



JOINT INSTITUTE FOR NUCLEAR RESEARCH

LABORATORY OF INFORMATION TECHNOLOGIES

FINAL REPORT ON THE SUMMER STUDENT PROGRAM

INVESTIGATION OF EFFECTIVE METHODS AND THEIR
STABILITY FOR SOLVING BAND MATRIX SLES AFTER
ELLIPTIC AND PARABOLIC NONLINEAR PDES

Supervisor:

Alexander Ayriyan

Student:

Milena Plamenova Veneva

Bulgaria, Sofia Univerity

“St. Kliment Ohridski”

Participation period:

June 04–July 13

Dubna, 2017

JOINT INSTITUTE FOR NUCLEAR RESEARCH
Laboratory of Information Technologies

Abstract

FINAL REPORT ON THE SUMMER STUDENT PROGRAM

from Milena Plamenova Veneva

Bulgaria, Sofia Univerity

“St. Kliment Ohridski”

Part 1. A model which consists of an elliptic partial differential equation and Dirichlet boundary conditions is considered. Three different finite difference methods (FDM) are suggested for its discretization. The resulting systems of linear equations are solved, using the successive overrelaxation (SOR) method. For that purpose, the conditions for its stability and convergence are found, using analytical and numerical approaches. The analytical approach requires the usage of a spectrum analysis, while for the numerical one a golden section search method is applied so as the number of the SOR iterations to be minimized. The results from the two technics show a negligibly small difference. Comparison between the obtained results of the investigated FDM, as well as their features, is made. Their upsides and downsides are discussed.

Part 2. A class of models of heat transfer processes in a multilayer domain is considered. The governing equation is a nonlinear heat-transfer equation with different temperature-dependent densities and thermal coefficients in each layer. Homogeneous Neumann boundary conditions and ideal contact ones are applied. A finite difference scheme on a special uneven mesh with a second-order approximation in the case of a piecewise constant spatial step is built. This discretization leads to a pentadiagonal system of linear equations (SLEs) with a matrix which is neither diagonally dominant, nor positive definite. Three different methods for solving such a SLE are developed – diagonal dominantization, symbolic algorithms, and iterative procedure. Computer simulations are conducted and the results from them are summarized and discussed.

Contents

Abstract	i
Table of Contents	ii
List of Figures	iv
List of Tables	v
1 Introduction	1
1.1 Introduction to the First Part	1
1.2 Introduction to the Second Part	1
1.3 Approbation of the Results and Publications	2
1.3.1 Approbation of the Results	2
1.3.2 Publications	2
1.4 Outline of the Report	2
2 Investigation of Numerical Methods for Solving Elliptic Equations with High Order of Approximation	4
2.1 The Problem – Mathematical Model	4
2.2 Discretization of the Problem	5
2.2.1 Five-point Stencil Difference Method	5
2.2.2 Five-point Stencil Difference Method with an Increased Order of Approximation	6
2.2.2.1 Algorithm	7
2.2.3 Nine-point Stencil Difference Method	8
2.3 Matrix Form of the Discretized Problem	10
2.4 Successive Overrelaxation Method	10
2.5 Optimum Relaxation Factor Problem for Elliptic PDEs	11
2.5.1 Optimum ω Problem - Numerical Approach	12
2.5.1.1 The Idea	12
2.5.1.2 Golden Section Search	12
2.5.1.3 Test Functions	13
2.5.1.4 Results	13
2.5.2 Optimum ω Problem - Analytical Approach	17
2.5.2.1 Spectrum Analysis	18
2.5.3 Optimum ω – Results	20
2.6 Number of SOR Iterations	21
2.7 Runge’s Rule for a Practical Estimation of the Error	21

2.8	Comparison between the Methods	24
3	Investigation of Effective Methods and Their Stability for Solving Band Matrix SLEs after Parabolic Nonlinear PDEs	26
3.1	Mathematical Model	26
3.2	Discretization of the Problem	27
3.3	Numerical and Symbolic Algorithms	28
3.4	Implementation and Results	29
3.5	Additional Results	30
3.6	Iterative Approach	31
3.6.1	Matrix Implementation	32
3.6.2	Array Implementation	32
3.6.3	Comparison between the Implementations	34
4	Discussion, Conclusions, and Future Work	35
4.1	Conclusions and Discussions	35
4.2	Future Work	36
A	Appendix A	37
A.1	Diagonal Dominantization for the Tridiagonal Matrix	37
	Acknowledgements	39
	Bibliography	40

List of Figures

2.1	Five-point stencil.	6
2.2	Nine-point stencil.	9
2.3	Golden section method.	12
2.4	Exact solution u_1^*	14
2.5	Exact solution u_2^*	14
2.6	Right-hand side f_1	15
2.7	Right-hand side f_2	15
2.8	Optimum ω for the five-point stencil method.	16
2.9	Optimum ω for the five-point stencil method with an increased order of approximation.	16
2.10	Optimum ω for the nine-point stencil method.	17
2.11	α coefficient for the five-point stencil method.	22
2.12	α coefficient for the five-point stencil method with an increased order of approximation.	23
2.13	α coefficient for the nine-point stencil method.	23
2.14	Comparison between the five-point stencil method with an increased order of approximation and the nine-point stencil method	24
3.1	Example of a rectangular domain in cylindrical coordinates (a longitudinal section of a multilayer cylinder), where the thermal coefficients are different in all the subdomains and they have a discontinuity of the first kind on the borders in-between the subdomains.	27

List of Tables

2.1	Optimum ω for the SOR method.	20
2.2	Optimum ρ for the SOR method.	20
2.3	Optimum ρ_{ACF} for the SOR method.	20
2.4	Number of SOR iterations.	21
2.5	Runge’s rule for a practical estimation of the error – results.	22
2.6	Comparison between the methods.	24
3.1	Results from solving SLE	30
3.2	Results from solving SLE on the cluster “HybriLIT”.	31
3.3	Results from the ILU(0) method and the numerical method for inverting matrices (matrix implementation), using the cluster “HybriLIT”.	32
3.4	Results from solving SLE, using SIP (matrix implementation), on the cluster “HybriLIT”.	32
3.5	Number of Iterations Needed for the Stone and the Hotelling-Bodewig methods (matrix implementation), using the cluster “HybriLIT”.	33
3.6	Results from the ILU(0) method and the numerical method for inverting matrices (array implementation), using the cluster “HybriLIT”.	33
3.7	Results from solving SLE, using SIP (array implementation), on the cluster “HybriLIT”.	33
3.8	Number of Iterations Needed for the Stone and the Hotelling-Bodewig methods (array implementation), using the cluster “HybriLIT”.	34

Chapter 1

Introduction

1.1 Introduction to the First Part

Elliptic partial differential equations can arise in many different fields, e.g.:

- Electrostatics: to describe the electrostatic potential field caused by a given electric charge;
- Astronomy: to describe Newtonian gravity;
- Fluid dynamics: to describe the fluid potential;
- Thermodynamics: to describe the steady-state heat equation.

For that reason, methods for their numerical solving with a high order of approximation are needed. Here, we investigate finite difference methods (FDM) for the approximation of the partial differential equation. As a result from these approximating schemes, systems of linear equations (SLEs) are obtained. In order to solve them, we consider successive overrelaxation (SOR) method and conditions for its stability.

1.2 Introduction to the Second Part

Systems of linear algebraic equations (SLEs) with pentadiagonal (PD) and tridiagonal (TD) coefficient matrices arise after discretizing partial differential equations (PDEs), using finite difference methods (FDM) or finite element methods (FEM). Methods for numerical solving of SLEs with such matrices which take into account the band structure of the matrices are needed. The methods known in the literature usually require

the matrix to possess special characteristics so as the method to be stable, e.g. diagonally dominance, positive definiteness, etc. which are not always feasible. For instance, such a problem was solved in [1]. There, a finite difference scheme with first-order approximation was built that leads to a TD SLE with a diagonally dominant coefficient matrix. The system was solved using the Thomas method ([2]). However, a difference scheme with second-order approximation leads to a matrix which does not have any of the above-mentioned special characteristics. Our aim is to build and explore effective methods and their stability which methods are not that restrictive. For that reason, we consider three different approaches: diagonal dominantization, symbolic algorithms, and iterative procedure.

1.3 Approbation of the Results and Publications

1.3.1 Approbation of the Results

Parts of the report were presented at the following scientific seminars and conferences:

- Chapter 2 – Seminar of the Sector of Methods for Modelling Nonlinear Systems of the Division of Computational Physics at the Laboratory of Information Technologies, Joint Institute for Nuclear Research;
- parts of Chapter 3 – International Conference “Mathematical Modeling and Computational Physics, 2017” (MMCP2017), 3–7 July 2017, Dubna, Russia (during the poster session).

1.3.2 Publications

Parts of Chapter 3 of this report are about to be published, as follows:

- Milena Veneva and Alexander Ayriyan, *Effective Methods for Solving Band SLEs after Parabolic Nonlinear PDEs*, submitted to European Physics Journal – Web of Conferences (EPJ-WoC), arXiv: 1710.00428 [math.NA].

1.4 Outline of the Report

The layout of the report is as follows:

- Chapter 1 is an introduction to the topics that are considered in this report;

-
- in Chapter 2 different FDM with a high order of approximation for numerical solving of elliptic PDEs are investigated. Comparison between the considered methods is made. A thorough study of SOR method and its stability is conducted;
 - in Chapter 3 different methods for solving pentadiagonal and tridiagonal SLE are built. Numerical experiments are conducted and the results are compared;
 - Chapter 4 draws the conclusions of the whole report and presents some insights for future work on the same topics;
 - in Appendix A the formulas for the diagonal dominantization used in Chapter 3 are written in details.

Chapter 2

Investigation of Numerical Methods for Solving Elliptic Equations with High Order of Approximation

2.1 The Problem – Mathematical Model

Let us have the following domain:

$$\bar{G} = G \cup \Gamma = \{(x, z) : x_{left} \leq x \leq x_{right}, z_{left} \leq z \leq z_{right}\}.$$

A solution of the following problem is searched over the already defined domain:

- governing equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} = f(x, z), \quad (x, z) \in G; \quad (2.1)$$

- Dirichlet boundary conditions:

$$u|_{\Gamma} = u_{exact}. \quad (2.2)$$

Equation (2.1) is a nonhomogeneous partial differential equation of an elliptic type (Poisson's equation). This equation is a generalization of Laplace's equation $\Delta u = 0$, where Δ stands for the Laplace operator. Dirichlet boundary conditions (2.2) are applied, i.e. the exact solution is known on the boundary of the considered domain.

2.2 Discretization of the Problem

Using the following mesh $\overline{\omega}_h = \omega_h \cup \gamma_h$, where:

$$\overline{\omega}_h = \left\{ x_i = ih_x, i = \overline{0, N}, h_x = \frac{1}{N}, z_j = jh_z, j = \overline{0, M}, h_z = \frac{1}{M} \right\},$$

and applying the central-point difference scheme with a second-order approximation:

$$\Lambda_x y_{i,j} = \frac{y_{i-1,j} - 2y_{i,j} + y_{i+1,j}}{h_x^2} + O(h_x^2), \quad (2.3)$$

the considered problem (2.1)–(2.2) can be discretized as follows:

$$\Lambda y_{i,j} = f_{i,j} + O(h_x^2 + h_z^2); \quad (2.4)$$

$$y_{i,j} = u_{exact}(x_i, z_j), \quad (x, z) \in \Gamma, \quad (2.5)$$

where

$$\Lambda y_{i,j} := \Lambda_x y_{i,j} + \Lambda_z y_{i,j}. \quad (2.6)$$

Remark: from this point on, everywhere in this Chapter 2 we are going to assume that we have a homogeneous spatial step in both the directions, so as: $h_x = h_z = h$.

2.2.1 Five-point Stencil Difference Method

Using the five-point stencil which can be seen in Figure 2.1:

$$St(x_i; z_j) = \{(x_i; z_{j-1}), (x_{i-1}; z_j), (x_i; z_j), (x_{i+1}; z_j), (x_i; z_{j+1})\}, \quad (2.7)$$

Equation (2.4) has the following form:

$$\Lambda_x y_{i,j} + \Lambda_z y_{i,j} = f_{i,j} + O(h^2). \quad (2.8)$$

Then, in canonic form the five-point stencil method is:

$$y_{i,j} = \frac{1}{4} \left[y_{i,j-1} + y_{i-1,j} + y_{i+1,j} + y_{i,j+1} - h^2 f_{i,j} \right], \quad i, j = \overline{1, \dots, N-1}. \quad (2.9)$$

This method has a second-order approximation.

