also included in the `Optimization Toolbox`.

3.6 Exam tasks

Below you can find some authentic exam tasks.

1. We want to find the intersection of two circles: one with centre at $C_1 = [1, 2]$ and perimeter $r_1 = 3$, the other with center at $C_2 = [2, -1]$ and perimeter $r_2 = 4$. We are interested in the intersection both coordinates of which are positive. Start looking for them using the Newton's method. Make two steps. Do not verify convergence criteria. Use integers for the initial approximation.

2. Let $f(z) = \Re(z) + j\Im(z) + |z|^2 - 2z - 3jz^2$ be a complex function defined for all $z \in \mathbb{C}$. We are looking for such a $z \in \mathbb{C}$ that $f(z) = 4 + j$. Start looking for it using either the Newton's method or the simple fixed-point iteration. Make two steps. Do not verify convergence criteria. Use $z_0 = 1 + j$ for the initial approximation.

3. Make two steps of the Newton's method for

$$\begin{aligned} \mathrm{e}^{2x+3y} + 4\sin 2x &= 5 \\ \ln(2x - 3y) + \cos(x^2 - y) &= 1 \end{aligned}$$

   Use $\mathbf{x}^{(0)} = (2, -1)$ as the initial approximation.

4. Make two steps of either the Newton's method or the simple fixed-point iteration for

$$\begin{aligned} (x-2)^2 + (y-3)^2 &= 9 \\ (x-2)^2 + (y+1)^2 &= 16 \end{aligned}$$

   Use $\mathbf{x}^{(0)} = (-1, 2)$ as the initial approximation.

5. Make two steps of either the Newton's method or the simple fixed-point iteration for

$$\begin{aligned} x^3 + 2y^3 - x + y + 1 &= 0 \\ x^3 + 4y^2 + x - y - 5 &= 0 \end{aligned}$$

   Use $\mathbf{x}^{(0)} = (1, -1)$ as the initial approximation.

3.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, MATLAB Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on MATLAB Compiler Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that MATLAB starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of MATLAB output) as far as indexing is concerned.

1. Numerical solution of system of non-linear equations

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Partial derivatives
2. Analytical solution of system of linear equations

4 Aproximations of functions: interpolations

4.1 Wording the problem

The problem we are going to discuss now can be formulated in two ways.

1. There is a set of points $[x_i, y_i]$, $i = 0, \ldots, n$, $n \geq 1$. Find the function $f(x)$ which passes through the given points.
2. *(alternatively)* There is a function $f(x)$ and an interval $I = \langle a, b \rangle$.
   On the interval I replace the function $f(x)$ by a "simpler" function.

Obviously, when we replace a function by a "simpler" one, we can (and will) find some points on it and then look for a function which goes through the given points.

Based on the context we will denote the given points instead of $[x_i, y_i]$ by $[x_i, f_i]$ or $[x_i, f(x_i)]$. To be more precise,

- f_i will denote the value of the searched function f in x_i, i.e. here we presume that we look for an unknown function $f(x)$ (see wording 1),
- $f(x_i)$ will denote the functional value of the function f in the point x_i, i.e. here we presume that to the known function $f(x)$ we look for some other, "simpler" one. This notation is used for cases covered by wording 2.

Points x_i will be called *node points*, *nodes* of interpolation, or *interpolation points*.

4.2 A brief summary of the subject matter

4.2.1 Possible approaches

We can approach the task in two different ways: either we look for *one function* which interpolates all points or we look for a set of functions, or rather *a piecewise defined function*, where every piece of the function connects two neighbouring points.

The application of the former idea results in the concept of the **interpolation polynomial** in *Mathematics 3* while the latter one results in the concept of **spline**.

4.2.2 Interpolation polynomial

We look for a polynomial of a yet unknown grade which interpolates, i.e. passes through, the given points. The polynomial exists exactly one for an arbitrary number of points. Its grade depends on the number of points given.

Theorem 4.1. *Let* $[x_i, f_i], i = 0, \dots n$ *be a set of given points, the* x *coordinates of which are mutually different. Then there exists exactly one polynomial* P_n *of degree at most* n *such that* $P_n(x_i) = f_i, i = 0, \dots n$.

We seek the polynomial using two different strategies: the *Lagrange's* and the *Newton's* one.

The **Lagrange's form of the interpolation polynomial** (or **Lagrange polynomial**) is

$$\begin{aligned} P_n(x) &= f_0 \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} + f_1 \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} + \\ &\quad \dots + f_n \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})} \end{aligned} \tag{4.1}$$

The **Newton's form of the interpolation polynomial** (or **Newton polynomial**) makes use of the concept of divided differences and has the form

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}), \tag{4.2}$$

where $a_0 = f_(x_0)$ and coefficients a_i, $i = 0, \dots, n$, are divided differences of a proper order. For a way of calculating them, cf. the teaching text.

4.2.3 Spline

In *Mathematics 3* we use the following definition of a spline.

Definition 4.2. **Spline of order** k for nodes $a = x_0 < x_1 < \dots < x_n = b$ is a function which

1. is in each interval $\langle x_i, x_{i+1} \rangle$, $i = 0, 1, \dots n-1$ a polynomial $S_i(x)$ of grade at most k such that $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ and
2. has in $\langle a, b \rangle$ continuous derivatives of order $k-1$ including.

Based on k we distinguish linear, quadratic, cubic, etc. splines.

Then we construct **natural cubic splines**.

Definition 4.3. The **natural cubic spline** for a function f with nodes $x_0, x_1, \ldots, x_n$ is the function $S(x)$ which is a cubic polynomial denoted as $S_i(x)$ on every subinterval $\langle x_i, x_{i+1}\rangle$, $i = 0, 1, \ldots, n-1$ fulfilling conditions

$$\begin{aligned} S_i(x_i) &= f(x_i),\ i = 0, \ldots, n-1, \quad S_{n-1}(x_n) = f(x_n) \\ S_i(x_{i+1}) &= S_{i+1}(x_{i+1}),\ i = 0, \ldots, n-2 \\ S_i'(x_{i+1}) &= S_{i+1}'(x_{i+1}),\ i = 0, \ldots, n-2 \\ S_i''(x_{i+1}) &= S_{i+1}''(x_{i+1}),\ i = 0, \ldots, n-2 \end{aligned}$$

and boundary conditions

$$S''(x_0) = S''(x_n) = 0$$

On the respective intervals $\langle x_i, x_{i+1}\rangle$, $i = 0, 1, \ldots, n-1$, we search for the natural cubic spline in the form

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \tag{4.3}$$

The coefficients a_i, b_i, c_i, d_i, $i = 0, \ldots n-1$ are obtained in the following way. First, $a_i = f(x_i)$ (or, based on the context, $a_i = f_i$), for all $i = 0, \ldots n-1$. Then $c_0 = c_n = 0$.

Then we calculate remaining coefficients c_i, i.e. c_i for $i = 1, \ldots, n-1$. These are solutions of the system of linear equations in the form

$$\begin{array}{ccccccccc} 2(h_0+h_1)c_1 & + & h_1c_2 & & & & & = & 3(\frac{\Delta f_1}{h_1} - \frac{\Delta f_0}{h_0}) \\ h_1c_1 & + & 2(h_1+h_2)c_2 & + & h_2c_3 & & & = & 3(\frac{\Delta f_2}{h_2} - \frac{\Delta f_1}{h_1}) \\ & & & \ddots & & & & \vdots & \\ & & h_{n-2}c_{n-2} & + & 2(h_{n-2}+h_{n-1})c_{n-1} & = & 3(\frac{\Delta f_{n-1}}{h_{n-1}} - \frac{\Delta f_{n-2}}{h_{n-2}}) \end{array}$$

or, in short,

$$\begin{aligned} &h_{i-1}c_{i-1} + 2(h_{i-1}+h_i)c_i + h_ic_{i+1} = 3\left(\frac{\Delta f_i}{h_i} - \frac{\Delta f_{i-1}}{h_{i-1}}\right), \quad i = 1, \ldots, n-1 \\ &c_0 = c_n = 0 \end{aligned} \tag{4.4}$$

where $h_i = x_{i+1} - x_i$ and $\Delta f_i = f(x_{i+1}) - f(x_i)$, $i = 0, \ldots, n-1$.

Finally,

$$b_i = \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{c_{i+1} + 2c_i}{3} h_i \qquad i = 0, \ldots, n-1 \tag{4.5}$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \qquad i = 0, \ldots, n-1 \tag{4.6}$$

4.3 Exercises

4.3.1 Prior to starting calculations

1. For the following sets of points $[x_i, f_i]$ decide whether the polynomial $P(x)$ is their interpolation polynomial.

   (a) $P_3(x) = x^3 + 2x^2 - 4x + 1$ for points $[-1; 6]$, $[2; 9]$, $[3; 34]$

   (b) $P_3(x) = x^3 + 2x^2 - 4x + 1$ for points $[-2; 9]$, $[-1; 6]$, $[2; 9]$, $[3; 34]$

   (c) $P_3(x) = x^3 + 2x^2 - 4x + 1$ for points $[-2; 9]$, $[-1; 6]$, $[2; 9]$, $[3; 34]$, $[5; 130]$

   (d) $P_3(x) = x^3 + 2x^2 - 4x + 1$ for points $[-2; 9]$, $[-1; 6]$, $[2; 9]$, $[3; 34]$, $[5; 156]$

   (e) $P_1(x) = x + 1$ for points $[2; 3]$, $[3; 4]$, $[5; 6]$, $[7; 8]$

2. Decide whether the following functions are splines of the indicated type.

   (a) quadratic

   $$S(x) = \begin{cases} x^2 + 2x + 1, & \text{for } x \in \langle -1, 0\rangle, \\ 2x^2 + 2x + 1, & \text{for } x \in \langle 0, 1\rangle, \\ 4x^2 + 1, & \text{for } x \in \langle 1, 2\rangle. \end{cases}$$

   (b) quadratic

   $$S(x) = \begin{cases} x^2 + 2x + 1, & \text{for } x \in \langle -1, 0\rangle, \\ 2x^2 + 2x + 1, & \text{for } x \in \langle 0, 1\rangle, \\ 4x^2 - 2x + 3, & \text{for } x \in \langle 1, 2\rangle. \end{cases}$$

   (c) quadratic

   $$S(x) = \begin{cases} x^2 + 2x + 1, & \text{for } x \in \langle -1, 0\rangle, \\ 2x^2 + 2x + 1, & \text{for } x \in \langle 0, 2\rangle, \\ 4x^2 + 1, & \text{for } x \in \langle 2, 4\rangle. \end{cases}$$

   (d) quadratic

   $$S(x) = \begin{cases} x^2 + 2x + 1, & \text{for } x \in \langle -1, 0\rangle, \\ 4x^2 + 1, & \text{for } x \in \langle 1, 2\rangle. \end{cases}$$

   (e) linear

   $$S(x) = \begin{cases} 0, & \text{for } x \in \langle -3, 0\rangle, \\ x, & \text{for } x \in \langle 0, 1\rangle, \\ 1, & \text{for } x \in \langle 1, 4\rangle. \end{cases}$$

4.3.2 Exercising the algorithms: interpolation polynomial and natural cubic splines

Below you will find 5 sets of points. Find the interpolation polynomials and natural cubic splines to interpolate them. Simplify the polynomial to the canonical form $P_n(x) = \sum_{i=0}^{n} a_i x^i$. *You may choose to construct either Lagrange or Newton polynomial. Below, however, we include the table of divided differences only, i.e. construct the Newton polynomial only. Should you choose the Lagrange form, your polynomial should naturally simplify to the result obtained at the end our calculations for the Newton polynomial. As far as natural cubic splines are concerned, we relate the form of the spline and of its coefficients to section 4.2.3.*

1. Fit the following data

x_i	1	2	2.5	3
f_i	2	1.5	0.8	2.8

first with the interpolation polynomial and then with a natural cubic spline.

Interpolation polynomial (Newton): The table of divided differences (where k is the order of the difference) is

x_i	f_i	$k=1$	$k=2$	$k=3$
1	2	-0.5	-0.6	3
2	1.5	-1.4	5.4	$---$
2.5	0.8	4	$---$	$---$
3	2.8	$---$	$---$	$---$

Thus the polynomial is

$$P_3(x) = 2 - 0.5(x-1) - 0.6(x-1)(x-2) + 3(x-1)(x-2)(x-2.5),$$

which can be simplified to

$$P_3(x) = 3x^3 - 17.1x^2 + 29.8x - 13.7.$$

Natural cubic spline: The system (4.4) is

$$\begin{aligned} 3c_1 + 0.5c_2 &= -2.7 \\ 0.5c_1 + 2c_2 &= 16.2 \end{aligned}$$

and the spline is

$$S(x) = \begin{cases} -0.7826(x-1)^3 + 0.2826(x-1) + 2 & \text{for } x \in \langle 1;2\rangle \\ 7.3565(x-2)^3 - 2.3478(x-2)^2 - 2.0652(x-2) + 1.5 & \text{for } x \in \langle 2;2.5\rangle \\ -5.7913(x-2.5)^3 + 8.6870(x-2.5)^2 + 1.1043(x-2.5) + 0.8 & \text{for } x \in \langle 2.5;3\rangle \end{cases}$$

2. Fit the following data

x_i	2	2.5	3	4	5
f_i	3	-1	3	-4	0

first with the interpolation polynomial and then with a natural cubic spline.

