Faculty of Computing and Telecommunications

Tomasz Hoffmann

Solving the Poisson equation in proper and directed interval arithmetic

PhD Thesis

Supervisor: Prof. Andrzej Marciniak

Supporting supervisor: Dr Habil. Małgorzata Jankowska

Poznan 2021

Proofreading
mgr Kinga Turowska
kinga.turowska@ue.man.poznan.pl

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank my supervisor, Professor Andrzej Marciniak, whose expertise was invaluable in formulating the research questions and methodology. I want to thank you for your patient support and for all of the opportunities I was given to further my research.

I would also like to thank Dr Habil. Malgorzata Jankowska and Ph. D. Barbara Szyszka, for their valuable guidance throughout my work on this thesis. You helped me to choose the right direction and successfully complete my dissertation.

In addition, I would like to thank my parents for their wise, counsel and empathic hearth. I would also like to thank my sister Iza and my dear sainted memory brother Przemek. You are always there for me. Finally, I could not have completed this dissertation without the support of my most beloved wife – Tetiana and my friends who provided stimulating discussions as well as happy distractions to rest my mind outside of my work and research.

Table of contents

Abstract

This dissertation presents the problem of estimating exact solutions for elliptic partial differential equations, on the example of Poisson's equation and its generalizations. As it is known, PDEs (Partial Differential Equations) allow modeling of physical phenomena, many scientific and engineering problems. If such equations cannot be, at least in an easy way, solved If such equations cannot be, at least easily, solved analytically, then we attempt to apply numerical methods. However results obtained by computer are only approximate solutions. It should be noted that within such a broad field as partial differential equations, analytical methods for their solution are still being developed. The question of the existence of a solution to an equation is also extremely important here – in many cases it may not exist. Thus, the problem itself must satisfy certain conditions for the search for approximate solutions to be meaningful.

The interval methods presented in this paper belong to the class FDM (Finite Difference Methods) and allow finding estimates of exact solutions for boundary problems defined for selected PDEs elliptic. In total, methods based on five different differential schemes are presented, for three types of PDEs, which are implemented in three types of arithmetic, i.e., floating point arithmetic, ordinary interval arithmetic, and directed interval arithmetic. The main research hypothesis is that the application of interval arithmetic to the solution of the Poisson equation makes it possible to automatically account for various numerical errors inside the obtained interval solutions. Furthermore, it is shown that the interval methods developed in the verification of this hypothesis can be generalised for the case of linear elliptic PDEs of order two.

An attempt has also been made to refer to a method allowing strict verification of the existence of solutions of the types of PDEs considered in the paper and finding their estimates supported by a mathematical proof. Such a method for elliptic equations is the Nakao method, which uses the FEM (Finite Element Methods) model. The results obtained with both methods, i.e. the interval-based FDM methods proposed in this work and the Nakao method which uses intervals (but not fully interval-based – as pointed out in this work), were compared.

The results obtained allowed positive verification of the research hypothesis. For all the developed methods it was presented how to estimate experimentally the errors of the method. Computational examples, in turn, showed that the exact solution was inside the interval solutions. An interesting prelude to further research seems to be the use of existing VC (verified-computing) methods as a tool for initial error estimation, and then the use of the method presented in this paper for the construction of fully interval

methods (i.e. methods in which the entire computation, i.e. all arithmetic operations are performed on the intervals). As a result, the obtained estimates of exact solutions could not only be more accurate, but also, each time, supported by a mathematical proof – resulting directly from the given VC method.

Abstract (PL)

W tej rozprawie przedstawiono problematykę szacowania rozwiązań dokładnych dla równań różniczkowych cząstkowych eliptycznych, na przykładzie równania Poissona i jego uogólnień. Jak wiadomo równania różniczkowe cząstkowe umożliwiają modelowanie zjawisk fizycznych, wielu problemów naukowych i inżynierskich. Jeśli równania takie nie mogą być, przynajmniej w łatwy sposób rozwiązane analitycznie, to wówczas podejmujemy próbę zastosowania metod numerycznych. Jednakże otrzymywane komputerowo wyniki stanowią jedynie rozwiązanie przybliżone. Należy zauważyć, że w obrębie tak szerokiej dziedziny, jak równania różniczkowe cząstkowe, wciąż opracowywane są analityczne metody ich rozwiązywania. Niezwykle istotna jest tutaj również kwestia istnienia rozwiązania dla danego równania – w wielu przypadkach może ono nie istnieć. Tak więc już samo zagadnienie musi spełniać określone warunki, by poszukiwanie dla niego rozwiązań przybliżonych było sensowne.

Przedstawione w pracy metody przedziałowe należą do klasy FDM (z ang. *finite difference methods*) i pozwalają na znajdowanie oszacowań rozwiązań dokładnych dla zagadnień brzegowych określonych dla wybranych PDE (z ang. *partial differential equations*) eliptycznych. Łącznie przedstawiono metody oparte na pięciu różnych schematach różnicowych, dla trzech typów PDE, które zostały zaimplementowane w trzech rodzajach arytmetyki. tj. arytmetyce zmiennopozycyjnej, przedziałowej zwykłej, i przedziałowej skierowanej. Główna hipoteza badawcza brzmi: zastosowanie arytmetyki przedziałowej do rozwiązywania równania Poissona umożliwia automatyczne uwzględnienie różnych błędów numerycznych wewnątrz otrzymanych rozwiązań przedziałowych. Ponadto wykazano, że metody przedziałowe opracowane w ramach weryfikacji tej hipotezy można uogólnić dla przypadku liniowych PDE eliptycznych rzędu drugiego.

Podjęto również próbę odniesienia się do metody pozwalającej na ścisłą weryfikację istnienia rozwiązań rozważanych w pracy rodzajów PDE oraz znajdowania ich oszacowania popartego dowodem matematycznym. Taką metodą dla równań eliptycznych jest korzystająca z modelu FEM (z ang. *finite element methods*) metoda Nakao. Porównano wyniki uzyskane obiema metodami, tj. zaproponowanymi w tej pracy przedziałowymi metodami FDM oraz korzystającą z przedziałów (lecz nie w pełni przedziałową – na co zwrócono uwagę w tej pracy) metodą Nakao.

Uzyskane wyniki pozwoliły na pozytywną weryfikację postawionej hipotezy badawczej. Dla wszystkich opracowanych metod zaprezentowano sposób w jaki eksperymentalnie można szacować błędy metody. Przykłady obliczeniowe z kolei pokazały, iż rozwiązanie dokładne znajdowało się wewnątrz rozwiązań przedziałowych. Interesującym przedmio-

tem dalszych badań wydaje się użycie istniejących metod VC (z ang. *verified-computing*) jako narzędzia do wstępnego oszacowania błędów, a następnie wykorzystanie zaprezentowanego w tej pracy sposobu konstrukcji metod w pełni przedziałowych (tj. takich, w których całe obliczenia, czyli wszystkie operacje arytmetyczne wykonywane są na przedziałach). W efekcie uzyskiwane oszacowania rozwiązań dokładnych mogłyby być nie tylko dokładniejsze, ale również, każdorazowo, poparte dowodem matematycznym – wynikającym bezpośrednio z danej metody VC.

1 Introduction

Many scientific and engineering problems are described by partial differential equations [12, 13]. They allow modeling of physical phenomena [93], many issues related to control and automation [58], through process optimization [27], to economic issues [82]. If such equations cannot be, at least in an easy way, solved analytically, then we attempt to apply numerical methods [7, 45]. The results obtained using them are approximate solutions [7, 38]. It should be noted that within such a broad domain as partial differential equations, even just finding an approximate solution can be difficult, both due to the complexity of the problem itself and the methods dedicated to it [33, 98]. The question of existence of solution for the given equation is also important here - in many cases it may not exist [55], so the problem itself must meet certain conditions to make the search for approximate solutions meaningful. In the case of numerical methods, the time required to perform the calculations and the required amount of memory are also crucial [45, 49].

From the mathematical point of view, the existence of solutions to the PDEs is still a subject of research and for many classes of such equations, conditions have been defined that they must satisfy to have a solution [12, 13]. On the other hand, in the case of numerical methods, continuous technological development, lasting essentially since the 1950s, contributed significantly to the increase in both the size of operating memory and computing power of computers [50]. This resulted in the development of numerical methods for solving complex mathematical problems, such as partial differential equations [3]. In numerical calculations, floating point arithmetic is most often used, which is defined by IEEE-754 standards [34–36]. However, it has specific limitations [99]. The computer memory stores the representation of a number in binary system with a strictly defined, and thus limited, number of bytes to store it, hence it is necessary to use rounding to be able to represent the real number in such a form, which results in representation errors [7, 45]. Moreover, the result of each arithmetic operation is also rounded, thus rounding errors accumulate during the calculation [15, 19]. Another type of errors are those related to the method we use to solve the problem. Here, a broad issue of their estimation arises, both *a priori* [46, 80] and *a posteriori* [44, 57]. Recently, an area of intensive research is also the field involving attempts to construct methods of so-called *verified computing* (VC). We should mention such works as [23], [75] and [96], but in the author's opinion, the book [81], is the most important, because it deals directly with VC problems solved using interval arithmetic. It should also be emphasized that a very similar term – *verifiable computing* – is used for some algorithms in cryptography [18], but their subject is checking the authenticity (origin) of results, not the mathematical

correctness or accuracy of obtained results, so they are in no way related to the methods described in this paper.

Let us note that even in the situation when for a given problem and the chosen method it is possible to determine the exact error estimate [77], during the computation unexpected problems may still arise, such as *rounding-off effect* [97]. A key and very interesting issue, therefore, seems to be the attempt to find the answer to the question: is it possible to develop such methods which would take into account all the previously mentioned numerical errors already during the computation? Is it possible to estimate these errors not only in general - by e.g. defining the order of the method or mathematical formula for a certain constant constituting their estimation - but also in a detailed way - for a given problem, a given representation of numbers, a given method and certain values of its parameters? Interval arithmetic seems to meet these expectations. Although, as mentioned earlier, the application of this kind of arithmetic used to be too expensive in terms of time complexity, nowadays it is becoming not only possible, but also effective [50].

According to the authors of the book [49],the first works on interval arithmetic date back to 1914, but it was popularized as a model of computation by the work of R. Moore [72] published in 1966. At that time, however, the subject was not widely taken up by other researchers, although it is worth mentioning quite important works from that period [2], [91] and [95]. This was probably due to the high complexity of the proposed model of computation and the limited capabilities of computers at that time. However, in recent years, this subject has been revisited, which is confirmed by the fact that it is more and more often taken up in publications, for instance, such as [26], [49], [52], [81], [86], as an effective tool extending the possibilities of floating-point arithmetic. The worldwide interest in the topic is so great that in 2008 a separate IEEE group was formed to develop a new standard for this arithmetic, and its work culminated in 2015 with the publication of IEEE Standard 1788-2015 (for more on this topic, see [50], [51] and [84]). As for the research conducted by Polish scientists dealing with interval arithmetic, it is worth noting the items [59–61] and [65].

1.1. Scope of work and basic assumptions

The information presented above concerning computer arithmetic is important for the problems covered by the subject of this dissertation. The author's research focuses on solving elliptic partial differential equations on the example of Poisson's equation and its generalized forms [29, 32]. They essentially concern methods from the class of finite differences [4, 59], which lead to systems of linear equations of a very large size. In practice, this raises two problems: first, concerning the optimization of memory usage, and in particular the storage of sparse matrices in memory, and second, related to the need to perform a significant number of arithmetic operations to obtain the results, which is a source of numerical errors and may lead to an erroneous result. The problem of storing sparse matrices and limitations of memory usage is already well known and there are many algorithms that effectively address this type of issue [21], [56] and [92], implemented in a number of well-known libraries for numerical computing [6, 25]. As for the second problem, concerning rounding errors, it was addressed from the theoretical angle in Wilkinson's book [99] and from the practical side in the work [23].In both cases, the analytical approach is used, which makes it possible to mathematically estimate the error in advance, with quite high imprecision. Therefore, an interesting research topic is the problem of auto-mathematical error estimation, so that the obtained results accurately inform about the accuracy of the performed calculations. Therefore, the author proposes the use of interval arithmetic, which makes it possible to collect information about errors arising during calculations.

The preliminary assumptions of the work are defined by the following points.

1. Solving elliptic partial differential equations by methods from the class of finite differences leads to large systems of linear equations the solution of which is subject to the accumulation of rounding errors and the possibility of *rounding-off* effect.
2. The errors of the method as well as rounding errors are not directly taken into account in calculations performed with the existing models of solving partial differential equations in floating point arithmetic.
3. Interval arithmetic gives the possibility to automatically collect information about all types of errors by taking them into account during the calculation.

The aim of the author's research was to analyse the usefulness of the application of interval arithmetic for calculations performed by methods belonging to the class of finite differences within elliptic PDEs, with particular emphasis on Poisson's equation and its generalised forms.

1.2. Research hypothesis

Based on the assumptions listed in the previous section, the main research hypothesis is:

Application of interval arithmetic to solve the Poisson's equation enables automatic inclusion of numerical errors of various types inside the obtained intervals - solutions.	(H1)

he aim of this paper is to verify the hypothesis by attempting to develop a method for solving Poisson's equation, such as specified in hypothesis (H1). The research included also generalizations of Poisson's equation and a certain class of elliptic PDEs. The use of different types of interval arithmetic was also considered. As a result, attempts were made to verify the auxiliary hypotheses mentioned further.

Methods for automatic solution estimation developed in verification of hypothesis (H1) can be generalised for the case of linear elliptic PDEs of order two.	(H2)

The use of directed interval arithmetic in calculations allowsobtaining better (more accurate) estimates than in the usual arithmetic interval metrics.	(H3)

In general, the methods presented in this paper should be treated as heuristics, since the development of a proof that exact solutions are contained in the resulting intervals is a separate mathematical problem. It is taken up by researchers ealing with problems on the borderline between advanced mathematics and numerical methods, and the most important work, which is a very good summary of the current state of this research, is the book [81]. A successful attempt to construct an interval method for verifying the existence of solutions of partial differential equations, together with the proof of their being contained in the obtained intervals, has so far been made by M. T. Nakao [76, 77]. As a starting point, he took a method belonging to the class of finite element methods. Due to the great importance of the method developed by him as a tool allowing to obtain a guarantee of the correctness of the results, the author of this dissertation, within the framework of his research, has attempted to reproduce these results and their application to the class of equations considered by him.

The method for verifying the existence and for estimating solutions of elliptic PDEs, using interval arithmetic and developed by M. T. Nakao, can be successfully applied to the Poisson equation and its generalizations. (H4)

As part of the experimental portion of the dissertation:

- the errors of the method for equations, for which the exact solution is known, are estimated in an analytical way and it is experimentally demonstrated that the exact solution is contained in the resulting intervals,
- the results obtained in both considered interval arithmetic were compared,
- possible directions for further research were identified.

It is worth noting that M. T. Nakao limited his considerations only to equations of a definite form. Such an approach enabled him to derive a mathematical proof, however, it limited the area of applicability of the method proposed by him precisely to this class of equations[1]. In this respect, the methods presented in this work seem to have an advantage that, although they do not give a mathematical guarantee that the solutions will hold, they have been shown experimentally to be effective for a much wider class of elliptic partial differential equations. However, the development of mathematical proofs remains an interesting subject for further research.

[1] We are talking about a certain class of elliptic partial differential equations. This class is described more broadly in Chapter 4 on the Nakao method.

2
Boundary problem for elliptic equations

This chapter describes the problem of the paper from the mathematical point of view. At the beginning, more important information concerning partial differential equations is presented[1]. Then, on the basis of the literature on the subject, basic definitions concerning linear partial differential equations of the second order - their types and types of boundary conditions - are collected. Special emphasis has been placed on PDEs of elliptic type with Dirichlet boundary conditions. Forms of equations, for which methods of solution by interval arithmetic are proposed in the following chapters, are defined. Other approaches to the problem, including analytic ones, are mentioned. However, this has been done in a very limited way and only in order to better define the area of applicability of the numerical methods presented in the dissertation.

2.1. Basic definitions

In general, a partial differential equation is an equation involving an unknown function of two or more variables and some of its partial derivatives. The classification of these equations and the analytical methods to solve them are well known and widely described in literature [12, 14]. Therefore, we refer here only to the most important definitions and terms, which will allow us to place the problem of this work within the broad field of methods for solving PDEs.

Let us first define a general form of the PDE. This requires the introduction of the auxiliary notion of *multi-index*) – it allows writing the partial derivatives in a simplified way.

Definition 1. *Let $\hat{x} = (x_1, x_2, \dots x_n)$ denote a point in the n–dimensional space $\mathbb{R}^n$. We call the vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ with nonnegative integer components a multi-index of length $|\alpha| = \sum_{i=1}^{n} \alpha_i$. Then the set of all partial derivatives of the function $u(\hat{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ defined by multi-indices of length $|\alpha| = k$ an be denoted by*

$$\mathcal{D}^k u(\hat{x}) := \mathcal{D}^{|\alpha|} u(\hat{x}) := \left\{ \frac{\partial^{|\alpha|} u(\hat{x})}{\partial {x_1}^{\alpha_1} \partial {x_2}^{\alpha_2} \dots \partial {x_n}^{\alpha_n}} \right\}.$$

Now, we can write down a general and most commonly used definition of PDEs, which at the same time best illustrates how broad a set of issues they address.

[1] As an abbreviation we will use letters PDE (from: partial differential equations) or the word equation – unless it is clear from the text that a different equation or type of equation is involved.

Definition 2. *Let k denote an integer satisfying the condition $k \geq 1$, and let symbol Ω — denote an open subset of the space $\mathbb{R}^n$ and let $\hat{x} \in \Omega$. Then an equation of the form*

$$L(\mathcal{D}^k u(\hat{x}), \mathcal{D}^{k-1} u(\hat{x}), \ldots, \mathcal{D}u(\hat{x}), u(\hat{x}), \hat{x}) = 0, \quad (\hat{x} \in \Omega) \tag{2.1}$$

is called the k-th order partial differential equation, where the form of the function

$$L : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \ldots \times \mathbb{R}^n \times \mathbb{R} \times \overline{\Omega} \mapsto \mathbb{R}$$

is defined[2]*, while $u : \Omega \mapsto \mathbb{R}$ denotes the unknown of the equation.*

We include a separate definition of a solution, primarily due to the fact that this concept is understood in different ways.

Definition 3. *The solution of a partial differential equation in a general sense is called the set of all functions $u(\hat{x})$ satisfying equation (2.1).*

Definition 4. *The solution of a partial differential equation in the classical sense is called the set of all functions $u(\hat{x})$ satisfying equation (2.1) and continuous together with partial derivatives up to and including order k in the region $\Omega \subset \mathbb{R}^n$.*

From the above definitions, it is clear that both a general and the classical solution of a partial differential equation consists of a set of functions. Most often, it is not possible to obtain simple and explicit formulas for solutions of a given equation [12]. In such a case, attempts are made to prove their existence or that they satisfy certain properties [81]. Finding one function which is a solution of the equation requires limiting the problem by introducing additional conditions. Therefore, let us introduce the necessary terms.

Definition 5. *A boundary value problem (BVP for short) for equation (2.1) is called the problem of finding a function $u(\hat{x})$which on the boundary part of the region $\overline{\Omega} = \partial\Omega \cup \Omega$ — et us denote it by $\Gamma = \partial\Omega$ — satisfies certain conditions $u(\hat{x})|_\Gamma$, hereafter called boundary conditions.*

Definition 6. *For problems where one of the independent variables is time t the concept of initial value problems (IVP) is used, which is written $u(t)|_{t=0}$. If the function u depends also on spatial variables $u = u(\hat{x}, t)$ then initial conditions can be used in conjunction with initial boundary value problems (IBVP).*

After imposing the above restrictions on the function $u(\hat{x})$ we speak of finding a solution not for the equation itself, but for a particular boundary problem. The following types of boundary conditions are most widely used in practice:

- Dirichlet, when we assume certain values for the function $u(\hat{x})$ at the edge Γ, which we denote as: $u(\hat{x})|_\Gamma = g(x, y)$, where g denotes the continuous function at the edge given in the problem Γ,
- Neumann, when we assume certain values for the derivative of the function $u(\hat{x})$ at the edge Γ, i.e. $\frac{\partial u(\hat{x})}{\partial \vec{n}}|_\Gamma = g(x, y)$, where $\vec{n}$ denotes the normal vector [3] external to the area $\overline{\Omega}$,
- Robin, being the linear combination of the two previous conditions, which means, that $\left(a\frac{\partial u}{\partial n} + bu\right)|_\Gamma = g(x, y)$, where a, b, like g, denote continuous functions on the edge Γ.

[2]The notation $\mathbb{R}^{n^k}$ may require some explanation. It results from the fact that all partial derivatives of order k in the set $\mathcal{D}^k u(\hat{x})$ are n^k. For example, for $n = 2$ and $k = 3$ we have $n^k = 2^3 = 8$ combinations of independent variables: $x_1^3, x_1^2 x_2,\ x_2 x_1^2,\ x_1 x_2 x_1,\ x_2^2 x_1, x_1 x_2^2, x_2 x_1 x_2, x_2^3$, which determines 8 partial derivatives of order three of the function $u(\hat{x}) = u(x_1, x_2)$.

[3]This is a normalized vector defined at any point $p \in \Gamma$, perpendicular to the plane tangent to the surface at any point and pointing outside the region $\overline{\Omega}$.

In this context, it is worth noting that not for every boundary problem there is an unambiguous solution. That is why the concept of *well posed boundary problems* is used in literature.

Definition 7. *The boundary problem for the partial differential equation* (2.1) *is called well posed if the following requirements are satisfied for it:*

- *under certain boundary conditions there is a solution for it,*
- *the solution is clear,*
- *the solution is stable, that is, it depends continuously on the boundary conditions.*

In the methods presented in the further, experimental part of this work, we shall refer to well posed problems for which an exact solution is known, as well as to such problems for which we do not know the analytical solution and about which we have no guarantee that they are well posed. We assume that well posed problems will serve us to demonstrate the correctness of the methods presented next. Let us now look at techniques for solving PDEs. The basic ones are the following (the list is based on the book [13]).

1. Variable separation method - a partial differential equation with n independent variables is converted into n proper differential equations.
2. Integral transforms — an equation with n variables can be converted to an equation with $n-1$ variables, and consequently equations with two variables can be reduced to proper differential equations.
3. Changing the coordinate system - the equation can be simplified by, for example, rotating the system axes or converting to polar coordinates.
4. Transformation of the dependent variable - the dependent variable is replaced by a new variable, such one that the form of the equation is simplified.
5. Perturbation methods - a nonlinear equation is converted into a sequence of linear equations that approximates it.
6. Conversion to integral equation - PDE is converted to integral form in which the dependent variable is under the integral. It is then solved using integral equation techniques
7. Impulse-response technique[4] – the boundary and/or initial conditions are converted into a series of simple impulses, then the response to each impulse is noted. The total response is obtained by adding the component responses together.
8. Variational methods – inding solutions of an equation is presented as a minimization problem (according to the so-called minimum energy principle[5], in which solutions for certain PDEs of elliptic type are shown to exist). It is then assumed that the minimum of the expression equivalent to the equation simultaneously expresses its solution.
9. Series expansion of eigenfunctions – the solution of the equation is written as the sum of the eigenfunctions [6].
10. Numerical methods - there are many approaches to the numerical solution of PDEs, the most important being *finite differences methods* (FDM for short) and *finite element methods* (FEM for short) (see, e.g., [10] and [37]). Other well-known approaches include *spectral methods* [5], *meshless methods* [11], and gradient discretization methods [1].

[4] A chapter on this technique can be found in [85].
[5] A more extensive description of such methods can be found in the works [53, 100].
[6] Methods of this kind are discussed in [9].

This dissertation focuses on boundary problems with Dirichlet conditions for second order linear elliptic equations and their solution using numerical methods from the FDM class.

2.2. Equation types analysed

We will describe here the classes of partial differential equations for which interval methods have been designed – so the focus is only on equations of elliptic type. However, we will begin by defining the most general form of linear PDEs of order two, since they are the starting point for all classes of equations considered within this work.