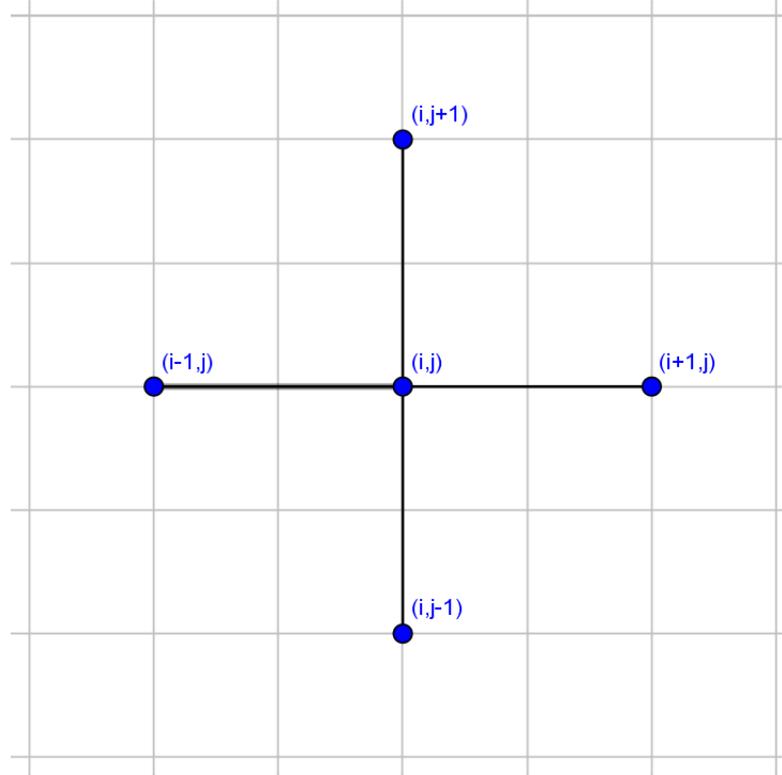


FIGURE 2.1: Five-point stencil.

2.2.2 Five-point Stencil Difference Method with an Increased Order of Approximation

Applying Taylor series to the function u taken in the points $x_{i\pm 1}$ around the point x_i (for any z), it follows that:

$$u_{i+1} = u_i + h_x \left. \frac{\partial u}{\partial x} \right|_{x=x_i} + \frac{h_x^2}{2!} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x=x_i} + \frac{h_x^3}{3!} \left. \frac{\partial^3 u}{\partial x^3} \right|_{x=x_i} + \frac{h_x^4}{4!} \left. \frac{\partial^4 u}{\partial x^4} \right|_{x=x_i} + \frac{h_x^5}{5!} \left. \frac{\partial^5 u}{\partial x^5} \right|_{x=x_i} + O(h_x^6); \quad (2.10)$$

$$u_{i-1} = u_i - h_x \left. \frac{\partial u}{\partial x} \right|_{x=x_i} + \frac{h_x^2}{2!} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x=x_i} - \frac{h_x^3}{3!} \left. \frac{\partial^3 u}{\partial x^3} \right|_{x=x_i} + \frac{h_x^4}{4!} \left. \frac{\partial^4 u}{\partial x^4} \right|_{x=x_i} - \frac{h_x^5}{5!} \left. \frac{\partial^5 u}{\partial x^5} \right|_{x=x_i} + O(h_x^6). \quad (2.11)$$

Summing Equations (2.9) and (2.10) up and dividing the sum by $h_x^2 \Rightarrow$

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{x=x_i} = \frac{u_{i-1} + 2u_i + u_{i+1}}{h_x^2} - \frac{h_x^2}{4!} \left. \frac{\partial^4 u}{\partial x^4} \right|_{x=x_i} + O(h_x^4). \quad (2.12)$$

Using the same technic with respect to $z \Rightarrow$

$$\left. \frac{\partial^2 u}{\partial z^2} \right|_{z=z_j} = \frac{u_{j-1} + 2u_j + u_{j+1}}{h_z^2} - \frac{h_z^2}{4!} \left. \frac{\partial^4 u}{\partial z^4} \right|_{z=z_j} + O(h_z^4). \quad (2.13)$$

Going to the discrete function y and substituting Equations (2.12) and (2.13) into (2.8), it follows that:

$$\Lambda_x y_{i,j} + \Lambda_z y_{i,j} = f_{i,j} + \underbrace{\frac{h_x^2}{12} \left(\frac{\partial^4 u}{\partial x^4} \right)_{i,j} + \frac{h_z^2}{12} \left(\frac{\partial^4 u}{\partial z^4} \right)_{i,j}}_{:=\varphi_{i,j}} + O(h_x^4 + h_z^4), \quad (2.14)$$

where $\varphi_{i,j}$ in Equation (2.14) can be derived, differentiating the governing equation (2.1) twice with respect to x and twice with respect to z :

$$\varphi_{i,j} = \frac{h_x^2}{12} \left(\frac{\partial^2 f}{\partial x^2} \right)_{i,j} + \frac{h_z^2}{12} \left(\frac{\partial^2 f}{\partial z^2} \right)_{i,j} - \left(\frac{h_x^2}{12} + \frac{h_z^2}{12} \right) \left(\frac{\partial^4 u}{\partial x^2 \partial z^2} \right)_{i,j}. \quad (2.15)$$

Discretizing the last term in Equation (2.15), using the central-point difference scheme with a second-order approximation, and taking the remark from the end of Section 2.2, the following equation for $\varphi_{i,j}$ is obtained:

$$\begin{aligned} \varphi_{i,j} = & \frac{h^2}{12} \Delta f_{i,j} - \\ & - \frac{1}{6h^2} (y_{i-1,j-1} + y_{i+1,j-1} + y_{i-1,j+1} + y_{i+1,j+1} - \\ & - 2(y_{i,j-1} + y_{i-1,j} + y_{i+1,j} + y_{i,j+1}) + 4y_{i,j}). \end{aligned} \quad (2.16)$$

Remark: If we do not have the function f in an analytic form, the central-point difference scheme with a second-order approximation can be used so as the Laplacian of f to be approximated.

Thus, the difference scheme has the following form:

$$\Lambda_x y_{i,j} + \Lambda_z y_{i,j} = f_{i,j} + \varphi_{i,j} + O(\|h\|^4). \quad (2.17)$$

2.2.2.1 Algorithm

In the case of the five-point stencil difference method with an increased order of approximation the computation algorithm is as follows:

1. using the five-point stencil method (2.8) (in canonic form – (2.9)), an initial approximation \tilde{y} of the searched solution u^* is found. The error of this approximation

is $O(\|h\|^2)$, i.e.

$$\tilde{y}_{i,j} = u^*(x_i, z_j) + O(\|h\|^2); \quad (2.18)$$

2. as a next step, using \tilde{y} , a number of corrections to the right-hand side is made. More precisely, the procedure is the following:

- (a) $\varphi(\tilde{y}_{i,j})$ is calculated and added to the right-hand side;
- (b) the approximating function \hat{y} of the solution u^* is calculated, substituting the corrected right-hand side into the five-point stencil method's Equation (2.8):

$$\hat{y}_{i,j} = u^*(x_i, z_j) + O(\|h\|^4). \quad (2.19)$$

Then, the procedure is repeated. The number of the needed corrections so as a prescribed margin error to be satisfied is usually ≤ 2 .

Although f is approximated, using a second-order scheme, the overall approximation error of the method is $O(\|h\|^4)$, because this second-order approximation is used only as an initial value. What is peculiar about this method is that a five-point stencil is used for the Laplacian and a nine-point one for the right-hand side.

Remark: In the Russian literature the five-point stencil difference method with an increased order of approximation is known as the Volkov method.

2.2.3 Nine-point Stencil Difference Method

Using the nine-point stencil which can be seen in Figure 2.2:

$$St(x_i; z_j) = \{(x_{i-1}; z_{j-1}), (x_i; z_{j-1}), (x_{i+1}; z_{j-1}), (x_{i-1}; z_j), (x_i; z_j), (x_{i+1}; z_j), \\ (x_{i-1}; z_{j+1}), (x_i; z_{j+1}), (x_{i+1}; z_{j+1})\}, \quad (2.20)$$

and applying Taylor series like we did in Equations (2.14) and (2.15), but moving the fourth derivative of u to the left-hand side (LHS), the following difference scheme is obtained ([3]):

$$\Lambda_x y_{i,j} + \Lambda_z y_{i,j} + \left(\frac{h_x^2}{12} + \frac{h_z^2}{12} \right) \Lambda_x \Lambda_z y_{i,j} = f_{i,j} + \frac{h_x^2}{12} \Lambda_x f_{i,j} + \frac{h_z^2}{12} \Lambda_z f_{i,j} + O(h_x^4 + h_z^4). \quad (2.21)$$

Sticking to the remark from the end of Section 2.2, it follows:

$$\Lambda_x y_{i,j} + \Lambda_z y_{i,j} + \frac{h^2}{6} \Lambda_x \Lambda_z y_{i,j} = f_{i,j} + \frac{h^2}{12} \Lambda_x f_{i,j} + \frac{h^2}{12} \Lambda_z f_{i,j} + O(\|h\|^4). \quad (2.22)$$

In canonic form the nine-point stencil method has the form:

$$\begin{aligned}
 y_{i,j} = \frac{1}{40} & \left[8(y_{i,j-1} + y_{i-1,j} + y_{i+1,j} + y_{i,j+1}) + \right. \\
 & + 2(y_{i-1,j-1} + y_{i+1,j-1} + y_{i-1,j+1} + y_{i+1,j+1}) - \\
 & \left. - 12h^2 f_{i,j} - h^4 \Delta f_{i,j} \right], \quad i, j = \overline{1, \dots, N-1}.
 \end{aligned}
 \tag{2.23}$$

The overall approximation error of the method is $O(\|h\|^4)$.

Remark 1: On the order of what was said in Subsection 2.2.2, if we do not have the

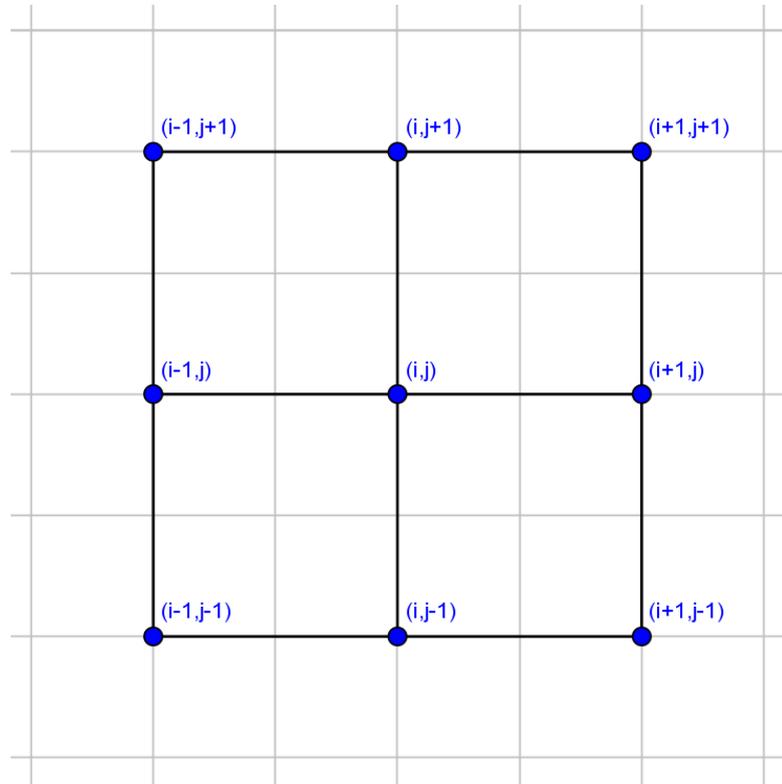


FIGURE 2.2: Nine-point stencil.

function f in an analytic form, the central-point difference scheme with a second-order approximation can be used so as the Laplacian of f to be approximated, namely:

$$\Delta f_{i,j} = \frac{f_{i-1,j} - 2f_{i,j} + f_{i+1,j}}{h_x^2} + \frac{f_{i,j-1} - 2f_{i,j} + f_{i,j+1}}{h_z^2},
 \tag{2.24}$$

or when $h_x = h_z = h$, then:

$$\Delta f_{i,j} = \frac{f_{i,j-1} + f_{i-1,j} - 4f_{i,j} + f_{i+1,j} + f_{i,j+1}}{h^2}.
 \tag{2.25}$$

Remark 2: In the Russian literature the nine-point stencil difference method is known as the Samarskii method or the “jashchik” (from the Russian word for “box”) method.

previous iterate and the computed Gauss-Seidel iterate successively for each component:

$$x_i^{(k)} = \omega \bar{x}_i^{(k)} + (1 - \omega) x_i^{(k-1)}, \quad (2.27)$$

where \bar{x} denotes a Gauss-Seidel iterate and ω is the extrapolation (relaxation) factor. The method was devised simultaneously by two authors – [4] and [5]. If $\omega = 1$, the SOR method simplifies to the Gauss-Seidel method. In a matrix form the SOR method is the following [6]:

$$A^- x_{k+1} + A^+ x_k + D \left(\frac{1}{\omega} x_{k+1} + \left(1 - \frac{1}{\omega}\right) x_k \right) = b, \quad k = 0, 1, \dots \quad (2.28)$$

or

$$\left(A^- + \frac{1}{\omega} D \right) (x_{k+1} - x_k) + A x_k = b, \quad k = 0, 1, \dots, \quad (2.29)$$

where A^- , D , and A^+ are the strictly lower-triangular, diagonal, and strictly upper-triangular parts of A , respectively.