Interpolation polynomial (Newton): The table of divided differences (where k is the order of the difference) is

x_i	f_i	$k=1$	$k=2$	$k=3$	$k=4$
2	3	-8	16	-13	6.4
2.5	-1	8	-10	6.2	$---$
3	3	-7	5.5	$---$	$---$
4	-4	4	$---$	$---$	$---$
5	0	$---$	$---$	$---$	$---$

Thus the polynomial is

$$P_4(x) = 3 - 8(x-2) + 16(x-2)(x-2.5) + 13(x-2)(x-2.5)(x-3) + \\ +6.4(x-2)(x-2.5)(x-3)(x-4),$$

which can be simplified to

$$P_4(x) = 6.4x^4 - 88.6x^3 + 423.9x^2 - 890.1x + 678.$$

Natural cubic spline: The system (4.4) is

$$\begin{aligned} 2c_1 + 0.5c_2 &= 48 \\ 0.5c_1 + 3c_2 + c_3 &= -45 \\ c_2 + 4c_3 &= 33 \end{aligned}$$

and the spline is

$$S(x) = \begin{cases} 20.1429(x-2)^3 - 13.0357(x-2) + 3 & \text{for } x \in \langle 2; 2.5\rangle \\ -36.7143(x-2.5)^3 + 30.2143(x-2.5)^2 + 2.0714(x-2.5) - 1 & \text{for } x \in \langle 2.5; 3\rangle \\ 13.1071(x-3)^3 - 24.8571(x-3)^2 + 4.75(x-3) + 3 & \text{for } x \in \langle 3; 4\rangle \\ -4.8214(x-4)^3 + 14.4643(x-4)^2 - 5.6429(x-4) - 4 & \text{for } x \in \langle 4; 5\rangle \end{cases}$$

3. Fit the following data

x_i	2	2.5	3	4	5	8
f_i	3	-1	3	-4	0	3

first with the interpolation polynomial and then with a natural cubic spline.

Interpolation polynomial (Newton): The table of divided differences (where k is the order of the difference) is

x_i	f_i	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
2	3	-8	16	-13	6.4	-1.2924
2.5	-1	8	-10	6.2	-1.3545	$---$
3	3	-7	5.5	-1.25	$---$	$---$
4	-4	4	-0.75	$---$	$---$	$---$
5	0	1	$---$	$---$	$---$	$---$
8	3	$---$	$---$	$---$	$---$	$---$

Thus the polynomial is

$$P_4(x) = 3 - 8(x-2) + 16(x-2)(x-2.5) + 13(x-2)(x-2.5)(x-3) + \\ +6.4(x-2)(x-2.5)(x-3)(x-4) - 1.2924(x-2)(x-2.5)(x-3)(x-4)(x-5),$$

which can be simplified to

$$P_4(x) = -1.2924x^5 + 27.725x^4 - 223.597x^3 + 852.3386x^2 - 1542.7742x + 1065.7273.$$

Natural cubic spline: The system (4.4) is

$$\begin{aligned} 2c_1 + 0.5c_2 &= 48 \\ 0.5c_1 + 3c_2 + c_3 &= -45 \\ c_2 + 4c_3 + c_4 &= 33 \\ c_3 + 8c_4 &= -9 \end{aligned}$$

and the spline is

$$S(x) = \begin{cases} 20.1911(x-2)^3 - 13.0478(x-2) + 3 & \text{for } x \in \langle 2; 2.5\rangle \\ -36.9553(x-2.5)^3 + 30.2866(x-2.5)^2 + 2.0955(x-2.5) - 1 & \text{for } x \in \langle 2.5; 3\rangle \\ 13.4807(x-3)^3 - 25.1464(x-3)^2 + 4.6656(x-3) + 3 & \text{for } x \in \langle 3; 4\rangle \\ -6.1109(x-4)^3 + 15.2958(x-4)^2 - 5.1849(x-4) - 4 & \text{for } x \in \langle 4; 5\rangle \\ 0.3374(x-5)^3 - 3.037(x-5)^2 + 7.074(x-5) & \text{for } x \in \langle 5; 8\rangle \end{cases}$$

4. Fit the following data

x_i	1.2	2.3	5.4	6.5	7.6
f_i	3.2	2.3	9.8	8.9	5.5

first with the interpolation polynomial and then with a natural cubic spline.

Interpolation polynomial (Newton): The table of divided differences (where k is the order of the difference) is

x_i	f_i	$k=1$	$k=2$	$k=3$	$k=4$
1.2	3.2	-0.8182	0.7708	-0.2909	0.0377
2.3	2.3	2.4194	-0.7708	-0.0495	$---$
5.4	9.8	-0.8182	-1.0331	$---$	$---$
6.5	8.9	-3.0909	$---$	$---$	$---$
7.6	5.5	$---$	$---$	$---$	$---$

Thus the polynomial is

$$P_4(x) = 3.2 - 0.8182(x-1.2) + 0.7708(x-1.2)(x-2.3) - \\ -0.2909(x-1.2)(x-2.3)(x-5.4) + 0.0377(x-1.2)(x-2.3)(x-5.4)(x-6.5),$$

which can be simplified to

$$P_4(x) = 0.0377x^4 - 0.8718x^3 + 6.3589x^2 - 15.6895x + 14.2989.$$

Natural cubic spline: The system (4.4) is

$$\begin{aligned} 8c_1 + 3.1c_2 &= 9.7126 \\ 3.1c_1 + 8.4c_2 + 1.1c_3 &= -9.7126 \\ 1.1c_2 + 4.4c_3 &= -6.8128 \end{aligned}$$

and the spline is

$$S(x) = \begin{cases} 0.5361(x-1.2)^3 - 1.4669(x-1.2) + 3.2 & \text{for } x \in \langle 1.2; 2.3\rangle \\ -0.3688(x-2.3)^3 + 1.7691(x-2.3)^2 + 0.4792(x-2.3) + 2.3 & \text{for } x \in \langle 2.3; 5.4\rangle \\ 0.1594(x-5.4)^3 - 1.6606(x-5.4)^2 + 0.8155(x-5.4) + 9.8 & \text{for } x \in \langle 5.4; 6.5\rangle \\ 0.3437(x-6.5)^3 - 1.1344(x-6.5)^2 - 2.259(x-6.5) + 8.9 & \text{for } x \in \langle 6.5; 7.6\rangle \end{cases}$$

5. Fit the following data

x_i	0.2	2.3	5.4	6.5	7.6
f_i	3.2	2.3	9.8	8.9	5.5

first with the interpolation polynomial and then with a natural cubic spline.

Interpolation polynomial (Newton): The table of divided differences (where k is the order of the difference) is

x_i	f_i	$k=1$	$k=2$	$k=3$	$k=4$
0.2	3.2	-0.4286	0.5477	-0.2093	0.0216
2.3	2.3	2.4194	-0.7708	-0.0495	$---$
5.4	9.8	-0.8182	-1.0331	$---$	$---$
6.5	8.9	-3.0909	$---$	$---$	$---$
7.6	5.5	$---$	$---$	$---$	$---$

Thus the polynomial is

$$P_4(x) = 3.2 - 0.4286(x - 0.2) + 0.5477(x - 0.2)(x - 2.3) - \\ -0.2093(x - 0.2)(x - 2.3)(x - 5.4) + 0.0216(x - 0.2)(x - 2.3)(x - 5.4)(x - 6.5),$$

which can be simplified to

$$P_4(x) = 0.0216x^4 - 0.5203x^3 + 3.6115x^2 - -6.7328x + 4.4062.$$

Natural cubic spline: The system (4.4) is

$$\begin{aligned} 8c_1 + 3.1c_2 &= 9.7126 \\ 3.1c_1 + 8.4c_2 + 1.1c_3 &= -9.7126 \\ 1.1c_2 + 4.4c_3 &= -6.8128 \end{aligned}$$

and the spline is

$$S(x) = \begin{cases} 0.1998(x - 0.2)^3 - 1.3095(x - 0.2) + 3.2 & \text{for } x \in \langle 0.2; 2.3\rangle \\ -0.2929(x - 2.3)^3 + 1.2584(x - 2.3)^2 + 1.3332(x - 2.3) + 2.3 & \text{for } x \in \langle 2.3; 5.4\rangle \\ 0.0856(x - 5.4)^3 - 1.4657(x - 5.4)^2 + 0.6905(x - 5.4) + 9.8 & \text{for } x \in \langle 5.4; 6.5\rangle \\ 0.3585(x - 6.5)^3 - 1.1832(x - 6.5)^2 - 2.2233(x - 6.5) + 8.9 & \text{for } x \in \langle 6.5; 7.6\rangle \end{cases}$$

6. Fit the following data

x_i	-2.5	0.3	1.2	2.9	3.7
f_i	-6	-0.4	1.4	4.8	6.4

first with the interpolation polynomial and then with a natural cubic spline.

Interpolation polynomial (Newton): The table of divided differences (where k is the order of the difference) is

x_i	f_i	$k = 1$	$k = 2$	$k = 3$	$k = 4$
-2.5	-6	2	0	0	0
0.3	-0.4	2	0	0	$---$
1.2	1.4	2	0	$---$	$---$
2.9	4.8	2	$---$	$---$	$---$
3.7	6.4	$---$	$---$	$---$	$---$

Thus the polynomial is

$$P_4(x) = -6 + 2(x + 2.5) = 2x - 1,$$

i.e. it is a line.

Remark: Notice that even though *after* finding the interpolation polynomial one can easily verify that the points all belong to the same line, discovering this fact *before* the calculations – and especially discovering the correct coefficients – is rather difficult from just the points themselves. Obviously, here we have a simple case of five points on one line. Therefore consider what would change if we had 10 points on a parabola.

Natural cubic spline: The system (4.4) is

$$\begin{aligned} 7.4c_1 + 0.9c_2 &= 0 \\ 0.9c_1 + 5.2c_2 + 1.7c_3 &= 0 \\ 1.7c_2 + 5c_3 &= 0 \end{aligned}$$

and the spline is

$$S(x) = \begin{cases} 2(x+2.5) - 6 & \text{for } x \in \langle -2.5; 0.3 \rangle \\ 2(x-0.3) - 0.4 & \text{for } x \in \langle 0.3; 1.2 \rangle \\ 2(x-1.2) + 1.4 & \text{for } x \in \langle 1.2; 2.9 \rangle \\ 2(x-2.9) + 4.8 & \text{for } x \in \langle 2.9; 3.7 \rangle \end{cases}$$

7. Verify that all interpolation polynomials are indeed interpolation polynomials and that all splines are indeed splines.

8. In each exercises choose an arbitrary $a \in \langle x_i; x_{i+1} \rangle$, $i = 1, \ldots, n-1$, and compare values $P(a)$ and $S(a)$.

9. Notice the relation between sets of points in Exercise 2 and Exercise 3, which is in fact Exercise 2 with one more point added.

10. Notice the relation between sets of points in Exercise 4 and Exercise 5, which is in fact Exercise 4 with the first point changed.

4.4 Related issues

Suppose we find the interpolation polynomial in either Lagrange's or Newton's form. How exactly do we simplify it to $P_n(x) = \sum_{i=1}^{n} a_i x^i$? At what places (if any) are there risks of round-off or other errors? Also notice that the interpolation polynomial is often used to calculate function values at points other than nodes. How exactly do we (or rather software) calculate it?

As far as splines, or rather natural cubic splines, are concerned, notice that when calculating the coefficients c_i we use the system of linear equations. In a general case such systems need not have one solution. Neither can we be certain that the methods we choose will converge. However, the system (4.4) is constructed in such a way that the matrix of

coefficients is strictly row diagonally dominant. As a result, the system (4.4) has exactly one solution and both Jacobi and Gauss-Seidel methods converge to it for an arbitrary choice of the initial approximation.

We have defined spline in Definition 4.3 as a piecewise function, each piece of which is a cubic polynomial. It is to be noted that this is meant in a general case. Naturally, during our calculations some of the coefficients d_i, i.e. coefficients of x^3, may turn to equal zero. As a result, the piece of spline in question will not be a cubic polynomial. For an example of this see Example 6 where not only coefficients d_i but even all coefficients c_i equal zero.

4.5 Interpolation using Matlab

Interpolation using the interpolation polynomial or splines is dealt with in `User Guide` > `Mathematics` > `Interpolation` > `Interpolating Uniform Data` (exact location and name of the help entry depends on the version of Matlab). Commands `interp1` and `interp1q` may be used. Vectors of x– and y–values and the vector of x–coordinates of points in which the interpolation is to be performed is the input of these commands. (In this text we assume that the two sets of x values are identical.) Finally, we must include the method, or rather the type of spline as an input parameter. The setting `cubic` stands for the cubic spline. Command `interp1q` performs faster yet requires data in a specific format. The interpolation using splines may be also obtained using the `spline` command.

Generally speaking, Matlab understands interpolation to be a problem of the best estimate of function values at points between known nodes. As a result, Matlab does not use specialized commands for interpolation polynomials. Instead, it focuses on finding splines of various types or on interpolating data with polynomials of different grades between two neighboroughing nodes. For details, see documentation of commands such as `mkpp`, `ppval` or `unmkpp`.

4.6 Exam tasks

Below you can find some authentic exam tasks.

1. Construct Newton polynomial for:

x_i	-5	-2	0	1	4
f_i	-3	3	7	9	15

   Simplify it. Comment on the result. Calculate the value of the polynomial in $x = 2$.

2. Construct interpolation polynomial (in either Lagrange's or Newton's form) for

x_i	4	2	1	3
f_i	2	5	6	8

and calculate its values in $x = 0$ and $x = 1.5$.