Definition 8. *Let us denote the partial derivatives as follow:*

$$u_{xx} = \frac{\partial^2}{\partial x^2}u(x,y), \quad u_{xy} = \frac{\partial^2}{\partial x \partial y}u(x,y), \quad u_{yy} = \frac{\partial^2}{\partial y^2}u(x,y),$$
$$u_x = \frac{\partial}{\partial x}u(x,y), \quad u_y = \frac{\partial}{\partial y}u(x,y).$$

A linear partial differential equation of order two, defined at points$(x,y) \in \Omega \subseteq \mathbb{R}^2$*, is called an equation of the form*

$$a_1 u_{xx} + b u_{xy} + a_2 u_{yy} + d u_x + e u_y + c u = f, \tag{2.2}$$

where

$$a_1 = a_1(x,y), \quad a_2 = a_2(x,y), \quad b = b(x,y), \quad c = c(x,y),$$
$$d = d(x,y), \quad e = e(x,y), \quad f = f(x,y)$$

denote continuous functions of class $\mathbb{C}^2$ *that are coefficients of the equation, and* $u = u(x,y)$ *denote a certain function, also continuous of class* $\mathbb{C}^2$*, of unknown form, whose finding is the solution of the equation.*

Note that only the derivative of u_{xy}, is written in this equation, and there is no derivative of u_{yx}. This is legitimate by virtue of Schwarz's theorem [7] , which states that once the function $u = u(x,y)$ satisfy the continuity condition (as in the definition above), there is the equality $u_{xy} = u_{yx}$.

Definition 9. *Depending on the value of the determinant* $W = b^2 - 4a_1a_2$ *we say that equation* (2.2) *is elliptic if* $W < 0$*, parabolic if* $W = 0$ *and hyperbolic if* $W > 0$*.*

The simplest and one of the most representative of the elliptic-type equations is Poisson's equation, and interval methods were developed for it, which then were generalized by constructing successive ones for wider and wider classes of equations. Below, the ones for which tests were carried out are presented. Subsequent generalizations tend towards the form (2.2), although this, the most general, form was not taken into account. For all equations under consideration we assume that the ellipticity condition is satisfied. All of them are considered on a rectangular region denoted as $\overline{\Omega} = [\alpha_1, \alpha_2] \times [\beta_1, \beta_2]$. Furthermore, in this paper we restrict ourselves to the cases where $b = 0 \wedge d = 0 \wedge e = 0$. The ellipticity condition then takes the form

$$-4a_1a_2 < 0. \tag{2.3}$$

[7] The theorem on the symmetry of partial derivatives of order two is sometimes called *Clairaut's theorem*. However, researchers of history of mathematics admit that the author of the first error–free proof of this theorem is H. A. Schwartz (1843–1921). The proof and more information on this theorem can be found in the work [71].

For the sake of further notation, let us define two more auxiliary operators.

Definition 10. *The Laplace operator for a function of n variables is defined as follows:*

$$\triangle u(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2}. \tag{2.4}$$

Definition 11. *The gradient of a function of n variables is defined as follows:*

$$\nabla u(x_1, x_2, \ldots, x_n) = \left(\frac{\partial u}{x_1}, \ldots, \frac{\partial u}{x_n} \right)^T. \tag{2.5}$$

We may now proceed to define further elliptic equations analysed in this paper, starting from their simplest form up to their generalised form. We will make use here of the simplified notation introduced in Definition 8. As previously mentioned, the *Poisson Equation* (PE) was the starting point for the methods developed.

Definition 12. *Poisson's equation is an equation of the form*

$$u_{xx} + u_{yy} = f. \tag{2.6}$$

Numerical methods were then prepared to solve a generalized form of the above equation, which we will abbreviate as GPE (*Generalized Poisson Equation*).

Definition 13. *The generalized Poisson's equation will be called an equation of the form*

$$a_1 u_{xx} + a_2 u_{yy} = f. \tag{2.7}$$

In the final stage of the research, the equations from the class of elliptic equations were considered, for which the method (belonging to the FEM class) was presented by M. T. Nakao [77]. These equations can be generally written in the form

$$\triangle u + b \nabla u + cu = -f. \tag{2.8}$$

In the above notation $b = [b_i(x_1, \ldots x_n)]$ is a vector of functions being coefficients of the equation. Let us note that equation (2.6) may also be presented using the Laplace operator and then written in the form

$$\triangle u = f \text{ in area } \overline{\Omega}$$

resembles a simplified version of the equation from Nakao's method, in which the coefficient $b = 0$ and $c = 0$. Thus it is possible to propose a hypothesis (see (H4)), that Nakao's method can be applied to Poisson's equation – as it operates on more general equations. It is one of the reasons for the implementation of this method, the results for the general equations were reproduced, and then an attempt was made to apply this method to the Poisson equation. The following sections describe the results of this research and determine the applicability of Nakao's method for the class of equations described in this work, i.e. the Poisson equation and its generalised form.

Definition 14. *An elliptic equation of the class Nakao will be called an equation of the form*

$$a_1 u_{xx} + b u_{xy} + a_2 u_{yy} + cu = f. \tag{2.9}$$

For all the equation types described in this section, differential and interval schemes of the finite difference method are presented later in this paper.

2.3. Finite Difference Method

The theory related to the construction of differential schemes for PDEs is widely presented in papers such as [22], [54] and [94]. Among the studies of Polish authors we should mention [38] and [16]. However, these papers deal with floating point arithmetic and in this paper we refer only to the most important notions concerning the design of FDM methods themselves. Let us also note that in order to organize the way the methods themselves are constructed and to improve the subsequent verification of the results, we restrict ourselves here only to two-dimensional problems defined on a rectangular domain with Dirichlet boundary conditions. The problem under consideration can be formulated as follows: find the function $u = u(x, y)$ satisfying equation (2.2) together with Dirichlet boundary conditions of the form

$$u(x,y) = \begin{cases} \varphi_1(x), & \text{jeśli } y = \alpha_1, \\ \varphi_2(y), & \text{jeśli } x = \beta_1, \\ \varphi_3(x), & \text{jeśli } y = \alpha_2, \\ \varphi_4(y), & \text{jeśli } x = \beta_2, \end{cases}$$

where $(x, y) \in \overline{\Omega} = \{(x, y) : \alpha_1 \leq x \leq \alpha_2 \wedge \beta_1 \leq y \leq \beta_2\}$.

Design of the finite difference method requires discretization of the problem. It consists in covering the area $\overline{\Omega}$ with a grid of isolated points, the so–called nodes [38, p. 159], and then writing a differential problem for them. In the methods presented in this paper we always assume a regular grid (see Fig. 2.1) with nodes distant by h and k from the x and y axes, respectively.

Definition 15. *Let n and m denote arbitrary integers. Then the grid of nodes $\overline{\Omega}_{h,k}$ is called the set of points (x_i, y_j) such that*

$$(x_i, y_j) = (ih, jk) \in \overline{\Omega},$$

where $h = (\alpha_2 - \alpha_1)/n$, $k = (\beta_2 - \beta_1)/m$, with $i = 0, 1, \ldots, n$, $j = 0, 1, \ldots, m$.

For the methods described in this work, we assumed a rectangular region $\overline{\Omega} = [0, 1]\times$ $\times[0, 1]$ and a uniform grid where $m = n$, implying $h = k$.

Definition 16. *Let us formulate any two-dimensional differential problem as follows: determine the function $u(x, y)$ defined in area $\overline{\Omega}$ and satisfying the differential equation*

$$Lu(x,y) = f(x,y), \quad (x,y) \in \Omega$$

and the boundary conditions

$$u(x,y) = \varphi(x,y), \quad (x,y) \in \Gamma,$$

where f, φ are given functions, and the operator L can be nonlinear.

Definition 17. *The differential problem from def. 16 can be represented in the form of a differential approximation task defined as follows: find a function $u_h(x, y)$ defined in area $\overline{\Omega}_h$ such that*

$$\begin{aligned} L_h u_h(x,y) &= f_h(x,y), \quad (x,y) \in \Omega_h, \\ u_h(x,y) &= \varphi_h(x,y), \quad (x,y) \in \Gamma_h, \end{aligned}$$

where f_h and φ_h are given functions approximating f and φ.

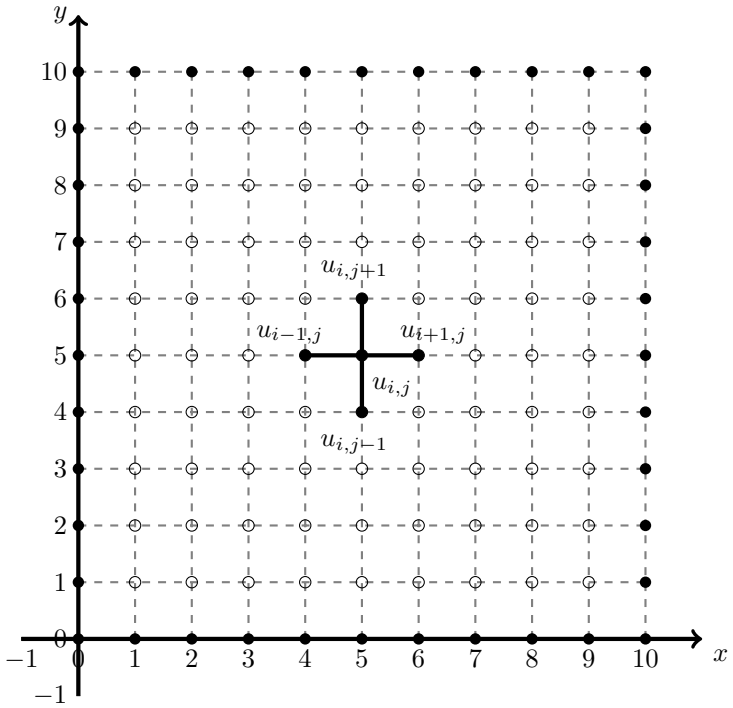

Figure 2.1. The mesh grid 11×11 for the finite difference method.

Since we only analyse two-dimensional tasks, let us consider the following approximations of the partial derivatives of the function $u(x, y)$ using central differences:

$$D^2_{xx} = \frac{\partial^2 u}{\partial x^2}(x, y) = \frac{u(x + h, y) - 2u(x, y) + u(x - h, y)}{h^2} + O(h^2),$$

$$D^2_{yy} = \frac{\partial^2 u}{\partial y^2}(x, y) = \frac{u(x, y + k) - 2u(x, y) + u(x, y - k)}{k^2} + O(k^2).$$

For a given grid node (x_i, y_j) he central differences can be written in a simplified, contracted form

$$\sigma^2_{xx} u_{ij} = \frac{u(x_i + h, y_j) - 2u(x_i, y_j) + u(x_i - h, y_j)}{h^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2},$$

$$\sigma^2_{yy} u_{ij} = \frac{u(x_i, y_j + k) - 2u(x_i, y_j) + u(x_i, y_j - k)}{k^2} = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j+1})}{k^2}.$$

Note that the difference scheme based on the above two differences, is a five-point scheme (see Fig. 2.1), zn principle, all the methods described in subsequent chapters will operate on five-point schemes and only the central differences will be used. Using the introduced notation, we can more easily write values of operators L and L_h. For example, for the Poisson equation they will be of the form $L = -(D^2_{xx} + D^2_{yy})$ i $L_h = -(\sigma^2_{xx} + \sigma^2_{yy})$.

Transformation of a differential problem into a differential task leads — from the algebraic side — to obtaining the system of linear equations given by the formula

$$A\vec{u}_h = \vec{f}_h \tag{2.10}$$

where

$$\vec{u}_h = [u_{h11}, u_{h21}, \ldots, u_{h,m-1,1}, u_{h12}, \ldots, u_{h,m-1,n-1}]^T,$$

$$\vec{f}_h = [f_{h11} + \frac{1}{k^2}\varphi_{10} + \frac{1}{h^2}\varphi_{01}, \ldots, f_{h22}, \ldots, f_{h,m-1,n-1} + \frac{1}{k^2}\varphi_{m-1,n} + \frac{1}{h^2}\varphi_{m,n-1}]^T$$

and

$$u_{hij} = u_h(ih, jk), \quad f_{hij} = f_h(ih, jk), \quad i = 1, 2 \ldots m, \quad j = 1, 2, \ldots n.$$

In principle, the system of equations given by equation (2.10) can be solved by any method. However, it should be noted that the matrix A is a sparse, banded matrix, and the number of bands depends on the number of points considered in the differential diagram, so for the diagram in Fig. 2.1 it will be a matrix with five bands. Moreover, as the grid size increases, the size of the array matrix increases rapidly. If we assume $m = n$, then we obtain a matrix of coefficients of the size expressed by the formula $(m-1)^2 \times (m-1)^2$. For grids of the size $m = n = 101$ we have exactly $10\,000 \times 10\,000 = = 100\,000\,000$ elements of matrix A. Storing such arrangements in computer memory requires optimization of the way the matrix itself is stored as well as the use of appropriate elimination methods.

3
Fundamentals of proper and directed interval arithmetic

This chapter presents basic information about proper and interdigitated floating point arithmetic. It also describes the types of numerical errors.

3.1. Floating-point arithmetic

The majority of computer calculations, which involve operations on real numbers, are performed using the so-called floating point arithmetic (more information about this arithmetic can be found, e.g., in [15], [19] and [74]).

Definition 18. *Let some integer $p \geq 2$ hereafter called a subbase, be fixed. Then, any nonzero real number l can be uniquely represented by an ordered triple (s, e, d) chosen in such a way that*

$$l = s \cdot d \cdot p^e, \tag{3.1}$$

where $s = +1$ or -1 is the sign of the number l, e – lis an integer with a sign called the feature or exponent, and the number $d \in [1/p, 1)$ is called the mantissa.

In the general case numbers α and d may have infinitely many digits, e.g. when $p = 10$ and $l = 1/3$ or $p = 2$ and $l = 1/10$. However, for technical reasons, restrictions are imposed on the number of digits of the mantissa d and on the size of the feature e. This results in real numbers being represented in machine notation using their expansions of the form (3.1), restricted to a finite number of digits.

When a real number, hereafter also called a floating point number, can be accurately represented in a computer, we say that it is a machine number. Other real numbers are rounded to machine numbers, provided their exponent e satisfies the constraint

$$e_{min} \leq e \leq e_{max}, \tag{3.2}$$

whereby

$$e_{min} \leq 0 \leq e_{max}. \tag{3.3}$$

If in calculations there occurs a number whose feature does not belong to the interval defined by the non-equality (3.3), we speak about the emergence of surplus or deficit. This problem rarely occurs in modern computers because the range of represented numbers is so large that the occurrence of, e.g., an excess is rather the effect of an error in data or in

the program. Let us denote by $\overline{\omega}$ the smallest positive machine number. Let us assume

$$\overline{\omega} = p^{-t} p^{e_{min}},$$

where t denotes the number of digits of the mantissa. We speak of an undershoot when, during calculations, a number appears whose absolute value is smaller than $\overline{\omega}$.

The assignment of a number $x \in \mathbb{R}$ to its floating point representation $fl(x)$ is usually done by so-called rounding. In general, the following types of rounding are distinguished:

- to the nearest machine number or, in the case of equal distances,to the even number (standard used),
- up (in the $+\infty$ direction),
- down (in the $-\infty$ direction),
- truncation (towards 0).

The floating-point arithmetic control unit [34–36] is responsible for the settings regarding the type of rounding used.

Definition 19. *Let a number $x \in \mathbb{R}$ and $fl(x)$ denote its floating-point representation. Then the accuracy of the floating-point arithmetic in which the number x has been expressed is given by the formula*

$$u = max_{\lceil log_\beta |x| \rceil \in [e_{min}, e_{max}]} \frac{|x - fl(x)|}{|x|}$$

(*based on* [70]).

The standard rounding rule [35, 42] states that if there is no overflow, then the floating point representation of the number x is assumed to be

$$fl(x) = x(1 + \delta),$$

with $|\delta| < u$.

More on the representation of floating-point numbers can be found in IEEE-754 standards [34–36] available on the organization's website at [34]. The implementation of the methods presented in the following chapters was done in C++ language, and to represent real numbers we chose *long double* type. It is the type which allows to store floating point numbers, with the highest precision, available in C++ language standard. It should be noted that there are libraries that allow storing numbers in any precision, i.e. limited only by the size of available computer memory (see [17] and [41]). Table 3.1 provides basic information about the data types for representing floating-point numbers in these languages. It is worth noticing, see Table 3.1, that while for particular types the number of bits used to represent the property and the mantissa is - theoretically - strictly determined, i.e. compliant with the IEEE-754, standard, in practice - the number of bytes used to store them in the computer's memory depends both on the machine itself (specifically, on the size of the so-called „word",meaning the computer's memory unit) and on the compiler with which we build the program. This results from the fact that compilers of various producers implement different solutions, see Table 3.2. These are important points to pay attention to when creating and running a program.

Table 3.1. Data types for floating-point number representation in C++

Język	Typ danych	Mantysa	Cecha
C++	float	24	8
	double	53	11
	long double	64	16

Table 3.2. Number of bytes used to represent floating point numbers in C++ depending on the compiler and the word size in the computer's memory

Segment word size	16-bit			32-bit						64-bit			
Compiler	Microsoft	Borland	Watcom	Microsoft	Intel Windows	Borland	Watcom	GCC v. 4.x	Intel Linux	Microsoft	Intel Windows	GCC	Intel Linux
float	4	4	4	4	4	4	4	4	4	4	4	4	4
double	8	8	8	8	8	8	8	8	8	8	8	8	8
long double	10	10	8	8	16	10	8	12	12	8	16	16	16

3.2. Numerical errors

All inaccuracies arising during calculations on a digital machine are called numerical errors. According to [16] three types of numerical errors can be distinguished:

a) representation errors,

b) truncation errors,

c) rounding errors,

d) input data errors.

Representation errors. The computer's memory is limited, whereas the set of real numbers is an infinite set. Moreover, a real number may have an infinite representation, which often depends on the number system in which it is expressed. The exceptions are infinitesimal numbers e.g. $\sqrt{2}$, π, e, which have an infinite representation in every system. As a result, numbers entered as input data may have a representation in the computer's memory and registers that differs from their exact values. We refer to this situation as representation errors.

Truncation errors. Often determining the exact solution would require an infinite number of operations. It is then limited, causing so-called truncation errors. This happens e.g. in the case of calculating the values of infinite sums, when one applies an approximation taking into account the sum of a finite number of components, sufficiently close to the exact value.

Example 3.1. Calculating the expression e^x can be reduced to the task of finding the sum of the series

$$1 + x + \frac{1}{2}x^2 + \ldots + \frac{1}{n}x^n + \ldots.$$

For a sufficiently large value of n the sum

$$1 + x + \frac{1}{2}x^2 + \ldots + \frac{1}{n}x^n$$

could be equal to the pre-rounded e^x, i.e. the misrepresentation. Due to the long computation time of such sums, a restriction to a relatively small number of components is applied and hence truncation errors arise.

Rounding errors. These errors occur during calculations and are generally difficult to avoid. They result from the fact that the result of each arithmetic operation is rounded to a machine value. They may be reduced by skillfully changing the order of operations. A detailed analysis of rounding errors occurring during calculations on floating point numbers was undertaken by J. H. Wilkinson in his work [99].

Input errors. Errors resulting from the difference between the value entered into computer calculations and the exact value. They are most often caused by the inaccuracy of the measuring equipment analysing specific physical parameters.

3.3. Proper interval arithmetic

In interval arithmetic, each number is represented as a pair - the lower and upper ends of an interval. The issues connected with performing arithmetic operations and defining functions operating on intervals are the subject of research in a separate field, the so-called interval analysis. In this section, basic information about it is presented, which is taken from the works [40, 73].

Definition 20. *Assuming that values $a^-, a^+ \in \mathbb{R}$, the set of ordered pairs $[a^-, a^+]$ defined as follows:*

$$A = [a^-, a^+] = \{a \in \mathbb{R} : a^- \leq x \leq a^+\}$$

is called the set of proper intervals.

The following rounding off shall be used in calculating the ends of the interval:

- $a^- := \nabla a^-$ – rounding down (in the $-\infty$ direction),
- $a^+ :=\triangle\ a^+$ – round up (in the $+\infty$ direction).

Definition 21 (equality of intervals)**.** *Let X and Y denote the proper intervals. Then*

$$X = Y \Leftrightarrow x^- = y^- \wedge x^+ = y^+.$$

Definition 22. *The intersection of intervals is defined as follows:*

$$X \cap Y = \{z : z \in X \wedge z \in Y\} = [\max\{x^-, y^-\}, \min\{x^+, y^+\}].$$

If $y^+ < x^- \vee x^+ < y^-$, then

$$X \cap Y = \oslash.$$

Definition 23. *The union of itervals is given by*

$$X \cup Y = \{z \in \mathbb{R} : z \in X \vee z \in Y\}.$$

In general, such a union is not a range, but a set. Therefore, in calculations, we use union of the form

$$X \underline{\cup} Y = [\min\{x^-, y^-\}, \max\{x^+, y^+\}].$$

There is therefore

$$X \cup Y \subseteq X \underline{\cup} Y.$$

Example 3.2. Take the intervals $X = [-1, 0]$ and $Y = [1, 2]$. Then $X \underline{\cup} Y = [-1, 2]$, while $X \cup Y = [-1, 2] - (0, 1)$.

Intersection plays an important role in interval analysis. If we assume that two intervals X and Y contain an exact solution, then the interval $X \cap Y$, which may be narrower, also contains a solution..

Example 3.3. Suppose that two experiments were performed independently to measure the value of the acceleration of the Earth, and the results were $r_1 = 9.8$ and $r_2 = 9.9$, both with error $\epsilon < 0.1$. In the interval representation, we have $R_1 = [9.7, 9.9]$ and $R_2 = [9.8, 10.0]$, respectively, so $R_1 \cap R_2 = [9.8, 9.9]$, which is a narrower interval than both input intervals. If the intersection result was empty, it would mean that one of the measurements may have been made incorrectly or the errors were estimated incorrectly.

The basic values which characterize the interval are

a) the width of the interval, denoted by $w(X)$, defined as the difference

$$w(X) = x^+ - x^-,$$

b) the absolute value of $|X|$ defined as

$$|X| = max\{|x^-|, |x^+|\},$$

c) the centre of the interval, denoted by $m(X)$, given by

$$m(X) = \frac{1}{2}(x^- + x^+).$$

Just as on numbers, arithmetic operations can be performed on intervals, but because of the way intervals are specified, they are similar to operations on elements of sets.

Definition 24. *Let $\odot$ denote a binary arithmetic operation, while X i Y denote intervals. Then*

$$X \odot Y = \{x \odot y : \ \ x \in X, \ \ y \in Y\}. \tag{3.4}$$

Based on the general definition (3.4) basic arithmetic operations and the ends of the resulting intervals have the following forms:

a) addition

$$X + Y = \{x + y : x \in X, y \in Y\} = [x^- + y^-, x^+ + y^+],$$
$$x^- + y^- \leq x + y \leq x^+ + y^+,$$

b) subtraction

$$X - Y = \{x - y : x \in X, y \in Y\} = [x^- - y^+, x^+ - y^-],$$
$$x^- - y^+ \leq x - y \leq x^+ - y^-,$$

c) multiplication

$$X \cdot Y = \{x \cdot y : x \in X, y \in Y\} = [\min \ S, \max \ S],$$

where

$$S = \{x^-y^-, x^-y^+, x^+y^-, x^+y^+\},$$

d) division

$$X/Y = \{x/y : x \in X, y \in Y\} = X \cdot (1/Y),$$
$$1/Y = \{1/y : \ \ y \in Y \wedge 0 \notin Y\} = [1/y^+, 1/y^-].$$

Just as for floating-point numbers, functions can be defined for intervals.

Definition 25. *Let a real function f of a real variable x and an interval X be given. We define the interval function $f(X)$ as follows:*

$$f(X) = \{f(x) : x \in X\}.$$

Arithmetic operations should not be confused with operations on sets, because

$$\begin{aligned} X + Y &\neq X \cup Y, \\ X - Y &\neq X \backslash Y, \\ X \cdot Y &\neq X \cap Y. \end{aligned}$$

In the implementation of interval arithmetic, it is important that each action is performed in the following order: setting the rounding down, computing the left end of the interval, setting the rounding up, computing the right end of the interval.