A theorem from [7] shows that SOR fails to converge if ω is outside the interval $(0, 2)$:

Theorem 2.1. (*Ostrowski*) *If A is symmetric positive-definite matrix, $B = \left(A^- + \frac{1}{\omega} D\right) > \frac{1}{2}A$, and B is invertible, then the SOR method is convergent with a rate of a geometric progression.*

Hence, the condition so as the SOR method to be convergent is $\omega \in (0; 2)$. If $\omega > 1$, the number of iterations is $O(N)$. For that reason, we are going to consider $\omega \in (1; 2)$.

2.5 Optimum Relaxation Factor Problem for Elliptic PDEs

The idea is to choose a value for ω that will accelerate the rate of convergence of the iterates to the solution. Let us define a function $n_{it}(\omega)$ of ω . It takes values in the interval $(1; 2)$ and gives the number of needed SOR iterations, i.e.:

$$n_{it}(\omega) : (1; 2) \rightarrow \mathbb{N}. \quad (2.30)$$

We are looking for the value $\omega \in (1; 2)$ which minimizes the number of SOR iterations, i.e.:

$$\min_{\omega \in (1; 2)} n_{it}. \quad (2.31)$$

Hence, this is an optimization problem for finding the relaxation factor which minimizes the number of iterations needed so as the SOR method to be convergent to the solution.

From this moment on, the ω which satisfies this condition is going to be called “optimum relaxation factor” or “ ω^{opt} ”.

Two different approaches to this optimization problem are suggested – numerical and analytical. Comparison between the two approaches is conducted.

2.5.1 Optimum ω Problem - Numerical Approach

2.5.1.1 The Idea

The idea of the numerical approach is to divide the interval (1;2) by a number of points and then to calculate the solution of the problem (2.1)–(2.2), using the three suggested discretization technics for all these values of ω . Then, the minimum functional values of the function $n_{it}(\omega)$ for all the three methods are searched. For that purpose, an algorithm for seeking an extremum is needed. The one we use is the golden section search.

2.5.1.2 Golden Section Search

The golden section search is a technique for finding the extremum of a strictly unimodal function by successively narrowing the range of values inside which the extremum is known to exist ([8]). Let, for instance, $n_{it}(\omega)$ is defined in $[\omega_1; \omega_3]$ and the extremum which is sought is minimum. Then, the procedure is the following (see Figure 2.3):

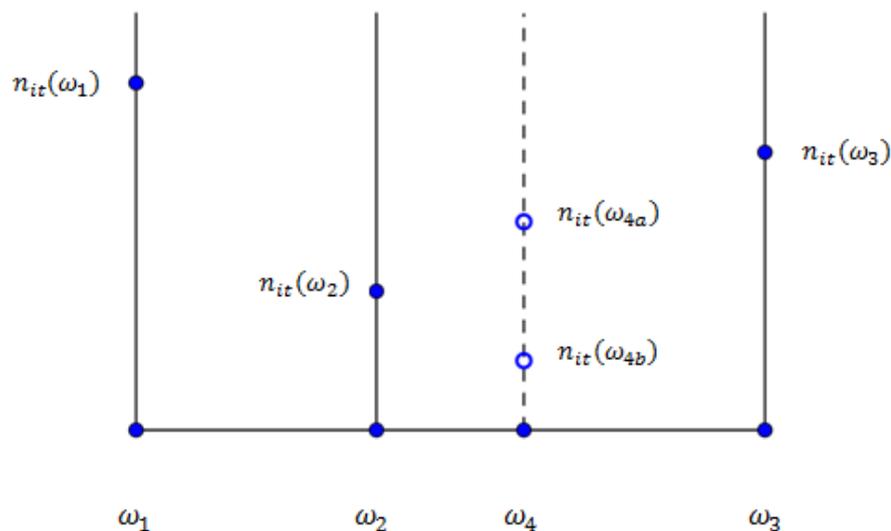


FIGURE 2.3: Golden section method.

- $\omega_2 \in [\omega_1; \omega_3]$ is chosen so that:

$$\frac{\omega_3 - \omega_1}{\omega_3 - \omega_2} = \varphi, \quad \text{where} \quad \varphi = \frac{1 + \sqrt{5}}{2}.$$

- if $(n_{it}(\omega_2) < n_{it}(\omega_1) \ \&\& \ n_{it}(\omega_2) < n_{it}(\omega_3))$, then the $\min \in [\omega_1; \omega_3]$;
- we make a probe in the larger interval (because it is more optimal). ω_4 is chosen so that:

$$\frac{\omega_3 - \omega_1}{\omega_4 - \omega_1} = \varphi;$$

- if $(n_{it}(\omega_4) > n_{it}(\omega_2))$, then the $\min \in [\omega_1; \omega_4]$;
else the $\min \in [\omega_2; \omega_3]$;
- repeat the procedure with the new interval;
- if the length of the interval is $< \varepsilon \Rightarrow \text{STOP}$.
- Complexity of the algorithm: $O(\log(\frac{1}{\varepsilon}))$, where ε is the margin error.

Remark 1: If one starts out with a triplet whose segments are not in the golden ratios, the procedure of choosing successive points at the golden mean point of the larger segment will quickly converge you to the proper ratios.

2.5.1.3 Test Functions

So as to find $\omega_{\text{numerical}}^{\text{opt}}$, we consider two different exact solutions of the problem (2.1)–(2.2) u_1^* and u_2^* . They lead to two different right-hand sides f_1 and f_2 :

$$\begin{aligned} u_1^*(x, z) &= x^6 + z^6; & f_1(x, z) &= 30x^4 + 30z^4; \\ u_2^*(x, z) &= (x - 1)^2(z - 1)^2; & f_2(x, z) &= 2(x - 1)^2 + 2(z - 1)^2. \end{aligned}$$

The graphics of these four functions are depicted in Figures 2.4, 2.6, 2.5, and 2.7, respectively.

2.5.1.4 Results

Graphics of the numerically evaluated function $n_{it}(\omega)$ against ω for all the three discretization technics are depicted in Figures 2.8, 2.9, and 2.10, respectively. The optimum values of $\omega_{\text{numerical}}^{\text{opt}}$ are summarized in Table 2.1.

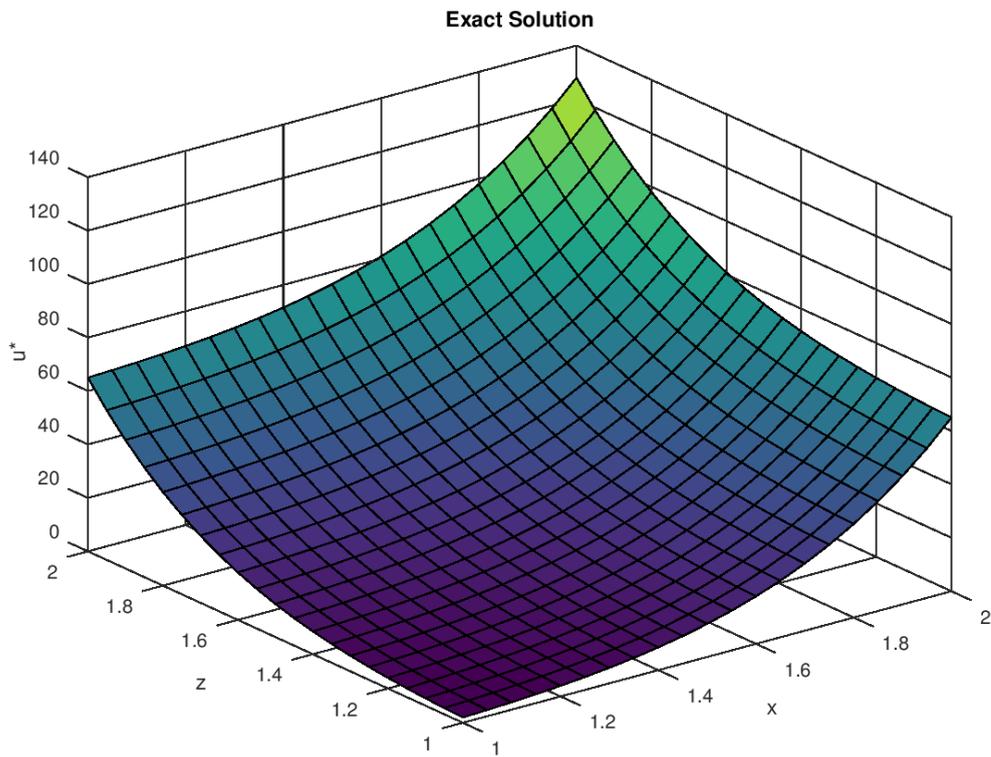


FIGURE 2.4: Exact solution u_1^* .

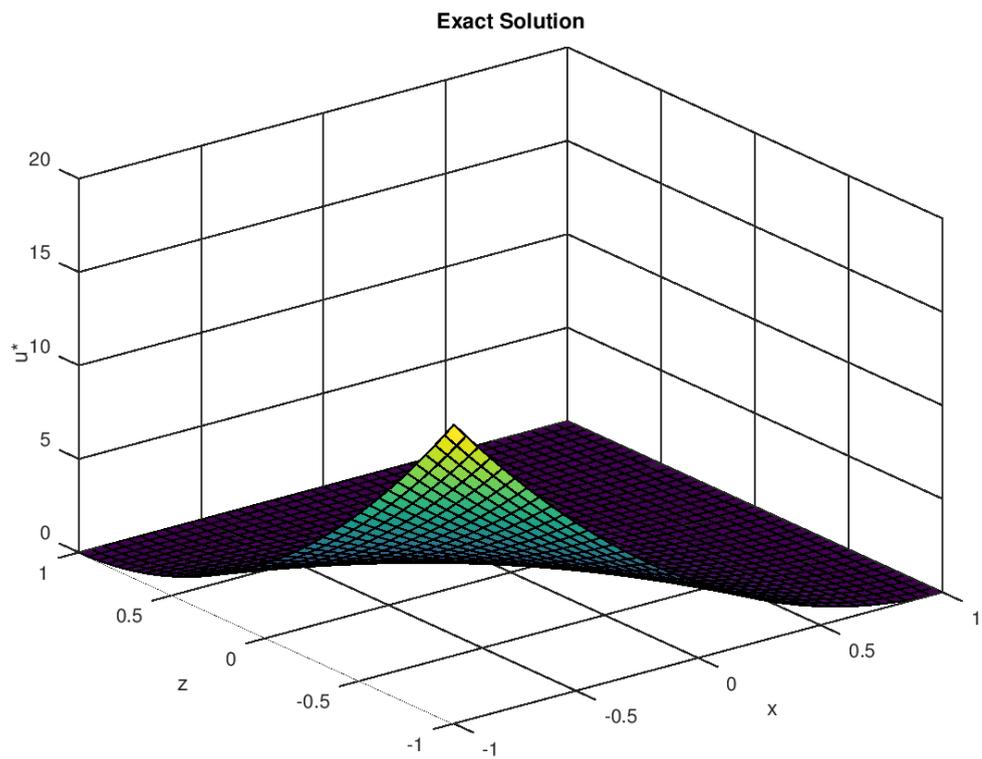


FIGURE 2.5: Exact solution u_2^* .

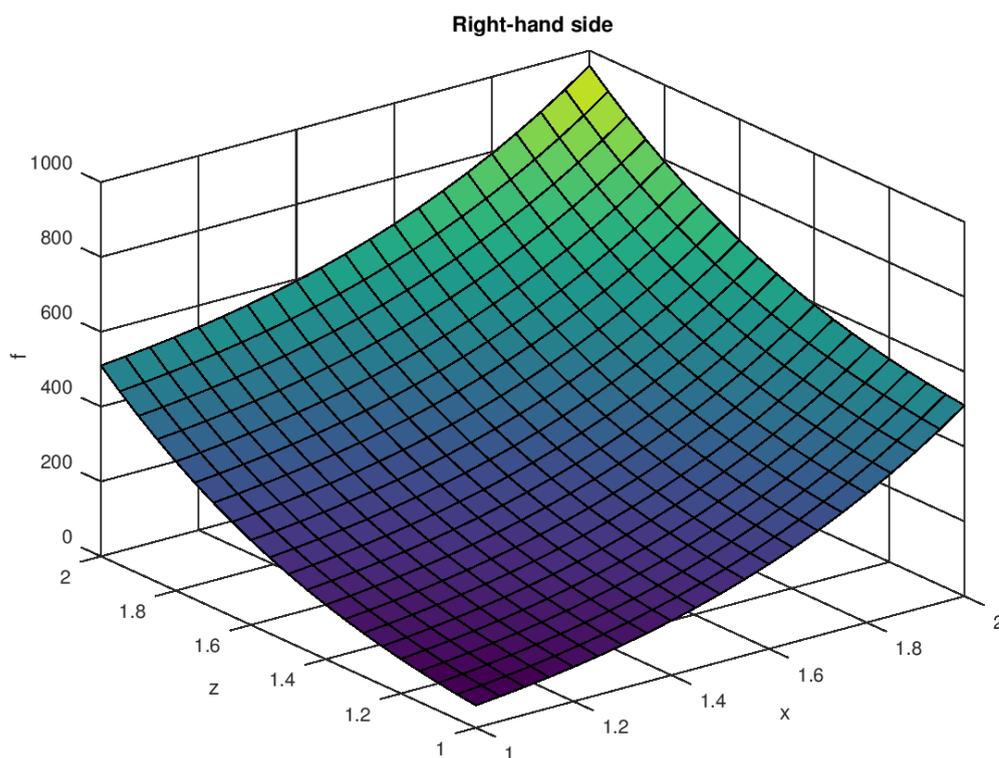


FIGURE 2.6: Right-hand side f_1 .