3. Choose six points $[x_i, f_i]$. Construct the interpolation polynomial for them. The polynomial *must* be in simplified to $P_n(x) = \sum_{i=0}^{n} a_i x^i$. Explain your choice and comment on your reasoning.

4. Decide whether
$$S(x) = \begin{cases} x^2 + 2x + 1, & \text{for } x \in \langle -1, 0\rangle, \\ 2x^2 + 2x + 1, & \text{for } x \in \langle 0, 1\rangle, \\ 4x^2 + 1, & \text{for } x \in \langle 1, 2\rangle. \end{cases}$$
is a quadratic spline. Explain your conclusion.

5. We want to construct the natural cubic spline for

x_i	-3	-1	0	1	3
f_i	-3	3	7	9	5

Regard the form of the spline as polynomials
$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$
valid for all $x \in \langle x_i, x_{i+1}\rangle$. Find arbitrary five coefficients a_i, b_i, c_i, d_i, $i = 0, \ldots, n-1$.

4.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, MATLAB Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on MATLAB Compiler Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that MATLAB starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of MATLAB output) as far as indexing is concerned.

1. Interpolation polynomial and spline

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Calculating function values
2. Simplification of algebraic expressions
3. Graphs of function

5 Least squares method

5.1 Wording the problem

There is a set of points $[x_i, y_i]$, $i = 0, \ldots, n, n \geq 1$. We known that the values y_i are inaccurate (or there are too many of them so that it is obvious that connecting them – either using the interpolation polynomial or spline – is of no practical use). Find the function $y = f(x)$ which "best" expresses the way y depends on x.

5.2 A brief summary of the subject matter

5.2.1 Idea of the method and its use

We start with a set of points which usually represent data acquired during some experiments, measurements of certain variables, etc. Regardless of the meaning of the data or the way they were acquired, they have one thing in common – we do not assume their being accurate. Also, measurements at certain points may be repeated, which means that for some $i, j \in \{0, \ldots, n-1\}$ there may be $x_i = x_j$. (Notice that this results in division by zero when attempting to construct the interpolation polynomial or spline.) The data are acquired based on a "hidden" principle. The relation between x and y is either unknown completely (this is true e.g. for most sets of statistical data) or partially as we may know the type of dependency (e.g. linear, quadratic, etc.) but not the exact coefficients.

Interpolating the data in the way of connecting the points (interpolation polynomial or spline) has no practical meaning in such situations. Therefore, we

1. decide on the type of relation between x and y and
2. among all possible coefficients in the given relation we choose the "best" one to fit the data.

The decision on the type of relation usually follows from the knowledge of the nature of the problem we face or from the visual representation of the data. In our introductory course *Mathematics 3* we focus on the following cases: *polynomial relation*, namely *line*, i.e. $y = c_0 + c_1 x$ and *parabola*, i.e. $y = c_0 + c_1 x + c_2 x^2$, and the most usual nonlinear case

of *exponential*, i.e. $y = aeb^x$, which is in fact equivalent to $y = ac^x$. These cases can be exercised using this text. Some further cases are discussed in the subject as well yet they are included mainly to give the general view of the method.

The preferred way of finding the "best" coefficients for the chosen approximation is the way of minimizing the function of the error of approximation. Notice that another approach is discussed in the subject as well.

5.2.2 Calculation of coefficients

When performing the **least-square fit of a straight line** using the least squares method, we look for a line $y = c_0 + c_1 x$, where coefficients c_0 and c_1 are solutions of the linear system

$$\begin{aligned} c_0(n+1) &+ c_1 \sum_{i=0}^{n} x_i &= \sum_{i=0}^{n} y_i \\ c_0 \sum_{i=0}^{n} x_i &+ c_1 \sum_{i=0}^{n} x_i^2 &= \sum_{i=0}^{n} x_i y_i \end{aligned} \tag{5.1}$$

where $n+1$ is the number of nodes, i.e. points available.

When performing the **least-square fit of a parabola** using the least squares method, we look for a quadratic polynomial $y = c_0 + c_1 x + c_2 x^2$, where coefficients c_0, c_1 and c_2 are solutions of the linear system

$$\begin{aligned} c_0(n+1) &+ c_1 \sum_{i=0}^{n} x_i &+ c_2 \sum_{i=0}^{n} x_i^2 &= \sum_{i=0}^{n} y_i \\ c_0 \sum_{i=0}^{n} x_i &+ c_1 \sum_{i=0}^{n} x_i^2 &+ c_2 \sum_{i=0}^{n} x_i^3 &= \sum_{i=0}^{n} x_i y_i \\ c_0 \sum_{i=0}^{n} x_i^2 &+ c_1 \sum_{i=0}^{n} x_i^3 &+ c_2 \sum_{i=0}^{n} x_i^4 &= \sum_{i=0}^{n} x_i^2 y_i \end{aligned} \tag{5.2}$$

where $n+1$ is again the number of nodes, i.e. points available.

In the case of **least-square fit of an exponential** using the least squares method we *linearize* the task by transforming the desired exponential $y = a\mathrm{e}^{bx}$ to

$$\ln y = \ln a + bx, \tag{5.3}$$

substitute $c_0 = \ln a$, $c_1 = b$ and then search for the straight line to approximate points $[x_i, \ln y_i]$, $i = 0, \ldots, n$. Notice that when working with the exponential $y = ac^x$, we get

$$\ln y = \ln a + x \ln c \tag{5.4}$$

and substitute $c_0 = \ln a$, $c_1 = \ln c$.

5.3 Exercises

5.3.1 Prior to starting the calculation

1. In the manual of your calculator and in the documentation of your mathematical software study the way vectors are entered and manipulated. Given two vectors $\mathbf{x} = (x_0, \ldots, x_n)$, $\mathbf{y} = (y_0, \ldots, y_n)$ find the most efficient way of computing $\sum_{i=0}^{n} x_i$, $\sum_{i=0}^{n} x_i^2$, $\sum_{i=0}^{n} x_i^3$, $\sum_{i=0}^{n} x_i y_i$ or $\sum_{i=0}^{n} x_i^2 y_i$.

2. Solve

$$\begin{aligned} 3x + 4y &= -0,75 \\ 4x + 5y &= 13,25 \end{aligned}$$

   using the Cramer's rule.

3. Solve the linear system included in Exercise 2 using the Gaussian elimination method.

4. Solve

$$\begin{aligned} 3x + 4y - 5z &= -0,75 \\ 4x + 5y + 3z &= 13,25 \\ -5x + 4y - 3z &= -7,75 \end{aligned}$$

   using the Cramer's rule.

5. Solve the linear system included in Exercise 4 using the Gaussian elimination method.

5.3.2 Calculation of coefficients

In the exercises below fit the data with the best curve specified. Solve the linear systems of equations using a direct method such as Cramer's rule or Gaussian elimination.

1. Fit the data with the best least-square line. Then repeat the exercise for a parabola and an exponential $y = ac^x$.

x	1	2	2.6	3.4	7.3	8.5
y	3.1	5.1	6	10.7	20.8	27.5

Fit of a line: We are looking for a line $y = c_0 + c_1 x$, where c_0, c_1 are solutions of

$$\begin{aligned} 6c_0 + 24.8c_1 &= 73.2 \\ 24.8c_0 + 148.86c_1 &= 450.87 \end{aligned}$$

We get $c_0 = -1.0248, c_1 = 3.1996$, i.e. the line is $y = -1.0248 + 3.1996x$.

Fit of a parabola: We are looking for a parabola $y = c_0 + c_1 x + c_2 x^2$, where c_0, c_1, c_2 are solutions of

$$\begin{aligned} 6c_0 + 24.8c_1 + 148.86c_2 &= 73.2 \\ 24.8c_0 + 148.86c_1 + 1069.022c_2 &= 450.87 \\ 148.86c_0 + 1069.022c_1 + 8256.2178c_2 &= 3283.059 \end{aligned}$$

We get $c_0 = 0.9343, c_1 = 1.9742$, $c_2 = 0.1252$, i.e. the parabola is $y = 0.9343 + + 1.9742x + 0.1252x^2$.

Fit of an exponential: We are looking for an $y = ac^x$. First, we have to transform the exercise using (5.4) or (5.3). In case of (5.4) we get a linear system

$$\begin{aligned} 6c_0 + 24.8c_1 &= 13.2718 \\ 24.8c_0 + 148.86c_1 &= 67.433 \end{aligned}$$

the solution of which is $c_0 = 1.0905$, $c_1 = 0.2713$. Since $c_0 = \ln a$, $c_1 = \ln c$, we get that the exponential is $y = 2.9759 \cdot 1.3177^x$.

Remark: The points and the three curves are depicted in Fig. 5.1. Notice that technically speaking, i.e. when all we have are the points, all three approaches are equivalent. However, this is not the case of real-life applications. There, we must always decide on the type of the fit *before* we start any calculations. The decision making is influenced by factors from beyond mathematics.

2. Fit the data with the best least-square line.

x	1	2.5	4	5	5.5	7
y	3.1	5.1	6	8	8.5	9

Fit of a line: We are looking for a line $y = c_0 + c_1 x$, where c_0, c_1 are solutions of

$$\begin{aligned} 6c_0 + 25c_1 &= 39.7 \\ 25c_0 + 127.5c_1 &= 189.6 \end{aligned}$$

We get $c_0 = 2.2982, c_1 = 1.0364$, i.e. the line is $y = 2.2982 + 1.0364x$.

3. Fit the data with the best least-square line.

x	1	2	3	5	6	8
y	4	5	6	8	9	12

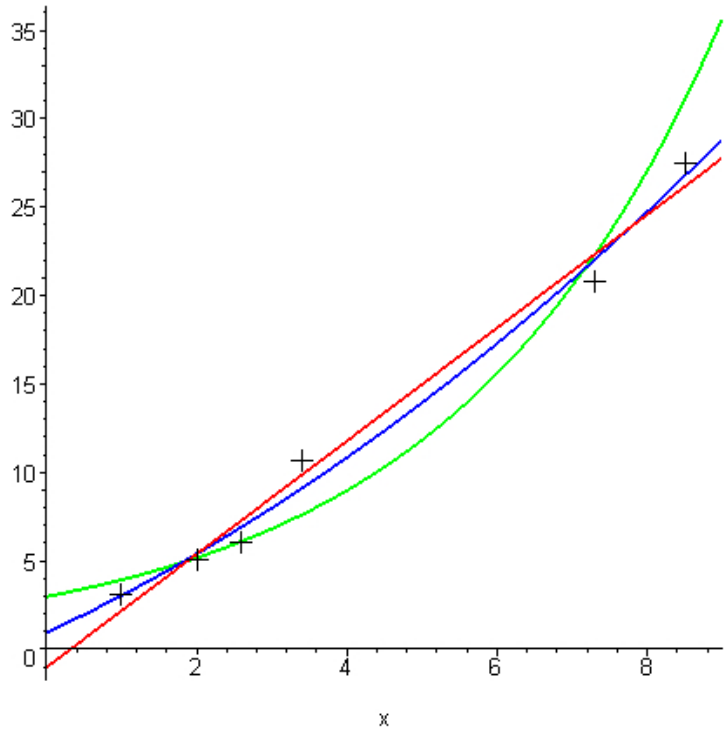


Fig. 5.1: Plots for Exercise 1: The given points fitted by a line (red), a parabola (blue) and an exponential (green)

Fit of a line: We are looking for a line $y = c_0 + c_1 x$, where c_0, c_1 are solutions of

$$\begin{aligned} 6c_0 + 25c_1 &= 44 \\ 25c_0 + 139c_1 &= 222 \end{aligned}$$

We get $c_0 = 2.7081, c_1 = 1.11$, i.e. the line is $y = 2.7081 + 1.11$.

4. Fit the data with the best least-square line. Then change the point $[3; 12]$ to $[3; 10]$ and compare the results.

x	-3	-2	-1	1	2	3
y	4	5	6	8	9	12

Fit of a line: We are looking for a line $y = c_0 + c_1 x$, where c_0, c_1 are solutions of

$$\begin{aligned} 6c_0 &= 44 \\ 28c_1 &= 34 \end{aligned}$$

We get $c_0 = 7.3333, c_1 = 1.2143$, i.e. the line is $y = 7.3333 + 1.2143$. When $[3; 12]$ is changed to $[3; 10]$, we get

$$6c_0 = 42$$

$$28c_1 = 28$$

the solution of which is $c_0 = 7$ and $c_1 = 1$. The points and the lines are depicted in Fig. 5.2.