3.4. Directed interval arithmetic

In this section, by directed interval arithmetic we mean the arithmetic described in the works of E. Popova [86] and S. Markov [69].We present selected information from those works, concerning the most important definitions, which turned out to be crucial for the implementation of this arithmetic in the C++ language (made available as the *Interval.h* module). From the theoretical point of view, the most important is that we generalise here the notion of interval, allowing it to include also the situations where the left end is greater than the right end.

Definition 26. *The set $\mathbb{H}$ of all directed intervals is defined as follows:*

$$\begin{aligned} \mathbb{H} = \left\{[a,b] : a,b \in \mathbb{R} = \mathbb{IR} \cup \overline{\mathbb{IR}},\ \textit{gdzie } \overline{\mathbb{IR}}\right\} &= \left\{[a^-,a^+] : a^- \leq a^+ \ \wedge \ a^-,a^+ \in \mathbb{R}\right\} \\ &\cup \left\{[a^-,a^+] : a^- > a^+ \ \wedge \ a^-,a^+ \in \mathbb{R}\right\}. \end{aligned}$$

Next, let us enter the set of intervals containing zero as well as the interval sign and direction operators – they will be necessary to define arithmetic operations on directed intervals.

Definition 27. *The set of directed intervals containing zero is called the set*

$$\mathbb{T} = \left\{A \in \mathbb{IR} : a^-a^+ \leq 0\right\} \cup \left\{A \in \overline{\mathbb{IR}} : a^-a^+ \leq 0\right\} = \mathbb{Z} \cup \overline{\mathbb{Z}}.$$

Definition 28. *For any directed interval $A = [a^-, a^+]$ its sign is called the quantity*

$$\sigma(A) = \begin{cases} +, \ \textit{jeżeli } 0 \leq a^- \cdot a^+, \\ -, \ \textit{jeżeli } a^- \cdot a^+ \leq 0, \ \textit{ale } [a^-,a^+] \neq [0,0]. \end{cases}$$

Definition 29. *The direction of the directed interval $A = [a^-, a^+]$ is called the quantity*

$$\tau(A) = \begin{cases} +, \ \textit{jeżeli } a^- \leq a^+, \\ -, \ \textit{otherwise.} \end{cases}$$

Let us now define basic arithmetic operations in directed interval arithmetic.

Definition 30. *We define addition of directed intervals as follows:*

$$A + B = [a^- + b^-, a^+ + b^+] \ \textit{dla } A, B \in \mathbb{H}.$$

Definition 31. *Multiplication of intervals is defined as*

$$A \times B = \begin{cases} [a^{-\sigma(B)}b^{-\sigma(A)}, a^{\sigma(B)}b^{\sigma(A)}] \ dla\ A, B \in \mathbb{H}\backslash\mathbb{T}, \\ [a^{\sigma(A)\tau(B)}b^{-\sigma(A)}, a^{\sigma(A)\tau(B)}b^{\sigma(A)}] \ dla\ A \in \mathbb{H}\backslash\mathbb{T}, B \in \mathbb{T}, \\ [a^{-\sigma(B)}b^{\sigma(B)\tau(A)}, a^{\sigma(B)}b^{\sigma(B)\tau(A)}] \ dla\ A \in \mathbb{T}, B \in \mathbb{H}\backslash\mathbb{T}, \\ [\min\{a^-b^+, a^+b^-\}, \max\{a^-b^-, a^+b^+\}] \ dla\ A, B \in \mathbb{Z}, \\ [\min\{a^-b^-, a^+b^+\}, \max\{a^-b^+, a^+b^-\}] \ dla\ A, B \in \overline{\mathbb{Z}}, \\ 0, \ dla\ A \in \mathbb{Z}, B \in \overline{\mathbb{Z}}\ lub\ A \in \overline{\mathbb{Z}}, B \in \mathbb{Z}. \end{cases}$$

From the definition of multiplication for an interval $B \in \mathbb{H}$ we obtain

$$(-1) \times B = [-b^+, -b^-] = -B.$$

Hence, subtraction can be defined as

$$A - B = A + (-B) = [a^- - b^+, a^+ - b^-], \quad A, B \in \mathbb{H}.$$

Note that any directed improper interval, i.e., $A = [a^-, a^+]$, satisfying the conditions $a^+ \leq b \leq a^-$, is contained in the point interval $B = [b, b]$, $A \subseteq B$.

Definition 32. *With respect to the operations of addition and multiplication, there are opposite and inverse elements defined as follows:*

$$-_h A = [-a^-, -a^+] \ dla\ A \in \mathbb{H},$$
$$1/_h A = [1/a^-, 1/a^+] \ dla\ A \in \mathbb{H}\backslash\mathbb{T}.$$

For intervals $A = [a^-, a^+] \in \mathbb{H}\backslash\mathbb{T}$ there is also an operator

$$1/A = 1/_h A\_ = [1/a^+, 1/a^-],$$

where $A\_ = [a^+, a^-]$, with $1/_h(1/A) = 1/(1/_h A) = A\_$. The unary-argument operator $A\_$ is called a dual operator because of the following property:

$$A\_ = [a^+, a^-] = -_h(-A) = -(-_h A).$$

Moreover, for $A, B \in \mathbb{H}$ the following properties occur:

$$A \subset B \Longleftrightarrow A\_ \supseteq B\_, \quad (A \circ B)\_ = A\_ \circ B\_, \quad \circ \in \{+, -, \times, /\}.$$

Definition 33. *We define division of directed intervals as follows:*

$$A/B = A \times (1/B) = \begin{cases} [a^{-\sigma(B)}/b^{\sigma(A)}, a^{\sigma(B)}/b^{-\sigma(A)}] \ dla\ A, B \in \mathbb{H}\backslash\mathbb{T}, \\ [a^{-\sigma(B)}/b^{-\sigma(B)\tau(A)}, a^{\sigma(B)}/b^{-\sigma(B)\tau(A)}] \ dla\ A \in \mathbb{T}, B \in \mathbb{H}\backslash\mathbb{T}. \end{cases}$$

Note also that for each interval $A = [a^-, a^+] \in \mathbb{H}$ we can assign a proper interval using the function

$$pro(A) = \begin{cases} [a^-, a^+], \ jeśli\ \tau(A) = +, \\ [a^+, a^-], \ jeśli\ \tau(A) = -. \end{cases}$$

3.5. Interval systems of linear equations

Solving systems of linear equations in interval arithmetic is a problem whose complexity results from the very posing of the problem. Let us consider a system of linear equations

of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \text{ gdzie } \boldsymbol{A} \in \mathbb{IR}^{n\times n}, \quad \boldsymbol{b}, \boldsymbol{x} \in \mathbb{IR}^{n}. \tag{3.5}$$

Equation (3.5) defines the entire set of systems of linear equations defined as follows:

$$Ax = b, \text{gdzie } A \in \boldsymbol{A}, b \in \boldsymbol{b} \text{ oraz } x \in \boldsymbol{x}.$$

While for classical systems of linear equations we can verify the existence of a univariate solution by the *Kroneckera-Capellego*[1], for interval systems of equations this has been shown to be an NP-hard problem in general (see [48], [87] and [88]). According to [28] and [89] we can say that $x \in \mathbb{R}^n$ and at the same time $x \in \boldsymbol{x}$ is a solution which is:

- weak if $\exists A \in \boldsymbol{A}, \exists b \in \boldsymbol{b} : Ax = b$,
- strong if $\forall A \in \boldsymbol{A}, \forall b \in \boldsymbol{b} : Ax = b$,
- tolerable if $\forall A \in \boldsymbol{A}, \exists b \in \boldsymbol{b} : Ax = b$,
- controllable if $\exists A \in \boldsymbol{A}, \forall b \in \boldsymbol{b} : Ax = b$.

Crucial for the methods described in this paper, however, is the theorem given by Moore and Kearfott (see p. 89 of theorem 7.1 in [73]), reproduced below.

Theorem 1. *If the division by an interval containing zero occurs in no step of the exact method used for solving the system of equations (3.5) and no overflow or underflow exception arises, then for every matrix $A \in \boldsymbol{A}$ and for every vector $b \in \boldsymbol{b}$ there exists a solution $x \in \boldsymbol{x}$.*

In the implementation of interval arithmetic, the *Interval.h* module takes into account the situations mentioned in Theorem 1 by aborting the computation and raising a *Run-time Exception*, with an appropriate message, whenever a given arithmetic operation causes them. Therefore, we can assume that for the systems of linear equations considered in this work we obtain a strong solution, as long as the computation ends without the above exceptions. However, this is not equivalent to finding solutions for the given differential equations solved by finite difference methods. In order to be certain that the resulting interval $\boldsymbol{x}$ contains the exact solution we would have to be guaranteed that we have correctly estimated the error of the method. The error which, as described in the following sections, we include in the vector $\boldsymbol{b}$. Since this error, described in the following sections, is estimated experimentally, all the proposed methods are heuristic in nature.

Let us note that if we were to obtain exact methods from the area of *verified computing* – i.e., full automatic verification of existence of solutions to elliptic PDEs – using FDM methods, then the existence of an exact solution to a given PDE within an interval $\boldsymbol{x}$ requires a suitable mathematical proof, which in interval arithmetic is no trivial task, also due to the fact that in general solving a system of equations resulting from a differential scheme may itself be an NP-hard problem. It also involves the necessity of making many assumptions about the form of the equation as well as about the function that is the solution. The first non-heuristic interval method for verification of existence and estimation of solutions for a family of elliptic equations was proposed by Nakao. However, it is based on the theory of FEM methods and does not require solving interval systems of linear equations, but only classical systems in floating point arithmetic. Only intermediate and final solution estimates are written as intervals. The method is described in detail in the next chapter.

[1] One of the basic theorems of elementary linear algebra – says that if the row of matrix A is equal to the row of the same matrix extended by a vector of free expressions, i.e. $[A|b]$ hen the solution of the system of equations exists and is unambiguous (on the basis of [47])

3.6. Implementation of interval arithmetic

Details of the implementation of directed interval arithmetic are presented in the works [43], [69] and [86]. Basically, its implementation comes down to determining for intervals A and B and each arithmetic operation $\circ \in \{+, -, \times, /\}$ the resulting intervals of the form

$$\Diamond(A \circ B) = [\nabla(A \circ B)^-, \Delta(A \circ B)^+],$$
$$\boxtimes(A \circ B) = [\Delta(A \circ B)^-, \nabla(A \circ B)^+],$$

where the operator $\Diamond$ denotes a rounding to the outside and the operator $\boxtimes -$ denotes a rounding to the inside. The symbol ∇ is used for rounding to $-\infty$, and the symbol Δ – for rounding in the $+\infty$ direction. An example implementation of such specified arithmetic operations in interval directed arithmetic is available in the PASCAL-XSC and C-XSC languages supporting the IEEE-1788 standard for floating-point arithmetic, described in detail in [34].

From the above formulas, it follows that during implementation we obtain two resultant intervals for each of the arithmetic operations. Thus, there arises the problem of choosing the interval which is to be used in further calculations. From the theoretical point of view, we should also consider two more intervals as potential results of arithmetic operations:

$$\left[\nabla(A \circ B)^-, \nabla(A \circ B)^+\right] \text{ i } \left[\Delta(A \circ B)^-, \Delta(A \circ B)^+\right].$$

Let us define the way to determine the width w of the interval $A = [a^-, a^+]$. It will be a starting point for the definitions given below.

Algorytm 3.1. Interval width

$w := \Delta(a^+ - a^-)$
if $w < 0$ **then**
  $w := -w$
end if
$\bar{w} := \nabla(a^+ - a^-)$
if $\bar{w} < 0$ **then**
  $\bar{w} := -\bar{w}$
end if
if $w < \bar{w}$ **then**
  $w := \bar{w}$
end if

The way addition and subtraction operations are implemented for the intervals $A = [a^-, a^+]$ and $B = [b^-, b^+]$ is shown in pseudocode 3.2. and 3.3..

Algorytm 3.2. Execution of an addition operation in interval arithmetic

1: $w := \Delta(a^+ - a^-)$
2: **if** $a^- \leq a^+$ **and** $b^- \leq b^+$ (proper intervals) **then**
3: $A + B := [\nabla(a^- + b^-), \Delta(a^+ + b^+)]$
4: **else**
5: $c^- := \nabla(a^- + b^-), \quad c^+ := \Delta(a^+ + b^+)$
6: $d^- := \Delta(a^- + b^-), d^+ := \nabla(a^+ + b^+)$
7: calculate the width w_1 of $[c^-, c^+]$
8: calculate the width w_2 of $[d^-, d^+]$
9: **if** $w_1 \geq w_2$ **then**
10: $A + B := [c^-, c^+]$
11: **else**
12: $A + B := [d^-, d^+]$

end if
end if

Algorytm 3.3. Realization of subtraction operations in interval arithmetic

if $a^- \leq a^+$ and $b^- \leq b^+$ **then**
  $A - B := [\nabla(a^- - b^+), \Delta(a^+ - b^-)]$
else
  $c^- := \nabla(a^- - b^+), c^+ := \Delta(a^+ - b^-)$
  $d^- := \Delta(a^- - b^+), d^+ := \nabla(a^+ - b^-)$
  calculate the width w_1 of $[c^-, c^+]$
  calculate the width w_2 of $[d^-, d^+]$
  if $w_1 \geq w_2$ **then**
    $A - B := [c^-, c^+]$
  else
    $A - B := [d^-, d^+]$
  end if
end if

Note that among the possible resulting intervals, the widest one is always chosen. Implementations of multiplication and addition operations are much more complicated, and their realizations are presented below in algorithms 3.4. and 3.5..

Algorytm 3.4. Implementation of multiplication operations in interval arithmetic

if $a^- \leq a^+$ **and** $b^- \leq b^+$ (proper intervals) **then**
  $A \times B = [\min\{\nabla(a^-b^-), \nabla(a^-b^+), \nabla(a^+b^-), \nabla(a^+b^+)\}$
  $\max\{\Delta(a^-, b^-), \Delta(a^-b^+), \Delta(a^+b^-), \Delta(a^+b^+)\}]$
else
  if ($a^- < 0$ **and** $a^+ < 0$ **or** $a^- > 0$ **and** $a^+ > 0$) **and** ($b^- < 0$ **and** $b^+ < 0$ **or** $b^- > 0$ **and** $b^+ > 0$) **then**
    if $a^- > 0$ **and** $a^+ > 0$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
      $c^- := \nabla(a^-b^-), c^+ := \Delta(a^+b^+)$
      $d^- := \Delta(a^-b^-), d^+ := \nabla(a^+b^+)$
      calculate $C \times D$
    else
      if $a^- > 0$ and $a^+ > 0$ and $b^- < 0$ and $b^+ < 0$ **then**
        $c^- = \nabla(a^+b^-), c^+ := \Delta(a^-b^+)$
        $d^- := \Delta(a^+b^-), d^+ := \nabla(a^-b^+)$
        calculate $C \times D$
      else
        if $a^- < 0$ and $a^+ < 0$ and $b^- > 0$ and $b^+ > 0$ **then**
          $c^- := \nabla(a^-b^+), c^+ := \Delta(a^+b^-)$
          $d^- := \Delta(a^-b^+), d^+ := \nabla(a^+b^-)$
          calculate $C \times D$
        else
          $c^- = \nabla(a^+b^+), c^+ := \Delta(a^-b^-)$
          $d^- := \Delta(a^+b^+), d^+ := \nabla(a^-b^-)$
          calculate $C \times D$
        end if
      end if
    end if
  else

28: **if** $(a^- < 0$ **and** $a^+ < 0$ **or** $a^- > 0$ **and** $a^+ > 0)$ **and** $(b^- \leq 0$ **and** $b^+ \geq 0$ **or** $b^- \geq 0$ **and** $b^+ \leq 0)$ **then**
29: **if** $a^- > 0$ **and** $a^+ > 0$ **and** $b^- \leq b^+$ **then**
30: $c^- := \nabla(a^+b^-), c^+ := \Delta(a^+b^+)$
31: $d^- := \Delta(a^+b^-), d^+ := \nabla(a^+b^+)$
32: calculate $C \times D$
33: **else**
34: **if** $a^- > 0$ **and** $a^+ > 0$ **and** $b^- > b^+$ **then**
35: $c^- := \nabla(a^-b^-), c^+ := \Delta(a^-b^+)$
36: $d^- := \Delta(a^-b^-), d^+ := \nabla(a^-b^+)$
37: calculate $C \times D$
38: **else**
39: **if** $a^- < 0$ and $a^+ < 0$ and $b^- \leq b^+$ **then**
40: $c^- := \nabla(a^-b^+), \quad c^+ := \Delta(a^-b^-)$
41: $d^- := \Delta(a^-b^+), d^+ := \nabla(a^-b^-)$
42: calculate $C \times D$
43: **else**
44: $c^- := \nabla(a^+b^+), c^+ := \Delta(a^+b^-)$
45: $d^- := \Delta(a^+b^+), d^+ := \nabla(a^+b^-)$
46: calculate $C \times D$
47: **end if**
48: **end if**
49: **end if**
50: **else**
51: **if** $(a^- \leq 0$ **and** $a^+ \geq 0$ **or** $a^- \geq 0$ **and** $a^+ \leq 0)$ **and** $\left(b^{-1} < 0\right.$ **and** $b^+ < 0$ **or** $b^- > 0$ **and** $\left.b^+ > 0\right)$ **then**
52: **if** $a^- \leq a^+$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
53: $c^- := \nabla(a^-b^+), c^+ := \Delta(a^+b^+)$
54: $d^- := \Delta(a^-b^+), d^+ := \nabla(a^+b^+)$
55: calculate $C \times D$
56: **else**
57: **if** $a^- \leq a^+$ **and** $b^- < 0$ **and** $b^+ < 0$ **then**
58: $c^- := \nabla(a^+b^-), \quad c^+ := \Delta(a^-b^-)$
59: $d^- := \Delta(a^+b^-), d^+ := \nabla(a^-b^-)$
60: calculate $C \times D$
61: **else**
62: **if** $a^- > a^+$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
63: $c^- := \nabla(a^-b^-), c^+ := \Delta(a^+b^-)$
64: $d^- := \Delta(a^-b^-), d^+ := \nabla(a^+b^-)$
65: calculate $C \times D$
66: **else**
67: $c^+ := \nabla(a^+b^+), c^+ := \Delta(a^-b^+)$
68: $d^- := \Delta(a^+b^+), d^+ := \nabla(a^-b^+)$
69: calculate $C \times D$
70: **end if**
71: **end if**
72: **end if**
73: **else**
74: **if** $a^- \geq 0$ **and** $a^+ \leq 0$ **and** $b^- \geq 0$ **and** $b^+ \leq 0$ **then**
75: $c_1^2 := \nabla(a^-b^-), c_2^- := \nabla(a^+b^+)$
76: **if** $c_1^- \leq c_2^-$ **then**

77: $c^- := c_2^-$
78: **else**
79: $c^- := c_1^-$
80: $c_1^+ := \Delta\left(a^- b^+\right), c_2^+ := \Delta\left(a^+ b^-\right)$
81: **end if**
82: **if** $c_1^+ \leq c_2^+$ **then**
83: $c^+ := c_1^+$
84: **else**
85: $c^+ := c_2^+$
86: $d_1^- := \Delta\left(a^- b^-\right), d_2^- := \Delta\left(a^+ b^+\right)$
87: **end if**
88: **if** $d_1^- \leq d_2^-$ **then**
89: $d^- := d_2^-$
90: **else**
91: $d^- := d_1^-$
92: $d_1^+ := \nabla\left(a^- b^+\right), d_2^+ := \nabla\left(a^+ b^-\right)$
93: **end if**
94: **if** $d_1^+ \leq d_2^+$ **then**
95: $d^+ := d_1^+$
96: **else**
97: $d^+ := d_2^+$
98: calculate $C \times D$
99: **end if**
100: **else**
101: $A \times B := [0, 0]$
102: **end if**
103: **end if**
104: **end if**
105: **end if**
106: **end if**

Algorytm 3.5. Execution of division operations in interval arithmetic

1: **if** $a^- \leq a^+$ and $b^- \leq b^+$ (proper intervals) **then**
2: $A/B := [\min\{\nabla(a^-/b^-), \nabla(a^-/b^+), \nabla(a^+/b^-), \nabla(a^+/b^+)\}$
3: $\max\{\Delta(a^-/b^-), \Delta(a^-/b^+), \Delta(a^+/b^-), \Delta(a^+/b^+)\}]$
4: **else**
5: **if** $(a^- < 0$ **and** $a^+ < 0$ **or** $a^- > 0$ **and** $a^+ > 0)$ **and** $(b^- < 0$ **and** $b^* < 0$ **or** $b^- > 0$ **and** $b^+ > 0)$ **then**
6: **if** $a^- > 0$ **and** $a^+ > 0$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
7: $c^- := \nabla(a^-/b^+), c^+ := \Delta(a^+/b^-)$
8: $d^- := \Delta(a^-/b^+), d^+ := \nabla(a^+/b^-)$
9: calculate C/D
10: **else**
11: **if** $a^- > 0$ **and** $a^+ > 0$ **and** $b^- < 0$ **and** $b^+ < 0$ **then**
12: $c^- := \nabla(a^+/b^+), c^+ := \Delta(a^-/b^-)$
13: $d^- := \Delta(a^+/b^+), d^+ := \nabla(a^-/b^-)$
14: calculate C/D
15: **else**
16: **if** $a^- < 0$ **and** $a^+ < 0$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
17: $c^- := \nabla(a^-/b^-), \quad c^+ := \Delta(a^+/b^+)$
18: $d^- := \Delta(a^-/b^-), d^+ := \nabla(a^+/b^+)$
19: calculate C/D
20: **else**
21: $c^- := \nabla(a^+/b^-), c^+ := \Delta(a^-/b^+)$
22: $d^- := \Delta(a^+/b^-), d^+ := \nabla(a^-/b^+)$
23: calculate C/D
24: **end if**
25: **end if**
26: **end if**
27: **else**
28: **if** $(a^- \leq 0$ **and** $a^+ \geq 0$ **or** $a^- \geq 0$ **and** $a^+ \leq 0)$ **and** $(b^- < 0$ **and** $b^+ < 0$ **or** $b^- > 0$ **and** $b^+ > 0)$ **then**
29: **if** $a^- \leq a^+$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
30: $c^- = \nabla(a^-/b^-), c^+ := \Delta(a^+/b^-)$
31: $d^- := \Delta(a^-/b^-), d^+ := \nabla(a^+/b^-)$
32: calculate C/D
33: **else**
34: **if** $a^- \leq a^+$ **and** $b^- < 0$ **and** $b^+ < 0$ **then**
35: $c^- := \nabla(a^+/b^+), c^+ := \Delta(a^-/b^+)$
36: $d^- := \Delta(a^+/b^+), d^+ := \nabla(a^-/b^+)$
37: calculate C/D
38: **else**
39: **if** $a^- > a^+$ **and** $b^- > 0$ **and** $b^+ > 0$ **then**
40: $c^- := \nabla(a^-/b^+), \quad c^+ := \Delta(a^+/b^+)$
41: $d^- := \Delta(a^-/b^+), d^+ := \nabla(a^+/b^+)$
42: calculate C/D
43: **else**
44: $c^- := \nabla(a^+/b^-), c^+ := \Delta(a^-/b^-)$
45: $d^- := \Delta(a^+/b^-), d^+ := \nabla(a^-/b^-)$
46: calculate C/D
47: **end if**

```
48:             end if
49:           end if
50:         else
51:           error "division by interval containing zero"
52:         end if
53:       end if
54: end if
```

It is worth noting that the implementation of proper interval arithmetic, based on algorithms 3.1. do 3.5. is consistent with the later IEEE-1788 standard and related publications such as [51] and [84]. It should be noted, however, that in the developed module *Interval.h* it is not possible to divide by the intervals containing zero, they are handled in a different way than presented in [51], because, instead of the resulting interval with one or both ends equal to $+/-\infty$ the execution time exception is returned (see [62]). This is closely related to the fact that the computation takes into account Theorem 1 (Rump) of Section 3.5.

4
Nakao's method

In his works, M. T. Nakao presented numerical methods of verifying the existence of solutions for selected types of partial differential equations. These methods are based on the theory related to the finite element method. Thus, they belong to a different class of methods than those presented within this work. The chapter is limited to the description of those that deal with linear elliptic PDEs.