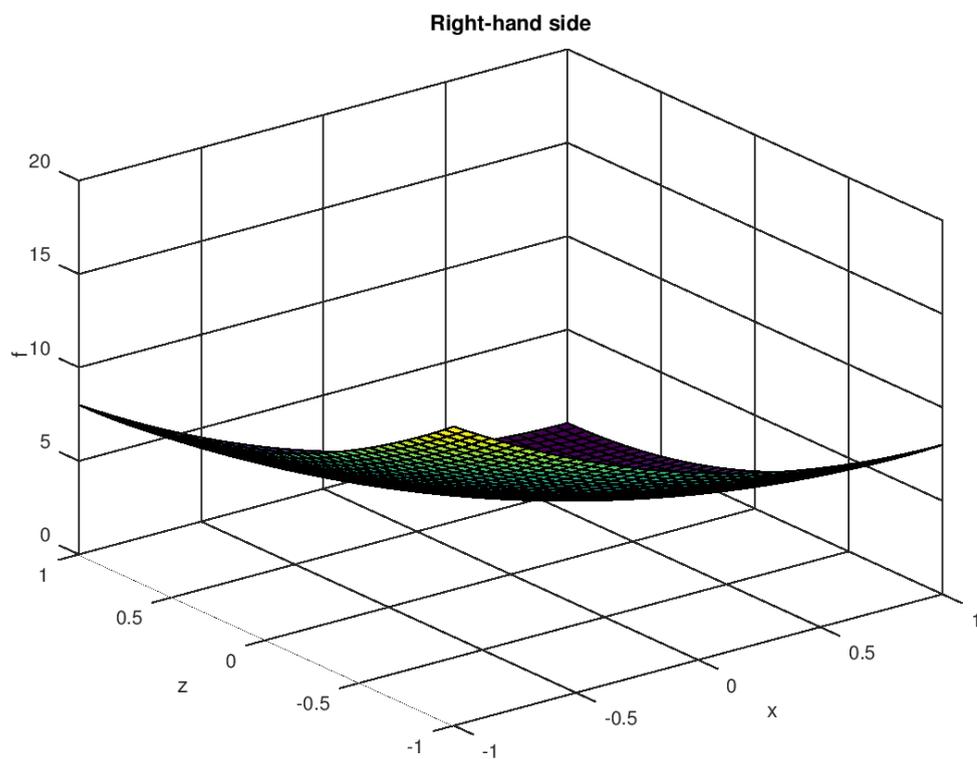


FIGURE 2.7: Right-hand side f_2 .

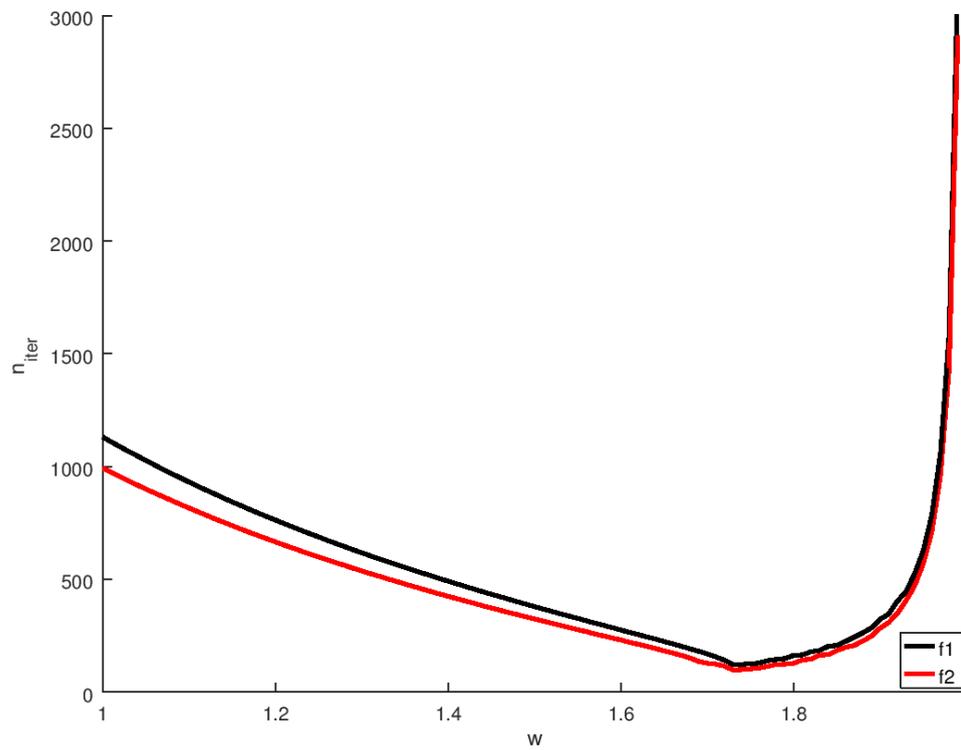


FIGURE 2.8: Optimum ω for the five-point stencil method.

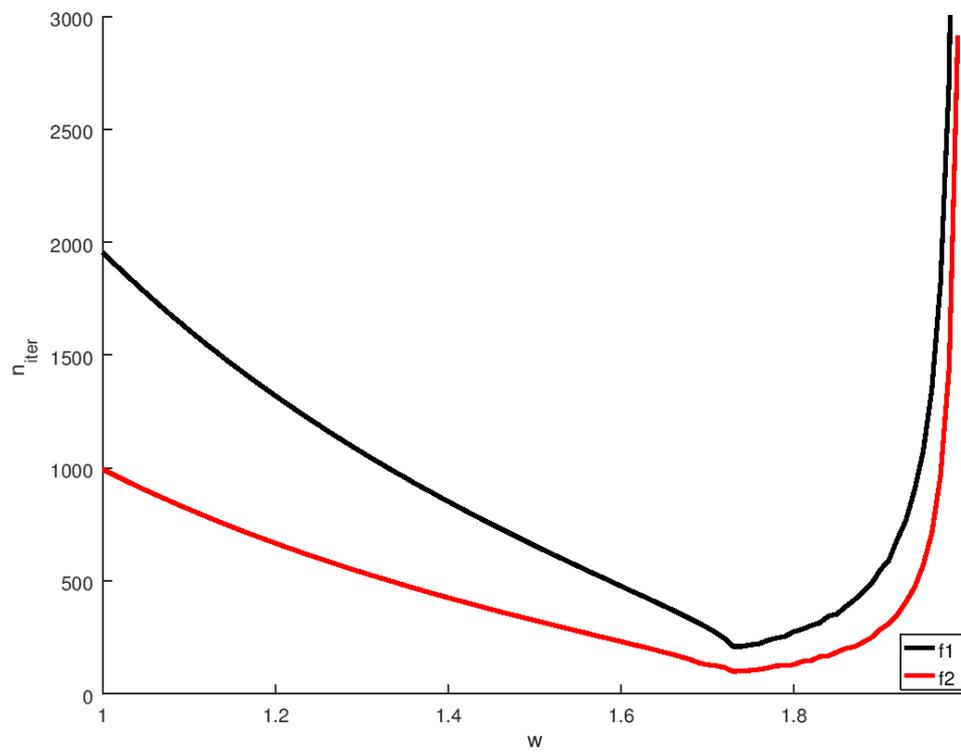
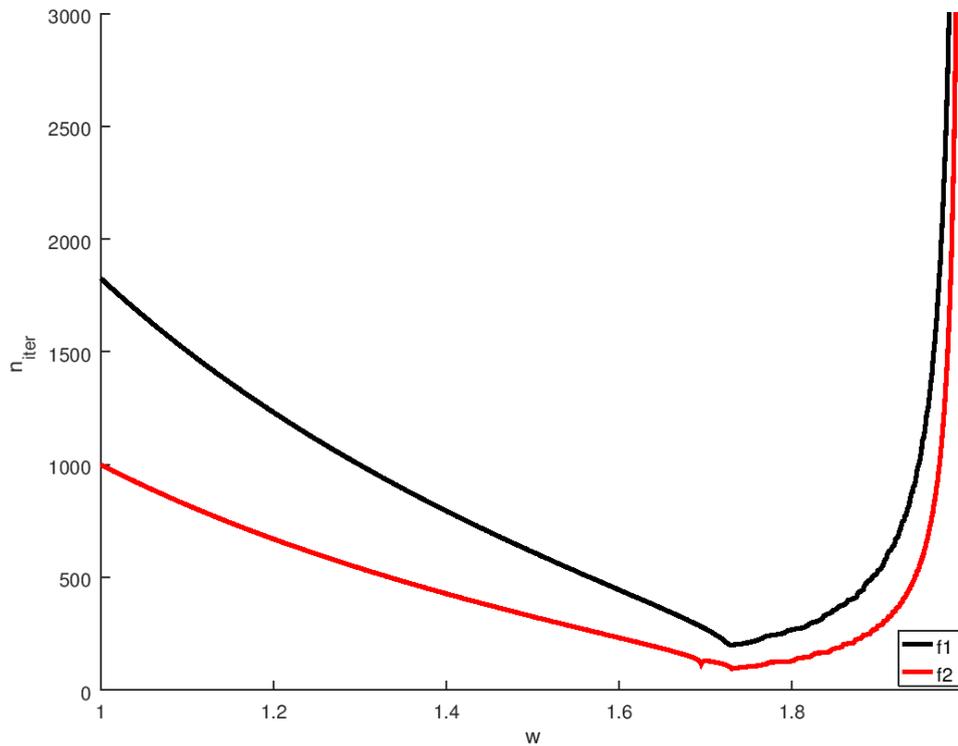


FIGURE 2.9: Optimum ω for the five-point stencil method with an increased order of approximation.

FIGURE 2.10: Optimum ω for the nine-point stencil method.

2.5.2 Optimum ω Problem - Analytical Approach

The optimum relaxation factor is known ([9], [10], and [11]) to satisfy the following formulae:

$$\omega_{\text{analytical}}^{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho^2}}, \quad (2.32)$$

where ρ is the spectral radius of the matrix. The optimum ω is corresponding to the asymptotic convergence factor ([11]), which defines the rate of convergence:

$$\rho_{\text{ACF}} = \frac{1 - \sqrt{1 - \rho^2}}{1 + \sqrt{1 - \rho^2}}. \quad (2.33)$$

So as to find $\omega_{\text{analytical}}^{\text{opt}}$ and ρ_{ACF} , the spectrum factor ρ must be evaluated. For that purpose, a spectrum analysis should be made.

columns are the eigenvectors of A , Λ is a diagonal matrix whose entries are the eigenvalues of A , it follows that:

$$\begin{aligned} A_{m,k} &= Q \Lambda Q^T; \\ A_{m,k+1} &= Q M Q^T. \end{aligned} \tag{2.38}$$

Remark: Here, the transformation matrices are the same, since the eigenvalues do not depend on the entries of the matrix ([9]).