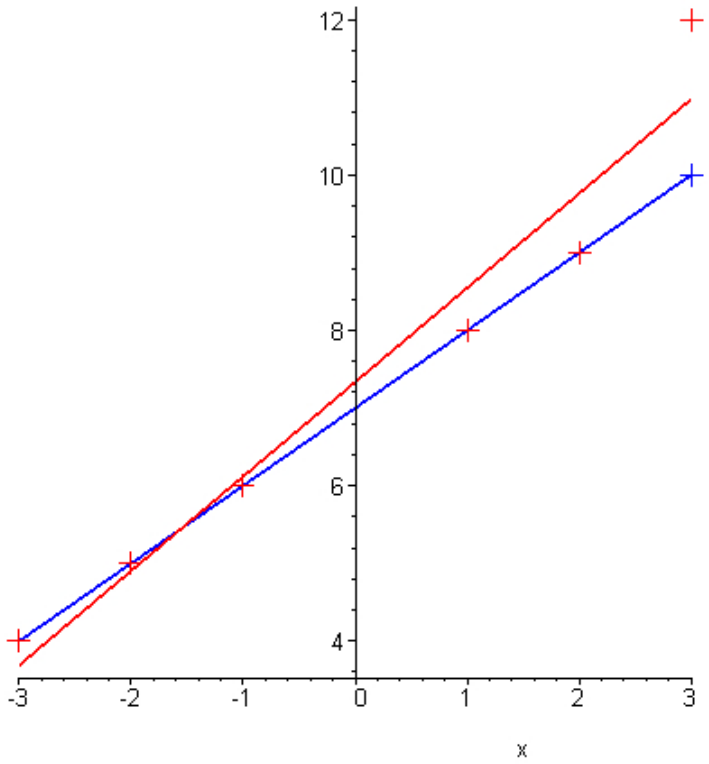


Fig. 5.2: Plots for Exercise 4: Points and line of the original exercise (red) and its modification (blue)

5. Fit the data with the best least-square parabola.

x	0.5	1.7	3.2	3.9	3.9	4.1
y	-3.5	-1.05	6	11.2	10.7	12.9

6. Fit the data with the best least-square parabola.

x	-1	-0.5	1.2	2.8	3.1	3.7
y	-1.5	-1.9	-1.3	1.9	2.8	4.9

Fit of a parabola: We are looking for a parabola $y = c_0 + c_1 x + c_2 x^2$, where c_0, c_1, c_2 are solutions of

$$\begin{aligned} 6c_0 + 9.3c_1 + 33.83c_2 &= 4.9 \\ 9.3c_0 + 33.83c_1 + 102.999c_2 &= 33.02 \end{aligned}$$

$$33.83c_0 + 102.999c_1 + 344.3699c_2 = 105.038$$

We get $c_0 = -2.0303, c_1 = -0.0191$, $c_2 = 0.5102$, i.e. the parabola is $y = -2.0303 - 0.0191x + 0.5102x^2$.

7. Fit the data with the best least-square parabola.

x	-1	-0.5	1.2	2.8
y	-1.5	-1.9	-1.3	1.9

Fit of a parabola: We are looking for a parabola $y = c_0 + c_1x + c_2x^2$, where c_0, c_1, c_2 are solutions of

$$\begin{aligned} 4c_0 + 2.5c_1 + 10.53c_2 &= -2.8 \\ 2.5c_0 + 10.53c_1 + 22.555c_2 &= 6.21 \\ 10.53c_0 + 22.555c_1 + 64.6017c_2 &= 11.049 \end{aligned}$$

We get $c_0 = -2.0193, c_1 = -0.0087$, $c_2 = 0.5032$, i.e. the parabola is $y = -2.0193 - 0.0087 + 0.5032x^2$.

Remark: Notice that this is in fact the previous exercise from which the last two points were removed.

8. Fit the data with the best least-square parabola.

x	-13.2	-5.6	1.12	5.95	8.3	14.2	16.3
y	238.5	50.1	-19.8	-15.7	3	98.5	153.4

Fit of a parabola: We are looking for a parabola $y = c_0 + c_1x + c_2x^2$, where c_0, c_1, c_2 are solutions of

$$\begin{aligned} 7c_0 + 27.07c_1 + 778.4769c_2 &= 508 \\ 27.07c_0 + 778.4769c_1 + 5502.2878c_2 &= 379.669 \\ 778.4769c_0 + 5502.2878c_1 + 148593.6355c_2 &= 103371.7756 \end{aligned}$$

We get $c_0 = -15.1216, c_1 = -6.0457$, $c_2 = 0.9988$, i.e. the parabola is $y = -15.1216 - 6.0457x + 0.9988x^2$.

9. Fit the data with the best least-square exponential $y = ac^x$.

x	-2	-0.8	1.2	2.4	6
y	-1.4	-0.8	-0.5	0	3

Fit of an exponential: We are looking for an $y = ac^x$. First, we have to transform the exercise using (5.4) or (5.3). However, this is not possible here for two reasons: the y coordinate of one of the points is 0 and the y values do not have the same sign. This means that transforming the exercise into (5.4) or (5.3) is not possible.

10. Fit the data with the best least-square exponential $y = ac^x$.

x	-2	-0.8	1.2	2.4	6	7.2
y	1.4	2.8	3.2	4.1	6	8.5

Fit of an exponential: We are looking for an $y = ac^x$. First, we have to transform the exercise using (5.4) or (5.3). In case of (5.4) we get a linear system

$$\begin{aligned} 6c_0 + 14c_1 &= 7.8721 \\ 14c_0 + 99.68c_1 &= 29.4445 \end{aligned}$$

the solution of which is $c_0 = 0.9263$, $c_1 = 0.1653$. Since $c_0 = \ln a$, $c_1 = \ln c$, we get that the exponential is $y = 2.523 \cdot 1.1797^x$.

5.4 Related issues

When applying the least squares method, one must always keep in mind that there are two steps to be done: *decision on the nature of the approximation* and *finding the respective coefficients*. Naturally, in *Mathematics 3* we concentrate on the latter step and leave the former one for the specialized non-mathematical subjects.

As regards the approximation itself, it is important to realize that its coefficients are solutions of a linear system. Thus adding or ignoring some nodes of approximation changes the system which naturally influences its solution, i.e. coefficients of the approximation. Even small changes in the solution may be important as can be seen in Fig. 5.2 or in the following example.

Example 5.1. Suppose that our task is to approximate a certain number of nodes on interval $\langle 40; 50\rangle$ by a parabola. Slight changes in the data set result in two "best" parabolas: $y_1 = -3 + 5x + 0,1x^2$ and $y_2 = -2,95 + 4,92x + 0,0001x^2$. Coefficients of both of them are "almost the same" because they differ less than $\varepsilon = 0,1$ each. Yet as is shown in Fig. 5.3, on the interval $\langle 40; 50\rangle$, which is where we work, the parabolas vary considerably.

Finally, we must consider the exact way of calculating coefficients of the approximation. When searching for a line or a parabola, we will most likely use direct methods to solve the system. However, approximation by polynomials of higher grades results in systems larger than 3×3. For these, all issues discussed in section 1.4 are relevant, and the issue suggested by Example 5.1 becomes crucial.

5.5 The least squares method using Matlab

Commands using the least squares method are summed up in specialized toolboxes `Curve Fitting` and `Optimization Toolbox`. MATLAB knows the whole set of tasks (linear, nonlinear, with non-negative values, etc.) which are solved using special commands

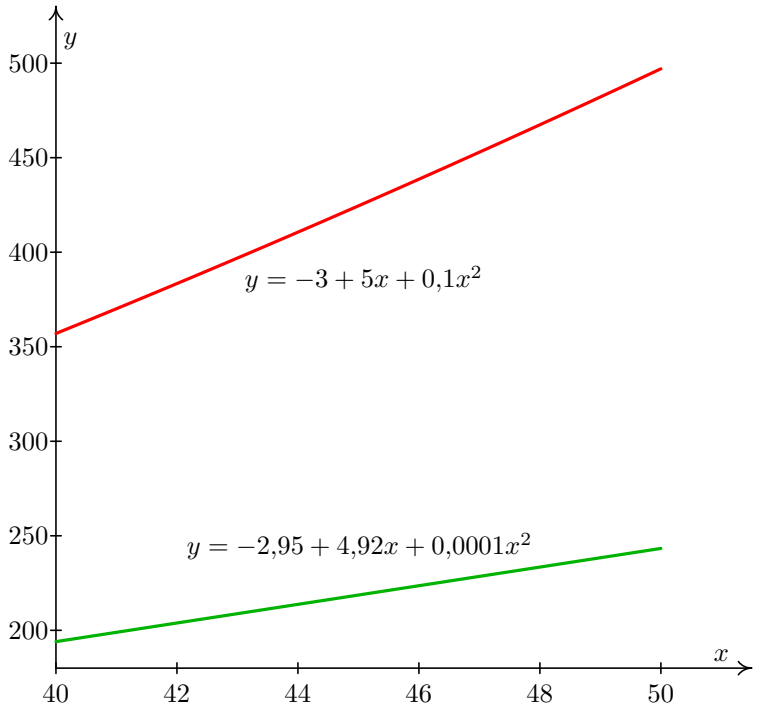


Fig. 5.3: An example of inaccuracies in computing coefficients of the least squares method

such as `lsqcurvefit`, `lsqlin`, `lsqnonneg`, etc. Before using a specific command, analyze the task and choose the apropriate way of solution. Each command has a number of optional parameters which further affect the way of solving the task – their description and explanation of their function is beyond the scope of this text.

5.6 Exam tasks

Below you can find some authentic exam tasks.

1. Find the best line to fit the following data:

x	1	2	2.6	3.4	7.2	8.1
y	3.2	5.5	6	10.1	20	26.6

2. Find either the best line or the best parabola to fit the following data:

x	1.5	2.4	2.9	3.7	4.5	6.7
y	3	4.5	6.1	8.2	14.1	17.8

   When choosing the type of the fit, you do not have to consider the actual position of the points. However, if you choose the fit of a straight line, solve the system and find the line. If you choose the fit of a parabola, stop when finding the system – you do not have to solve it.

3. The following data has been gathered

t_i	-5	-2	0	1	4
v_i	-3	3	7	9	15

We know that v depends on t linearly. Find the least–square fit of a straight line for the gathered data.

4. Find the best exponential $y = ac^x$ to fit the following data:

x	1	2	3	4	5
y	0.9	1.62	2.92	5.25	9.45

5. The following data has been gathered:

x	1	2	3	4	5
y	0.9	1.62	2.92	5.25	9.45

We want to perform the least–square fit of the data by a parabola $y = c_0 + c_1 x + c_2 x^2$. Find at least one of coefficients c_0, c_1, c_2.

5.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, MATLAB Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on MATLAB Compiler Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that MATLAB starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of MATLAB output) as far as indexing is concerned.

1. Least squares method

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Calculating function values
2. Simplification of algebraic expressions
3. Graphs of function

6 Numerical integration

6.1 Wording the problem

For a real function f of a real variable x and numbers $a, b \in \mathbb{R}$ such that $a < b$ calculate $\int_a^b f(x)\mathrm{d}x$.

6.2 A brief summary of the subject matter

6.2.1 The idea of numerical integration and its advantages

In *Mathematics 1* you learned basic integration techniques such as *substitution* or *integration by parts*. All these techniques were based on the same principle – after the analysis of the task a suitable method of integration (e.g. a suitable substitution, decomposition of the integrand into partial fractions, etc.) was chosen. Calculating the definite integral was done in two steps, the first of which was finding the antiderivative of the integrand. This strategy has two main limitations: the analysis is a lengthy and difficult process, and the antiderivative need not exist at all.

Numerical integration bypasses these two obstacles by relying on the geometrical meaning of the definite integral. We simply calculate the area bounded by the x–axis, lines $x = a$ and $x = b$, and function $f(x)$. This can be done in a number of ways. In *Mathematics 3* we make use of the interpolation polynomial and use methods which assume that

$$\int_a^b f(x)\mathrm{d}x \doteq \int_a^b P_n(x)\mathrm{d}x. \tag{6.1}$$

Because of the known issues with the interpolation polynomial (possible oscilation for polynomials of higher grades) we focus on methods with a small number of nodes.

6.2.2 Methods of calculation

In *Mathematics 3* we distinguish between **simple** and **composite** methods (or "rules").

The **simple trapezoidal rule** calculates the integral as

$$\int_a^b f(x)\mathrm{d}x \doteq \int_a^b L(x)\mathrm{d}x = \frac{b-a}{2}\Big(f(b)+f(a)\Big) \tag{6.2}$$

while the **simple Simpson's rule** calculates it as

$$\int_a^b f(x)\mathrm{d}x \doteq \int_a^b S(x)\mathrm{d}x = \frac{b-a}{6}\Big(f(a)+4f(\tfrac{a+b}{2})+f(b)\Big), \tag{6.3}$$

where a, b are bounds of integration and c is the midpoint of $\langle a, b\rangle$.

When applying the composite methods, we first partition the interval $\langle a, b\rangle$ with step h, i.e. each subinterval of I is of length $h = \frac{b-a}{m}$, where m is the number of subintervals. Then we apply the given method on each subinterval. The formula for calculating $\int_a^b f(x)\mathrm{d}x$ using the **composite trapezoidal rule** with step h is

$$\int_a^b f(x)\,\mathrm{d}x \doteq h\left(\tfrac{1}{2}\,f(x_0)+f(x_1)+\cdots+f(x_{m-1})+\tfrac{1}{2}\,f(x_m)\right) = L_m \tag{6.4}$$

while the integral using the **composite Simpson's rule** is computed as

$$\int_a^b f(x)\,\mathrm{d}x \doteq \tag{6.5}$$
$$\doteq \frac{h}{3}\Big(f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+\cdots+2f(x_{m-2})+4f(x_{m-1})+f(x_m)\Big) = S_m$$

6.2.3 Precision and stopping the algorithms

Generally speaking, there are two strategies to achieve the result with a given precision ε. We may either calculate such m, i.e. the number of subintervals of interval $\langle a, b\rangle$, that the numerical result L_m – given by (6.2) – or S_m – given by (6.5) – gives the result with a given precision, or we may work on calculating members of the the sequence $\{L_m\}_{m=1}^{\infty}$ (or $\{S_m\}_{m=1}^{\infty}$) and search for the limit of the sequence or stop calculating its members "at some suitable time". Both of these strategies are discussed in the subject.