4.1. Theoretical assumptions

Let us represent second order linear PDEs in the following form:

$$-\sum_{i,j=1}^{n}\frac{\partial}{\partial x_j}\left(a_{ij}(x)\frac{\partial u}{\partial x_i}\right)+\sum_{i=1}^{n}b_i(x)\frac{\partial u}{\partial x_i}+c(x)u=f(x),\quad x\in\Omega, \tag{4.1}$$

where Ω denotes the bounded open set in the space $\mathbb{R}^n$, and the coefficients $a_{ij}(x)$, $b_i(x)$ and $c(x)$ and the function $f(x)$ satisfy the following conditions:

$$\begin{aligned} a_{ij}(x)\in\mathbf{C}^1(\overline{\Omega}),&\quad i,j=1,2\ldots,n,\\ b_i(x)\in\mathbf{C}(\overline{\Omega}),&\quad i=1,2\ldots,n,\\ c(x)\in\mathbf{C}(\overline{\Omega}),&\quad f(x)\in\mathbf{C}(\overline{\Omega}) \end{aligned} \tag{4.2}$$

and

$$\sum_{i,j=1}^{n}a_{ij}\xi_i\xi_j\geq\tilde{c}\sum_{i=1}^{n}\xi_i^2,\quad\forall\xi=(\xi_1,\xi_2,\ldots,\xi_n)\in\mathbb{R}^n,\quad x\in\overline{\Omega},$$

where $\tilde{c}$ denotes a positive constant independent of x and ξ, while $\overline{\Omega}$ denotes the closure of the set Ω, and $\mathbf{C}(\overline{\Omega})$ the space of continuous functions on the set $\overline{\Omega}$, $\mathbf{C}^1(\overline{\Omega})$ the space of continuous functions together with first order derivatives on the same set. As already mentioned in Chapter 2, finding solutions of such equations is not a trivial task; very often it is not possible to find their analytic form.

Therefore, in his papers, Nakao simplifies the problem and restricts himself to considering a certain case of this type of equation with Dirichlet boundary conditions. The

problem under consideration is then written as follows:

$$\begin{gathered} -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_j}\left(a_{ij}(x)\frac{\partial u}{\partial x_i}\right)+\sum_{i=1}^{n} b_i(x)\frac{\partial u}{\partial x_i}+c(x)u=f(x), \quad x\in\Omega, \\ u=0, \quad x\in\partial\Omega. \end{gathered} \tag{4.3}$$

This problem has an explicit classical solution if the functions $a_{ij}(x)$, $b_i(x)$, $c(x)$ and $f(x)$ and the edge $\partial\Omega$ are sufficiently smooth. However, these requirements and the requirements for the differentiability of the function u can be reduced by introducing the notion of a weak solution. For this purpose, we need to cite the necessary notions from functional analysis[1]. Let $\mathbf{L}^2(\Omega)$ denote the space of integrable functions with square in the set Ω, i.e.

$$\mathbf{L}^2(\Omega)=\left\{ f : \int_\Omega |f(x)|^2\,dx<\infty \right\}.$$

The norm and the product in this space are defined by the formulas

$$\|f\|_{\mathbf{L}^2(\Omega)}=\sqrt{\int_\Omega |f(x)|^2\,dx} \text{ and } (f,g)=\int_\Omega f(x)g(x)dx.$$

By $\mathbf{L}^\infty(\Omega)$ we will denote the space of functions bounded almost everywhere with norm[2]

$$\|f\|=\operatorname*{ess\,sup}_{x\in\Omega}|f(x)|.$$

Let us further define the space denoted by $\mathbf{H}_0^1(\Omega)$ as follows:

$$\mathbf{H}_0^1(\Omega)=\left\{ u\in\mathbf{L}^2(\Omega) : \frac{\partial u}{\partial x_i}\in\mathbf{L}^2(\Omega)(i=1,2,\ldots,n), u=0 \text{ na brzegu } \partial\Omega \right\}.$$

In this dissertation we will consider only two-dimensional problems. For the rectangular region $\overline{\Omega}=[0,1]\times[0,1]$ the equation with Dirichlet homogeneous boundary condition has the form

$$\begin{gathered} -\frac{\partial}{\partial x}\left(a_{11}(x,y)\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial x}\left(a_{12}(x,y)\frac{\partial u}{\partial y}\right)-\frac{\partial}{\partial y}\left(a_{21}(x,y)\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(a_{22}(x,y)\frac{\partial u}{\partial y}\right) \\ +b_1(x,y)\frac{\partial u}{\partial x}+b_2(x,y)\frac{\partial u}{\partial y}+c(x,y)u(x,y)=f(x,y),(x,y)\in\Omega, \\ u(x,0)=u(x,1)=u(0,y)=u(1,y)=0, \quad (x,y)\in\partial\Omega, \end{gathered} \tag{4.4}$$

where individual functions belong to the following classes:

$$a_{ij},b_i(x,y),c(x,y)\in\mathbf{C}(\Omega), f(x,y)\in\mathbf{L^2}(\Omega)$$

and $a_{ij}(x,y)\geq\tilde{c}>0$ for $i,j=1,2$ and $(x,y)\in\Omega$. In contrast, in the papers [76], [78] and [79], to which we are going to refer, a boundary problem of the form

$$\begin{gathered} \triangle u+b\nabla u+cu=-f \text{ in the } \Omega \text{ region}, \\ u=0 \text{ at the edge } \partial\Omega, \end{gathered} \tag{4.5}$$

[1] That is, the branch of mathematics concerned with the study of the properties of function spaces.

[2] The notation ess sup denotes the supremum of the function $f(x)$ for $x\in\Omega$ except for a finite number of points $x_i\in\Omega$, where $i=1,2,\ldots N$.

where, in general, Ω denotes a bounded convex set in the space $\mathbb{R}^n (1 \leq n \leq 3))$ with piecewise smooth edges, $u = u(\hat{x})$, and b denotes the vector of functions that are coefficients of the equation, which we write as $b = (b_i)(1 \leq n \leq 3)$ where $b_i = b_i(\hat{x})$, $c = c(\hat{x})$ and $f = f(\hat{x})$ for elements $\hat{x} \in \mathbb{R}^n$.

The following sections present the two basic steps of Nakao's method. Step one, which is the Galerkin approximation from which we obtain an initial approximation of the solution, and step two, the iterative method of estimating the interval containing the exact solution of the problem. the exact problem.

4.2. Galerkin approximation

Let us consider a boundary problem of the form

$$\begin{gathered} -\frac{\partial u}{\partial x^2} - \frac{\partial u}{\partial y^2} + c(x,y)u(x,y) = f(x,y), \quad (x,y) \in \Omega, \\ u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0, \quad x,y \in \{0,1\}, \\ \text{where } \Omega = (0,1) \times (0,1). \end{gathered} \tag{4.6}$$

According to the Galerkin approximation, this problem can be defined as the problem of finding such a real function $u(x,y) \in \mathbf{H}_0^1(\Omega)$, that

$$\begin{gathered} \int_0^1 \int_0^1 \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) dxdy + \int_0^1 \int_0^1 c(x,y)u(x,y)v(x,y)dxdy \\ = \int_0^1 \int_0^1 f(x,y)v(x,y)dxdy \end{gathered} \tag{4.7}$$

for each function $v(x,y) \in \mathbf{H}_0^1(\Omega)$.

We will use this approximation for equation (4.5). Thus, in a weak form, the problem can be formulated as follows: find a real function $u(x) \in \mathbf{H}_0^1(\Omega)$, such that

$$(\nabla u, \nabla \varphi) = (b\nabla u + cu, \varphi) + (f, \varphi), \tag{4.8}$$

where $(\cdot,\cdot)$ denotes the scalar product in $\mathbf{L}^2(\Omega)$ space defined as follows:

$$(u,v) = \int_\Omega u(x,y)v(x,y)dxdy.$$

Note that equations (4.5) and (4.8) are not the same. Unification requires taking the function $-c(x,y)$ from equation (4.5) as the function $c(x,y)$ in the Galerkin approximation. Hence, in fact, using the coefficients from the Nakao equations, we will consider an equation of the form

$$\begin{gathered} \int_0^1 \int_0^1 \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) dxdy - \int_0^1 \int_0^1 c(x,y)u(x,y)v(x,y)dxdy \\ = \int_0^1 \int_0^1 f(x,y)v(x,y)dxdy. \end{gathered} \tag{4.9}$$

To construct a finite element-based approximation for the rectangular region $\overline{\Omega} = [0,1]\times[0,1]$, we divide each interval $[0,1]$ into n subintervals, i.e.into subintervals $[x_i, x_{i+1}]$

and $[y_j, y_{j+1}]$ with points $x_i = ih$ $(i = 0, 1, \ldots, n)$ and $y_j = jh$ $(j = 0, 1, \ldots, n)$. The nodes thus defined form a so-called stretched grid over the area $\overline{\Omega}$. On this mesh we can define the key finite elements for the method described. In Nakao's work, triangular elements were chosen. Such a triangulation for area $\overline{\Omega}$ is shown in Fig. 4.1.

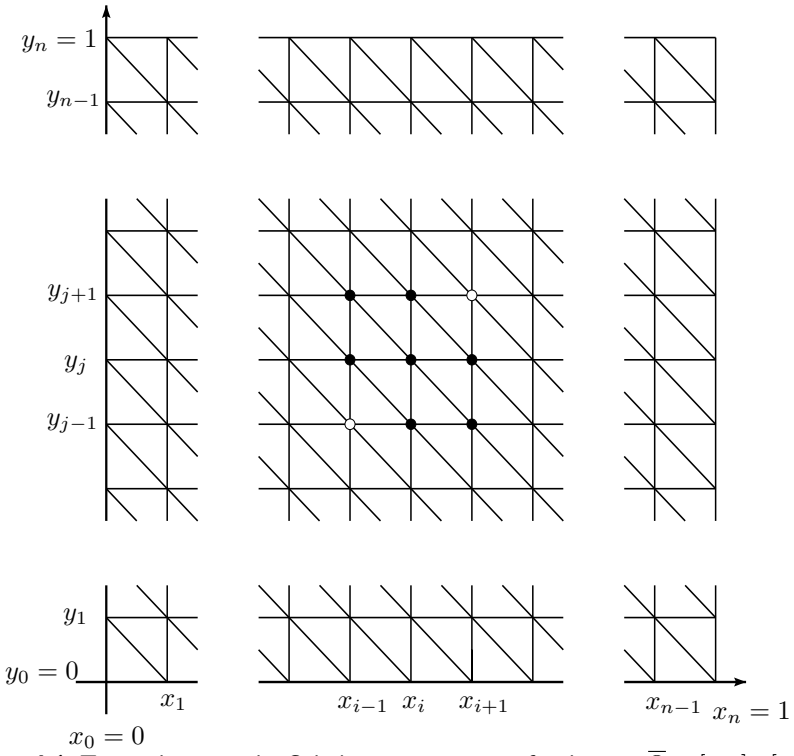

Figure 4.1. Triangulation in the Galerkin approximation for the area $\overline{\Omega} = [0, 1] \times [0, 1]$

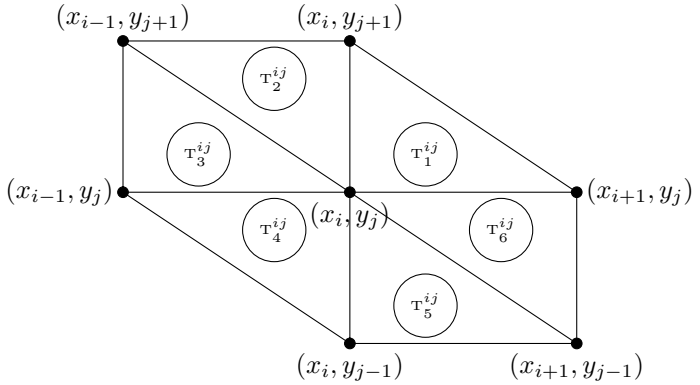

Figure 4.2. Triangulation for node (x_i, y_j), mesh with neighboring nodes

For each node (x_i, y_j) the surrounding areas T_k^{ij} $(k = 1, 2, \ldots, 6)$ are defined as follows:

$$
\begin{aligned}
T_1^{ij} &= \{(x, y) : ih \le x \le (i+1)h, \quad jh \le y \le (i+j+1)h - x\}, \\
T_2^{ij} &= \{(x, y) : (i-1)h \le x \le ih, \quad (i+j)h - x \le y \le (j+1)h\}, \\
T_3^{ij} &= \{(x, y) : (i-1)h \le x \le ih, \quad jh \le y \le (i+j)h - x\}, \\
T_4^{ij} &= \{(x, y) : (i-1)h \le x \le ih, \quad (i+j-1)h - x \le y \le jh\}, \\
T_5^{ij} &= \{(x, y) : ih \le x \le (i+1)h, \quad (j-1)h \le y \le (i+j)h - x\}, \\
T_6^{ij} &= \{(x, y) : ih \le x \le (i+1)h, \quad (i+j)h - x \le y \le jh\}.
\end{aligned}
$$

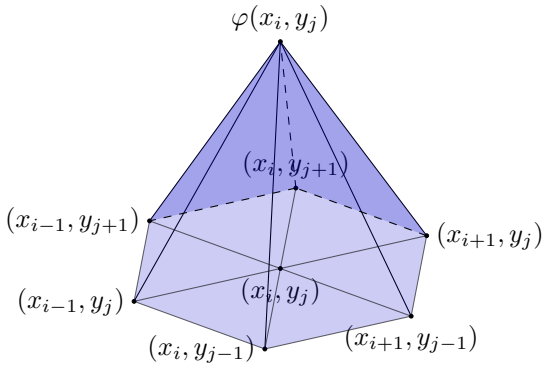

Figure 4.3. Pyramidal basis function φ_{ij} defined for node (x_i, y_j)

We will approximate the weak solution $u \in \mathbf{H}_0^1(\Omega)$ by a linear combination of continuous functions on the set $\overline{\Omega}$,which are linear in each of the resulting triangles. These functions with indices i and j are pyramidal and take value 1 at each node (x_i, y_j) and value 0 at the other nodes. They are defined as follows:

$$\varphi_{ij} = \begin{cases} 1 - \frac{x-x_i}{h} - \frac{y-y_j}{h}, & (x,y) \in T_1^{ij}, \\ 1 - \frac{y-y_j}{h}, & (x,y) \in T_2^{ij}, \\ 1 - \frac{x_i-x}{h}, & (x,y) \in T_3^{ij}, \\ 1 - \frac{x_i-x}{h} - \frac{y_j-y}{h}, & (x,y) \in T_4^{ij}, \\ 1 - \frac{y_j-y}{h}, & (x,y) \in T_5^{ij}, \\ 1 - \frac{x-x_i}{h}, & (x,y) \in T_6^{ij}, \\ 0, & \text{in other cases.} \end{cases} \tag{4.10}$$

The functions φ_{ij} are elements of the space $\mathbf{H}_0^1(\Omega)$. Due to their shape, they can be called pyramidal (Fig. 4.3).

The partial derivatives of the basis functions φ_{ij}, defined in equation (4.10), can be easily determined. They are given by the following formulas:

$$\frac{\partial \varphi_{ij}(x,y)}{\partial x} = \begin{cases} -\frac{1}{h}, & (x,y) \in T_1^{ij}, \\ 0, & (x,y) \in T_2^{ij}, \\ \frac{1}{h}, & (x,y) \in T_3^{ij}, \\ \frac{1}{h}, & (x,y) \in T_4^{ij}, \\ 0, & (x,y) \in T_5^{ij}, \\ -\frac{1}{h}, & (x,y) \in T_6^{ij}, \\ 0, & \text{in other cases,} \end{cases} \tag{4.11}$$

$$\frac{\partial \varphi_{ij}(x,y)}{\partial y} = \begin{cases} -\frac{1}{h}, & (x,y) \in T_1^{ij}, \\ -\frac{1}{h}, & (x,y) \in T_2^{ij}, \\ 0, & (x,y) \in T_3^{ij}, \\ \frac{1}{h}, & (x,y) \in T_4^{ij}, \\ \frac{1}{h}, & (x,y) \in T_5^{ij}, \\ 0, & (x,y) \in T_6^{ij}, \\ 0, & \text{in other cases.} \end{cases} \tag{4.12}$$

Definition 34. *Let $\mathbf{V}_h$ denote any set which is a linear combination of φ_{ij} functions, that is, $\mathbf{V}_h = \operatorname{span} \varphi_{ij}$. Then, the Galerkin approximation of the problem (4.8) is called*

the problem of finding a function $u_h \in \mathbf{V}_h$*, such that*

$$\int_0^1 \int_0^1 u_h'(x,y)v_h'(x,y)dxdy - \int_0^1 \int_0^1 c(x,y)u_h(x,y)v_h(x,y)dxdy = \\ = \int_0^1 \int_0^1 f(x,y)v_h(x,y)dxdy, \tag{4.13}$$

for each function $v_h \in \mathbf{V}_h$.

Note that the function $u_h \in \mathbf{V}_h$ can be written as

$$u_h(x,y) = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} u_{ij}\varphi_{ij}(x,y). \tag{4.14}$$

After substituting equation (4.14) into equation (4.13) we obtain the equivalent problem: find a vector $\vec{u} = (u_{1,1}, \dots u_{1,n-1}, \dots, u_{n-1,1}, \dots u_{n-1,n-1})^T$, such that

$$\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} u_{ij}\left[\int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{kl}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{kl}}{\partial y}\right)dxdy \right. \\ \left. + \int_0^1\int_0^1 c(x,y)\varphi_{ij}(x,y)\varphi_{kl}(x,y)dxdy\right] = \int_0^1\int_0^1 f(x,y)\varphi_{kl}(x,y)dxdy, \\ k,l = 1,2\dots,n-1. \tag{4.15}$$

The indices k and l are the indices of the internal elements of the triangulated area (see Fig. 4.1). We write equation (4.15) for each node (x_k, y_l) and then obtain a system of $(n-1)(n-1)$ equations. The integrals on the left hand side define the coefficient matrix $A = \{a_{kl}\}$, where its individual elements are defined as follows:

$$a_{ij} = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\left[\int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{kl}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{kl}}{\partial y}\right)dxdy \right. \\ \left. + \int_0^1\int_0^1 c(x,y)\varphi_{ij}(x,y)\varphi_{kl}(x,y)dxdy\right]. \tag{4.16}$$

Determining the integrals on the left-hand side of equation (4.15), and consequently finding the values of the coefficients a_{ij}, requires consideration of several cases given below.

- $k = i$ and $l = j$, then the first integral occurring in formula (4.16) is definied as follows:

$$\int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{ij}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{ij}}{\partial y}\right)dxdy = \int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\right)^2 dxdy + \int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial y}\right)^2 dxdy.$$

Determining each of the two component integrals comes down to the following calculations:

$$\int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\right)^2 dxdy = \iint\limits_{(x,y)\in T_1^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial x}\right)^2 dxdy + \iint\limits_{(x,y)\in T_3^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial x}\right)^2 dxdy$$

$$+ \iint\limits_{(x,y)\in T_4^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial x}\right)^2 dxdy + \iint\limits_{(x,y)\in T_6^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial x}\right)^2 dxdy = \iint\limits_{(x,y)\in T_1^{ij}} \left(\frac{1}{h^2}\right)^2 dxdy$$
$$+ \iint\limits_{(x,y)\in T_3^{ij}} \left(\frac{1}{h^2}\right)^2 dxdy + \iint\limits_{(x,y)\in T_4^{ij}} \left(\frac{1}{h^2}\right)^2 dxdy + \iint\limits_{(x,y)\in T_6^{ij}} \left(\frac{1}{h^2}\right)^2 dxdy$$
$$= \frac{1}{h^2}\left(\int_{ih}^{(i+1)h}\int_{jh}^{(i+j+1)h-x} dydx + \int_{(i-1)h}^{ih}\int_{(i+j)h-x}^{(j+1)h} dydx\right.$$
$$\left. + \int_{(i-1)h}^{ih}\int_{(i+j-1)h-x}^{jh} dydx + \int_{ih}^{(i+1)h}\int_{(j-1)h}^{(i+j)h-x} dydx\right) = 2$$

and similarly

$$\int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial y}\right)^2 dxdy = \iint\limits_{(x,y)\in T_1^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial y}\right)^2 dxdy + \iint\limits_{(x,y)\in T_2^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial y}\right)^2 dxdy$$
$$+ \iint\limits_{(x,y)\in T_4^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial y}\right)^2 dxdy + \iint\limits_{(x,y)\in T_5^{ij}} \left(\frac{\partial\varphi_{ij}}{\partial y}\right)^2 dxdy = 2,$$

from where

$$\int_0^1\int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{ij}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{ij}}{\partial y}\right) dxdy = 2 + 2 = 4.$$

The second integral in equation (4.16) has the form

$$I_1^c = \int_0^1 \int_0^1 c(x,y)\varphi_{ij}(x,y)\varphi_{ij}(x,y)dxdy$$

$$= \int_0^1 \int_0^1 c(x,y)\varphi_{ij}^2(x,y)dydx = \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y)\left[1+i+j-\frac{1}{h}(x+y)\right]^2 dydx$$

$$+ \int_{(i-1)h}^{ih} c(x,y) \int_{(j+1)h-x}^{(i+j)h} \left(1+j-\frac{y}{h}\right)^2 dydx + \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h} c(x,y)\left(1-i+\frac{x}{h}\right)^2 dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c(x,y)\left[1-i-j+\frac{1}{h}(x+y)\right]^2 dxdy$$

$$+ \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} c(x,y)\left(1-j+\frac{y}{h}\right)^2 dydx + \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c(x,y)\left(1+i-\frac{x}{h}\right)^2 dydx.$$

Although the most accurate method is finding the analytic form of individual integrals, due to the fact that the function $c(x,y)$, which is a parameter of the equation, may take a complicated form, an effective solution may be finding the value of the integral I_1^c numerically. This remark also applies to the integrals $I_2^c, I_3^c, \ldots, I_7^c$ described further on.