Λ and M are diagonal matrices and consist of the eigenvalues of the matrices $A_{i,k}$, and $A_{i,k\pm 1}$. Using the result from the theorem in [9], it follows that the elements of these diagonal matrices are:

$$\begin{aligned} \lambda_j &= \alpha + 2\beta \cos(j\pi h), \quad j = \overline{1, \dots, (N-1)^2}; \\ \nu_l &= \beta + 2\gamma \cos(l\pi h), \quad l = \overline{1, \dots, (N-1)^2}. \end{aligned} \tag{2.39}$$

Substituting (2.38) into (2.37), multiplying with Q^{-1} from left, and denoting $Q^T u_k$ with y_k , it follows that:

$$\nu_m y_{m,k-1} + (\lambda_m - \mu I) y_k + \nu_m y_{k+1} = 0, \quad k = \overline{1, \dots, N-1}, \tag{2.40}$$

or in a matrix form:

$$\begin{bmatrix} \lambda_m & \nu_m & & & \\ \nu_m & \lambda_m & \nu_m & & \\ & & \ddots & \ddots & \ddots \\ & & & \nu_m & \lambda_m & \nu_m \\ & & & & \nu_m & \lambda_m \end{bmatrix} \begin{bmatrix} y_{m,1} \\ y_{m,2} \\ \vdots \\ y_{m,N-2} \\ y_{m,N-1} \end{bmatrix} = \mu_m \begin{bmatrix} y_{m,1} \\ y_{m,2} \\ \vdots \\ y_{m,N-2} \\ y_{m,N-1} \end{bmatrix}. \tag{2.41}$$

Using the theorem from [9] once again:

$$\mu_{m,n} = \lambda_m + 2\nu_m \cos(n\pi h), \quad m, n = \overline{1, \dots, N-1}. \tag{2.42}$$

Substituting (2.39) into (2.42), \Rightarrow

$$\begin{aligned} \mu_{m,n} &= \alpha + 2\beta \cos(m\pi h) + 2(\beta + 2\gamma \cos(m\pi h)) \cos(n\pi h), \\ m, n &= \overline{1, \dots, N-1} \Leftrightarrow \\ \mu_{m,n} &= \alpha + 2\beta (\cos(m\pi h) + \cos(n\pi h)) + 4\gamma \cos(m\pi h) \cos(n\pi h), \\ m, n &= \overline{1, \dots, N-1}. \end{aligned}$$

□

2.5.3 Optimum ω – Results

After conducting the spectrum analysis, finding the maximum by an absolute value eigenvalue, and substituting into (2.32), the following results are obtained (see Table 2.1).

ω^{opt}	Five-point	Five-point with an IOA	Nine-point
$\omega_{\text{numerical}}^{\text{opt}}$	1.73390	1.73799	1.73373
$\omega_{\text{analytical}}^{\text{opt}}$	1.72945	1.72945	1.73057

TABLE 2.1: Optimum ω for the SOR method.

There exists a perfect correspondence between the numerical and the analytical results. Comparing the results for $\omega_{\text{analytical}}^{\text{opt}}$, in the case of the five-point stencil method and the five-point stencil method with an increased order of approximation (IOA), with the ones which exists in the literature (e.g. [9] and [12]), a perfect correspondence is achieved. Some differences are observed when it comes to the nine-point stencil method. In the Table 2.1 one can see the $\omega_{\text{analytical}}^{\text{opt}}$ which was obtained, using the `Matlab` function for an eigendecomposition. It must be noted that the spectrum analysis gave us the following result $\omega_{\text{analytical}}^{\text{opt}} \approx 1.70781$. On the other hand, the authors in [12] have obtained $\omega^{\text{opt}} \approx 1.66758$, using a combination between an analytical and a numerical approaches. As

	Five-point	Five-point with an IOA	Nine-point
ρ	$\cos(\pi h)$	$\cos(\pi h)$	$\frac{4}{5} \cos(\pi h) + \frac{1}{5} (\cos(\pi h))^2$
evaluated value	0.98769	0.98769	0.98526

TABLE 2.2: Optimum ρ for the SOR method.

it can be seen in Table 2.2, the spectral radius $\rho_{\text{nine-point}}$ is a little bit smaller than $\rho_{\text{five-point with an IOA}}$ which determines the slower convergence of the nine-point method (see Table 2.3). The two five-point stencil methods are obtained to have one and the same optimum relaxation factors and hence, one and the same spectral radii and asymptotic convergence factors.

	Five-point	Five-point with an IOA	Nine-point
ρ_{ACF}	0.72945	0.72945	0.70782

TABLE 2.3: Optimum ρ_{ACF} for the SOR method.

2.6 Number of SOR Iterations

Using the evaluated values for $\omega_{\text{numerical}}^{\text{opt}}$, numerical experiments are conducted. Three different numbers of points for discretization of the domain of the initial problem are considered and for each of them, the number of SOR iterations so as a prescribed margin error to be obtained were found. The results are summarized in Table 2.4. There, the last column gives the quotient between the number of needed SOR iterations for two consecutive number of points. The results show that the five-point stencil method with an increased order of approximation, and the nine-point stencil method require more iterations than the general five-point method so as they to converge, which is expected, having in mind that the former ones are more complicated methods. On the other hand, the nine-point stencil method requires less number of iterations in comparison with the five-point stencil method with an increased order of approximation. The explanation for that behaviour is the nature of the two algorithms – in the case of the five-point stencil method with an increased order of approximation, the algorithm starts from a zero approximation for the searched solution, while in the case of the nine-point stencil method, the first iteration of the relaxation method is used as an initial value.

Points	Five-point	Five-point with an IOA	Nine-point	Coefficient
N	122	210	200	-
$2 * N - 1$	652	1096	1059	≈ 5
$4 * N - 1$	2588	4185	4051	≈ 4

TABLE 2.4: Number of SOR iterations.

2.7 Runge's Rule for a Practical Estimation of the Error

There exist two different formulas for a practical estimation of the approximation error called Runge's rule ([6]):

$$\alpha = \ln \left| \frac{y_h(x, z) - y_{h/2}(x, z)}{y_{h/2}(x, z) - y_{h/4}(x, z)} \right| / \ln 2$$

$$\alpha = \ln \left| \frac{u(x, z) - y_h(x, z)}{u(x, z) - y_{h/2}(x, z)} \right| / \ln 2$$

The idea is the following: if we have a certain method for an approximate solving of a problem, then taking the approximating function on three different meshes (the second is taken with a twice smaller step, and the third — with four times smaller), the approximation error can be found. In the case when the exact solution is known, the

approximating function on just two meshes is needed. Using the first formulae from the two listed above, the Runge’s coefficient α for the three considered methods was found. As it can be seen in Table 2.5, the approximate order of approximation is second, fourth, and fourth for the five-point stencil method, the five-point stencil method with an increased order of approximation, and the nine-point stencil method, respectively. This means that what was expected from the theory is achieved.

The Runge’s coefficient α for the three methods is depicted over the whole domain in Figures 2.11, 2.12, and 2.13, respectively. One can see that in the case of the five-point stencil methods the order of approximation around the boundaries is lower than in the middle of the domain, while for the nine-point stencil method it is higher. The reason for that is the following: as an initialization of the solution in the inner part of the domain for all the methods zero value is used, but while the first two methods use just two out of five points from the boundary (where the exact solution is applied) and three from the zero initialization so as to approximate the solution around the boundary, the third method uses five out of nine from the boundary and four from the initialization.

Method	Five-point	Five-point with an IOA	Nine-point
α	≈ 2	≈ 4	≈ 4

TABLE 2.5: Runge’s rule for a practical estimation of the error – results.

Runge Error of the Five-point Laplacian (Cross) Method without Correction of the Right-hand Side

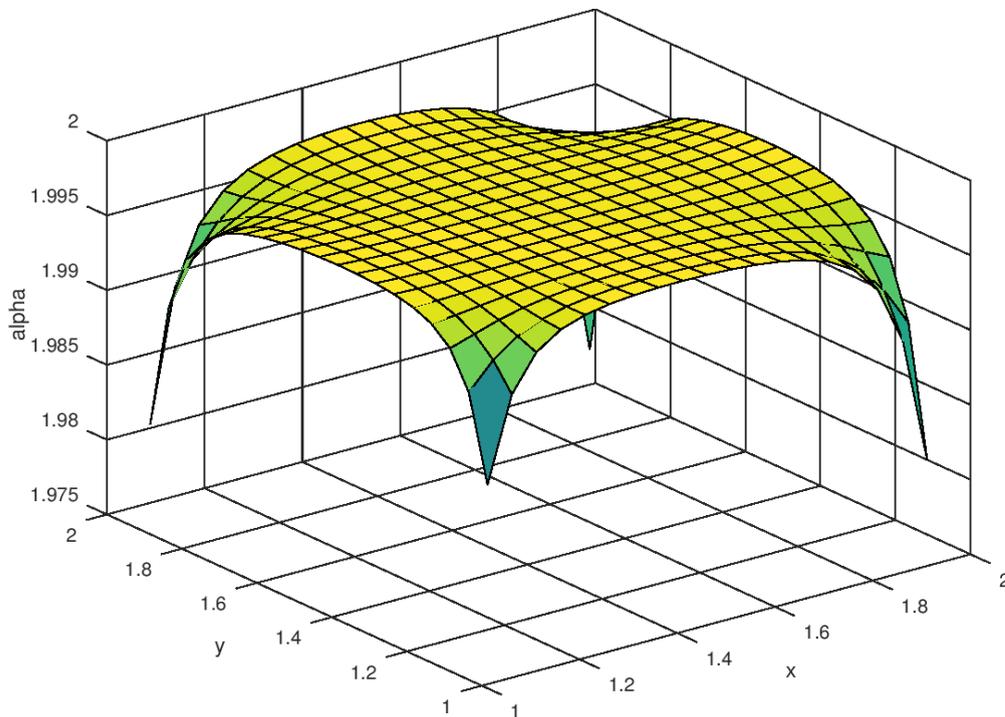


FIGURE 2.11: α coefficient for the five-point stencil method.

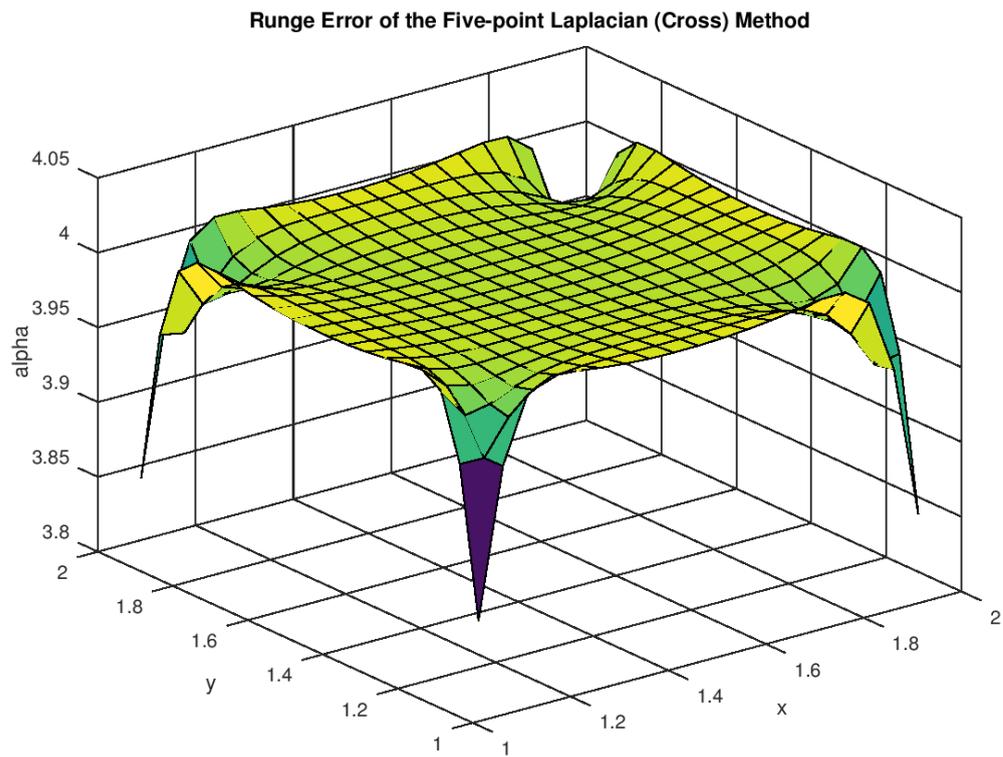


FIGURE 2.12: α coefficient for the five-point stencil method with an increased order of approximation.