Here in this text we use the latter approach and stop our calculations when for some $m \in \mathbb{N}$ there is $|S_{m+1} - S_m| < \varepsilon$. Since for the trapezoidal rule we have

$$L_{2m} = \frac{1}{2}L_m + \frac{b-a}{2m}\Big(f(x_1)+f(x_3)+\cdots+f(x_{2m-1})\Big), \tag{6.6}$$

it is better to stop when for some $m \in \mathbb{N}$ there is $|L_{2m} - L_m| < \varepsilon$.

6.3 Exercises

6.3.1 Prior to starting the algorithms

1. Using the techniques of *Mathematics 1*, i.e. using the antiderivative of the integrand, calculate $\int_{-2}^{6} f(x)\mathrm{d}x$, where

$$f(x) = \begin{cases} -1, & \text{for } x \in \langle -3, 0\rangle, \\ x, & \text{for } x \in \langle 2, 3\rangle, \\ 1, & \text{for } x \in \langle 5, 7\rangle, \\ 0, & \text{otherwise.} \end{cases}$$

2. Calculate

$$\int_5^5 \frac{x^2 + \cos\left(\ln\left(x^5 - 4\right)\right) + \mathrm{e}^x}{x^3 - \sin x}\mathrm{d}x.$$

3. We know that

$$\int_0^1 \mathrm{e}^{-x^2}\mathrm{d}x \doteq 0.746824.$$

   Give the value of

$$\int_{-1}^0 \mathrm{e}^{-x^2}\mathrm{d}x.$$

4. Give an example of a function $f(x)$ and an interval $I = \langle a, b\rangle$ such that $f(x)$ is not continuous on I yet $\int_a^b f(x)\mathrm{d}x$ converges. Calculate it.

5. Let k be the number of letters of your first name and p the number of letters of your surname and c the decimal number $k.p$. Give an example of a function $f(x)$ and an interval $I = \langle a, b\rangle$ such that

$$\int_a^b f(x)\mathrm{d}x = c$$

   Verify your answer by calculating the integral.

6.3.2 Composite trapezoidal and Simpson's rules

Using the composite trapezoidal rule and the composite Simpson rule find $\int_a^b f(x)\mathrm{d}x$. *In the results below the lower index* m *suggests that the interval* $\langle a, b\rangle$ *has been partitioned to* m *subintervals. Since the simple Simpson's method already partitions the interval into two subintervals, the indices for the Simpson method are even only. In the exercises below proceed from* $m = 1$ *(or* $m = 2$ *in case of the Simpson's method), i.e. from the "simple" rule, to* $m = 8$, *i.e. the composite rule partitioning the interval* $\langle a, b\rangle$ *into 8 subintervals.*

1. $f(x) = x^3 + x^2 - x + \frac{1}{x} - \frac{1}{x^2}$, $a = -3$, $b = -1$

   Solution using trapezoidal rule: $x_1 = -16.4444$, $x_2 = -10.9722$, $x_3 = -9.9371$, $x_4 = -9.5717$, $x_5 = -9.4018$, $x_6 = -9.3094$, $x_7 = -9.2536$, $x_8 = -9.2173$

   Solution using the Simpson's rule: $x_2 = -9.1481$, $x_4 = -9.1048$, $x_6 = -$ -9.1002, $x_8 = -9.0992$.

   Modification: Change the interval of integration to $\langle -3, 1 \rangle$.

   Solution: The function is not continuous on the interval in question, i.e. the request is incorrect. From the integral calculus we know that the integral diverges. However, the numerical calculation can easily miss this fact if not specifically asked to check continuity of the integrand on the interval. Notice that e.g. $L_2 = -16.4444$ or $L_6 = -21.7889$ without any signs of possible problems. Do not forget that a problem in this particular exercise can be discovered easily. However, e.g. for terms such as $\frac{1}{ax+\pi}$, where a was obtained numerically, we run into serious problems.

2. $f(x) = \sin x + \cos x + e^x$, $a = -1$, $b = 1$

   Solution using trapezoidal rule: $x_1 = 4.1668$, $x_2 = 4.0834$, $x_3 = 4.0570$, $x_4 = = 4.0469$, $x_5 = 4.0421$, $x_6 = 4.0395$, $x_7 = 4.0378$, $x_8 = 4.0368$

   Solution using the Simpson's rule: $x_2 = 4.0556$, $x_4 = 4.0347$, $x_6 = 4.0336$, $x_8 = 4.0334$

3. $f(x) = x \sin x \cos x$, $a = -1$, $b = 1$

   Solution using trapezoidal rule: $x_1 = 0.9093$, $x_2 = 0.4546$, $x_3 = 0.4405$, $x_4 = = 0.4377$, $x_5 = 0.4367$ $x_6 = 0.4362$, $x_7 = 0.4360$, $x_8 = 0.4358$

   Solution using the Simpson's rule: $x_2 = 0.3031$, $x_4 = 0.4320$, $x_6 = 0.4348$, $x_8 = 0.4352$

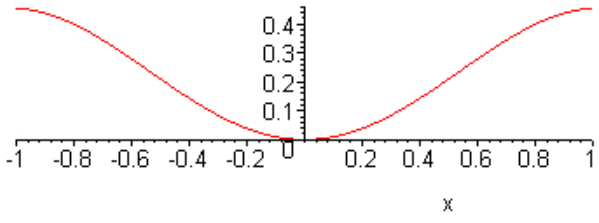


Fig. 6.1: Plots for Exercise 3: Comment on the results for both methods and various m

4. $f(x) = \operatorname{arctg} x$, $a = 2$, $b = 3$

   Solution using trapezoidal rule: $x_1 = 1.1781$, $x_2 = 1.1842$, $x_3 = 1.1853$, $x_4 = = 1.1857$, $x_5 = 1.1859$ $x_6 = 1.1860$, $x_7 = 1.1861$, $x_8 = 1.1861$

Solution using the Simpson's rule: $x_2 = 1.1862$, $x_4 = 1.1863$, $x_6 = 1.1863$, $x_8 = 1.1863$

5. $f(x) = \frac{10}{x^2+4} + 2\sin x$, $a = -3$, $b = -1$

   Solution using trapezoidal rule: $x_1 = 0.8040$, $x_2 = -0.1666$, $x_3 = -0.3338$, $x_4 = -0.3914$, $x_5 = -0.4180$, $x_6 = -0.4323$, $x_7 = -0.4410$, $x_8 = -0.4466$

   Solution using the Simpson's rule: $x_2 = -0.4901$, $x_4 = -0.4664$, $x_6 = -$ -0.4652, $x_8 = -0.4650$

6. $f(x) = \ln \frac{x-4}{sinx} + e^x$, $a = -3$, $b = -1$

   Solution using trapezoidal rule: $x_1 = 6.1038$, $x_2 = 5.0741$, $x_3 = 4.8313$, $x_4 =$ $= 4.7359$, $x_5 = 4.6884$, $x_6 = 4.6614$, $x_7 = 4.6446$, $x_8 = 4.6334$

   Solution using the Simpson's rule: $x_2 = 4.7308$, $x_4 = 4.6231$, $x_6 = 4.6048$, $x_8 = 4.5992$

7. $f(x) = \frac{x}{e^x+\sin x}$, $a = -2$, $b = 2$

   Solution using trapezoidal rule: $x_1 = 5.6502$, $x_2 = 2.8251$, $x_3 = 10.6992$, $x_4 =$ $= 3.8050$, $x_5 = 1.9085$, $x_6 = 6.7910$, $x_7 = 3.4414$, $x_8 = 1.1585$ – *yet* BEWARE*! as these values – although formally correct, are a mere nonsence. Explain.*

8. $f(x) = \frac{x}{e^x}$, $a = 1$, $b = 2$

   Solution using trapezoidal rule: $x_1 = 0.3193$, $x_2 = 0.3270$, $x_3 = 0.3285$, $x_4 =$ $= 0.3291$, $x_5 = 0.3293$, $x_6 = 0.3294$, $x_7 = 0.3295$, $x_8 = 0.3296$

   Solution using the Simpson's rule: $x_2 = 0.32962$, $x_4 = 0.3297$, $x_6 = 0.3298$, $x_8 = 0.3298$

9. $f(x) = x^2 - x + \cos 3x - 4$, $a = 0$, $b = 1$

   Solution using trapezoidal rule: $x_1 = -0.0567$, $x_2 = -0.5539$, $x_3 = -0.6358$, $x_4 = -0.6638$, $x_5 = -0.6767$, $x_6 = -0.6836$, $x_7 = -0.6878$, $x_8 = -0.6906$

   Solution using the Simpson's rule: $x_2 = -0.7197$, $x_4 = -0.7004$, $x_6 = -$ -0.6996, $x_8 = -0.6995$

10. $f(x) = x^2 - x + \cos 3x - 4$, $a = 0$, $b = 2$

    Solution using trapezoidal rule: $x_1 = 0.9302$, $x_2 = 1.0054$, $x_3 = 0.8475$, $x_4 =$ $= 0.7909$, $x_5 = 0.7646$, $x_6 = 0.7502$, $x_7 = 0.7416$, $x_8 = 0.7359$

    Solution using the Simpson's rule: $x_2 = 1.0305$, $x_4 = 0.7194$, $x_6 = 0.7178$, $x_8 = 0.7176$

6.4 Related issues

When performing numerical integration we look for alternative strategies of how to *integrate* a function over an interval. It is therefore important to realize whether integration in the given bounds is possible at all. Regard the following tasks:

1. $\int_0^5 \frac{1}{x-\pi}\mathrm{d}x$
2. $\int_0^{10} f(x)\mathrm{d}x$, where $f(x) = x^2$ for $x \in \mathbb{R} \setminus \mathbb{N}$ and $f(x) = 5x$ for $x \in \mathbb{N}$
3. $\int_{-5}^5 f(x)\mathrm{d}x$, where $f(x) = x^3$ for $x < 0$ and $f(x) = x + 4$ for $x > 0$.

In which cases does the integral converge?

The geometrical meaning and limitations of the respective rules must be also considered. For example, in item 1 the point of discontinuity is such that no reasonable partition of the interval will be such that the $x_i = \pi$ for some i. Therefore, the only way to discover such a problem when algorithmically performing the task is to perform the continuity test before actually calculating the integral. Yet will this be the case? Can we be sure that the command we use in some future version of some future computer software will have the test implemented? Notice that continuity cannot be tested in a purely algorithmic way as every test not based on analytical knowledge of elementary functions is a compromise only.

Also, notice the above functions and try to apply formulas (6.4) and (6.5) on them. If you feel that problems will disappear sooner or later, adjust item 2 so that there are infinitely many points of discontinuity on the given interval (does this have any implications for the analytical way of integration?) or divide the function in case 3 into more segments.

6.5 Numerical integration using Matlab

There is a variety of commands for numerical integration in MATLAB. These usually apply a specific method. For example, the `trapz` command calculates the integral using the (composite) trapezoidal rule while `quad` performs the (composite) Simpson's rule. In these commands, one must enter two vectors – x_i and y_i, where x_i is a partition of interval $\langle a, b\rangle$ and $y_i = f(x_i)$ for the task $\int_a^b f(x)\mathrm{d}x$. The commands may be used for the partition of the interval $\langle a, b\rangle$ with step h (as is assumed in this text) as well as for random partition.

Naturally, we assume that $f(x)$ is integrable on $\langle a, b\rangle$. In numerical calculation of the integral $\int_a^b f(x)\mathrm{d}x$, points of discontinuity might be detect only by chance when a value of x_i equals the point where $f(x)$ is not defined. In MATLAB, when not even a very small

step produces reasonable precision, the user is warned that the function might not be continuous on interval $\langle a, b\rangle$.

6.6 Exam tasks

Below you can find some authentic exam tasks.

1. Calculate
$$\int_{-1}^{1} x + \sin x^2 \mathrm{d}x$$
using the composite trapezoidal rule for $m = 4$. Then do the same using the simple Simpson's rule.

2. Compare numerical results for
$$\int_{0.5}^{1.5} \ln x \mathrm{d}x$$
achieved when using the simple trapezoidal rule and the simple Simpson's rule to the analytical result. Explain any differencies or similarities.

3. With precision $\varepsilon = 0.0001$ apply the composite trapezoidal rule on
$$\int_{0}^{\frac{\pi}{4}} \sin \frac{x}{8} \mathrm{d}x.$$

4. Using a method of your choice calculate
$$\int_{-0.5}^{0.7} 2 \sin 3x \mathrm{d}x.$$
Any result which differs less than 20% from the exact value of the integral will be accepted.

5. Give an example of a function $f(x)$ and bounds $a, b \in \mathbb{N}$, where $a < b$, such that $\int_a^b f(x)\mathrm{d}x$ computed using the simple trapezoidal rule equals 2 and the result computed using the composite trapezoidal rule for $m = 2$ equals the analytical result of the integration (which differs from 2).

6.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, Matlab Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on Matlab Compiler

Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that MATLAB starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of MATLAB output) as far as indexing is concerned.

1. Numerical calculations of the definite integral

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Calculating function values
2. Integration: calculating the definite integral
3. Integration: the primitive function

7 Ordinary differential equations

7.1 Wording the problem

In this part of the subject we deal with initial value problems for both one ordinary differential equation and for systems of ordinary differential equations. In this chapter of the text we focus on the initial value problem for *one ordinary differential equation of the first order*.

We want to find the numerical solution of the initial value problem

$$y' = f(x, y), \quad y(x_0) = y_0, \tag{7.1}$$

where y is a real function of a real variable x.