- $i = k$ and $j = l + 1$ (areas 4 and 2' and 5 and 1' overlap - see Fig. 4.4)

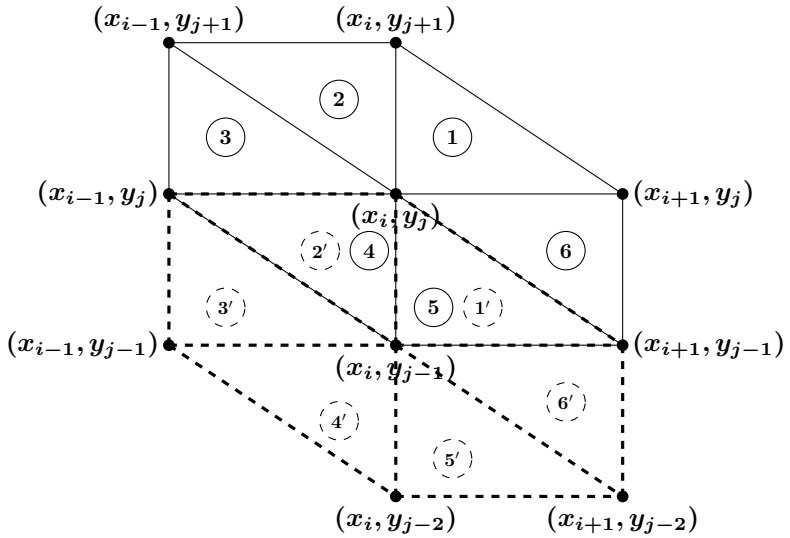

Figure 4.4. Triangles surrounding the nodes (x_i, y_j) and (x_i, y_{j-1})

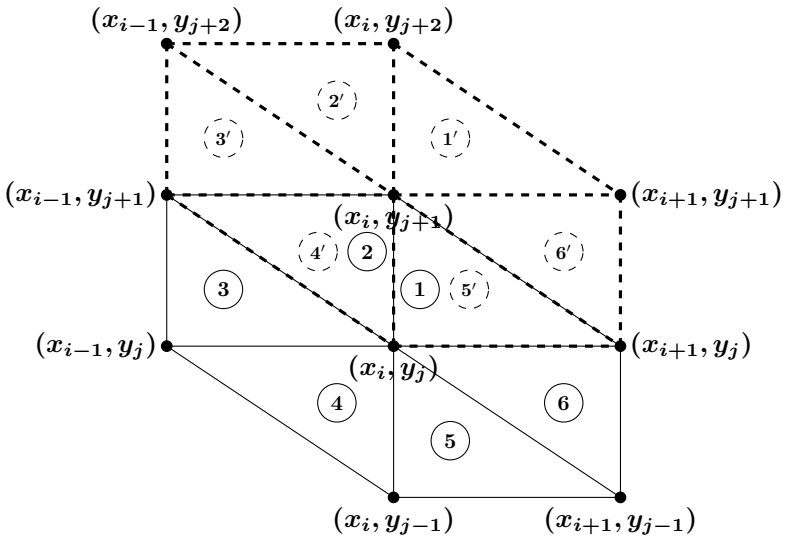

Figure 4.5. Triangles surrounding the nodes (x_i, y_j) and (x_i, y_{j+1})

After substituting $k = i$ and $l = j - 1$ into formula (4.16) we obtain

$$\begin{aligned}
&\int_0^1 \int_0^1 \left(\frac{\partial \varphi_{ij}}{\partial x} \frac{\partial \varphi_{i,j-1}}{\partial x} + \frac{\partial \varphi_{ij}}{\partial y} \frac{\partial \varphi_{i,j-1}}{\partial y} \right) dxdy \\
&= \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} \left(\frac{1}{h} \cdot 0 + \frac{1}{h} \left(-\frac{1}{h} \right) \right) dydx \\
&+ \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} \left(0 \cdot \left(-\frac{1}{h} \right) + \frac{1}{h} \left(-\frac{1}{h} \right) \right) dydx \\
&= -\frac{1}{h^2} \left(\int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} dydx + \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} dydx \right) = -1,
\end{aligned}$$

and

$$\begin{aligned}
I_2^c &= \int_0^1 \int_0^1 c(x,y) \varphi_{ij}(x,y) \, \varphi_{i,j-1}(x,y) \, dxdy \\
&= \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c(x,y) \left[1 - i - j + \frac{1}{h}(x+y) \right] \left(j - \frac{y}{h} \right) dydx \\
&+ \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h} c(x,y) \left(1 - j + \frac{y}{h} \right) \left[i + j - \frac{1}{h}(x+y) \right] dxdy \\
&= I_1^{i,j-1} + I_2^{i,j-1}.
\end{aligned}$$

- $i = k$ and $j = l - 1$, i.e. $k = i$ and $l = j + 1$ (areas 1 and 5' and 2 and 4' overlap – see Fig. 4.5)

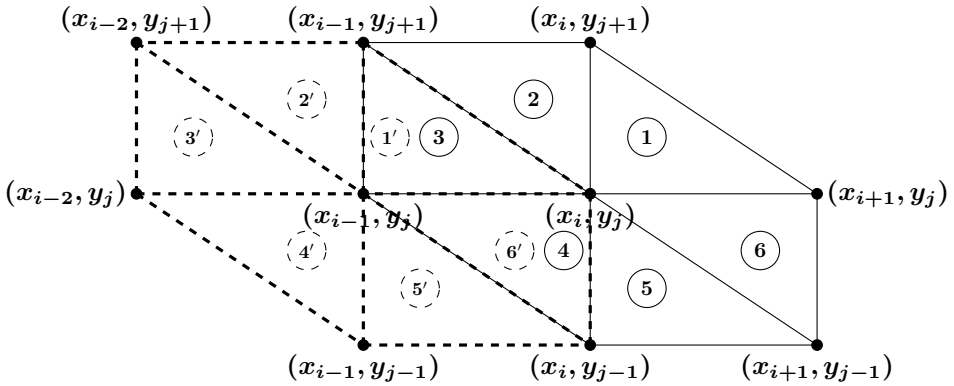

Figure 4.6. Triangles surrounding the nodes (x_i, y_j) and (x_{i-1}, y_j)

$$\int_0^1 \int_0^1 \left(\frac{\partial \varphi_{ij}}{\partial x} \frac{\partial \varphi_{i,j+1}}{\partial x} + \frac{\partial \varphi_{ij}}{\partial y} \frac{\partial \varphi_{i,j+1}}{\partial y} \right) dxdy$$

$$= \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} \left(\left(-\frac{1}{h} \right) \cdot 0 + \left(-\frac{1}{h} \right) \frac{1}{h} \right) dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} \left(0 \cdot \frac{1}{h} x + \left(-\frac{1}{h} \right) \frac{1}{h} \right) dydx = \ldots = -1,$$

$$I_3^c = \int_0^1 \int_0^1 c(x,y) \varphi_{ij}(x,y) \, \varphi_{i,j+1}(x,y) \, dxdy$$

$$= \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y) \left[1 + i + j - \frac{1}{h}(x+y) \right] \left(-j + \frac{y}{h} \right) dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} c(x,y) \left(1 + j - \frac{y}{h} \right) \left[-i - j + \frac{1}{h}(x+y) \right] dydx.$$

- $i = k+1$ and $j = l$, i.e. $k = i-1$ and $l = j$ (areas 3 and 1' and 4 and 6' overlap – see Fig. 4.6)

$$\int_0^1 \int_0^1 \left(\frac{\partial \varphi_{ij}}{\partial x} \frac{\partial \varphi_{i-1,j}}{\partial x} + \frac{\partial \varphi_{ij}}{\partial y} \frac{\partial \varphi_{i-1,j}}{\partial y} \right) dxdy =$$

$$= \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h} \left(\frac{1}{h} \left(-\frac{1}{h} \right) + 0 \cdot \left(-\frac{1}{h} \right) \right) dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} \left(\frac{1}{h} \left(-\frac{1}{h} \right) + \frac{1}{h} \cdot 0 \right) dydx = 1,$$

$$I_4^c = \int_0^1 \int_0^1 c(x,y)\varphi_{ij}(x,y)\,\varphi_{i-1,j}(x,y)\,dydx$$

$$= \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h} c(x,y)\left(1-i+\frac{x}{h}\right)\left[i+j-\frac{1}{h}(x+y)\right]dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c(x,y)\left[1-i-j+\frac{1}{h}(x+y)\right]\left(i-\frac{x}{h}\right)dydx.$$

- $i = k-1$ and $j = l$, i.e. $k = i+1$ and $l = j$ (areas 1 and 3' and 6 and 4' overlap – see Fig. 4.7)

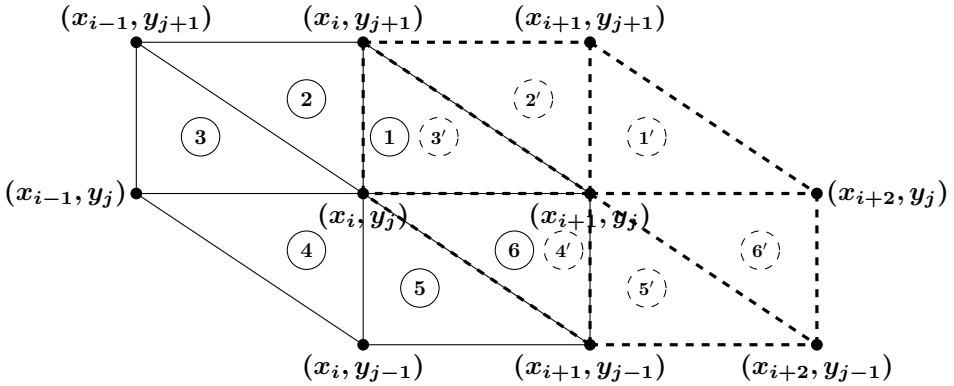

Figure 4.7. Triangles surrounding the nodes (x_i, y_j) and (x_{i+1}, y_j)

$$\int_0^1 \int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{i+1,j}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{i+1,j}}{\partial y}\right)dydx$$

$$= \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} \left(\left(-\frac{1}{h}\right)\frac{1}{h} + \left(-\frac{1}{h}\right)\cdot 0\right)dydx$$

$$+ \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} \left(\left(-\frac{1}{h}\right)\frac{1}{h} + 0\cdot\frac{1}{h}\right)dydx = -1,$$

$$I_5^c = \int_0^1 \int_0^1 c(x,y)\varphi_{ij}(x,y)\,\varphi_{i+1,j}(x,y)\,dxdy$$

$$= \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y)\left[1+i+j-\frac{1}{h}(x+y)\right]\left(-i+\frac{x}{h}\right)dydx$$

$$+ \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y)\left(1+i-\frac{x}{h}\right)\left[-i-j+\frac{1}{h}(x+y)\right]dydx.$$

- $i = k+1$ and $j = l-1$, i.e. $k = i-1$ and $l = j+1$ (areas 2 and 6' and 3 and 5' overlap – see Fig. 4.8)

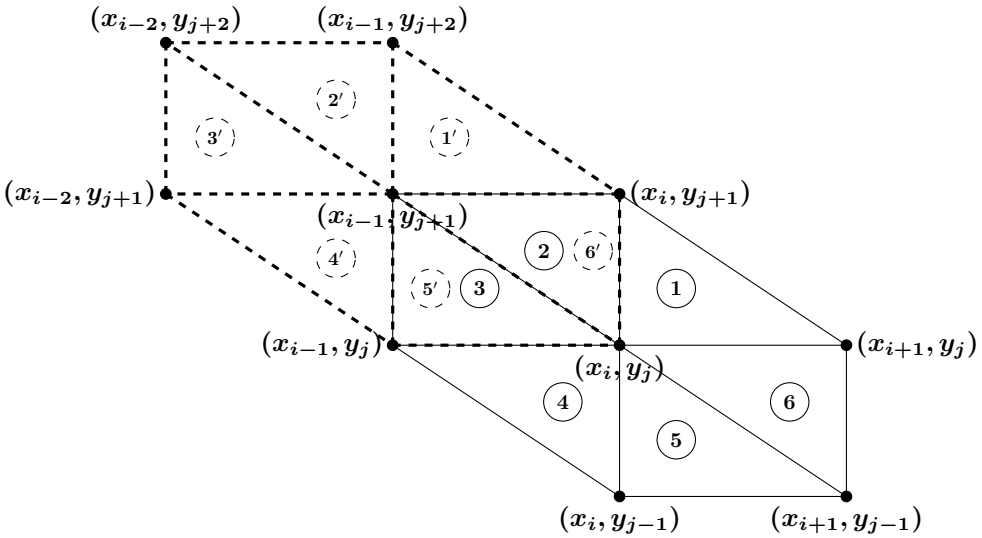

Figure 4.8. Triangles surrounding the nodes (x_i, y_j) and (x_{i-1}, y_{j+1})

$$\int_0^1 \int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{i-1,j+1}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{i-1,j+1}}{\partial y}\right)dxdy$$

$$= \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} \left(0\cdot\left(-\frac{1}{h}\right) + \left(-\frac{1}{h}\right)\cdot 0\right)dydx +$$

$$+ \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} \left(\frac{1}{h}\cdot 0 + 0\cdot\frac{1}{h}\right)dydx = 0,$$

$$I_6^c = \int_0^1 \int_0^1 c(x,y)\varphi_{ij}\,(x,y)\,\varphi_{i-1,j+1}\,(x,y)\,dydx$$
$$= \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} c(x,y)\left(1+j-\frac{y}{h}\right)\left(i-\frac{x}{h}\right)dydx$$
$$+ \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} c(x,y)\left(1-i+\frac{x}{h}\right)\left(-j+\frac{y}{h}\right)dydx.$$

- $i = k-1$ and $j = l+1$, i.e. $k = i+1$ and $l = j-1$ (areas 5 and 3' and 6 and 2' overlaps – see Fig. 4.9)

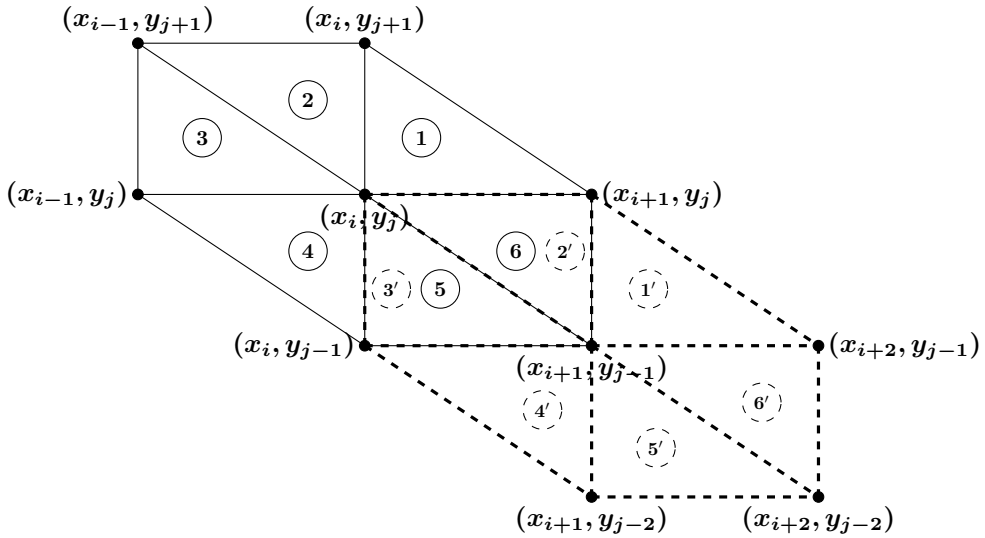

Figure 4.9. Triangles surrounding the nodes (x_i, y_j) and (x_{i+1}, y_{j-1})

$$\int_0^1 \int_0^1 \left(\frac{\partial\varphi_{ij}}{\partial x}\frac{\partial\varphi_{i+1,j-1}}{\partial x} + \frac{\partial\varphi_{ij}}{\partial y}\frac{\partial\varphi_{i+1,j-1}}{\partial y}\right)dxdy$$
$$= \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} \left(0\cdot\frac{1}{h} + \frac{1}{h}\cdot 0\right)dydx$$
$$+ \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} \left(\left(-\frac{1}{h}\right)\cdot 0 + 0\cdot\left(-\frac{1}{h}\right)\right)dydx = 0,$$

$$I_7^c = \int_0^1 \int_0^1 c(x,y)\varphi_{ij}(x,y)\,\varphi_{i+1,j-1}(x,y)\,dydx$$
$$= \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} c(x,y)\left(1-j+\frac{y}{h}\right)\left(-i+\frac{x}{h}\right)dydx$$
$$+ \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c(x,y)\left(1+i-\frac{x}{h}\right)\left(j-\frac{y}{h}\right)dydx.$$

It follows from the above equations that, in order to determine the exact values of the coefficients a_{ij}, given by equation (4.16), the following integrals must be calculated (in different limits) for each task:

$$\int\int c(x,y)dydx, \quad \int x\int c(x,y)dydx, \quad \int\int yc(x,y)dydx,$$
$$\int x\int yc(x,y)dydx, \quad \int x^2\int c(x,y)dydx, \quad \int\int y^2c(x,y)dydx.$$

For other values of k and l all considered integrals are 0.

For the integral on the right-hand side of the system of equations (4.15) we have $(k=i,\ \ l=j)$

$$\int_0^1 \int_0^1 f(x,y)\,\varphi_{ij}(x,y)\,dxdy$$
$$= \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} f(x,y)\left(1+i+j-\frac{1}{h}(x+y)\right)dydx$$
$$+ \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} f(x,y)\left(1+j-\frac{y}{h}\right)dydx$$
$$+ \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} f(x,y)\left(1-i+\frac{x}{h}\right)dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{(jh} f(x,y)\left(1-i-j+\frac{1}{h}(x+y)\right)dydx$$
$$+ \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} f(x,y)\left(1-j+\frac{y}{h}\right)dydx$$
$$+ \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} f(x,y)\left(1+i-\frac{x}{h}\right)dydx$$
$$= I_1^f + I_2^f + I_3^f + I_4^f + I_5^f + I_6^f.$$

The integrals of I_k^f for $k = 1, 2, \ldots 6$ are shown below.

$$I_1^f = \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} f(x,y) \left(1 + i + j - \frac{1}{h}(x+y)\right) dydx$$

$$= (1+i+j) \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} f(x,y)\, dydx$$

$$-\frac{1}{h}\left(\int_{ih}^{(i+1)h} x \int_{jh}^{(i+j+1)h-x} f(x,y)\, dydx + \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} yf(x,y)\, dydx \right),$$

$$I_2^f = \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} f(x,y) \left(1 + j - \frac{y}{h}\right) dydx$$

$$= (1+j) \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} f(x,y)\, dydx - \frac{1}{h} \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} yf(x,y)\, dydx,$$

$$I_3^f = \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} f(x,y) \left(1 - i + \frac{x}{h}\right) dydx$$

$$= (1-i) \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} f(x,y)\, dydx + \frac{1}{h} \int_{(i-1)h}^{ih} x \int_{jh}^{(i+j)h-x} f(x,y)\, dydx,$$

$$I_4^f = \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} f(x,y) \left(1 - i - j + \frac{1}{h}(x+y)\right) dydx$$

$$= (1-i-j) \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} f(x,y)\, dydx$$

$$+\frac{1}{h}\left(\int_{(i-1)h}^{ih} x \int_{(i+j-1)h-x}^{jh} f(x,y)\, dydx + \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} yf(x,y)\, dydx \right),$$

$$I_5^f = \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} f(x,y) \left(1 - j + \frac{y}{h}\right) dydx$$

$$= (1-j) \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} f(x,y)\, dydx + \frac{1}{h} \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} yf(x,y)\, dydx,$$

$$I_6^f = \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} f(x,y)\left(1+i-\frac{x}{h}\right) dydx$$
$$= (1+i)\int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} f(x,y)\, dydx - \frac{1}{h}\int_{ih}^{(i+1)h} x \int_{(i+j)h-x}^{jh} f(x,y)\, dydx.$$

It follows from the above formulas that for each task, the following three integrals must also be calculated (in different limits):

$$\int\int f(x,y)\, dydx, \quad \int x \int f(x,y)\, dydx, \quad \int\int y f(x,y)\, dydx.$$

In general, the above integrals are calculated using quadratures. This introduces an additional error (quadrature error). Therefore, it is recommended, if possible, to determine these integrals analytically.

In summary, the system of equations (4.15) has the form

$$(4-I_1^c)\,u_{11} - (1+I_3^c)\,u_{12} - (1+I_5^c)\,u_{21} = f_{11},$$
$$-(1+I_2^c)\,u_{1,j-1} + (4-I_1^c)\,u_{1j} - (1+I_3^c)\,u_{1,j+1} - I_7^c u_{2,j-1}$$
$$-(1+I_5^c)\,u_{2j} = f_{1j}, \quad j = 2,3,\ldots,n-2,$$

$$-(1+I_2^c)\,u_{1,n-2} + (4-I_1^c)\,u_{1,n-1} - I_7^c u_{2,n-2} - (1+I_5^c)\,u_{2,n-1} = f_{1,n-1},$$
$$-(1+I_4^c)\,u_{i-1,1} - I_6^c u_{i-1,2} + (4-I_1^c)\,u_{i1} - (1+I_3^c)\,u_{i2}$$
$$-(1+I_5^c)\,u_{i+1,1} = f_{i1}, \quad i = 2,3,\ldots,n-2,$$

$$-(1+I_4^c)\,u_{i-1,j} - I_6^c u_{i-1,j+1} - (1+I_2^c)\,u_{i,j-1} + (4-I_1^c)\,u_{ij}$$
$$-(1+I_3^c)\,u_{i,j+1} - I_7^c u_{i+1,j-1} - (1+I_5^c)\,u_{i+1,j} = f_{ij},$$
$$i = 2,3,\ldots,n-2, \quad j = 2,3,\ldots,n-2.$$
$$-(1+I_4^c)\,u_{i-1,n-1} - (1+I_2^c)\,u_{i,n-2} + (4-I_1^c)\,u_{i,n-1}$$
$$-I_7^c u_{i+1,n-2} - (1+I_5^c)\,u_{i+1,n-1} = f_{i,n-1}, \quad i = 2,3,\ldots,n-2,$$

$$-(1+I_4^c)\,u_{n-2,1} - I_6^c u_{n-2,2} + (4-I_1^c)\,u_{n-1,1}$$
$$-(1+I_3^c)\,u_{n-1,2} = f_{n-1,1},$$

$$-(1+I_4^c)\,u_{n-2,j} - I_6^c u_{n-2,j+1} - (1+I_2^c)\,u_{n-1,j-1}$$
$$+(4+I_1^c)\,u_{n-1,j} - (1+I_3^c)\,u_{n-1,j+1} = f_{n-1,j},$$
$$j = 2,3,\ldots,n-2,$$

$$-(1+I_4^c)\,u_{n-2,n-1} - (1+I_2^c)\,u_{n-1,n-2} + (4-I_1^c)\,u_{n-1,n-1} = f_{n-1,n-1},$$

where $f_{ij} = \int_0^1 \int_0^1 f(x,y)\,\varphi_{ij}(x,y)\, dxdy$.

After dividing both sides of the above equations by h^2 we get the following a system of equations:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{B}} & \mathbf{A} & \mathbf{B} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{B}} & \mathbf{A} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \hat{\mathbf{B}} & \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \hat{\mathbf{B}} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \\ \mathbf{u}_{n-3} \\ \mathbf{u}_{n-2} \\ \mathbf{u}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \vdots \\ \mathbf{d}_{n-3} \\ \mathbf{d}_{n-2} \\ \mathbf{d}_{n-1} \end{bmatrix}, \tag{4.17}$$

where

$$\mathbf{A} = \begin{bmatrix} a & a_+ & 0 & \cdots & 0 & 0 & 0 \\ a_- & a & a_+ & \cdots & 0 & 0 & 0 \\ 0 & a_- & a & \vdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & a & a_+ & 0 \\ 0 & 0 & 0 & \cdots & a_- & a & a_+ \\ 0 & 0 & 0 & \cdots & 0 & a_- & a \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} b & 0 & 0 & \cdots & 0 & 0 & 0 \\ b_- & b & 0 & \cdots & 0 & 0 & 0 \\ 0 & b_- & b & \vdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & b & 0 & 0 \\ 0 & 0 & 0 & \cdots & b_- & b & 0 \\ 0 & 0 & 0 & \cdots & 0 & b_- & b \end{bmatrix},$$

$$\hat{\mathbf{B}} = \begin{bmatrix} \hat{b} & \hat{b}_+ & 0 & \cdots & 0 & 0 & 0 \\ 0 & \hat{b} & \hat{b}_+ & \cdots & 0 & 0 & 0 \\ 0 & 0 & \hat{b} & \vdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & \hat{b} & \hat{b}_+ & 0 \\ 0 & 0 & 0 & \cdots & 0 & \hat{b} & \hat{b}_+ \\ 0 & 0 & 0 & \cdots & 0 & 0 & \hat{b} \end{bmatrix},$$

$$\mathbf{u}_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{i,n-1} \end{bmatrix}, \quad \mathbf{d}_i = \begin{bmatrix} d_{i1} \\ d_{i2} \\ \vdots \\ d_{i,n-1} \end{bmatrix}, \quad i = 1, 2, \ldots, n-1$$

whereby

$$a = \frac{1}{h^2}\left(4 - I_1^c\right), \quad a_- = -\frac{1}{h^2}\left(1 + I_2^c\right), \quad a_+ = -\frac{1}{h^2}\left(1 + I_3^c\right),$$
$$b = -\frac{1}{h^2}\left(1 + I_5^c\right), \quad b_- = -\frac{1}{h^2} I_7^c, \quad \hat{b} = -\frac{1}{h^2}\left(1 + I_4^c\right), \quad \hat{b}_+ = -\frac{1}{h^2} I_6^c,$$
$$d_{ij} = \frac{1}{h^2} f_{ij}, \quad i, j = 1, 2, \ldots, n-1.$$

The system can be solved using an exact method, e.g. the Gaussian elimination method (without choosing a fundamental element, since such an element is located in every place of the main diagonal) or the Cholesky method (since the system matrix is positively determined). Since the array matrix is sparse, it will be advisable to use algorithms which reduce its occupancy in computer memory (regularly spaced elements with value 0 should be omitted in this notation).