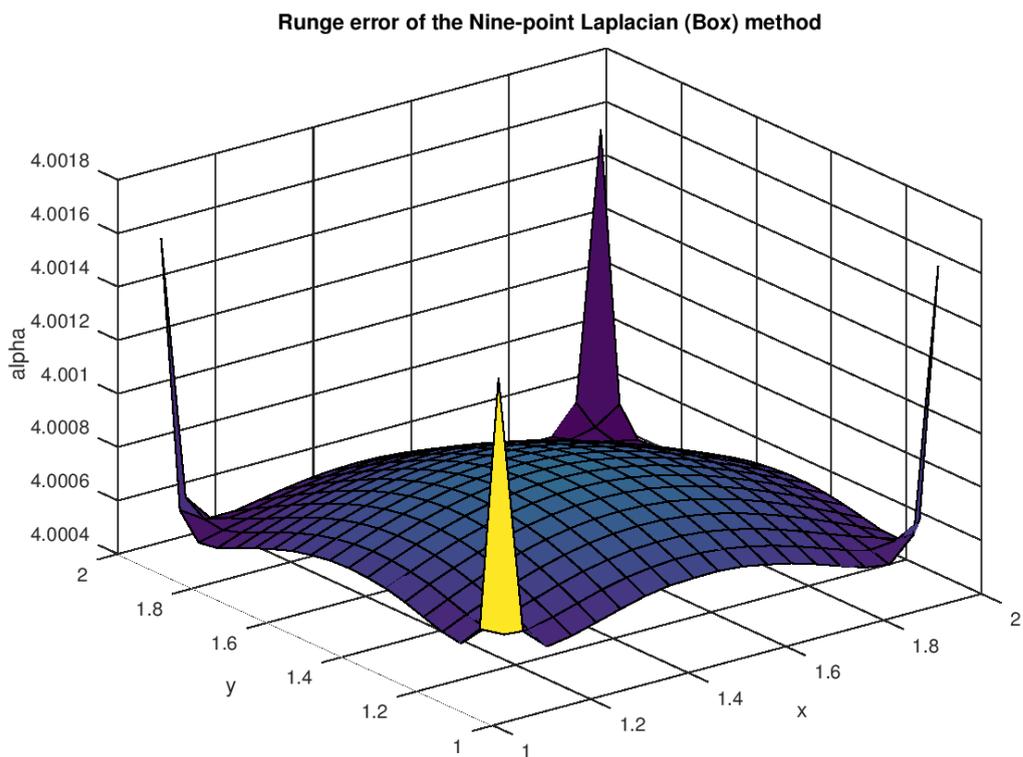


FIGURE 2.13: α coefficient for the nine-point stencil method.

2.8 Comparison between the Methods

In Figure 2.14 the relative errors $\frac{|y-u^*|}{|u^*|}$ for the nine-point stencil method, and the five-point stencil method with an increased order of approximation are shown. It can be noticed that the practical order of approximation of the nine-point stencil method is higher than the one of the five-point stencil method with an increased order of approximation, namely $\approx 1 * 10^{-7}$ vs. $\approx 2 * 10^{-6}$.

In conclusion, both the methods theoretically have a fourth-order of approximation, but

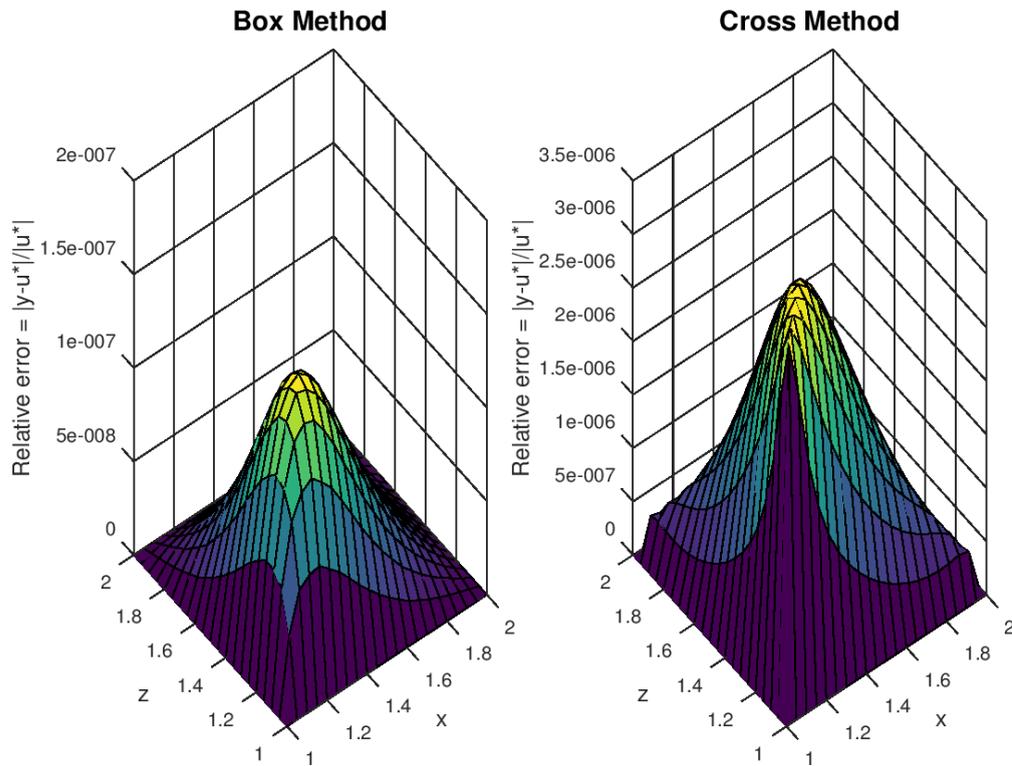


FIGURE 2.14: Comparison between the five-point stencil method with an increased order of approximation and the nine-point stencil method

Method	Five-point with an IOA	Nine-point
theoretical order of approximation	$O(h^4)$	$O(h^4)$
higher practical order of approx.	$(\approx 2 * 10^{-6})$	$\sqrt{(\approx 1 * 10^{-7})}$
higher complexity of the algorithm		✓
faster convergence	✓	
smaller number of SOR iterations		✓

TABLE 2.6: Comparison between the methods.

practically, the nine-point stencil method has a higher one in comparison with the other

method. The nine-point stencil method possesses higher complexity of the algorithm in comparison with the other method on one hand, but needs a smaller number of SOR iterations so as to converge. On the other hand, the five-point stencil method with an increased order of approximation converges faster than the nine-point stencil method. All these features can be read in Table [2.6](#).

Chapter 3

Investigation of Effective Methods and Their Stability for Solving Band Matrix SLEs after Parabolic Nonlinear PDEs

3.1 Mathematical Model

The following nonlinear model of a cylindrical multilayer structure is considered:

$$\rho^m(u) c_v^m(u) \frac{\partial u}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda^m(u) \frac{\partial u}{\partial r} \right) + \sum_{\alpha=1}^n \frac{\partial}{\partial z_\alpha} \left(\lambda^m(u) \frac{\partial u}{\partial z_\alpha} \right) + \varphi^m(u); \quad (3.1)$$

$$\frac{\partial u}{\partial r} = 0 \quad \forall r \in [r_{\min}, r_{\max}]; \quad (3.2)$$

$$-\lambda^m(u) \frac{\partial u}{\partial r} \Big|_{r=r_{i^*}^m-0} = -\lambda^{m+1}(u) \frac{\partial u}{\partial r} \Big|_{r=r_{i^*}^m+0} \quad \text{and} \quad u|_{r=r_{i^*}^m-0} = u|_{r=r_{i^*}^m+0}, \quad (3.3)$$

where $(r, \vec{z}) \in \Omega \cup \partial\Omega$, $t \geq 0$; m – index of the subdomain. Equation (3.1) represents the conservation of heat inside a multilayer structure. It is an energy equation with conduction heat transfer, where the densities, the specific heat capacities, and the thermal conductivities depend on the temperature. For instance, in the two-dimensional case, Equation (3.1) could be defined in a domain similar to the one shown in Figure 3.1. Homogeneous Neumann boundary conditions (3.2) are applied on the outer boundaries in the radial direction. The ideal contact conditions (3.3) model the heat flux on the inner boundaries in the radial direction, where $r_{i^*}^m$ denotes the point of discontinuity. The

numerical algorithms for solving the multidimensional governing equation (GE), using FDM (e.g. ADI algorithms ([13], [14])), ask for a repeated SLE solution.

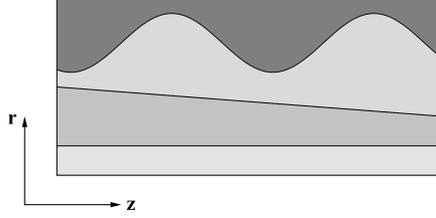


FIGURE 3.1: Example of a rectangular domain in cylindrical coordinates (a longitudinal section of a multilayer cylinder), where the thermal coefficients are different in all the subdomains and they have a discontinuity of the first kind on the borders in-between the subdomains.

3.2 Discretization of the Problem

Focusing on the radial term of the GE with the assumption that the other terms will be moved to the right-hand side (RHS), we consider the following special mesh with grid points on the inner boundaries: $\bar{\omega}_r = \{(t, r) \mid t_k = k h_t, k \in \mathbb{N}_0; r_0 = r_{\min}, r_{i+1} = r_i + h_{i+1}, i = 0, \dots, N-2\}$. A finite difference scheme with a second-order approximation has the following form (three-point stencils are taken for the GE and the outer boundary conditions (BC), and five-point stencil for the inner BC):

$$\hat{\rho}_i \hat{c}_{v_i} \frac{\hat{u}_i - u_i}{\tau} = \frac{1}{r_i} \frac{1}{\bar{h}_i} \left[r_{i+1/2} \hat{\lambda}_{i+1/2} \frac{\hat{u}_{i+1} - \hat{u}_i}{h_{i+1}} - r_{i-1/2} \hat{\lambda}_{i-1/2} \frac{\hat{u}_i - \hat{u}_{i-1}}{h_i} \right] + \hat{\varphi}_i; \quad (3.4)$$

$$-\frac{h_2(2h_1 + h_2)\hat{u}_0 - (h_1 + h_2)^2\hat{u}_1 + h_1^2\hat{u}_2}{h_1 h_2 (h_1 + h_2)} = 0; \quad (3.5)$$

$$\hat{\lambda}_{i^*}^m \frac{h_{i^*-1}(2h_{i^*} + h_{i^*-1})\hat{u}_{i^*} - (h_{i^*} + h_{i^*-1})^2\hat{u}_{i^*-1} + h_{i^*}^2\hat{u}_{i^*-2}}{h_{i^*} h_{i^*-1} (h_{i^*} + h_{i^*-1})} = \quad (3.6)$$

$$= -\hat{\lambda}_{i^*}^{m+1} \frac{h_{i^*+2}(2h_{i^*+1} + h_{i^*+2})\hat{u}_{i^*} - (h_{i^*+1} + h_{i^*+2})^2\hat{u}_{i^*+1} + h_{i^*+1}^2\hat{u}_{i^*+2}}{h_{i^*+1} h_{i^*+2} (h_{i^*+1} + h_{i^*+2})}; \quad \hat{u}_{i^*-0} = \hat{u}_{i^*+0}; \quad (3.7)$$

$$\frac{h_{N-2}(2h_{N-1} + h_{N-2})\hat{u}_{N-1} - (h_{N-1} + h_{N-2})^2\hat{u}_{N-2} + h_{N-1}^2\hat{u}_{N-3}}{h_{N-1} h_{N-2} (h_{N-1} + h_{N-2})} = 0, \quad (3.8)$$

where

$$\hat{\lambda}_{i\pm 1/2} = \lambda \left(\frac{\hat{u}_i + \hat{u}_{i\pm 1}}{2} \right), \quad \bar{h}_i = \frac{h_{i+1} + h_i}{2}, \quad r_{i\pm 1/2} = \frac{r_i + r_{i\pm 1}}{2}.$$

The matrix form of the considered system is: $A\vec{\hat{u}} = \vec{\varphi}(\hat{u})$, where A is a PD sparse matrix which does not have any special properties, e.g. diagonally dominance or positive definiteness.

In order to preserve the band structure of the matrix, we cannot use the Gaussian elimination with pivoting. Within this, we could obtain division by zero at some point of the procedure, which is going to make our algorithm unstable. For that reason, we alter the initial PD matrix by adding the minimum values to the diagonal elements so as to transform the matrix into a weakly diagonally dominant one:

$$A_{DD} \vec{u} = \vec{\varphi}(\hat{u}) + P \vec{u}, \quad \text{where} \quad A_{DD} = A + P;$$

$$P = \text{diag} \left(2h_1^2, \delta_{i^*,j} p_{i^*,j}, 2h_{N-1}^2 \right),$$

where

$$p_{i^*,i^*} = \frac{2\lambda_{i^*}^m h_{i^*}}{h_{i^*-1} (h_{i^*} + h_{i^*-1})} + \frac{2\lambda_{i^*}^{m+1} h_{i^*+1}}{h_{i^*+2} (h_{i^*+1} + h_{i^*+2})}.$$

The Gaussian elimination with pivoting (the procedure we use to transform the PD matrix into a TD one) does not preserve the diagonal dominance of the matrix. The use of the Gaussian elimination to the initial matrix A (not A_{DD}) yields a transformed matrix \tilde{A} . Then, the diagonal dominantization method is used in order to transform the obtained TD matrix \tilde{A} into a diagonally dominant one. To that purpose, the nondiagonal elements are added to the diagonal ones:

$$\tilde{A}_{DD} \vec{u} = \vec{\varphi}(\hat{u}) + \tilde{P} \vec{u}, \quad \text{where} \quad \tilde{A}_{DD} = \tilde{A} + \tilde{P};$$

$$\tilde{P} = \text{diag} \left(|\tilde{A}_{0,1}|, \delta_{i^*,j} \tilde{p}_{i^*,j}, |\tilde{A}_{N-2,N-1}| \right), \quad \text{where} \quad \tilde{p}_{i^*,i^*} = \sum_{\beta \in \{-1,1\}} |\tilde{A}_{i^*,i^*+\beta}|.$$

3.3 Numerical and Symbolic Algorithms

Two different approaches for solving the SLE are considered – numerical and symbolic. The complexity of all the suggested numerical algorithms is $O(N)$. Since it is unknown what stands behind the symbolic library, evaluating the complexity of the symbolic algorithms is a very complicated task.