7.2 A brief summary of the subject matter

You were introduced to the theory of ordinary differential equations (or ODE's, for short) in *Mathematics 2*. Thus you are (among other concepts) familiar with the concepts of *order* of the differential equation and its *solution*. You should also know the difference between the *general* and *particular* solution of an arbitrary ODE. It is to be noted that since in this section we deal with initial value problems formulated as (7.1), we look for a *particular solution* of the given equation, i.e. for one particular function.

When working analytically, we get this function in the form of a function formula. However, when working numerically, we get function values at points chosen before the calculation starts. First of all, we have to identify the interval of our interest – denote it by $I = \langle a, b \rangle$. Then we partition the interval with step h. What we regard as the numerical solution of the initial value problem (7.1) on interval I are in fact pairs $[x_i, y_i]$, for i such that I has been partitioned with step h. In *Mathematics 3* we assume that $x_0 = a$ and denote $x_n = b$, i.e. we aim at completing the following table:

i	0	1	...	n
x_i	x_0	x_1	...	x_n
y_i	y_0	y_1	...	y_n

(7.2)

Values $y_1, \ldots, y_n$ can be calculated using various methods which –generally speaking – fall into two different groups: *single-step* and *multistep methods*. Both of these are discussed in the subject. Furhter on in this text we focus on *single-step methods*.

When using the **Euler method** we, for $i = 0, \ldots n-1$, calculate the values of y_{i+1} from y_i using

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{7.3}$$

In the other methods discussed in this section we do not calculate y_{i+1} directly. We first calculate some interim values (denoted k with a suitable index) and only then calculate y_{i+1} using these. When using the **first modification of the Euler method** we use

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \\ y_{i+1} &= y_i + hk_2, \end{aligned} \tag{7.4}$$

while when using the **second modification of the Euler method** by work with

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + h\,, y_i + hk_1) \\ y_{i+1} &= y_i + \frac{1}{2}h(k_1 + k_2), \end{aligned} \tag{7.5}$$

where in both cases $i = 0, \ldots n-1$.

Strictly speaking, the modifications of the Euler method are the simplest cases of the Runge–Kutta methods. In this text we discuss the **classical Runge–Kutta method**, known also as "RK4". With this we calculate

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1) \\ k_3 &= f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2) \\ k_4 &= f(x_i + h, y_i + hk_3), \end{aligned} \tag{7.6}$$

where again $i = 0, \ldots n-1$.

7.2.1 Existence and uniqueness of solution, stability

What has been discussed above are *algorithms* of finding the numerical solution of the initial value problem (7.1) in the form of the table (7.2). However, there are far more important issues to be considered when looking for the solution: we must be sure that on the interval of our choice the solution does exist and is unique. We must also be sure that the solution we obtain falls within the stability region of the equation.

Stability is not discussed in *Mathematics 3*. When testing existence and uniqueness of solutions we use the same criterion as in *Mathematics 2*.

Theorem 7.1. *If the function $f(x, y)$ is continuous on a rectangle*

$$R = \{(x, y); |x - x_0| \leq a, |y - y_0| \leq b\}, a > 0, b > 0,$$

then there exists a solution of the initial value problem (7.1) on the interval $\langle x_0 - \alpha, x_0 + \alpha \rangle$, where $\alpha = \min(a, \frac{b}{M})$, $M = \max_R |f(x, y)|$. Moreover, if the function $\frac{\partial f(x,y)}{\partial y}$ is bounded on the rectangle R, then this solution is unique.

7.3 Exercises

7.3.1 Prior to starting the algorithms

We have the following ordinary differential equations:

1. $y' = x + y$
2. $y'' + 2y' + 3y = x$
3. $y''' = x^2$
4. $y' = y^2 + \sin(x + y)$
5. $y' + \ln(xy) - x = 0$
6. $y' + \ln x + y = 0$
7. $y' + \sin x \cos y = 0$
8. $y' + y \sin x = x - y$
9. $y'' - y \cos x = 2$
10. $y' - xy = \tan x$

where y is always a function of x. Now do the following:

1. Identify equations of the first order.
2. Identify all equations solvable by methods discussed in *Mathematics 2*.
3. Identify all equations which can – after we add a suitable initial condition – be solved using the single–step methods discussed in *Mathematics 3*.
4. Choose one separable differential equation and find its general solution.
5. Choose one linear differential equation of the first order and find its general solution.

7.3.2 Exercising the algorithms

Find the particular solution of the given ordinary differencial equations on the respective intervals $I = \langle a; b\rangle$ with step h. Find the solution using all the four methods discussed in section 7.2, i.e. using the Euler method, its first and second modifications and using the classical Runge–Kutta method.

Notice that in the exercises below the initial condition $y(x_0) = y_0$ is always such that $x_0 = 0$. This is because in a great number of applications x denotes time. In this respect the initial condition then describes situation in time $x_0 = 0$, i.e. "at the beginning of the experiment". Naturally, we could choose an arbitrary x_0 permitted by Theorem 7.1.

> The exercises below are included for students to be able to exercise the algorithms themselves. The issue of stability of solution of the following initial value problems is not relevant in this respect. The equations are not to be treated as describing real-life problems.

1. $y' = x^2 - y$, $y(0) = 1$, $I = \langle 0; 1\rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	1	0.8	0.648	0.5504	0.5123	0.5379
Euler, 1^{st} modification	1	0.822	0.6912	0.6136	0.5940	0.6363
Euler, 2^{nd} modification	1	0.824	0.6949	0.6186	0.6001	0.6432
Runge–Kutta	1	0.8213	0.6897	0.6112	0.5907	0.6321

2. $y' = x^2 + 4y$, $y(0) = 0$, $I = \langle 0; 1\rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	0	0.0	0.0080	0.0464	0.1555	0.4079
Euler, 1^{st} modification	0	0.002	0.0254	0.1167	0.3743	1.0067
Euler, 2^{nd} modification	0	0.004	0.0317	0.1320	0.4086	1.0813
Runge–Kutta	0	0.0033	0.0334	0.1476	0.4731	1.2926

3. $y' = e^x + y$, $y(0) = 1$, $I = \langle 0; 1\rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	1	1.4	1.9243	2.6075	3.4934	4.6372
Euler, 1^{st} modification	1	1.4610	2.0769	2.8934	3.9691	5.3910
Euler, 2^{nd} modification	1	1.4621	2.0796	2.8983	3.9771	5.3910
Runge–Kutta	1	1.4657	2.0885	2.9154	4.0059	5.4365

4. $y' = e^{x+1} - 2y$, $y(0) = 1$, $I = \langle 0; 1\rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	1	1.1437	1.3502	1.6212	1.9633	2.3879
Euler, 1^{st} modification	1	1.1721	1.3981	1.6848	2.0424	2.4840
Euler, 2^{nd} modification	1	1.1751	1.4038	1.6932	2.0535	2.4983
Runge–Kutta	1	1.1697	1.3940	1.6794	2.0356	2.4759

5. $y' = e^{\frac{x}{2}} + 4y$, $y(0) = 0$, $I = \langle 0; 1\rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	0	0.2	0.581	1.2901	2.5922	4.9644
Euler, 1^{st} modification	0	0.2903	0.9361	2.3391	5.3507	11.7764
Euler, 2^{nd} modification	0	0.2905	0.9370	2.3412	5.3555	11.7871
Runge–Kutta	0	0.3192	1.0622	2.7506	6.5439	15.0193

6. $y' = e^{x^2} + y$, $y(0) = 1$, $I = \langle 0; 1\rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	1	1.4	1.8882	2.5005	3.2873	4.3240
Euler, 1^{st} modification	1	1.442	1.9989	2.7189	3.6722	4.9676
Euler, 2^{nd} modification	1	1.4441	2.0040	2.7291	3.6911	5.0026
Runge–Kutta	1	1.4456	2.0084	2.7379	3.7061	5.0257

7. $y' = e^{x^2} - y$, $y(0) = 1$, $I = \langle 0; 1\rangle$, $h = 0,2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	1	1	1.0082	1.0412	1.1197	1.2750
Euler, 1^{st} modification	1	1.002	1.0197	1.0695	1.1748	1.3750
Euler, 2^{nd} modification	1	1.0041	1.0240	1.0769	1.1873	1.3972
Runge–Kutta	1	1.0026	1.0204	1.0701	1.1754	1.3759

8. $y' = x^3 - 4y$, $y(0) = 1$, $I = \langle 0; 1 \rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	1	0.2	0.0416	0.0211	0.0474	0.1119
Euler, 1^{st} modification	1	0.5202	0.2753	0.1630	0.1361	0.1756
Euler, 2^{nd} modification	1	0.5208	0.2774	0.1671	0.1424	0.1843
Runge–Kutta	1	0.4521	0.2089	0.1137	0.1024	0.1524

9. $y' = \frac{1}{2}x + y$, $y(0) = 2$, $I = \langle 0; 1 \rangle$, $h = 0.2$

   Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	2	2.4	2.9	3.52	4.284	5.2208
Euler, 1^{st} modification	2	2.45	3.021	3.7396	4.6383	5.7568
Euler, 2^{nd} modification	2	2.45	3.021	3.7396	4.6383	5.7568
Runge–Kutta	2	2.4535	3.0295	3.7553	4.6638	5.7956

10. $y' = \frac{1}{10}x + y$, $y(0) = 2$, $I = \langle 0; 1 \rangle$, $h = 0.2$

    Solution: Results corresponding to table (7.2):

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1
Euler	2	2.4	2.884	3.4688	4.1746	5.0255
Euler, 1^{st} modification	2	2.4420	2.9856	3.6533	4.4722	5.4757
Euler, 2^{nd} modification	2	2.4420	2.9856	3.6533	4.4722	5.4757
Runge–Kutta	2	2.4449	2.9928	3.6664	4.4936	5.5083

7.4 Related issues

Not mentioning the issue of the existence and uniqueness of solution of the initial value problem (7.1) and the issue of stability of its solution, the most important thing to consider

when performing the numerical solution of (7.1) may be in short described as the problem of *method – interval – step*. In other words, we have to choose the right method, interval and step so that both the local and global truncation errors are sufficiently small. Notice that in each step of each method we make errors (known as *local truncation errors*) which cumulate into what is known as *global truncation errors*. This issue is thoroughly discussed in the subject.

When solving (7.1) numerically, we get pairs $[x_i, y_i]$ which – in case of one ODE – represent points in plane which are approximations of the exact solution – a planar curve. Thus we have to also consider the issue of converting these points into a curve. This is often done by connecting the points using polygons (linear splines, in fact). However, other techniques such as interpolation polynomials might be employed as well.

7.5 Solution using Matlab

This section of the text goes beyond the subject matter discussed in this chapter as when using professional mathematical sofwtare methods far more advanced then the basic versions of the four methods discussed in section 7.2 are used. However, it is included for the reader to have at least a basic insight in how the topic is treated in Matlab.

Ways of finding solutions of differential equations are discussed in Matlab help typically under `User Guide` > `Mathematics` > `Calculus` > `Ordinary Differential Equations`, resp. `Demos` > `Mathematics` > `Differential Equations`. Matlab uses a set of specialized solvers – in case of ODE initial value problems these are included as commands `odexxx`, in case of ODE boundary value problems as `bvpxxx`, where `xxx` specifies the way of solution.

The input of these solvers is a function defined using techniques discussed in Matlab help typically under `User Guide` > `Mathematics` > `Function Handles`. This function is the right-hand side of the ODE in the explicit form $y'(x) = f(x, y(x))$, or rather such that on the left-hand side there is only the highest order derivative of the unknown function $y(x)$. Moreover, the interval on which we seek the solution as well as the initial or boundary value conditions are the input of the solver.

Before choosing the solver we must analyze the task and based on its type and other information choose the suitable solver and set its input parameters. As far as first order ODE's, or rather their initial value problems, are concerned, `ode45` or `ode23` solvers are based on the Runge-Kutta methods.

7.6 Exam tasks

Below you can find some authentic exam tasks.

1. Using the Euler method solve the initial value problem

$$y' - 2y - x = 0, \quad y(1) = 2.$$

Work on $I = \langle 1, 2 \rangle$ with step $h = 0.2$.

2. Using a method of your choice you aim at solving the initial value problem

$$y' - y - x = \sin(x + y), \quad y(1) = 2.$$

With step $h = 0.2$ find $y(1.4)$. *For the sake of the exam, all methods will be treated as relevant. No considerations regarding precision are required.*

3. Using the classical Runge–Kutta method solve the initial value problem

$$y' = x^2 - \cos y, \quad y(2) = 3.$$

Work on $I = \langle 2, 2.4 \rangle$ with step $h = 0.2$.

4. We have the initial value problem

$$y' = 3x + \sin(x - y) + 2y, \quad y(2) = 3.$$

Compare values $y(2.2)$ obtained with step $h = 0.2$ using the Euler method, both of its modifications and the classical Runge-Kutta method.

5. We have the initial value problem

$$y' = xy + \mathrm{e}^x, \quad y(0) = 1.$$

Calculate $y(0.4)$ with step $h = 0.2$ using a method of your choice. *For the sake of the exam, all methods will be treated as relevant. No considerations regarding precision are required.* Then calculate $y(0.3)$ using a procedure of your choice. Comment your calculations and your choice of strategy.

7.7 Supplementary electronic materials

Executable files prepared with Matlab

Prior to executing these files, MATLAB Compiler Runtime version R2013a, 32-bit pro Windows (400 MB) must be installed. For detailed information on MATLAB Compiler Runtime see help on Mathworks website. Remember that the applications cannot (and do not!) discuss all subtleties of the subject matter. Also, in their explanatory mode, notice that they are algorithms only. Their "intelligence" is thus rather limited. Finally, notice that in some minor aspects of the subject matter they do not exactly follow all theorems included in *Mathematics 3*. Most importantly, the applications were prepared for training

purposes, i.e. "small, reasonable and nice" data are assumed as their input. Do not use numbers with many digits as the space for output display is limited.