4.3. Iterative solution verification procedure

Nakao's method is an iterative method for determining the intervals containing the exact solution. In [76] he proved that for a problem which is a weak form of equation (4.6), i.e.

$$(\nabla u, \nabla \varphi) = (b\nabla u + cu, \varphi) + (f, \varphi), \quad \varphi \in H_0^1(\Omega),$$

if there exists an unambiguous solution, which is the function $u(x)$, then it lies in the set $u_h + [\alpha]$, where

$$[\alpha] = \{\varphi \in H_0^1(\Omega) : \|\varphi\|_{H_0^1(\Omega)} \leq \alpha, \|\varphi\|_{L^2(\Omega)} \leq Ch\alpha\},$$

where $\alpha > 0$ and C denotes some constant independent of h. The quantities u_h and α are determined iteratively. In the case where Ω denotes a two-dimensional region, Nakao's method is based on the following formulas (due to the examples considered in Sec. 7, we assume $b = 0$):

$$\left(\nabla u_h^{(k)}, \nabla \varphi_{ij}\right) = \left(cu_h^{(k-1)} + f, \varphi_{ij}\right) + [-1, 1]\, Ch\alpha^{(k-1)} \|\varphi_{ij}\|_{L^2(\Omega)},$$

$$\alpha^{(k)} = Ch\left(\left\|cu_h^{(k-1)} + f\right\|_{L^2(\Omega)} + Ch\|c\|_{L^\infty(\Omega)}\alpha^{(k-1)}\right).$$

Then, the solution $u_h^{(0)}$ obtained from the Galerkin approximation is taken as the initial approximation, i.e.

$$\left(\nabla u_h^{(0)}, \nabla \varphi_{ij}\right) = \left(cu_h^{(0)} + f, \varphi_{ij}\right), \quad i, j = 1, 2, \ldots, n-1,$$

and vector $\alpha^{(0)} = \left(\alpha_{11}^{(0)}, \alpha_{12}^{(0)}, \ldots, \alpha_{n-1,n-1}^{(0)}\right)^T = (0, 0, \ldots, 0)^T$. In practice, the usual assumption $C = 1$ (see [79, p. 327]), so in the two-dimensional case for the area $\Omega = (0,1) \times (0,1)$ method can be written in the form

$$\begin{aligned} \left(\nabla u_h^{(k)}, \nabla \varphi_{ij}\right) &= \left(c u_h^{(k-1)} + f, \varphi_{ij}\right) + [-1,1]\, h \alpha^{(k-1)} \left\|\varphi_{ij}\right\|_{L^2(0,1)\times(0,1)}, \\ \alpha^{(k)} &= h \left(\left\| c u_h^{(k-1)} + f \right\|_{L^2(0,1)\times(0,1)} + h \left\| c \right\|_{L^\infty(0,1)\times(0,1)} \alpha^{(k-1)} \right). \end{aligned} \tag{4.18}$$

In the method (4.18) the quantities $u_h^{(k)} = \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} u_{ij}^{(k)} \varphi_{ij}(x,y)$ are intervals because the coefficient $u_{ij}^{(k)}$ are the intervals $\left(u_{ij}^{(k)} = [\underline{A}_{ij}^{(k)}, \overline{A}_{ij}^{(k)}]\right)$, and $\alpha^{(k)}$ is a vector with components that are real numbers.

Let us determine the norm appearing in the first formula (4.18). We have

$$\left\|\varphi_{ij}\right\|_{L^2(0,1)\times(0,1)} = \sqrt{\int_0^1 \int_0^1 \varphi_{ij}^2(x,y) dx dy} = \sqrt{I_1^{ij} + I_2^{ij} + I_3^{ij} + I_4^{ij} + I_5^{ij} + I_6^{ij}} = \sqrt{\frac{h^2}{2}},$$

where the integrals $I_p^{ij}, p = 1, 2, \ldots, 6$, occurring under the root were calculated in the article [63, str. 13–26]. Finally, we have

$$\left\|\varphi_{ij}\right\|_{L^2(0,1)\times(0,1)} = \frac{\sqrt{2}}{2} h.$$

Equation (4.18) therefore represents a system of equations of the form (cf. system of equations (4.17))

$$\begin{gathered} 4u_{11}^{(k)} - u_{12}^{(k)} - u_{21}^{(k)} = I_1^c u_{11}^{(k-1)} + I_3^c u_{12}^{(k-1)} + I_5^c u_{21}^{(k-1)} + f_{11} + [-1,1] h^2 \frac{\sqrt{2}}{2} \alpha_{11}^{(k-1)}, \\ -u_{1,j-1}^{(k)} + 4u_{1j}^{(k)} - u_{1,j+1}^{(k)} - u_{2j}^{(k)} = I_2^c u_{1,j-1}^{(k-1)} + I_1^c u_{1j}^{(k-1)} + I_3^c u_{1,j+1}^{(k-1)} \\ + I_7^c u_{2,j-1}^{(k-1)} + I_5^c u_{2j}^{(k-1)} + f_{1j} + [-1,1] h^2 \frac{\sqrt{2}}{2} \alpha_{1j}^{(k-1)}, \\ j = 2, 3, \ldots, n-2, \end{gathered}$$

$$\begin{gathered} -u_{1,n-2}^{(k)} + 4u_{1,n-1}^{(k)} - u_{2,n-1}^{(k)} = I_2^c u_{1,n-2}^{(k)} + I_1^c u_{1,n-1}^{(k-1)} + I_7^c u_{2,n-2}^{(k-1)} + I_5^c u_{2,n-1}^{(k-1)} \\ + f_{1,n-1} + [-1,1] h^2 \frac{\sqrt{2}}{2} \alpha_{1,n-1}^{(k-1)}, \\ -u_{i-1,1}^{(k)} + 4u_{i1}^{(k)} - u_{i2}^{(k)} - u_{i+1,2}^{(k)} = I_4^c u_{i-1,1}^{(k-1)} + I_6^c u_{i-1,2}^{(k-1)} + I_1^c u_{i1}^{(k-1)} + I_3^c u_{i2}^{(k-1)} \\ + I_5^c u_{i+1,1} + f_{i1} + [-1,1] h^2 \frac{\sqrt{2}}{2} \alpha_{i1}^{(k-1)}, \\ i = 2, 3, \ldots, n-2, \end{gathered}$$

$$\begin{gathered} -u_{i-1,j}^{(k)} - u_{i,j+1}^{(k)} + 4u_{ij}^{(k)} - u_{i,j+1}^{(k)} - u_{i+1,j}^{(k)} = I_4^c u_{i-1,j}^{(k-1)} + I_6^c u_{i-1,j+1}^{(k-1)} + I_2^c u_{i,j-1}^{(k-1)} \\ + I_1^c u_{ij}^{(k-1)} + I_3^c u_{i,j+1}^{(k-1)} + I_7^c u_{i+1,j-1}^{(k-1)} + I_5^c u_{i+1,j}^{(k-1)} + f_{ij} + [-1,1] h^2 \alpha_{ij}^{(k-1)}, \\ i = 2, 3, \ldots, n-2, \quad j = 2, 3, \ldots, n-2, \end{gathered}$$

$$-u^{(k)}_{i-1,n-1} - u^{(k)}_{i,n-2} + 4u^{(k)}_{i,n-1} - u^{(k)}_{i+1,n-1} = I^c_4 u^{(k-1)}_{i-1,n-1} + I^c_2 u^{(k-1)}_{i,n-2} + I^c_1 u^{(k-1)}_{i,n-1}$$
$$+I^c_7 u^{(k-1)}_{i+1,n-2} + I^c_5 u^{(k-1)}_{i+1,n-1} + f_{i,n-1} + [-1,1]h^2\alpha^{(k-1)}_{i,n-1},$$
$$i = 2, 3, \ldots, n-2,$$

$$-u^{(k)}_{n-2,1} + 4u^{(k)}_{n-1,1} - u^{(k)}_{n-1,2} = I^c_4 u^{(k-1)}_{n-2,1} + I^c_6 u^{(k-1)}_{n-2,2} + I^c_1 u^{(k-1)}_{n-1,1} + I^c_3 u^{(k-1)}_{n-1,2}$$
$$+f_{n-1,1} + [-1,1]h^2\frac{\sqrt{2}}{2}\alpha^{(k-1)}_{n-1,1},$$

$$-u^{(k)}_{n-2,j} - u^{(k)}_{n-1,j-1} + 4u^{(k)}_{n-1,j} - u^{(k)}_{n-1,j+1} = I^c_4 u^{(k-1)}_{n-2,j} + I^c_6 u^{(k-1)}_{n-2,j+1}$$
$$+I^c_2 u^{(k-1)}_{n-1,j-1} + I^c_1 u^{(k-1)}_{n-1,j} + I^c_3 u^{(k-1)}_{n-1,j+1} + f_{n-1,j} + [-1,1]h^2\frac{\sqrt{2}}{2}\alpha^{(k-1)}_{n-1,j},$$
$$j = 2, 3, \ldots, n-2,$$

$$-u^{(k)}_{n-2,n-1} - u^{(k)}_{n-1,n-2} + 4u^{(k)}_{n-1,n-1} = I^c_4 u^{(k-1)}_{n-2,n-1} + I^c_2 u^{(k-1)}_{n-1,n-2} + I^c_1 u^{(k-1)}_{n-1,n-1}$$
$$+f_{n-1,n-1} + [-1,1]h^2\frac{\sqrt{2}}{2}\alpha^{(k-1)}_{n-1,n-1}.$$

After dividing both sides of these equations by h^2 we get the following notation of the system of equations in matrix form:

$$\begin{bmatrix} \mathbf{A}' & \mathbf{B}' & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}' & \mathbf{A}' & \mathbf{B}' & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}' & \mathbf{A}' & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{A}' & \mathbf{B}' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{B}' & \mathbf{A}' & \mathbf{B}' \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{B}' & \mathbf{A}' \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(k)}_1 \\ \mathbf{u}^{(k)}_2 \\ \mathbf{u}^{(k)}_3 \\ \vdots \\ \mathbf{u}^{(k)}_{n-3} \\ \mathbf{u}^{(k)}_{n-2} \\ \mathbf{u}^{(k)}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{d}^{(k)}_1 \\ \mathbf{d}^{(k)}_2 \\ \mathbf{d}^{(k)}_3 \\ \vdots \\ \mathbf{d}^{(k)}_{n-3} \\ \mathbf{d}^{(k)}_{n-2} \\ \mathbf{d}^{(k)}_{n-1} \end{bmatrix}, \tag{4.19}$$

where

$$\mathbf{A}' = \begin{bmatrix} a' & b' & 0 & \cdots & 0 & 0 & 0 \\ b' & a' & b' & \cdots & 0 & 0 & 0 \\ 0 & b' & a' & \vdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & a' & b' & 0 \\ 0 & 0 & 0 & \cdots & b' & a' & b' \\ 0 & 0 & 0 & \cdots & 0 & b' & a' \end{bmatrix}, \quad \mathbf{B}' = \begin{bmatrix} b' & 0 & \cdots & 0 & 0 \\ 0 & b' & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b' & 0 \\ 0 & 0 & \cdots & 0 & b' \end{bmatrix},$$

$$\mathbf{u}_i^{(k)} = \begin{bmatrix} u_{i1}^{(k)} \\ u_{i2}^{(k)} \\ \vdots \\ u_{i,n-1}^{(k)} \end{bmatrix}, \quad \mathbf{d}_i^{(k)} = \begin{bmatrix} d_{i1}^{(k)} \\ d_{i2}^{(k)} \\ \vdots \\ d_{i,n-1}^{(k)} \end{bmatrix}, \quad i = 1, 2, \ldots, n-1,$$

whereby

$$\begin{gathered} a' = \frac{4}{h^2}, \;\; b' = -\frac{1}{h^2}, \\ d_{ij}^{(k)} = I_4^c u_{i-1,j}^{(k-1)} + I_6^c u_{i-1,j+1}^{(k-1)} + I_2^c u_{i,j-1}^{(k-1)} + I_1^c u_{ij}^{(k-1)} + I_3^c u_{i,j+1}^{(k-1)} \\ + I_7^c u_{i+1,j-1}^{(k-1)} + I_5^c u_{i+1,j}^{(k-1)} + \frac{1}{h^2} f_{ij} + [-1, 1] \frac{\sqrt{2}}{2} \alpha_{ij}^{(k-1)}, \\ i, j = 1, 2, \ldots, n-1 \end{gathered} \tag{4.20}$$

and $u_{0j}^{(k-1)} = u_{nj}^{(k-1)} = u_{i0}^{(k-1)} = u_{in}^{(k-1)} = 0$. The system of equations (4.19) is solved by one of the known exact methods taking into account the fact that the matrix of this system is a sparse matrix.

There are two norms in the formula for the quantity $\alpha^{(k)}$ The first one has the form

$$\left\| cu_h^{(k-1)} + f \right\|_{L^2(0,1)\times(0,1)} = \sqrt{\int_0^1 \int_0^1 \left(cu_h^{(k-1)} + f\,(x, y) \right)^2 dxdy}.$$

This norm should be calculated for each value of i and j, since

$$\alpha^{(k)} = \begin{bmatrix} \alpha_{11}^{(k)} \\ \vdots \\ \alpha_{1,n-1}^{(k)} \\ \vdots \\ \alpha_{n-1,1}^{(k)} \\ \vdots \\ \alpha_{n-1,n-1}^{(k)} \end{bmatrix}.$$

Denoting for each value of i and j the expression under the integral by β_{ij} and taking into account that

$$u_h^{(k-1)} = \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} u_{ij}^{(k-1)} \varphi_{ij}(x, y)$$

we have

$$\begin{aligned}\beta_{ij} &= \int_0^1\int_0^1 \left[c^2(x,y)\left(u_h^{(k-1)}\right)^2 + 2c(x,y)u_h^{(k-1)}f(x,y) + f^2(x,y)\right]dxdy \\ &= \int_0^1\int_0^1 c^2(x,y)\left(u_h^{(k-1)}\right)^2 dxdy + 2\int_0^1\int_0^1 c(x,y)u_h^{(k-1)}f(x,y)dxdy \\ &\qquad + \int_0^1\int_0^1 f^2(x,y)dxdy \\ &= \beta_{ij}^{(1)} + 2\beta_{ij}^{(2)} + \int_0^1\int_0^1 f^2(x,y)dxdy,\end{aligned} \tag{4.21}$$

where

$$\begin{aligned}\beta_{ij}^{(1)} = \left(u_{ij}^{(k-1)}\right)^2 \Bigg[&\int_{ih}^{(i+1)h}\int_{jh}^{(i+j+1)h-x} c^2(x,y)\left(1+i+j-\frac{1}{h}(x+y)\right)^2 dydx \\ &+ \int_{(i-1)h}^{ih}\int_{(i+j)h-x}^{(j+1)h} c^2(x,y)\left(1+j-\frac{y}{h}\right)^2 dydx \\ &+ \int_{(i-1)h}^{ih}\int_{jh}^{(i+j)h-x} c^2(x,y)\left(1-i+\frac{x}{h}\right)^2 dydx \\ &+ \int_{(i-1)h}^{ih}\int_{(i+j+1)h-x}^{jh} c^2(x,y)\left(1-i-j-\frac{1}{h}(x+y)\right)^2 dydx \\ &+ \int_{ih}^{(i+1)h}\int_{(j-1)h-x}^{(i+j)h-x} c^2(x,y)\left(1-j-\frac{y}{h}\right)^2 dydx \\ &+ \int_{ih}^{(i+1)h}\int_{(i+j)h-x}^{jh} c^2(x,y)\left(1+i-\frac{x}{h}\right)^2 dydx\Bigg] \\ = \left(u_{ij}^{(k-1)}\right)^2 &\left(I_{\beta 1}^c + I_{\beta 2}^c + I_{\beta 3}^c + I_{\beta 4}^c + I_{\beta 5}^c + I_{\beta 6}^c\right),\end{aligned}$$

whereby

$$I^c_{\beta 1} = \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c^2(x,y)\left(1+i+j-\frac{1}{h}(x+y)\right)^2 dydx$$
$$= (1+i+j)^2 \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c^2(x,y)dydx$$
$$-\frac{2}{h}(1+i+j)\left(\int_{ih}^{(i+1)h} x \int_{jh}^{(i+j+1)h-x} c^2(x,y)dydx\right.$$
$$\left.+\int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} yc^2(x,y)dydx\right)$$
$$+\frac{1}{h^2}\left(\int_{ih}^{(i+1)h} x^2 \int_{jh}^{(i+j+1)h-x} c^2(x,y)dydx + 2\int_{ih}^{(i+1)h} x \int_{jh}^{(i+j+1)h-x} yc^2(x,y)dydx\right.$$
$$\left.+\int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} y^2c^2(x,y)dydx\right),$$

$$I^c_{\beta 2} = \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(i+j)h} c^2(x,y)\left(1+j-\frac{y}{h}\right)^2 dydx$$
$$= (1+j)^2 \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} c^2(x,y)dydx - \frac{2}{h}(1+j)\int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} yc^2(x,y)dydx$$
$$+\frac{1}{h^2}\int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} y^2c^2(x,y)dydx,$$

$$I^c_{\beta 3} = \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} c^2(x,y)\left(1-i+\frac{x}{h}\right)^2 dydx$$
$$= (1-i)^2 \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} c^2(x,y)dydx + \frac{2}{h}(1-i)\int_{(i-1)h}^{ih} x \int_{jh}^{(i+j)h-x} c^2(x,y)dydx$$
$$+\frac{1}{h^2}\int_{(i-1)h}^{ih} x^2 \int_{jh}^{(i+j)h-x} c^2(x,y)dydx,$$

$$I^c_{\beta 4} = \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c^2(x,y)\left(1-i-j+\frac{1}{h}(x+y)\right)^2 dydx$$

$$= (1-i-j)^2 \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c^2(x,y)dydx + \frac{2}{h}(1-i-j)\left(\int_{(i-1)h}^{ih} x \int_{(i+j-1)h-x}^{jh} c^2(x,y)dydx + \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} yc^2(x,y)dydx\right)$$

$$+\frac{1}{h^2}\left(\int_{(i-1)h}^{ih} x^2 \int_{(i+j-1)h-x}^{jh} c^2(x,y)dydx + 2\int_{(i-1)h}^{ih} x \int_{(i+j-1)h-x}^{jh} yc^2(x,y)dydx + \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} y^2c^2(x,y)dydx\right),$$

$$I^c_{\beta 5} = \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} c^2(x,y)\left(1-j+\frac{y}{h}\right)^2 dydx$$

$$= (1-j)^2 \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} c^2(x,y)dydx + \frac{2}{h}(1-j)\int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} yc^2(x,y)dydx$$

$$+\frac{1}{h^2}\int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} y^2c^2(x,y)dydx,$$

$$I^c_{\beta 6} = \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c^2(x,y)\left(1+i-\frac{x}{h}\right)^2 dydx$$

$$= (1+i)^2 \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c^2(x,y)dydx - \frac{2}{h}(1+i)\int_{ih}^{(i+1)h} x \int_{(i+j)h-x}^{jh} c^2(x,y)dydx$$

$$+\frac{1}{h^2}\int_{ih}^{(i+1)h} x^2 \int_{(i+j)h-x}^{jh} c^2(x,y)dydx$$

and

$$\beta_{ij}^{(2)} = \int_0^1 \int_0^1 c(x,y) u_h^{(k-1)} f(x,y) dxdy$$

$$= u_{ij}^{(k-1)} \left[\int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y) \left(1 + i + j - \frac{1}{h}(x+y)\right) f(x,y) dydx \right.$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} c(x,y) \left(1 + j - \frac{y}{h}\right) f(x,y) dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} c(x,y) \left(1 - i + \frac{x}{h}\right) f(x,y) dydx$$

$$+ \int_{(i-1)h}^{ih} \int_{(i+j+1)h-x}^{jh} c(x,y) \left(1 - i - j - \frac{1}{h}(x+y)\right) f(x,y) dydx$$

$$+ \int_{ih}^{(i+1)h} \int_{(j-1)h-x}^{(i+j)h-x} c(x,y) \left(1 - j - \frac{y}{h}\right) f(x,y) dydx$$

$$\left. + \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c(x,y) \left(1 + i - \frac{x}{h}\right) f(x,y) dydx \right]$$

$$= u_{ij}^{(k-1)} \left(I_{\beta 1}^{cf} + I_{\beta 2}^{cf} + I_{\beta 3}^{cf} + I_{\beta 4}^{cf} + I_{\beta 5}^{cf} + I_{\beta 6}^{cf} \right),$$

whereby

$$I_{\beta 1}^{cf} = \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y) \left(1 + i + j - \frac{1}{h}(x+y)\right) f(x,y) dydx$$

$$= (1 + i + j) \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} c(x,y) f(x,y) dydx$$

$$- \frac{1}{h} \left(\int_{ih}^{(i+1)h} x \int_{jh}^{(i+j+1)h-x} c(x,y) f(x,y) dydx \right.$$

$$\left. + \int_{ih}^{(i+1)h} \int_{jh}^{(i+j+1)h-x} y c(x,y) f(x,y) dydx \right),$$

$$
\begin{aligned}
I_{\beta 2}^{cf} &= \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(i+j)h} c(x,y)\left(1+j-\frac{y}{h}\right) f(x,y)dydx \\
&= (1+j) \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} c(x,y)f(x,y)dydx \\
&\quad -\frac{1}{h} \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} yc(x,y)f(x,y)dydx \\
&\quad +\frac{1}{h^2} \int_{(i-1)h}^{ih} \int_{(i+j)h-x}^{(j+1)h} y^2c^2(x,y)dydx,
\end{aligned}
$$

$$
\begin{aligned}
I_{\beta 3}^{cf} &= \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} c(x,y)\left(1-i+\frac{x}{h}\right) f(x,y)dydx \\
&= (1-i) \int_{(i-1)h}^{ih} \int_{jh}^{(i+j)h-x} c(x,y)f(x,y)dydx \\
&\quad +\frac{1}{h} \int_{(i-1)h}^{ih} x \int_{jh}^{(i+j)h-x} (x,y)f(x,y)dydx,
\end{aligned}
$$

$$
\begin{aligned}
I_{\beta 4}^{cf} &= \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c(x,y)\left(1-i-j+\frac{1}{h}(x+y)\right) f(x,y)dydx \\
&= (1-i-j) \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} c(x,y)f(x,y)dydx \\
&\quad +\frac{1}{h}\left(\int_{(i-1)h}^{ih} x \int_{(i+j-1)h-x}^{jh} c(x,y)f(x,y)dydx \right. \\
&\quad \left. + \int_{(i-1)h}^{ih} \int_{(i+j-1)h-x}^{jh} yc(x,y)f(x,y)dydx \right),
\end{aligned}
$$

$$I_{\beta 5}^{cf} = \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} c(x,y)\left(1-j+\frac{y}{h}\right) f(x,y)dydx$$
$$= (1-j) \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} c(x,y)f(x,y)dydx$$
$$+\frac{1}{h} \int_{ih}^{(i+1)h} \int_{(j-1)h}^{(i+j)h-x} yc(x,y)f(x,y)dydx,$$

$$I_{\beta 6}^{cf} = \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c(x,y)\left(1+i-\frac{x}{h}\right) f(x,y)dydx$$

$$= (1+i) \int_{ih}^{(i+1)h} \int_{(i+j)h-x}^{jh} c(x,y)f(x,y)dydx$$
$$-\frac{1}{h} \int_{ih}^{(i+1)h} x \int_{(i+j)h-x}^{jh} c(x,y)f(x,y)dydx.$$

The second norm appearing in the formula for the quantity $\alpha^{(k)}$, namely

$$\left\|c\right\|_{L^\infty(0,1)\times(0,1)} = \left\|c(x,y)\right\|_{L^\infty(0,1)\times(0,1)} = \text{ess sup}_{(x,y)\in(0,1)\times(0,1)} \left|c(x,y)\right|,$$

depends solely on the function $c(x,y)$ and should be calculated for each problem. Therefore, for the components of the vector $\alpha_{ij}^{(k)}$ we obtain

$$\alpha_{ij}^{(k)} = h\left(\sqrt{\beta_{ij}} + h \cdot \text{ess sup}_{(x,y)\in(0,1)\times(0,1)} \left|c(x,y)\right| \cdot \alpha_{ij}^{(k-1)}\right),$$

where the quantities β_{ij} are given by formula (4.21). The process of iterative determination of $u_h^{(k)}$ i $\alpha^{(k)}$ terminates after N iterations when the following conditions hold:

$$\left\|u_h^{(N)} - u_h^{(N-1)}\right\| < \varepsilon, \tag{4.22}$$

where

$$\left\|u_h^{(N)} - u_h^{(N-1)}\right\| = \max_{i,j=1,2,\ldots,n-1} \left\{\left|\underline{A}_{ij}^{(N)} - \underline{A}_{ij}^{(N-1)}\right|, \left|\overline{A}_{ij}^{(N)} - \overline{A}_{ij}^{(N-1)}\right|\right\}$$

and

$$\left|\alpha^{(N)} - \alpha^{(N-1)}\right| < \varepsilon, \tag{4.23}$$

where ε denotes the required accuracy. The last inequality should hold for each component of the vector $\alpha^{(k)}$, which means that the foloowing inequality is satisfied:

$$\max_{i,j=1,2,\ldots,n-1} \left|\alpha_{ij}^{(N)} - \alpha_{ij}^{(N-1)}\right| < \varepsilon. \tag{4.24}$$

Then, the so-called δ-extension of the obtained solution defined as follows is introduced:

$$\widetilde{u}_h^{(N)} = \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} \widetilde{A}_{ij}^{(N)} \varphi_{ij}(x,y), \quad \widetilde{\alpha}^{(N)} = \alpha^{(N)} + \delta, \tag{4.25}$$

where $\widetilde{A}_{ij}^{(N)} = \left[\underline{A}_{ij}^{(N)} - \delta, \overline{A}_{ij}^{(N)} + \delta\right]$.