Numerical Algorithms. Two different numerical algorithms are applied to the system with a PD matrix. Both of them are based on LU decomposition. The first one ([15]) is intended for a dense PD matrix. In the case of the considered problem, the matrix is sparse. For that reason, a modified algorithm is built. The main idea is that after the mesh was defined, the indexes of the discontinuity points are known. Since these indexes coincide with those of the matrix rows which are not sparse, we can reference them to the algorithm and conduct the full calculation only for them. For the rest of the rows, the algorithm is reduced to a problem similar to solving a system with TD matrix. This way, the complexity of the algorithm is decreased but at the cost of additional $N+2$

check-ups for the non-sparse rows. In the case of the TD matrix, the system is solved using the Thomas method.

Symbolic Algorithms. The symbolic algorithm in the case of a PD matrix is also based on LU decomposition ([15]). For the TD matrix, a symbolic version of Thomas method is created. As it is known, Thomas method is not suitable for non diagonally dominant matrices ([2]). In order to cope with this problem, in the case of a zero quotient of two subsequent leading principal minors, a symbolic zero is assigned instead and the calculations are continued. At the end of the algorithm, this symbolic zero is substituted with zero. For a SLE of the form $\text{tridiag}(\mathbf{a}, \mathbf{c}, \mathbf{b}) \mathbf{x} = \mathbf{f}$, the algorithm in pseudocode is as follows:

```
d[0] = c[0];
if d[0] == 0
    d[0] = t; \\ 't' is a symb.\,variable
endif
for i=1:N
    d[i] = c[i]-b[i-1]*a[i-1]/d[i-1];
    if d[i] == 0
        d[i] = t;
    endif
endfor
y[0]=f[0];
for i=1:N
    y[i] = f[i]-y[i-1]*a[i-1]/d[i-1];
endfor
x[N] = y[N]/d[N];
for k=N-1:0
    x[k] = (y[k]-b[k]*x[k+1])/d[k];
endfor
Substitute t in x=(x[0],...,x[N]) with 0.
```

3.4 Implementation and Results

All the numerical algorithms are implemented using C++. The matrix needs to be non-singular and diagonally dominant so as the methods to be stable. Two symbolic algorithms are implemented, using the GiNaC library (version 1.7.2) ([16]) of C++. Our research showed that usage of the SymbolicC++ library (version 3-3.35) ([17]) of C++ is not possible, since in the implementation of this library the following statement was used: $t^{-n} = 0$, if $t = 0, n \in \mathbb{N}$. Also, this library does not maintain **Rational class** for symbolic variables. The symbolic algorithms require the matrix to be nonsingular

only. In Table 3.1 one can find the wall-clock time results from the conducted experiments. Since the largest supported precision in the **GiNaC** library is **double**, during all the experiments double data type is used. The notation is as follows: **NPDM** stands for numerical PD method, **MNPDM** – modified numerical PD method, **SPDM** – symbolic PD method, **NTDM** – numerical TD method, **STDM** – symbolic TD method. The achieved accuracy is summarized, using infinity norm. On the penultimate row of the table, one can find the complexity of all the considered methods. On the last row, the characteristics of the computer which is used are described. The number of considered discontinuity points is $K = 11$.

Wall-clock time [s]					
N	NPDM	MNPDM	SPDM	NTDM	STDM
10^3	0.000036	0.000034	0.088669	0.000021	0.043690
10^4	0.000403	0.000373	8.467241	0.000245	2.971745
10^5	0.004709	0.003916	3547.020851	0.002416	799.533587
10^8	3.159357	2.682258	–	1.652945	–
$\max_N \ y - \bar{y}\ _\infty$	2.22×10^{-16}	2.22×10^{-16}	0	2.22×10^{-16}	0
Complexity:	$19N - 29$	$13N + 7K - 14$	–	$9N + 2$	–
OS: Fedora 25; Processor: Intel Core i7-6700 (3.40 GHz); Compiler: GCC 6.3.1 (-O0).					

TABLE 3.1: Results from solving SLE

Remark 1: So as the nonsingularity of the matrices to be checked, a fast symbolic algorithm for calculating the determinant is implemented, using the method suggested in [18]. The complexity of the algorithm is $O(N)$.

Remark 2: The number of needed operations for the Gaussian elimination used in this Chapter 3 is $18 + 16K$.

Remark 3: Both the achieved computational times and accuracy for the **NPDM** and **SPDM** methods are much better than the ones in [15].

3.5 Additional Results

The numerical algorithms suggested in this Chapter 3 are also computed, using the heterogeneous cluster “HybriLIT” ([19]). The achieved computational times are summarized in Table 3.2. The results are coherent with the ones evaluated, using a PC. The compilers and the optimizations which are used are described on the last row of the table. The number of considered discontinuity points is again $K = 11$.

Wall-clock time [s]					
N	NPDM	MNPDM	SPDM	NTDM	STDM
10^3	0.000044	0.000043	0.118732	0.000030	0.054840
10^4	0.000410	0.000400	17.220277	0.000270	6.686015
10^5	0.004022	0.003899	5632.271746	0.002720	1937.355622
10^8	2.855820	2.697446	–	2.035205	–
$\max_N \ y - \bar{y}\ _\infty$	2.22×10^{-16}	2.22×10^{-16}	0	2.22×10^{-16}	0
Complexity:	$19N - 29$	$13N + 7K - 14$	–	$9N + 2$	–
Compiler: num: Intel 2017.2.050 ICPC (-O2) and symb: GCC 4.9.3 (-O0).					

TABLE 3.2: Results from solving SLE on the cluster “HybriLIT”.

3.6 Iterative Approach

Finally, an iterative procedure for solving the pentadiagonal matrix is considered, namely the strongly implicit procedure (SIP) ([20]), also known as the Stone method. It is an algorithm for solving sparse SLEs. The method uses the incomplete LU (ILU(0)) decomposition ([21]) which is an approximation of the exact LU decomposition in the case when a sparse matrix is considered. The idea of ILU is that the zero elements of L and U are chosen to be the same as of the initial matrix A . The Stone method needs the inverse matrices of L and U . We use a numerical procedure to compute these inverse matrices, namely the Hotelling-Bodewig iterative algorithm ([22]). Its form is as follows:

$$A_{n+1}^{-1} = A_n^{-1} (2I - AA_n^{-1}), \quad n = 0, 1, \dots, \quad (3.9)$$

where I is the identity matrix, A is the matrix whose inverse we are looking for. A diagonal matrix is used as an initial guess for the inverse matrix, as it is suggested in [23]. The obtained computational times for the ILU(0) method and the Hotelling-Bodewig iterative algorithm, using the heterogeneous cluster “HybriLIT”, are summarized in Tables 3.3 and 3.6.

The Stone method has the following form:

$$\begin{aligned}
 k = 0; & & \text{while}(\|\overrightarrow{residual}^{(k)}\|_\infty \geq \text{errorMargin})\{ \\
 \vec{x}^{(k)} = \vec{0}; & & \overrightarrow{newRH\dot{S}}^{(k)} = K \vec{x}^{(k)} + \vec{b}; \\
 \overrightarrow{newRH\dot{S}}^{(k)} = A \vec{x}^{(k)}; & & \vec{y}^{(k)} = L^{-1} \overrightarrow{newRH\dot{S}}^{(k)}; \\
 \overrightarrow{residual}^{(k)} = \overrightarrow{RH\dot{S}} - \overrightarrow{newRH\dot{S}}^{(k)}; & & \vec{x}^{(k+1)} = U^{-1} \vec{y}^{(k)}; \\
 K = LU - A; & & \overrightarrow{residual}^{(k+1)} = \vec{b} - A \vec{x}^{(k+1)}; \quad k++;\}
 \end{aligned}$$

Every iteration step of the Stone method consists of four matrix-vector multiplications

(two multiplications with the pentadiagonal matrix and two with the inverse triangular matrices of the ILU(0), which are also triangular) and two vector additions, i.e. the complexity of the algorithm on every iteration is $30N - 36 = O(N)$, where N is the number of rows of the initial matrix. Computer experiments are conducted, using “HybriLIT”. The results can be seen in Tables 3.4 and 3.7.

In Tables 3.5 and 3.8 one can see the number of needed iterations for the Stone and the Hotelling-Bodewig methods.

3.6.1 Matrix Implementation

N	Wall-clock time [s]		
	ILU(0)	L^{-1}	U^{-1}
10^2	0.000439	0.009864	0.009712
10^3	0.281497	105.130351	55.428357
2×10^3	2.203920	428.588584	299.559702
5×10^3	33.403451	6983.327956	6771.895344
7×10^3	91.059694	28264.604603	17814.543375
Compiler: Intel 2017.2.050 ICPC; Optimization: -O2.			

TABLE 3.3: Results from the ILU(0) method and the numerical method for inverting matrices (matrix implementation), using the cluster “HybriLIT”.

N	Wall-clock time [s]
	SIP
10^2	0.000031
10^3	0.003066
2×10^3	0.013564
5×10^3	0.087308
7×10^3	0.189665
$\max_N \ y - \bar{y}\ _\infty$	4.44×10^{-16}
Complexity:	$30N - 36$
Compiler: Intel 2017.2.050 ICPC; Optimization: -O2.	

TABLE 3.4: Results from solving SLE, using SIP (matrix implementation), on the cluster “HybriLIT”.

3.6.2 Array Implementation

Initially, a matrix implementation results of which can be seen in Subsection 3.6.1 was made. Since it was very demanding in regards to memory, it was impossible to

N	SIP Method	Hotelling-Bodewig Method L^{-1}	Hotelling-Bodewig Method U^{-1}
10^2	1	7	7
10^3	1	7	7
2×10^3	1	7	7
5×10^3	1	7	7
7×10^3	1	7	7

TABLE 3.5: Number of Iterations Needed for the Stone and the Hotelling-Bodewig methods (matrix implementation), using the cluster “HybriLIT”.

computational experiments for a matrix with more than 7×10^3 rows to be conducted. For that reason, the algorithms were redesigned, taking into account the band structure of our data, and so an array implementation was made.

N	Wall-clock time [s]		
	ILU(0)	L^{-1}	U^{-1}
10^2	0.000454	0.003654	0.003937
10^3	0.294193	4.781403	5.636033
2×10^3	2.252378	39.946383	45.180269
5×10^3	33.425096	1656.377837	1681.127071
7×10^3	91.030153	4835.771232	4670.087554
10^4	271.939590	24950.736479	13809.421644

Compiler: Intel 2017.2.050 ICPC; Optimization: -O2.

TABLE 3.6: Results from the ILU(0) method and the numerical method for inverting matrices (array implementation), using the cluster “HybriLIT”.

N	Wall-clock time [s]	
	SIP	
10^2	0.000012	
10^3	0.000661	
2×10^3	0.004064	
5×10^3	0.017032	
7×10^3	0.034007	
10^4	0.091237	
$\max_N \ y - \bar{y}\ _\infty$	4.44×10^{-16}	
Complexity:	$30N - 36$	

Compiler: Intel 2017.2.050 ICPC; Optimization: -O2.

TABLE 3.7: Results from solving SLE, using SIP (array implementation), on the cluster “HybriLIT”.

N	SIP Method	Hotelling-Bodewig Method L^{-1}	Hotelling-Bodewig Method U^{-1}
10^2	1	7	6
10^3	1	7	6
2×10^3	1	7	6
5×10^3	1	7	6
7×10^3	1	7	6
10^4	1	7	6

TABLE 3.8: Number of Iterations Needed for the Stone and the Hotelling-Bodewig methods (array implementation), using the cluster “HybriLIT”.

3.6.3 Comparison between the Implementations

Comparing the results, one can see that the array implementation not only decreased the computational times for the inverting of both the matrices L and U but also it decreases the number of iterations needed so as the matrix U to be inverted. As one can see, the time required for the SIP procedure is also improved by the new implementation approach. Finally, this second approach requires less amount of memory which allows experiments with bigger matrices to be conducted.