Keep in mind that MATLAB starts indexing vectors and matrices form 1 while most of the formulas in *Mathematics 3* start indexing from 0. Therefore, in some applications there are discrepancies between the explanatory mode (which has to follow the subject) and the notation of the results (which are elements of MATLAB output) as far as indexing is concerned.

1. Single-step numerical methods for initial value problems

Maplets

You should perform all tasks strictly in hand. However, the following maplets will help you to verify some auxiliary calculations.

1. Simplification of algebraic expressions
2. Analytical solution of systems of linear equations
3. Determining ODE types
4. Separable ODE
5. First order linear differential ODE
6. Is function an ODE solution?
7. Derivatives
8. Integration

8 Mathematical dictionary

Below you will find a Czech – English / English – Czech dictionary of numerical methods terminology used in *Mathematics 3*. You can also consult an online dictionary [2], which is tailored to use at FEEC BUT and covers more areas of mathematics. However, as far terminology used in numerical methods is concerned, we suggest using this text since it is a revised, rewritten and corrected version of [2], which in fact dates back to 2006 and has not been maintained since. For more translations of terminology use also e.g. [7], where switching between language versions might help. A dictionary with definitions and explanations can be found also in e.g. Maple.

Notice that terminology may vary. British English and American English variants are possible as well.

8.1 Czech – English

absolutní chyba absolute error

Adams–Bashforthovy metody Adams–Bashforth methods, Adams–Bashforth multistep method

Adams–Moultonovy metody Adams–Moulton methods

algebraická rovnice algebraic equation

algebraický polynom algebraic polynomial

algoritmus algorithm, *adj.* algorithmic

aproximace approximation

aproximace funkcí approximation of functions

aproximace křivkou curve fitting

aproximace metodou nejmenších čtverců pomocí přímky, paraboly atd. least-square fit of a straight line, parabola, etc.

aproximace parabolou fit of a parabola

aproximovat approximate, *n.* approximation

aritmetická operace arithmetic operation

Banachova věta o pevném bodu Banach fixed point theorem

Cauchyho (počáteční) úloha Cauchy (initial) value problem

cauchyovská posloupnost Cauchy sequence, *also* fundamental sequence

Cramerovo pravidlo Cramer's rule

částečný výběr hlavního prvku partial pivoting

číslo podmíněnosti úlohy condition number

člen posloupnosti term of a sequence

derivace derivative, *v.* differentiate

derivace řádu n derivative of order n, *also* n–th derivative

derivování differentiation, *n.* derivative

derivovat differentiate, *n.* derivative, *(process)* differentiation

determinant determinant

diagonála diagonal, *adj.* diagonal, *adv.* diagonally

diagonálně dominantní matice diagonally dominant matrix

diagonální dominance diagonal dominance

diference difference

diferenciál funkce differential of a function

diferenciální rovnice differential equation

diferenciální rovnice n–tého řádu n–th order differential equation

diferencovatelná funkce differentiable function

diferenční metoda method of difference, *synonym in Czech* metoda konečných diferencí, metoda sítí

diskretizace discretization

divergence divergence, *adj.* divergent, *v.* diverge

divergentní divergent, *n.* divergence, *v.* diverge

divergentní řada divergent series, *opposite* convergent series

divergovat diverge, *n.* divergence, *adj.* divergent

dobře podmíněný well-conditioned, *opposite* ill-conditioned, ill-posed

dolní trojúhelníková matice lower triangular matrix

druhá derivace second derivative

ekvidistantní equidistant, equispaced

ekvidistantní uzly equally spaced interpolation points, equally spaced nodes

ekvivalentní systém rovnic equivalent system of equations, *n.* equivalence

eliminace elimination, *v.* eliminate

eliminace s výběrem hlavního prvku pivoting, *v.* pivot

euklidovská norma matice Euclidean norm

Eulerova metoda Euler method, Euler's method

explicitní metoda explicit method

extrapolace extrapolation, *v.* extrapolate

Fourierova podmínka Fourier's condition

Gaussova eliminace Gaussian elimination algorithm, Gaussian elimination

Gaussova eliminační metoda Gaussian elimination algorithm, Gaussian elimination

Gauss–Seidelova iterační metoda Gauss–Seidel method, *synonym in Czech* Gauss–Seidelova metoda

Gauss–Seidelova metoda Gauss–Seidel method, *synonym in Czech* Gauss–Seidelova iterační metoda

globálně globally, *adj.* global

globální global, *n.* global

globální (diskretizační) chyba global truncation error, GTE

grafická metoda graphical method

hermitovská matice Hermitian matrix, self-adjoint matrix

hlavní prvek pivot element, *synonym in Czech* vedoucí prvek, pivotový prvek

homogenní homogeneous, *opposite* nonhomogeneous, non-homogeneous

homogenní soustava rovnic homogeneous system of equations, *opposite* nonhomogeneous, non-homogeneous

hornní trojúhelníková matice upper triangular matrix

hranice oblasti domain boundary

chyba error

chyba aproximace approximation error

chyba interpolace interpolation error

implicitní metoda implicit method

indefinitní indefinite

integrace (integrování) integration, *n.* integral, *adj.* integral, *v.* integrate

integrace per partes integration by parts

integrál integral, *adj.* integral, *v.* integrate

integrovaná funkce integrand

integrovat integrate, *n.* integral, *adj.* integral

interpolace interpolation

interpolace algebraickými polynomy polynomial interpolation

interpolační interval interpolation interval

interpolační polynom interpolation polynomial

interpolační polynom v Lagrangeově tvaru Lagrange polynomial, Lagrange's form of the interpolation polynomial

interpolační polynom v Newtonově tvaru Newton polynomial, Newton's form of the interpolation polynomial, Newton's divided differences interpolation polynomial

interpolace pomocí splajnů spline interpolation

inverzní matice inverse, matrix inverse

iterace iteration, *adj.* iterative, *v.* iterate

iterační funkce iterative function

iterační metoda iterative method

iterační metody řešení iterative methods of solution

iterační proces iterative process, recursive process

iterační tvar iterative form, recursive form

Jacobiho iterační metoda Jacobi method, Jacobi's method, *synonym in Czech* Jacobiho metoda

Jacobiho metoda Jacobi method, Jacobi's method, *synonym in Czech* Jacobiho iterační metoda

jednokroková metoda single–step method

kontrakce contraction mapping, *adj.* contractive

kontraktivní contractive, *n.* contraction

konvergence convergence, *adj.* converge, *v.* converge, *opposite* divergence

konvergentní convergent, *n.* convergence, *v.* converge, *opposite* divergent

konvergovat converge, *n.* convergence, *adj.* convergent, *opposite* divergene

kritérium criterion, *pl.* criteria

kritérium konvergence convergence test, convergence criterion, *pl.* criteria

krok algoritmu step

krok (vzdálenost mezi uzly) step

kubický splajn cubic spline

kvadratický splajn quadratic spline

Lagrangeův interpolační polynom Lagrange polynomial, Lagrange's form of the interpolation polynomial, *n.* interpolation, *v.* interpolate

lichoběžníková metoda trapezoid method, trapezoid rule, trapezoidal rule, trapezium rule, *synonym in Czech* lichoběžníkové pravidlo

limita limit

limita funkce limit of a function

lineární splajn linear spline

lokálně locally, *n.* local

lokální local, *adj.* locally

lokální (diskretizační) chyba local truncation error, LTE

LU rozklad LU decomposition, *v.* decompose

matematický model mathematical model

matice přidružená normě matrix corresponding to a norm, *pl.* matrices

matice soustavy matrix of a system, *pl.* matrices

maticová rovnice matrix equation

měření *(výsledek)* measurement, *(proces)* measuring, *adj.* measurable, *v.* measure

metoda ...

- **bisekce** bisection method, *synonym in Czech* půlení intervalů
- **konečných prvků** finite elements method
- **LU rozkladu** LU decomposition method
- **nejmenších čtverců** least squares method, LSM
- **prosté iterace** simple iteration method, simple fixed–point iteration
- **půlení intervalů** bisection method, *synonym in Czech* bisekce
- **regula falsi** false position method, regula falsi method
- **Runge–Kutta** Runge–Kutta method
- **Runge–Kutta 4. řádu** classical Runge–Kutta method, RK4
- **sdružených gradientů** conjugate gradient method
- **sečen** secant method
- **tečen** method of tangents, Newton's method, *synonym in Czech* Newtonova metoda
- **prediktor – korektor** predictor – corrector method, predictor – corrector approach, *v.* predict, correct

metrický prostor metric space

metrika metric, distance function

modifikace Eulerovy metody modification of (the) Euler method

modifikátor corrector modifier, *v.* modify

negativně definitní negative definite, *n.* definiteness

negativně semidefinitní negative semidefinite, *n.* semidefiniteness

nehomogenní nonhomogeneous, non-homogeneous, *opposite* homogeneous

nelineární rovnice nonlinear equation, non-linear equation

neklesající funkce nondecreasing function, non-decreasing function

nerostoucí funkce nonincreasing function, non-increasing function

nesouhlasná maticová norma negatively oriented matrix norm, *opposite* positively oriented matrix norm

nestabilní instable, unstable

Newton–Cotesovy kvadraturní vzorce Newton–Cotes quadrature formulas, *also* formulae, *synonym in Czech* Newton–Cotesův vzorec

Newton–Cotesův vzorec Newton–Cotes formula, *also* formulae, *synonym in Czech* Newton–Cotesovy kvadraturní vzorce

Newtonova metoda Newton method, Newton's method, tangent method, *synonym in Czech* metoda tečen

Newtonův interpolační polynom Newton polynomial, Newton's form of the interpolation polynomial, *n.* interpolation, *v.* interpolate

norma norm

normální rovnice normal equation, equation in the normal form

normovaný vektorový prostor normed space

nulový vektor zero vector

numerická metoda numerical method

numerická stabilita numerical stability

numerické derivování numerical differentiation

numerické integrování numerical integration

numericky stabilní algoritmus numerically stable algorithm

obdélníková metoda rectangular method, rectangular rule, *synonym in Czech* obdélníkové pravidlo

odchylka deviation

ohraničená funkce bounded function, *opposite* unbounded function

okrajová úloha boundary value problem, *synonym in Czech* okrajový problém

okrajové podmínky boundary conditions

okrajový problém boundary value problem, *synonym in Czech* okrajová úloha

parciální derivace partial derivative, *v.* differentiate

parciální derivace n–tého řádu n–th order partial derivative, *v.* differentiate

parciální diferenciální rovnice partial differential equation, PDE

parciální diferenciální rovnice prvního řádu first-order partial differential equation

pásová matice banded matrix

pevný bod fixed point

počáteční aproximace initial approximation, *v.* approximate

počáteční podmínka initial condition, Cauchy condition, *synonym in Czech* Cauchyho podmínka, Cauchyho počáteční podmínka

počáteční problém initial value problem, *synonym in Czech* počáteční úloha

počáteční úloha initial value problem, *synonym in Czech* počáteční problém

podmínka condition, *adj.* conditional, *adv.* conditioned

podmínky konvergence convergence criteria

poloviční krok half step

poměrná diference divided difference

poměrná diference k **–tého řádu** n–th order divided difference, divided difference of order n

pozitivně definitní positive definite, *n.* definiteness

pozitivně semidefinitní positive semidefinite, *n.* semidefiniteness

pravidelná síť equidistant net, *synonym in Czech* ekvidistantní síť

prediktor predictor, *predict*

prostá iterace simple fixed-point iteration, simple iteration, *v.* iterate

primitivní funkce antiderivative

první derivace first derivative, *v.* differentiate

přesné řešení exact solution

přesnost precision, accuracy, *adj.* precise, accurate

přímá metoda direct method

přirozený kubický splajn natural cubic spline

půlení intervalů bisection

regula falsi false position, regula falsi

relativní relative

relativní chyba relative error

relativní chyba vstupu relative entry error

relativní chyba výstupu relative output error

relaxační metoda relaxation, successive over-relaxation, SOR

Richardsonova extrapolace Richardson extrapolation

rostoucí funkce increasing function, *v.* increase

rozklad na kořenové činitele factorization, *v.* factor

rozšířená amtice soustavy augmented matrix, *v.* augment, *pl.* matrices

Rungovy–Kuttovy metody Runge–Kutta method, *(in case of order 4)* classical Runge–Kutta method, RK4

rychlost konvergence rate of convergence

řád diferenciální rovnice order of a differential equation

řád metody order of a method

řádková norma matice row–sum norm

řádkově diagonálně dominantní row diagonally dominant, diagonally dominant with respect to rows, *ostře* ... strictly ...

samoadjungovaný tvar self–adjoint form

sečna secant

separace kořenů root isolation

Simpsonova metoda Simspon's rule, Simpson's method, *synonym in Czech* Simpsonovo pravidlo

síť net, partition of interval

sloupcová norma matice column–sum norm

sloupcově diagonálně dominantní column diagonally dominant, diagonally dominant with respect to columns, *ostře ...* strictly ...