Nakao's main theorem given in [76] states that if we further determine the quantities u_h and α in relation (4.18), in which $u_h^{(k)}$ and $\alpha^{(k)}$ are replaced by u_h and α, respectively, and $u_h^{(k-1)}$ and $\alpha^{(k-1)}$ are replaced by $\widetilde{u}_h^{(N)}$ and $\widetilde{\alpha}^{(N)}$, respectively, then when

$$u_h \subset \widetilde{u}_h^{(N)} \quad i \quad \alpha < \widetilde{\alpha}^{(N)},$$

then there exists an unambiguous solution u of the problem (4.5) lying in the set $u_h + [\alpha]$.

The complete algorithm of Nakao's method is presented in the form of pseudocode below. The results obtained as a result of its implementation can be found in Chapter 7 in Examples 5 and 6.

Algorytm 4.6. Nakao's method for verification and estimation of elliptic PDE solutions

1: {Details of $GalerkinApprox(f, \varphi, n)$ can be found in Section 4.2}
2: $u_h^{(0)} := GalerkinApprox(f, \varphi, n)$
3: $k := 1$
4: {$EndCondition(u_h^{(k)}, u_h^{(k-1)})$ function checks condtions (4.22)–(4.24)}
5: **while not** $EndCondition(u_h^{(k)}, u_h^{(k-1)})$ **do**
6: $\left(\nabla u_h^{(k)}, \nabla\varphi_{ij}\right) := \left(cu_h^{(k-1)} + f, \varphi_{ij}\right) + [-1,1]\, h\alpha^{(k-1)} \left\|\varphi_{ij}\right\|_{L^2(0,1)\times(0,1)}$,
7: $\alpha^{(k)} := h\left(\left\|cu_h^{(k-1)} + f\right\|_{L^2(0,1)\times(0,1)} + h\|c\|_{L^\infty(0,1)\times(0,1)}\alpha^{(k-1)}\right)$
8: $k := k + 1$
9: **end while**
10: {function $DeltaExtension(u_h^{(k)})$ implements formula (4.25)}
11: $u_h^{(k)} := DeltaExtension(u_h^{(k)})$
12: **return** $u_h^{(k)}$

In Algorithm 4.6. it should be noted that only the intermediate results and the final result are represented as intervals, while all calculations - starting from the Galerkin approximation to numerical integration and determination of norms for functions - are performed in classical floating point arithmetic. Therefore, Nakao's method is not a typical interval method, i.e. one where all calculations are performed in interval arithmetic (according to appropriate definitions for particular arithmetic operations), but only uses interval arithmetic to store the results of calculations and to verify the stop condition.

5 Second-order interval methods

In this chapter and the next, a set of methods belonging to the FDM class is presented for the different forms of the Poisson equation (see Section p. 2.2). Each method was first designed for floating point arithmetic and then extended for interval arithmetic. The interval methods differ from the corresponding classical methods in that they include an estimate of the method error resulting, in the case of FDM methods, from truncation of the number of words in the Taylor series. For the purpose of further analyses and comparisons all the methods presented in this paper have been designated by abbreviations depending on the form of the equation for which they have been designed, the order of error as well as the type of arithmetic applied. These designations are derived from acronyms of names of particular classes of equations which were presented in Section 2.2. All designations are summarized in Table 5.1.

Table 5.1. Designation of second-order methods presented in this paper according to the form of the equation and type of arithmetic

Equation	Arithmetic type		
	floating-point	interval proper	interval directed
PE (2.6)	PE2	IPE2	DIPE2
GPE (2.7)	GPE2	IGPE2	DIGPE2

This chapter covers the description of four second-order interval FDM methods. Two of them are classical methods for floating point arithmetic, and two more are their interval counterparts. Higher order methods are presented in the next chapter.

Each method leads to a system of linear equations and requires finding its solutions. In general, such a system can be solved by any of the known numerical algorithms dedicated to this problem. However, for the author of this paper it was interesting to compare the proposed FDM methods with one another and not the very problem of efficient solution of the system of linear equations. Only the correctness of the solutions was important. Therefore, for the purpose of this work, the Gauss–Jordan method with full selection of the basis element (see [66, p. 110]) was used and consistently applied while implementing the solution of systems of equations obtained for each of the methods described in Chapters 5 and 6.

5.1. Methods for the elementary form of the Poisson equation

The method presented in this section together with the numerical results obtained has been published in papers [29] and [32]. Here, the method's construction is described and the most relevant formulas are given.

5.1.1. Classical method

We assume that in each of the grid points under consideration there exist partial derivatives of the function u up to and including order four. We discretize the area $\overline{\Omega}$ in the way described in Section 2.3. We obtain a $m \times n$ grid of nodes. Then, for each of the nodes lying inside the area Ω we use the Taylor series expansion of the function u with respect to the variable x in the neighbourhood of the point x_i and with respect to the variable y in the neighbourhood of the point y_j. Then

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j),$$

where $\xi_i \in (x_{i-1}, x_{i+1})$ oraz

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2} - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j),$$

where $\eta_j \in (y_{j-1}, y_{j+1})$. Using these formulas allows us to express the Poisson equation at the points (x_i, y_j) in the form

$$\begin{aligned} &\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \\ &+\frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2} \\ &= f(x_i, y_j) + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j) \\ &i = 1, 2, \ldots, n-1, \quad j = 1, 2, \ldots, m-1. \end{aligned} \tag{5.1}$$

If we write the central differences in a simplified way, i.e.

$$\delta_x^2 u_{ij} = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2}, \quad \delta_y^2 u_{ij} = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{k^2}, \tag{5.2}$$

where $u_{ij} = u(x_i, y_j)$, $f_{ij} = f(x_i, y_j)$ and where $\xi_i \in (x_{i-1}, x_{i+1})$, $\eta_i \in (y_{i-1}, y_{i+1})$ denote intermediate points, then equation (5.1) can be written in the form

$$\delta_x^2 u_{ij} + \delta_y^2 u_{ij} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j) = f_{ij}. \tag{5.3}$$

Let us notice that omission of partial derivatives in the formula (5.3) causes simplification of the method notation to the form

$$\delta_x^2 u_{ij} + \delta_y^2 u_{ij} = f_{ij}. \tag{5.4}$$

The boundary conditions are given by the following formulas::

$$\begin{aligned} u_{0j} &= u(0, y_j) = \varphi_1(y_j) \text{ dla } j = 0, 1, \ldots, m, \\ u_{i0} &= u(x_i, 0) = \varphi_2(x_i) \text{ dla } i = 1, 2, \ldots, n-1, \\ u_{nj} &= u(\alpha, y_j) = \varphi_3(y_j) \text{ dla } j = 0, 1, \ldots, m, \\ u_{im} &= u(x_i, \beta) = \varphi_4(x_i) \text{ dla } i = 1, 2, \ldots, n-1. \end{aligned} \tag{5.5}$$

PE2 Method. If in equation (5.1) we omit the error components, i.e., the partial derivatives of $\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j)$ and $\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j)$, hen we obtain explicit formulas for the classical 5-point method of central differences of the form

$$\begin{gathered} k^2 u_{i-1,j} + h^2 u_{i,j-1} - 2\left(h^2 + k^2\right) u_{ij} \\ + k^2 u_{i+1,j} + h^2 u_{i,j+1} = h^2 k^2 f_{ij}, \\ i = 1, 2, \ldots, n-1, \quad j = 1, 2, \ldots, m-1. \end{gathered} \tag{5.6}$$

As one can see, equations (5.4) together with conditions (5.5) lead to equations (5.6), which define a system of $(n-1)(m-1)$ linear equations with $(n-1)(m-1)$ unknowns u_{ij} which are approximations of the quantities $u(x_i, y_j)$, i.e., the sought values of the function u for the nodes lying inside the mesh. his system can be solved by one of the known methods for solving systems of linear equations – an exact or iterative one. The resulting method is characterized by a truncation error of order $O(h^2 + k^2)$ (see [7] and [45]). The method is denoted by the abbreviation PE2 (see Table 5.1).

5.1.2. Interval methods

The following central differences result from the Taylor series expansion of the real function u w in the neighbourhood of the point (x, y):

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{u(x-h, y) - 2u(x, y) + u(x+h, y)}{h^2} \\ &\quad - \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi, y), \xi \in (x-h, x+h), \\ \frac{\partial^2 u}{\partial y^2} &= \frac{u(x, y-k) - 2u(x, y) + u(x, y+k)}{k^2} \\ &\quad - \frac{h^2}{12} \frac{\partial^4 u}{\partial y^4}(x, \eta), \eta \in (y-k, y+k). \end{aligned} \tag{5.7}$$

We can find intervals that estimate the values $\frac{\partial^4 u}{\partial x^4}(\xi, y)$ and $\frac{\partial^4 u}{\partial y^4}(x, \eta)$.

Let us suppose first that there exists a constant M, such that

$$\left| \frac{\partial^4 u}{\partial x^2 \partial y^2} \right| \leq M \text{ for all values } 0 \leq x \leq \alpha \text{ and } 0 \leq y \leq \beta$$

and let

$$\frac{\partial^4 u}{\partial^2 x \partial^2 y}(x, y) = \frac{\partial^4 u}{\partial^2 y \partial^2 x}(x, y).$$

It follows directly from Poisson's equation (2.6)that

$$\frac{\partial^4 u}{\partial x^4}(x,y) = \frac{\partial^2 f}{\partial x^2} - \frac{\partial^4 u}{\partial x^2 \partial y^2}(x,y),$$
$$\frac{\partial^4 u}{\partial y^4}(x,y) = \frac{\partial^2 f}{\partial y^2} - \frac{\partial^4 u}{\partial y^2 \partial x^2}(x,y).$$

We shall try to estimate $\frac{\partial^4 u}{\partial x^4}$ and $\frac{\partial^4 u}{\partial y^4}$. The function f is a known parameter of the equation, on the right hand side, $\frac{\partial^4 u}{\partial x^2 \partial y^2}$ and $\frac{\partial^4 u}{\partial y^2 \partial x^2}$ remain unknown. Using formulas (5.7) we obtain

$$\frac{\partial^2}{\partial y^2}\left(\frac{\partial^2 u}{\partial x^2}\right) =$$
$$= \frac{u(x-h,y-k) - 2u(x-h,y) + u(x-h,y+k)}{h^2 k^2}$$
$$-2\frac{u(x,y-k) - 2u(x,y) + u(x,y+k)}{h^2 k^2}$$
$$+\frac{u(x+h,y-k) - 2u(x+h,y) + u(x+h,y+k)}{h^2 k^2}$$
$$-\frac{k^2}{12h^2}\left[\frac{\partial^4 u}{\partial y^4}(x-h,\eta_1) + \frac{\partial^4 u}{\partial y^4}(x,\eta_2) + \frac{\partial^4 u}{\partial y^4}(x+h,\eta_3)\right]$$
$$-\frac{h^2}{12}\frac{\partial^2}{\partial y^2}\left[\frac{\partial^4 u}{\partial x^4}(\xi,y)\right]$$

and

$$\frac{\partial^2}{\partial x^2}\left(\frac{\partial^2 u}{\partial y^2}\right) =$$
$$= \frac{u(x-h,y-k) - 2u(x,y-k) + u(x+h,y-k)}{h^2 k^2}$$
$$-2\frac{u(x-h,y) - 2u(x,y) + u(x+h,y)}{h^2 k^2}$$
$$+\frac{u(x-h,y+k) - 2u(x,y+k) + u(x+h,y+k)}{h^2 k^2}$$
$$-\frac{h^2}{12k^2}\left[\frac{\partial^4 u}{\partial x^4}(\xi_1,y-k) + \frac{\partial^4 u}{\partial x^4}(\xi_2,y) + \frac{\partial^4 u}{\partial x^4}(\xi_3,y+k)\right]$$
$$-\frac{k^2}{12}\frac{\partial^2}{\partial x^2}\left[\frac{\partial^4 u}{\partial y^4}(x,\eta)\right],$$

where $\xi, \xi_1, \xi_2, \xi_3 \in (x-h, x+h), \eta, \eta_1, \eta_2, \eta_3 \in (y-k, y+k)$. If the values of h and k are sufficiently small and the fourth order partial derivatives are not very large, it follows from the above equations that

$$\frac{\partial^2}{\partial y^2}\left(\frac{\partial^2 u}{\partial x^2}\right) \approx$$
$$\approx \frac{u(x-h,y-k) - 2u(x-h,y) + u(x-h,y+k)}{h^2 k^2}$$
$$-2\frac{u(x,y-k) - 2u(x,y) + u(x,y+k)}{h^2 k^2}$$
$$+\frac{u(x+h,y-k) - 2u(x+h,y) + u(x+h,y+k)}{h^2 k^2}$$

and

$$\frac{\partial^2}{\partial x^2}\left(\frac{\partial^2 u}{\partial y^2}\right) \approx$$
$$\approx \frac{u(x-h,y-k)-2u(x,y-k)+u(x+h,y-k)}{h^2k^2}$$
$$-2\frac{u(x-h,y)-2u(x,y)+u(x+h,y)}{h^2k^2}$$
$$+\frac{u(x-h,y+k)-2u(x,y+k)+u(x+h,y+k)}{h^2k^2}.$$

Note that the right-hand sides of the above approximations are equal, so we propose to estimate the constant M as follows:

$$\begin{aligned} M \approx \frac{1,5}{h^2k^2} \max_{\substack{i=1,2,\ldots,n-1\\ j=1,2,\ldots,m-1}} |4u_{ij} \\ -2(u_{i-1,j}+u_{i,j-1}+u_{i,j+1}+u_{i+1,j}), \\ +u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1}|, \end{aligned} \tag{5.8}$$

where the values of u_{ij} are obtained by the classical central difference method (5.5) – – (5.6), a factor of 1.5 (instead of 1.0) indicates that we are taking a value 50% larger. Thus

$$\frac{\partial^4 u}{\partial x^4}(\xi,y) \in \Psi(X+[-h,h],Y)+[-M,M],$$
$$\frac{\partial^4 u}{\partial x^4}(x,\eta) \in \Omega(X,Y+[-k,k])+[-M,M],$$

for each value $\xi \in (x-h,x+h)$ and each value $\eta \in (y-k,y+k)$, where X and Y denote the interval extensions of x and y, respectively, and $\Psi(X,Y)$ and $\Omega(X,Y)$ denote the interval extensions of the functions $\frac{\partial^2 f}{\partial x^2}(x,y)$ and $\frac{\partial^2 f}{\partial y^2}(x,y)$, respectively.

IPE2 Method. If we now return to the Poisson equation defined at the grid points, i.e. equation (5.3),and write the partial derivatives in it on the right hand side, then we obtain an interval analogue for this equation. Assuming that all interval expansions are proper intervals[1], we have

$$\begin{aligned} k^2U_{i-1,j}+h^2U_{i,j-1}-2(h^2+k^2)U_{i,j}+k^2U_{i+1,j}+h^2U_{i,j+1} \\ = h^2k^2\left(F_{i,j}+\frac{1}{12}\left(h^2\Psi(X_i+[-h,h],Y_j)+k^2\Omega(X_i,Y_j+[-k,k])\right.\right. \\ \left.\left.+(h^2+k^2)[-M,M]\right)\right), \\ i=1,2,\ldots,n-1, \quad j=1,2,\ldots,m-1, \end{aligned} \tag{5.9}$$

where $F_{ij}=F(X_i,Y_j)$ and where

$$\begin{gathered} U_{0j}=\Phi_1(Y_j),\ \ U_{i0}=\Phi_2(X_i),\ \ U_{nj}=\Phi_3(Y_j),\ \ U_{im}=\Phi_4(X_i) \\ \text{dla } j=0,1,\ldots,m \text{ oraz } i=1,2,\ldots,n-1, \end{gathered} \tag{5.10}$$

while $\Phi_1(Y),\Phi_2(X),\Phi_3(Y)$ and $\Phi_4(X)$ denote the interval expansions of the functions $\varphi_1(y)$, $\varphi_2(x)$, $\varphi_3(y)$ i $\varphi_4(x)$, respectively. The system of linear equations (5.9) – (5.10), hereafter abbreviated IPE2, can be solved using ordinary (proper) variable interval arithmetic since all the intervals defined here are proper intervals.

[1] i.e., those on which we operate in ordinary interval arithmetic, see Chapter 3, def. 20.

DIPE2 Method. Note that in interval form, equation (5.3) can also be written as follows:

$$\begin{gathered} k^2U_{i-1,j}+h^2U_{i,j-1}-2(h^2+k^2)U_{ij}+k^2U_{i+1,j}+h^2U_{i,j+1} \\ -\frac{h^2k^2}{12}\left(h^2\Psi(X_i+[-h,h],Y_j)+k^2\Omega(X_i,Y_j+[-k,k])\right. \\ \left.+(h^2+k^2)[-M,M]\right) \\ =h^2k^2F_{i,j}, \\ i=1,2,\ldots,n-1,\quad j=1,2,\ldots,m-1. \end{gathered}$$

Using directed interval arithmetic, we can add elements on both sides of equation (5.3) opposite to the elements associated with the error components. Then, we obtain

$$\begin{gathered} k^2U_{i-1,j}+h^2U_{i,j-1}-2(h^2+k^2)U_{i,j}+k^2U_{i+1,j}+h^2U_{i,j+1} \\ =h^2k^2\left(F_{i,j}+\frac{1}{12}\left(h^2\Psi(X_i+[-h,h],Y_j)+k^2\Omega(X_i,Y_j+[-k,k])\right.\right. \\ \left.\left.+(h^2+k^2)[M,-M]\right)\right), \\ i=1,2,\ldots,n-1,\quad j=1,2,\ldots,m-1. \end{gathered} \tag{5.11}$$

Equation (5.11) differs from equation (5.9) only by the last expression on the right-hand side, i.e., $[M,-M]$, which is an improper interval. Using directed variable interval arithmetic, we can solve the system of equations (5.11) (together with the boundary conditions (5.10)). We will denote this method by the abbreviation IDPE2. If the interval solutions of this system are in the form of improper intervals, then in order to obtain proper intervals one can apply the so-called interval projection, i.e., transform every interval $[a^-,a^+]$, for which $a^+<a^-$, to the interval $[a^+,a^-]$.

We should also add a remark concerning the constant M.In general, when the exact solution is not known and no conclusions can be drawn as for the value of this constant on the basis of physical or technical properties or the characteristics of the problem under consideration, we propose to find this constant using the formula

$$\begin{aligned} \frac{\partial^4u}{\partial x^2\partial y^2}(x_i,y_j)=\lim_{h\to0}\lim_{k\to0}\left(\frac{u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1}}{h^2k^2}\right. \\ \left.+\frac{4u_{i,j}-2(u_{i-1,j}+u_{i,j-1})+u_{i,j+1}+u_{i+1,j}}{h^2k^2}\right). \end{aligned}$$

We can calculate the quantities

$$\begin{aligned} M_{nm}=\frac{1}{h^2k^2}\max_{i,j}|u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1} \\ +4u_{ij}-2(u_{i-1,j}+u_{i,j-1}+u_{i,j+1}+u_{i+1,j})| \end{aligned}$$

for $i=1,2,\ldots,n-1$, $j=1,2,...,m-1$, where the values u_{ij} are obtained in the classical way for different values of n and m, let us assume $n=m=10,20,\ldots,N$ and where the number N is sufficiently large. We can then plot a curve of values of M_{nm} against different values of $n=m$, as long as there is no execution time exception during the computation due to handling the conditions in Theorem 1 (Rump). The constant M can be easily determined from the resulting graph, since the inequality $\lim\limits_{n\to\infty}\lim\limits_{m\to\infty}M_{nm}\leq M$.

Using such a constructed method for the Poisson equation, we can expect that at all interior grid points where we approximate the values of the function u by its interval expansions taking into account the intervals containing the error estimate of the method,

$u(x_i, y_j) \in U_{ij}$ where $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, m$. The computational experiments presented in the next section confirm that for all grid points the exact solution is within the range of the obtained interval solution. Let us emphasize again that in the presented interval method, very important is the component

$$\frac{1}{12}\left[h^2\Psi(X_i + [-h, h], Y_j) + k^2\Omega(X_i, Y_j + [-k, k]) + (h^2 + k^2)[-M, M]\right],$$

because it guarantees that the error of the method is taken into account in the obtained interval solution.

To find solutions at interior points of the grid, we need to solve the system $(n-1)(m-1)$ interval linear equations (5.11). This system may be written in the form

$$\mathbf{AU} = \mathbf{Q}, \tag{5.12}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{C} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{B} & \mathbf{C} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{B} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{B} & \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{C} & \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{C} & \mathbf{B} \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \vdots \\ \mathbf{U}_{n-3} \\ \mathbf{U}_{n-2} \\ \mathbf{U}_{n-1} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \end{bmatrix},$$

$$\mathbf{U_i} = \begin{bmatrix} U_{i,1} \\ U_{i,2} \\ \vdots \\ U_{i,m-1} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} Q_{i,1} \\ Q_{i,2} \\ \vdots \\ Q_{i,m-1} \end{bmatrix},$$

$$i = 1, 2, \ldots, n-1,$$

$$\mathbf{B} = \begin{bmatrix} \gamma & h^2 & 0 & \dots & 0 & 0 & 0 \\ h^2 & \gamma & h^2 & \dots & 0 & 0 & 0 \\ 0 & h^2 & \gamma & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & \gamma & h^2 & 0 \\ 0 & 0 & 0 & \dots & h^2 & \gamma & h^2 \\ 0 & 0 & 0 & \dots & 0 & h^2 & \gamma \end{bmatrix},$$

$$\gamma = -2(h^2 + k^2),$$

$$\mathbf{C} = \begin{bmatrix} k^2 & 0 & \dots & 0 & 0 \\ 0 & k^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & k^2 & 0 \\ 0 & 0 & \dots & 0 & k^2 \end{bmatrix},$$

$$\dim \mathbf{B} = \dim \mathbf{C} = (m-1) \times (m-1),$$

$$Q_{1,1} = \Lambda_{1,1} - k^2\Phi_1([k,k]) - h^2\Phi_2([h,h]),$$

$$Q_{1,j} = \Lambda_{1,j} - k^2\Phi_1([jk,jk]),$$

$$j = 2,3,\dots,m-2,$$

$$Q_{1,m-1} = \Lambda_{1,m-1} - k^2\Phi_1([(m-1)k,(m-1)k]) - h^2\Phi_4([h,h]),$$

$$Q_{i,1} = \Lambda_{i,1} - h^2\Phi_2([ih,ih]),$$

$$i = 2,3,\dots,n-2,$$

$$Q_{i,j} = \Lambda_{i,j},$$

$$i = 2,3,\dots,n-2, j = 2,3,\dots,m-2,$$

$$Q_{i,m-1} = \Lambda_{i,m-1} - h^2\Phi_4([ih,ih]),$$

$$i = 2,3,\dots,n-2,$$

$$Q_{n-1,1} = \Lambda_{n-1,1} - h^2\Phi_2([(n-1)h,(n-1)h]) - k^2\Phi_3([k,k]),$$

$$Q_{n-1,j} = \Lambda_{n-1,j} - k^2\Phi_3([jk,jk]),$$

$$j = 2,3,\dots,m-2,$$

$$Q_{n-1,m-1} = \Lambda_{n-1,m-1} - k^2\Phi_3([(m-1)k,(m-1)k]) - h^2\Phi_4([(n-1)h,(n-1)h]),$$

and where

$$\Lambda_{i,j} = h^2k^2\Bigg\{F_{i,j} + \frac{1}{12}\left[h^2\Psi(X_i + [-h,h], Y_j)\right.$$
$$\left.+k^2\Omega(X_i, Y_j) + [-k,k] + (h^2+k^2)[-M,M]\right]\Bigg\}.$$

5.2. Methods for the generalised form of the Poisson equation

This section presents methods for finding solutions to equations of the form

$$a(x,y)\frac{\partial^2 u}{\partial x^2}(x,y) + b(x,y)\frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y), \tag{5.13}$$

where

$$a(x,y)\cdot b(x,y) > 0,$$

with Dirichlet boundary conditions defined as follows:

$$\begin{gathered} u(x,y) = \varphi(x,y), \text{ for all } (x,y)\in\Gamma, \\ \Gamma = \{(x,y): \quad (x=\alpha_1,\alpha_2 \ \wedge \ \beta_1 \le y \le \beta_2) \\ \vee \quad (\alpha_1 \le x \le \alpha_2 \ \wedge \ y = \beta_1,\beta_2)\}, \end{gathered} \tag{5.14}$$

where

$$u|_\Gamma = \varphi(x,y) = \begin{cases} \varphi_1(y) \text{ for } x=\alpha_1, \\ \varphi_2(x) \text{ for } y=\beta_1, \\ \varphi_3(y) \text{ for } x=\alpha_1, \\ \varphi_4(x) \text{ for } y=\beta_2. \end{cases}$$

Let us point out that they were a step towards developing methods for the equations considered in Section 6.2 and, together with numerical results, were published in the paper [30].