Chapter 4

Discussion, Conclusions, and Future Work

4.1 Conclusions and Discussions

Part 1. Three different finite difference schemes for solving elliptic PDEs were considered — five-point stencil method, five-point stencil method with an increased order of approximation, and nine-point stencil method. Special attention was drawn to the latter two since they are methods of a high order of approximation and that feature of theirs is needed in the practice. It was shown that the theoretical order of approximation of both the methods is fourth, while in practice the nine-point stencil method achieves more accurate results than the other. On the other hand, the five-point stencil method with an IOA is easier to be understood and implemented. Moreover, there is a possibility for a technology transfer — since there exist inline functions for the general five-point stencil method, one can use them twice (usually, two is the number of needed corrections of the RHS) and they will obtain the solution. So as the SLEs which were obtained after the considered FDM schemes to be solved, SOR method was used. It is very important the optimum relaxation factor ω^{optimum} to be evaluated, because it decreases the number of the needed SOR iterations to a minimum. It was shown that the estimation of the ω^{optimum} numerically instead of analytically is easier and faster, while the difference between the two values is negligible. Finally, during the numerical estimation of ω^{optimum} the golden section search method for evaluation of an extremum of a function was used. The choice was prescribed by the lack of requirements for the function. In compression, methods like the Newton method or the bisection method put constraints over the derivative of the function which is a very limiting requirement.

Part 2. A nonlinear heat transfer equation in a multilayer domain was considered. The suggested discretization scheme always has a first-order approximation. In the case of piecewise constant thermal conductivities or when $\|\hat{\lambda}_{i+1/2} - \hat{\lambda}_{i-1/2}\|_\infty = O(\|h_i\|_\infty)$, the order of approximation is going to be second. Focusing on the radial term, a SLE with a PD matrix was obtained. Then, applying Gaussian elimination, a TD matrix was derived. For both these matrices, a diagonal dominantization procedure was suggested. This approach ensures the stability of the suggested methods. A modified version of the numerical method for solving a SLE with a PD matrix was built. Since the complexity of this method is lower than the complexity of the general algorithm (usually $K \ll N$), better computational time was achieved. The fastest numerical algorithm was found to come from the Thomas method. All the experiments gave an accuracy of an order of magnitude of 10^{-16} . As a next step symbolic algorithms were used. They do not require the matrices to be of a special form and are exact. However, they are not comparable with the numerical algorithms with respect to the required time in the case of a numerical solving of the heat equation when one needs to solve the SLE many times. On the other hand, these symbolic methods are not as restrictive as the numerical ones when it comes to the matrix properties. Another upside of the symbolic algorithms is that in the case of a piecewise linear equation, they do not add nonlinearity to the RHS of the system and hence, there is no need of iterations for the time step to be executed. In future, the approach suggested in this note will be investigated in detail. Finally, an iterative algorithm was built – the Stone method. For the needs of the method, additionally, ILU(0) and the Hotelling-Bodewig methods were used. Using a matrix form, the Stone method is not suitable for large matrices (with number of rows $> 7 \times 10^3$). This drawback was fixed, using an array representation of the data. An upside of the iterative procedure is that it requires the initial matrix to be nonsingular only. However, this method is not suitable for too large matrices (with number of rows 1×10^5), since the evaluation of the inverse of a matrix is computationally demanding on both time and memory.

4.2 Future Work

Part 1. As a future work, we are going to explore the reasons for the observed differences in the values of the optimum relaxation factor in the case of the nine-point stencil method.

Part 2. In future, the approach suggested in Section 3.1 is going to be implemented for a real problem. Also, modified versions of the approach suggested in Section 3.6 could be designed for the special case of a tridiagonal matrix, using an array implementation. This is going to make the SIP algorithm faster.

Appendix A

Appendix A

In this Appendix one can find the values of the diagonal matrix \tilde{P} which should be used in Chapter 3 so as the tridiagonal matrix \tilde{A} to be made diagonally dominant. Note that in Chapter 3 different values were used during the diagonal dominantization of the matrix. The reason is that the minimum values contain very complicated expressions.

A.1 Diagonal Dominantization for the Tridiagonal Matrix

$$\begin{aligned}
 p_0 &\geq 2h_1^2 - 2h_1 h_2 - h_2^2 + \frac{2\lambda_{1/2} r_{1/2} h_1 h_2}{\lambda_{3/2} r_{3/2}} + \frac{r_1 \hbar_1 h_1^2 h_2}{\lambda_{3/2} r_{3/2}} \frac{\rho c_v}{\tau} \Big|_{u_1}; \\
 p_{i^*} &\geq \Lambda_1 \left[h_{i^*-1} h_{i^*}^2 r_{i^*-1} \hbar_{i^*-1} A - (h_{i^*-1} + h_{i^*})^2 \lambda_{i^*-3/2} r_{i^*-3/2} - \right. \\
 &\quad \left. - h_{i^*-1} (h_{i^*-1} + 2h_{i^*}) \lambda_{i^*-3/2} r_{i^*-3/2} + h_{i^*-1} h_{i^*} \lambda_{i^*-1/2} r_{i^*-1/2} \right] + \\
 &\quad + \Lambda_2 \left[h_{i^*+1}^2 h_{i^*+2} r_{i^*+1} \hbar_{i^*+1} B - (h_{i^*+1} + h_{i^*+2})^2 \lambda_{i^*+3/2} r_{i^*+3/2} - \right. \\
 &\quad \left. - h_{i^*+2} (2h_{i^*+1} + h_{i^*+2}) \lambda_{i^*+3/2} r_{i^*+3/2} + h_{i^*+1} h_{i^*+2} \lambda_{i^*+1/2} r_{i^*+1/2} \right] + q_{i^*}; \\
 p_{N_r-1} &\geq -h_{N-1}^2 + \frac{h_{N-1} h_{N-2}}{\lambda_{N-5/2} r_{N-5/2}} \left[h_{N-1} \hbar_{N-2} r_{N-2} C - \lambda_{N-3/2} r_{N-3/2} \right],
 \end{aligned} \tag{A.1}$$

where

$$\begin{aligned}
 A &:= \frac{\lambda_{i^*-1/2} r_{i^*-1/2} h_{i^*-1} + \lambda_{i^*-3/2} r_{i^*-3/2} h_{i^*}}{h_{i^*-1} h_{i^*} \hbar_{i^*-1} r_{i^*-1}} + \frac{\rho c_v}{\tau} \Big|_{u_{i^*-1}} \\
 B &:= \frac{\lambda_{i^*+1/2} r_{i^*+1/2} h_{i^*+2} + \lambda_{i^*+3/2} r_{i^*+3/2} h_{i^*+1}}{h_{i^*+1} h_{i^*+2} \hbar_{i^*+1} r_{i^*+1}} + \frac{\rho c_v}{\tau} \Big|_{u_{i^*+1}} \\
 C &:= \frac{\lambda_{N-3/2} r_{N-3/2} h_{N-2} + \lambda_{N-5/2} r_{N-5/2} h_{N-1}}{h_{N-1} h_{N-2} \hbar_{N-2} r_{N-2}} + \frac{\rho c_v}{\tau} \Big|_{u_{N-2}}; \\
 \Lambda_1 &:= \frac{\lambda_{i^*}^{(m)}}{h_{i^*-1} h_{i^*} (h_{i^*-1} + h_{i^*}) \lambda_{i^*-3/2} r_{i^*-3/2}}; \\
 \Lambda_2 &:= \frac{\lambda_{i^*+1}^{(m)}}{h_{i^*+1} h_{i^*+2} (h_{i^*+1} + h_{i^*+2}) \lambda_{i^*+3/2} r_{i^*+3/2}};
 \end{aligned}$$

$$\begin{aligned}
q_{i^*} &:= \frac{2 \lambda_{i^*}^{(m)} (\lambda_{i^*-3/2} r_{i^*-3/2} h_{i^*} + \lambda_{i^*-1/2} r_{i^*-1/2} h_{i^*-1})}{h_{i^*-1} (h_{i^*-1} + h_{i^*}) \lambda_{i^*-3/2} r_{i^*-3/2}} + \\
&+ \frac{2 \lambda_{i^*}^{(m+1)} (\lambda_{i^*+3/2} r_{i^*+3/2} h_{i^*+1} + \lambda_{i^*+1/2} r_{i^*+1/2} h_{i^*+2})}{h_{i^*+2} (h_{i^*+1} + h_{i^*+2}) \lambda_{i^*+3/2} r_{i^*+3/2}}.
\end{aligned}$$

Acknowledgements

I would like to express my gratitude to my supervisor Alexander Ayriyan for his constant readiness to help, for all the things he taught me on, for the useful comments, remarks, and engagement through the working process of this project, for encouraging and believing in me. Thank you very much for all the opportunities you presented in front of me during the whole extremely intense and also highly beneficial time I spent at JINR. Finally, thank you very much for being the best supervisor ever! For me, it was a great honor to work together with you!

I also want to thank Ján Buša Jr. for the very useful discussions we had during my stay at JINR! Thank you very much also for your comments, remarks, and eagerness to share your knowledge and to help!

Another person I want to thank for his readiness to help and to share his knowledge is Andrey Lebedev (GSI/JINR). I really appreciate all the hard work you have done and the time you have spent so as to help me learning.

Also, I am really grateful to all the members of JINR for their warm welcome and attitude during my stay in Dubna, Russia.

Last but definitely not least, thank you very much to the Student Summer Program at JINR and its organizers! Thank you very much for the perfect organization and facilitating my stay in Russia! Excellent job! I was really happy to be part of JINR!

Bibliography

- [1] Alexander Ayriyan, Ján Buša Jr., Eugeny E. Donets, Hovik Grigorian, and Ján Pribiš. Algorithm and simulation of heat conduction process for design of a thin multilayer technical device. *Applied Thermal Engineering*, 94:151–158, 2016. ISSN 1359-4311. doi: 10.1016/j.applthermaleng.2015.10.095.
- [2] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, second edition, 2002. ISBN 978-0-89871-802-7.
- [3] Alexander Samarskii. *Vvedenie v Teoriju Raznostnyx Sxem*. Nauka, Moscow, 1971. (in Russian).
- [4] S. P. Frankel. Convergence rates of iterative treatments of partial differential equations. *Mathematical Tables and Other Aids to Computation*, 4:65–75, 1950.
- [5] D. M. Young. *Iterative methods for solving partial differential equations of elliptic type*. Harvard University, Cambridge, MA, 1950. Doctoral Thesis.
- [6] P. V. Vinogradova and A. G. Ereklincev. *Chislennyye Metody*. DVGUPS, Xabarovsk, second edition, 2011. (in Russian).
- [7] W. Kahan. *Gauss-Seidel Methods of Solving Large Systems of Linear Equations*. University of Toronto, Toronto, Canada, 1958. Doctoral Thesis.
- [8] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press, second edition, 1992. ISBN 0-521-43108-5.
- [9] Shiming Yang and Matthias K. Gobbert. The optimal relaxation parameter for the SOR method applied to the Poisson equation in any space dimensions. *Applied Mathematics Letters*, 22:325–331, March 2009.
- [10] J. K. Reid. A method for finding the optimum successive over-relaxation parameter. *The Computer Journal*, 9:200–204, 1966. doi: 10.1093/comjnl/9.2.200.
- [11] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Texts in Applied Mathematics. Numerical Mathematics*, volume 37. Springer, 2007. ISBN 978-1-4757-7394-1.

-
- [12] Loyce M. Adams, Randall J. LeVeque, and David M. Young. Analysis of the SOR iteration for the 9-point Laplacian. *SIAM Journal on Numerical Analysis*, 25:1156–1180, 1986. doi: 10.1137/0725066.
- [13] Alexander Samarskii and Alexei Goolin. *Chislennye Metody*. Nauka, Moscow, 1989. ISBN 5-02-013996-3. (in Russian).
- [14] Alexander A. Samarskii. *The Theory of Difference Schemes*. Taylor & Francis Inc, New York, 2001. ISBN 0-8247-0468-1.
- [15] S. S. Askar and A. A. Karawia. On solving pentadiagonal linear systems via transformations. *Mathematical Problems in Engineering*, 2015:9, 2015. doi: 10.1155/2015/232456. article ID 232456.
- [16] C. Bauer, A. Frink, and R. Kreckel. Introduction to the GiNaC framework for symbolic computation within the c++ programming language. 33:1–12, 2002. doi: 10.1006/jscs.2001.0494.
- [17] Y. Hardy, K. S. Tan, and W.-H. Steeb. *Computer Algebra With Symbolic C++*. World Scientific, first edition, 2008. ISBN 9789812823600.
- [18] M. El-Mikkawy. Fast and reliable algorithm for evaluating nth order pentadiagonal determinants. *Applied Mathematics and Computation*, 202(1):210–215, 2008.
- [19] L. Valova, V. Galaktionov, D. Golub, T. Zaikina, M. Kirakosyan, A. Prikhodko, and S. Torosyan. *Information Support of Users of the Heterogeneous Cluster HYBRILIT*.
- [20] H. L. Stone. Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM Journal of Numerical Analysis*, 4(3):530–538, 1968. doi: 10.1137/0705044.
- [21] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, second edition, 2003. ISBN 978-0-89871-534-7. doi: 10.1137/1.9780898718003.
- [22] G. Schulz. Iterative berechnung der reziproken matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13:57–59, 1933.
- [23] F. Soleymani. A rapid numerical algorithm to compute matrix inversion. *International Journal of Mathematics and Mathematical Sciences*, 2012:11, 2012. doi: 10.1155/2012/134653. article ID 134653.