složená lichoběžníková metoda composite trapezoid rule, composite trapezoid method, *synonym in Czech* složené lichoběžníkové pravidlo

složená Simpsonova metda composite Simpson's rule, composite Simpson's method, *synonym in Czech* složené Simpsonovo pravidlo

směrnice slope

směrové pole direction field, slope field

souhlasná maticová norma positvely oriented matrix norm, *opposite* negatively oriented matrix norm

soustava diferenciálních rovnic system of differential equations, *(for ordinary differential equations)* ODE system

soustava lineárních rovnic system of linear equations, linear system

soustava nelineárních rovnic system of non-linear equations

splajn spline

spojitá funkce continuous function, *opposite* discontinuous function, *n.* continuity

spojitost continuity, *adj.* continuous, *opposite* discontinuity

stabilita stability, *adj.* stable, *opposite* instability *(but beware of the concept!)*

stabilita řešení stability of a solution, *adj.* stable

stabilní stable, *n.* stability, *opposite* instable, unstable *(but beware of the concept!)*

střed intervalu midpoint

substituce substitution, *v.* substitute

symetrická matice symmetric matrix, *pl.* matrices

šířka pásu band width

špatně podmíněný ill–conditioned, ill–posed, *opposite* well-conditioned

Taylorův polynom Taylor polynomial

Taylorův rozvoj Taylor expansion, *v.* expand

Taylorova řada Taylor series

triangulace triangulation

trojúhelníková matice triangular matrix, *n.* triangle, *pl.* matrices

třídiagonální matice tridiagonal matrix, *pl.* matrices

úplný metrický prostor complete metric space, *n.* completeness

úplný výběr hlavního prvku complete pivoting, *n.* completeness

určit polohu kořenů localize roots, isolate roots, isolate intervals for roots

určitý integrál definite integral, *n.* integration, *v.* integrate

uzlový bod node, interpolation point, data point

vektorová norma maximum-magnitude norm, uniform-vector norm

vektorový prostor vector space

vícekroková metoda multiple step method

vstupní data initial data, input data

výstupní data output data

zaokrouhlovací chyba round-off error

zbytek v Taylorově rozvoji remainder

zmenšení kroku step–size control

zpětná substituce backward elimination, *v.* eliminate, *synonym in Czech* zpětný chod, zpětná substituce, *opposite* forward elimination

8.2 English – Czech

absolute error absolutní chyba

accuracy přesnost, *adj.* accurate, *(as a mathematical term)* precision, *adj.* precise

Adams–Bashforth methods Adams–Bashforthovy metody, *synonym in English* Adams–Bashforth multistep method

Adams–Moulton methods Adams–Moultonovy

algebraic equation algebraická rovnice

algorithm algoritmus

antiderivative primitivní funkce

approximation error chyba aproximace

approximate aproximovat

approximation aproximace

approximation of functions aproximace funkcí

arithmetic vector space aritmetický vektorový prostor

augmented matrix rozšířená matice soustavy

backward elimination zpětný chod, zpětná substituce, *opposite* forward elimination

band width šířka pásu

banded matrix pásová matice

bisection půlení intervalů, metoda půlení intervalů, bisekce, metoda bisekce

bisection method metoda půlení intervalů, metoda bisekce

boundary conditions okrajové podmínky

boundary value problem okrajový problém, okrajová úloha

bounded function ohraničená funkce

Cauchy problem Cauchyho problém, Cauchyho úloha, počáteční problém, počáteční úloha, *synonym in English* initial value problem

Cauchy sequence cauchyovská posloupnost, *synonym in English* fundamental sequence

classical Runge–Kutta method metoda Runge–Kutty 4. řádu, *synonym in English* RK4

column-sum row sloupcová norma matice

complete metric space úplný metrický prostor

complete pivoting úplný výběr hlavního prvku

composite Simpson's rule složené Simpsonovo pravidlo, složená Simpsonova metoda, *synonym in English* composite Simpson's method

composite trapezoid rule složené lichoběžníkové pravidlo, složená lichoběžníková metoda, *synonym in English* composite trapezoid method

condition podmínka

condition number číslo podmíněnosti úlohy

conjugate gradient method metoda sdružených gradientů

continuity spojitost

continuous spojitý

continuous function spojitá funkce

contraction mapping kontrakce, kontraktivní zobrazení

converge konvergovat, *opposite* diverge

convergence criteria kritéria konvergence

convergence test ověřování podmínek konvergence

convergent konvergentní, *opposite* divergence

convergent series konvergentní posloupnost

corrector korektor

corrector modifier modifikátor

Cramer's rule Cramerovo pravidlo

criterion kritérium

criteria kritéria

cubic spline kubický splajn

curve fitting aproximace křivkou

definite integral určitý integrál

derivative derivace

derivative of order n derivace řádu n, *synonym in English* n–th order derivative

determinant determinant

deviation odchylka

diagonal diagonála

diagonal dominance diagonální dominance

diagonally dominant matrix diagonálně dominantní matice

difference diference

differentiable function diferencovatelná funkce

differential equation diferenciální rovnice

differential (of a function) diferenciál (funkce)

differentiate derivovat

differentiation derivování

direct method přímá metoda

direction field směrové pole, *synonym in English* slope field

discretization diskretizace

diverge divergovat, *opposite* converge

divergence divergence, *opposite* diverge

divergent divergentní, *opposite* convergent

divergent series divergentní řada

divided difference poměrná diference

divided difference of order n poměrná diference n–tého řádu, *synonym in English* n–th order divided difference

divided difference of the first order poměrná diference prvního řádu

domain boundary hranice oblasti

elimination eliminace

equally spaced interpolation points ekvidistantní uzly, *synonym in English* equally spaced nodes

equidistant ekvidistantní, *synonym in English* equispaced

equidistant net pravidelná síť

equivalent system of equations ekvivalentní systém rovnic

error chyba

Euclidean norm euklidovská norma matice

Euler method Eulerova metoda

exact solution přesné řešení

explicit method explicitní metoda

extrapolation extrapolation

factorization rozklad na kořenové činitele

false position metoda regula falsi, *synonym in English* regula falsi method

finite difference method metoda konečných diferencí, metoda konečných prvků, metoda sítí

first derivative první derivace, *v.* differentiate

first-order partial derivative parciální derivace prvního řádu

first-order partial differential equation parciální diferenciální rovnice prvního řádu

fit of a parabola aproximace parabolou

fit of a straight line aproximace přímkou

fixed point pevný bod

Gaussian elimination algorithm Gaussova eliminace, Gaussova eliminační metoda

Gauss–Seidel method Gauss–Seidelova metoda

global globální

global (truncation) error globální (diskretizační) chyba

globally globálně

graphical method grafická metoda

half step poloviční krok

Hermitian matrix hermitovská matice, *synonym in English* self-adjoint matrix

homogeneous homogenní

homogeneous system of equations homogenní soustava rovnic

ill–conditioned špatně podmíněný, *synonym in English* ill–posed

implicit method implicitní metoda

increasing function rostoucí funkce

initial approximation počáteční aproximace

initial condition počáteční podmínka, Cauchyho podmínka, Cauchyho počáteční podmínka, *synonym in English* Cauchy condition

initial data vstupní data, *synonym in English* input data

initial value problem počáteční problém, počáteční úloha

instable nestabilní, *synonym in English* unstable *(but beware of the concept!)*

integral integrál, integrální

integrand integrand, integrovaná funkce

integrate integrovat

integration integrování

integration by parts integrace per partes

interpolation interpolace

interpolation error chyba interpolace

interpolation point uzlový bod, uzel interpolace, *synonym in English* node

interpolation polynomial interpolační polynom

inverse inverzní, inverzní matice, *synonym in English* matrix inverse

isolate intervals for roots určit polohu kořenů, *synonym in English* localize roots, isolate roots

iteration iterace

iterative form iterační tvar, *synonym in English* recursive form

iterative method iterační metoda

iterative methods of solution iterační metody řešení

iterative process iterační proces

Jacobi method Jacobiho metoda, *synonym in English* Jacobi's method

Lagrange polynomial Lagrangeův interpolační polynom, interpolační polynom v Lagrangeově tvaru, *synonym in English* Lagrange's form of the interpolation polynomial

least-square fit of a straight line, parabola, etc. aproximace metodou nejmenších čtverců pomocí přímky, paraboly atd.

limit limita

limit of a function limita funkce

limit of a sequence limita posloupnosti

linear spline lineární splajn

local lokální

local truncation error lokální (diskretizační) chyba, *synonym in English* LTE

lower triangular matrix dolní trojúhelníková matice

LU decomposition LU rozklad

mathematical model matematický model

matrix corresponding to a norm matice přidružená normě

matrix equation maticová rovnice

maximum-magnitude norm vektorová norma, *synonym in English* uniform-vector norm

measurement měření

method metoda

method of differences diferenční metoda, metoda konečných diferencí, metoda sítí

method of least squares metoda nejmenších čtverců, *synonym in English* least-squares method, LSM

method of tangents metoda tečen, Newtonova metoda, *synonym in English* Newton method

metric metrika, *synonym in English* distance function

metric space metrický prostor

midpoint střed intervalu

modification of Euler method modifikace Eulerovy metody

multiple step method vícekroková metoda

natural cubic spline přirozený kubický splajn

negative definite negativně definitní

negative semi-definite negativně semidefinitní

negatively oriented matrix norm nesouhlasně orientovaná maticová norma, *opposite* positively oriented matrix norm

net síť

Newton method Newtonova metoda, *synonym in English* method of tangents, Newton's method

Newton Cotes formula Newton-Cotesův vzorec, *synonym in English* Newton-Cotes quadrature formuals

Newton-Cotes quadrature formulas Newton-Cotesovy kvadraturní vzorce, *synonym in English* Newton-Cotes formula

Newton polynomial Newtonův interpolační polynom, interpolační polynom v Newtonově tvaru, *synonym in English* Newton's form of the interpolation polynomial

nonhomogeneous nehomogenní

nonincreasing function nerostoucí funkce

nonlinear equation nelineární rovnice

norm norma

normal equation normální rovnice, *synonym in English* equation in the normal form

n–th order differential equation diferenciální rovnice n–tého řádu

n–th order partial derivative parciální derivace n–tého řádu

numerical differentiation numerické derivování

numerical integration numerické integrování

numerical stability numerická stabilita

numerically stable algorithm numericky stabilní algoritmus

order of a differential equation řád diferenciální rovnice

partial derivative parciální derivace

partial differential equation parciální diferenciálnír rovnice , *synonym in English* PDE

partial pivoting částečný výběr hlavního prvku

perform a convergence test ověřit podmínky konvergence

pivot element hlavní prvek, vedoucí prvek, pivotový prvek

pivoting eliminace s výběrem hlavního prvku

polynomial interpolation interpolace algebraickými polynomy

positive definite pozitivně definitní

positive semi-definite pozitivně semidefinitní

positively oriented matrix norm souhlasně orientovaná maticová norma, *opposite* negatively oriented matrix norm

precision přesnost

predictor prediktor

predictor – corrector approach metoda prediktor – korektor

quadratic spline kvadratický splajn

rate of convergence rychlost konvergence

rectangular matrix obdélníková matice

rectangular method obdélníková metoda, *synonym in English* rectangular rule

relative relativní

relative entry error relativní chyba vstupu

relative error relativní chyba

relative output error relativní chyba výstupu

relaxation relaxační metoda

remainder zbytek *(v Taylorově rozvoji)*

Richardson extrapolation Richardsonova extrapolace

root isolation separace kořenů

round-off error zaokrouhlovací chyba

row-sum norm řádková norma matice

Runge-Kutta method metoda Runge-Kutty, Runge–Kuttovy metody, *synonym in English* RK4

secant sečna

secant method metoda sečen

second derivative druhá derivace

self-adjoint matrix samoadjungovaný tvar

simple fixed–point iteration metoda prosté iterace, *synonym in English* simple iteration method

Simpson's rule Simpsonovo pravidlo, Simpsonova metoda, *synonym in English* Simpson's method

single-step method jednokroková metoda

slope směrnice

spline splajn

stability stabilita

stability of a solution stabilita řešení

stable stabilní

step krok *(vzdálenost mezi uzly)*, krok algoritmu

step-size control zmenšení kroku

Sturm's sequence Sturmova posloupnost

substitution substituc

successive over–relaxation relaxační metoda *(řešení soustav lineárních rovnic)*

symmetric matrix symetrická matice

system of differential equations soustava diferenciálních rovnic

system of linear equations soustava lineárních rovnic

system of nonlinear equations soustava nelineárních rovnic

Taylor expansion Taylorův rozvoj

Taylor polynomial Taylorův polynom

Taylor series Taylorova řada

term of a sequence člen posloupnosti

trapezoid method lichoběžníková metoda, lichoběžníkové pravidlo, *synonym in English* trapezoidal rule, trapezium rule

triangular matrix trojúhelníková matice

triangulation triangulace

tridiagonal matrix třídiagonální matice

upper triangular matrix horní trojúhelníková matice

vector space vektorový prostor

well–conditioned dobře podmíněný

zero vector nulový vektor

Bibliography

[1] B.Fajmon, I. Hlavičková, M. Novák, *Mathematics 3 (Electrica, Electronic, Communication and Control Technology)*, Brno, VUT, 2014. (teaching text)

[2] M. Novák, P. Langerová, *English – Czech / Czech – English dictionary of mathematical terminology*

[3] M. Novák, *Matematika 3: Sbírka příkladů z numerických metod*, Brno, VUT, 2010. (accessible online after logging in the BUT information system)

[4] M. Novák, *Mathematics 3 (komentovaná zkoušková zadání pro kombinovanou formu studia)*, Brno, VUT, 2014.

[5] Courses at Brno University of Technology

[6] Texts and supplementary teaching materials for other mathematical subjects

[7] Wikipedia, The Free Encyclopedia