5.2.1. Classical method

Analogously to earlier situation (see Section 2.3) we shall define a grid of nodes defined on a rectangular region, with Dirichlet boundary conditions given by the general formula (5.14). Next, assuming that at each interior point of the grid there exist partial derivatives of the function $u = u(x,y)$ up to and including the fourth order, using the expansions of the function u at the point (x_i, y_j) in a Taylor series with respect to x and y we obtain

$$\begin{gathered} a(x_i,y_j)\cdot\left[\frac{u(x_{i+1},y_j) - 2u(x_i,y_j) + u(x_{i-1},y_j))}{h^2} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i,y_j)\right] \\ +b(x_i,y_j)\cdot\left[\frac{u(x_i,y_{j+1}) - 2u(x_i,y_j) + u(x_i,y_{j-1}))}{k^2} - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i,\eta_j)\right] \\ = f(x_i,y_j), \end{gathered} \tag{5.15}$$

where $\xi_i \in (x_{i-1}, x_{i+1})$, $\eta_j \in (y_{j-1}, y_{j+1})$. As for the values of the function u lying on the edge of Γ they can be written in the form

$$\begin{aligned} u_{0j} &= u(\alpha_1, y_j) = \varphi_1(y_j) \text{ dla } j = 0, 1, \ldots, m, \\ u_{i0} &= u(x_i, \beta_1) = \varphi_2(x_i) \text{ dla } i = 1, 2, \ldots, n-1, \\ u_{nj} &= u(\alpha_2, y_j) = \varphi_3(y_j) \text{ dla } j = 0, 1, \ldots, m, \\ u_{im} &= u(x_i, \beta_2) = \varphi_4(x_i) \text{ dla } i = 1, 2, \ldots, n-1. \end{aligned} \tag{5.16}$$

GPE2 Method. If in equation (5.15) we omit the components containing partial derivatives and simplify the notation by taking $u_{ij} = u(x_i, y_j)$, $a_{ij} = a(x_i, y_j)$ and $b_{ij} = = b(x_i, y_j)$, then we obtain, defining the classical method of central differences, the formula of the following form:

$$a_{ij} \cdot \left(\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} \right) + b_{ij} \cdot \left(\frac{u_{i,j+1} - 2u_{ij} + u_{i,j+1}}{k^2} \right) = f_{ij}. \tag{5.17}$$

The values of u_{ij}, for each $i = 1, 2, \ldots m-1$ and $j = 1, 2, \ldots n-1$, are obtained by solving the system $(m-1)(n-1)$ of linear equations defined by equation (5.17).

5.2.2. Interval methods

As in Section 5.1.2, here also we take into account the error of the method in the obtained interval solutions. For this purpose, we transform equation (5.15) to the following form:

$$\begin{aligned} &a(x_i, y_j) \cdot \left[\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} \right] \\ &+b(x_i, y_j) \cdot \left[\frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} \right] \\ &-a(x_i, y_j) \cdot \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) - b(x_i, y_j) \cdot \frac{k^2}{12} \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j) = f(x_i, y_j). \end{aligned} \tag{5.18}$$

If we place the error components on the right hand side, then we obtain the formula

$$\begin{aligned} &a(x_i, y_j) \cdot \left[\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} \right] \\ &+b(x_i, y_j) \cdot \left[\frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} \right] \\ &= f(x_i, y_j) + a(x_i, y_j) \cdot \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + b(x_i, y_j) \cdot \frac{k^2}{12} \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j). \end{aligned} \tag{5.19}$$

For the estimation of the error components, let us assume that there exist constants M and N, such that

$$\left| \frac{\partial^4 u}{\partial x^4}(x, y) \right| \leq M \text{ oraz } \left| \frac{\partial^4 u}{\partial y^4}(x, y) \right| \leq N \tag{5.20}$$

for all points (x, y), such that $\alpha_1 \leq x \leq \alpha_2$ and $\beta_1 \leq y \leq \beta_2$. We propose to determine the values of these constants using the approximation of derivatives of order four by central differences. Thus, if we denote

$$\begin{aligned} M_h &= \max_{ij} \frac{6u_{ij} - 4u_{i-1,j} - 4u_{i+1,j} + u_{i-2,j} + u_{i+2,j}}{h^4}, \\ N_k &= \max_{ij} \frac{6u_{ij} - 4u_{i,j-1} - 4u_{i,j+1} + u_{i,j-2} + u_{i,j+2}}{k^4}, \end{aligned} \tag{5.21}$$

where the values of u_{ij} can be obtained by the classical method defined by equation (5.17), then the values of the constants M and N can be determined as the following limits:

$$\begin{aligned} M &= \lim_{h\to 0} M_h, \\ N &= \lim_{k\to 0} N_h. \end{aligned} \tag{5.22}$$

Clearly $h \to 0$, when $m \to \infty$, and $k \to 0$, when $n \to \infty$. This means that by increasing the grid size we can determine M and N experimentally.

IGPE2 Method. Let $A(X,Y)$, $B(X,Y)$ and $U(X,Y)$ denote the interval expansions of the functions $a(x,y)$, $b(x,y)$ and $u(x,y)$ respectively. To simplify the notation let us assume

$$\begin{aligned} A_{ij} &= A(X_i, Y_j), \\ B_{ij} &= B(X_i, Y_j), \\ U_{ij} &= U(X_i, Y_j). \end{aligned}$$

Then, using equation (5.19) and values of constants M and N determined experimentally, we can write the following formula determining the method in ordinary interval arithmetic:

$$\begin{aligned} k^2 A_{ij} U_{i+1,j} + h^2 B_{ij} U_{i,j+1} - 2(k^2 A_{ij} + h^2 B_{ij}) U_{ij} + k^2 A_{ij} U_{i-1,j} + h^2 B_{ij} U_{i,j-1} \\ = h^2 k^2 \left\{ F_{ij} + \frac{h^2 A_{ij}}{12}[-M, M] + \frac{k^2 B_{ij}}{12}[-N, N] \right\}. \end{aligned} \tag{5.23}$$

DIGPE2 Method. As it is well known, there are opposite elements in directed interval arithmetic. In interval form, equation (5.18) can also be written as follows:

$$\begin{aligned} k^2 A_{ij} U_{i+1,j} + h^2 B_{ij} U_{i,j+1} - 2(k^2 A_{ij} + h^2 B_{ij}) U_{ij} \\ + k^2 A_{ij} U_{i-1,j} + h^2 B_{ij} U_{i,j-1} - \frac{h^4 A_{ij}}{12}[-M, M] - \frac{k^4 B_{ij}}{12}[-N, N] \\ = h^2 k^2 F_{ij}. \end{aligned}$$

If we add to both sides of the above equation the components opposite to those associated with the method error, then we get

$$\begin{aligned} k^2 A_{ij} U_{i+1,j} + h^2 B_{ij} U_{i,j+1} - 2(k^2 A_{ij} + h^2 B_{ij}) U_{ij} + k^2 A_{ij} U_{i-1,j} + h^2 B_{ij} U_{i,j-1} \\ = h^2 k^2 \left\{ F_{ij} + \frac{h^2 A_{ij}}{12}[M, -M] + \frac{k^2 B_{ij}}{12}[N, -N] \right\}. \end{aligned} \tag{5.24}$$

Note that equation (5.24) differs from equation (5.23) only in that the error components of the method are written as directed intervals $[M, -M]$ and $[N, -N]$.

The other methods that were developed within this thesis are orders of magnitude higher than the second one and are therefore placed in a separate chapter. The next one.

6 Higher order interval methods

In this chapter, finite difference methods of higher (than second) orders are described, the methods which allow finding interval solutions for the generalized Poisson equation (GPE) and for elliptic equations of the form $a\Delta u + cu = f$ (NE). The designations of the different methods described in this chapter are given in Table 6.1.

Table 6.1. Designations of higher order methods presented in the paper according to the form of Eq, order of method and type of arithmetic

Equation	Order of the method	Arithmetic type		
		flaoting-point	interval proper	interval directed
PE (2.6)	4	PE4	IPE4	DIPE4
NE (2.8)	3	NE3	INE3	DINE3
NE (2.8)	3	NE5C	INE5C	DINE5C

6.1. Methods for the elementary form of the Poisson equation

6.1.1. Classical method

Using Taylor series, we can write the Poisson equation (2.6) in terms of (x_i, y_j) of the form

$$\begin{aligned}
&\delta_x^2 u_{ij} + \delta_y^2 u_{ij} + \frac{1}{12}(h^2+k^2)\delta_x^2\delta_y^2 u_{ij} \\
&\quad - \frac{1}{240}\left(h^4 \frac{\partial^6 u}{\partial x^4 \partial y^2}(\xi_i, y_j) + k^4 \frac{\partial^6 u}{\partial x^2 \partial y^4}(x_i, \eta_j)\right) \\
&\quad - \frac{h^2k^2}{144}\left(\frac{\partial^6 u}{\partial x^4 \partial y^2}(\xi_i, \eta_j) + \frac{\partial^6 u}{\partial x^2 \partial y^4}(\xi_i, \eta_j)\right) \\
&= f_{ij} + \frac{1}{12}(h^2\delta_x^2 + k^2\delta_y^2) f_{ij} \\
&\quad - \frac{1}{240}\left(h^4 \frac{\partial^4 f}{\partial x^4}(\xi_i, y_j) + k^4 \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j)\right).
\end{aligned} \tag{6.1}$$

PE4 Method. If we neglect the partial derivatives in equation (6.1) we obtain the following classical fourth order finite difference method [24, 101]:

$$\delta_x^2 u_{ij} + \delta_y^2 u_{ij} + \frac{1}{12}(h^2+k^2)\delta_x^2\delta_y^2 u_{ij} = f_{ij} + \frac{1}{12}(h^2\delta_x^2 + k^2\delta_y^2) f_{ij}. \tag{6.2}$$

6.1.2. Interval methods

Based on the method defined in the previous section, two interval methods can be derived, for ordinary and directed interval arithmetic – we will denote them by the abbreviations IPE4 and DIPE4, respectively. Together with example numerical results, they have been published in [67].

Let $\Theta(X,Y)$ and $\Xi(X,Y)$ denote the interval expansions of the functions $\frac{\partial^4 f}{\partial x^4}$ and $\frac{\partial^4 f}{\partial y^4}$ respectively, and let us suppose that

$$\left|\frac{\partial^6 u}{\partial x^4 \partial y^2}\right| \le P \text{ i } \left|\frac{\partial^6 u}{\partial x^2 \partial y^4}\right| \le Q \text{ dla } 0 \le x \le \alpha \text{ i } 0 \le y \le \beta.$$

It is clear that

$$\frac{\partial^4 f}{\partial x^4}(\xi, y) \in \Theta(X + [-h,h], Y), \quad \frac{\partial^4 f}{\partial y^4}(x,\eta) \in \Xi(X, Y + [-k,k]),$$
$$\frac{\partial^6 u}{\partial x^4 \partial y^2} \in [-P,P] \quad \frac{\partial^6}{\partial x^2 \partial y^4} \in [-Q,Q].$$

IPE4 Method. If we write all partial derivatives on the right-hand side in equation (6.1), it is easy obtain an interval analogy for this equation. Thus, we have

$$\begin{aligned}
&(h^2+k^2)(U_{i-1,j-1}) + U_{i-1,j+1} + U_{i+1,j-1} + U_{i+1,j+1}) \\
&\quad + 2(5k^2-h^2)(U_{i-1,j} + U_{i+1,j}) + 2(5h^2-k^2)(U_{i,j-1} + U_{i,j+1}) \\
&\quad - 20(h^2+k^2)U_{i,j} \\
&= h^2k^2\Bigg(F_{i-1,j} + F_{i+1,j} + 8F_{i,j} + F_{i,j-1} + F_{i,j+1}) \\
&\quad - \frac{1}{20}\left(h^4\Theta(X_i + [-h,h], Y_j) + k^4\Xi(X_i, Y_j + [-k,k])\right) \\
&\quad + \frac{1}{20}\left(h^4[-P,P] + k^4[-Q,Q]\right) + \frac{h^2k^2}{12}[-P-Q, P+Q]\Bigg).
\end{aligned} \tag{6.3}$$

DIPE4 Method. On the other hand, if we leave the partial derivatives on the left-hand side in equation (6.1)write down the interval analogy for this equation, and then add the corresponding interval counter elements (they exist only in directed interval arithmetic), then we obtain

$$\begin{aligned}
&(h^2+k^2)(U_{i-1,j-1}) + U_{i-1,j+1} + U_{i+1,j-1} + U_{i+1,j+1}) \\
&\quad + 2(5k^2-h^2)(U_{i-1,j} + U_{i+1,j}) + 2(5h^2-k^2)(U_{i,j-1} + U_{i,j+1}) \\
&\quad - 20(h^2+k^2)U_{i,j} \\
&= h^2k^2\Bigg(F_{i-1,j} + F_{i+1,j} + 8F_{i,j} + F_{i,j-1} + F_{i,j+1}) \\
&\quad - \frac{1}{20}\left(h^4\Theta(X_i + [-h,h], Y_j) + k^4\Xi(X_i, Y_j + [-k,k])\right) \\
&\quad + \frac{1}{20}\left(h^4[-P,P] + k^4[-Q,Q]\right) + \frac{h^2k^2}{12}[P+Q, -P-Q]\Bigg).
\end{aligned} \tag{6.4}$$

The difference between equations (6.3) i (6.4) occurs only in the last line. If, for the problem under consideration, we have no information about the constants P and Q, we can determine them using the relation

$$\begin{aligned} P_{nm} = \frac{1}{h^4k^2} \max_{i,j} |&u_{i-2,j-1} + u_{i-2,j+1} + u_{i+2,j-1} + u_{i+2,j+1} - 2(u_{i-2,j} + u_{i+2,j}) \\ &- 4(u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1}) \\ &+8(u_{i-1,j} + u_{i+1,j}) + 6(u_{i,j-1} + u_{i,j+1}) - 12u_{ij}|\,, \\ Q_{nm} = \frac{1}{h^2k^4} \max_{i,j} |&u_{i-1,j-2} + u_{i+1,j-2} + u_{i-1,j+2} + u_{i+1,j+2} - 2(u_{i-2,j} + u_{i+2,j}) \\ &- 4(u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1}) \\ &+8(u_{i-1,j} + u_{i,j+1}) + 6(u_{i-1,j} + u_{i+1,j}) - 12u_{ij}| \end{aligned}$$

for $i = 2, 3, \ldots, n-2,\ \ j = 2, 3, \ldots, m-2$, where the values of u_{ij} are obtained by the classical method (in floating point arithmetic) for different values of n and m. Then, the values of the constants P and Q can be estimated from the fact that $\lim\limits_{n\to\infty} \lim\limits_{m\to\infty} P_{nm} \leq P$ and $\lim\limits_{n\to\infty} \lim\limits_{m\to\infty} Q_{nm} \leq Q$.

6.2. Methods for equations of the form $a\Delta u + cu = f$

This and the next section describe the methods for the elliptic equations considered by Nakao, i.e. of the form given by equation (2.8). Let us consider equation

$$a(x,y)\frac{\partial^2 u}{\partial x^2} + b(x,y)\frac{\partial^2 u}{\partial y^2} + c(x,y)u = f(x,y), \tag{6.5}$$

where

$$a(x,y)b(x,y) > 0$$

inside the rectangle Ω.

6.2.1. Classical method

The third order method described in this subsection was published in [68]. For the purpose of further references and comparisons, we denote it by the abbreviation NGPE3.

Assuming that there exist partial derivatives of order four of the function u and applying the Taylor series for the variable x in the neighbourhood of point x_i and for the variable $y$in the neighbourhood of point y_j, we can express equation (6.5) in points (x_i, y_j) in the form

$$a_{ij}\left[\delta_x^2 u_{ij} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j)\right] + b_{ij}\left[\delta_y^2 u_{ij} - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j)\right] + c_{ij}u_{ij} = f_{ij}, \tag{6.6}$$

where

$$\delta_x^2 u_{ij} = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2}, \quad \delta_y^2 u_{ij} = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{k^2},$$

$i = 1, 2, \ldots, n-1; j = 1, 2, \ldots, m-1, u_{ij} = u(x_i, y_j), a_{ij} = a(x_i, y_j), b_{ij} = b(x_i, y_j)$. Furthermore, $c_{ij} = c(x_i, y_j), f_{ij} = f(x_i, y_j)$, where $\xi_i \in (x_{i-1}, x_{i+1}), \eta_j \in (y_{j-1}, y_{j+1})$

denote the intermediate points and the boundary conditions take the following form:

$$\begin{aligned} u(0,y_j) &= \varphi_1(y_j) \quad \text{dla} \quad j=0,1,\ldots,m,\\ u(x_i,0) &= \varphi_2(x_i) \quad \text{dla} \quad i=0,1,\ldots,n-1,\\ u(\alpha,y_j) &= \varphi_3(y_j) \quad \text{dla} \quad j=0,1,\ldots,m,\\ u(x_i,\beta) &= \varphi_4(x_i) \quad \text{dla} \quad i=0,1,\ldots,n-1. \end{aligned} \tag{6.7}$$

Directly from equation (6.5) we obtain

$$\begin{aligned} a\frac{\partial^3 u}{\partial x^3} &= \frac{\partial f}{\partial x} - \frac{\partial a}{\partial x}\frac{\partial^2 u}{\partial x^2} - \frac{\partial b}{\partial x}\frac{\partial^2 u}{\partial y^2} - b\frac{\partial^3 u}{\partial x \partial y^2} - \frac{\partial c}{\partial x}u - c\frac{\partial u}{\partial x},\\ b\frac{\partial^3 u}{\partial y^3} &= \frac{\partial f}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial^2 u}{\partial x^2} - a\frac{\partial^3 u}{\partial x^2 \partial y} - \frac{\partial b}{\partial y}\frac{\partial^2 u}{\partial y^2} - \frac{\partial c}{\partial y}u - c\frac{\partial u}{\partial y} \end{aligned} \tag{6.8}$$

and

$$\begin{aligned} a\frac{\partial^4 u}{\partial x^4} &= \frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 a}{\partial x^2}\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial a}{\partial x}\frac{\partial^3 u}{\partial x^3} - \frac{\partial^2 b}{\partial x^2}\frac{\partial^2 u}{\partial y^2} - 2\frac{\partial b}{\partial x}\frac{\partial^3 u}{\partial x \partial y^2} - b\frac{\partial^4 u}{\partial x^2 \partial y^2}\\ &\quad - \frac{\partial^2 c}{\partial x^2}u - 2\frac{\partial c}{\partial x}\frac{\partial u}{\partial x} - c\frac{\partial^2 u}{\partial x^2},\\ b\frac{\partial^4 u}{\partial y^4} &= \frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 a}{\partial y^2}\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial a}{\partial y}\frac{\partial^3 u}{\partial x^2 \partial y} - a\frac{\partial^4 u}{\partial x^2 \partial y^2} - \frac{\partial^2 b}{\partial y^2}\frac{\partial^2 u}{\partial y^2} - 2b\frac{\partial b}{\partial y}\frac{\partial^3 u}{\partial y^3}\\ &\quad - \frac{\partial^2 c}{\partial y^2}u - 2\frac{\partial c}{\partial y}\frac{\partial u}{\partial y} - c\frac{\partial^2 u}{\partial y^2}. \end{aligned} \tag{6.9}$$

Taking into account equality (6.9) in equation (6.8), we have

$$\begin{aligned} a\frac{\partial^4 u}{\partial x^4} &= \frac{\partial^2 f}{\partial x^2} - \frac{2}{a}\frac{\partial a}{\partial x}\frac{\partial f}{\partial x}\\ &\quad - \left[\frac{\partial^2 a}{\partial x^2} - \frac{2}{a}\left(\frac{\partial a}{\partial x}\right)^2 + c\right]\frac{\partial^2 u}{\partial x^2} - \left(\frac{\partial^2 b}{\partial x^2} - \frac{2}{a}\frac{\partial a}{\partial x}\frac{\partial b}{\partial x}\right)\frac{\partial^2 u}{\partial y^2}\\ &\quad -2\left(\frac{\partial b}{\partial x} - \frac{b}{a}\frac{\partial a}{\partial x}\right)\frac{\partial^3 u}{\partial x \partial y^2} - b\frac{\partial^4 u}{\partial x^2 \partial y^2}\\ &\quad - \left(\frac{\partial^2 c}{\partial x^2} - \frac{2}{a}\frac{\partial a}{\partial x}\frac{\partial c}{\partial x}\right)u - 2\left(\frac{\partial c}{\partial x} - \frac{c}{a}\frac{\partial a}{\partial x}\right)\frac{\partial u}{\partial x} \end{aligned} \tag{6.10}$$

and

$$\begin{aligned} b\frac{\partial^4 u}{\partial y^4} &= \frac{\partial^2 f}{\partial y^2} - \frac{2}{b}\frac{\partial b}{\partial y}\frac{\partial f}{\partial y}\\ &\quad - \left(\frac{\partial^2 a}{\partial y^2} - \frac{2}{b}\frac{\partial a}{\partial y}\frac{\partial b}{\partial y}\right)\frac{\partial^2 u}{\partial x^2} - \left[\frac{\partial^2 b}{\partial y^2} - \frac{2}{b}\left(\frac{\partial b}{\partial y}\right)^2 + c\right]\frac{\partial^2 u}{\partial y^2}\\ &\quad -2\left(\frac{\partial a}{\partial y} - \frac{a}{b}\frac{\partial b}{\partial y}\right)\frac{\partial^3 u}{\partial x^2 \partial y} - a\frac{\partial^4 u}{\partial x^2 \partial y^2}\\ &\quad - \left(\frac{\partial^2 c}{\partial y^2} - \frac{2}{b}\frac{\partial b}{\partial y}\frac{\partial c}{\partial y}\right)u - 2\left(\frac{\partial c}{\partial y} - \frac{c}{b}\frac{\partial b}{\partial y}\right)\frac{\partial u}{\partial y}. \end{aligned} \tag{6.11}$$

Equation (6.10) should be considered at the intermediate point (ξ_i, y_j) and equation (6.11) at the intermediate point $((x_i, \eta_j)$. It is known that

$$\begin{gathered} b(\xi_i, y_j) = b_{ij} + O(h), \quad c(\xi_i, y_j) = c_{ij} + O(h), \quad \frac{1}{a(\xi_i, y_j)} = \frac{1}{a_{ij}} + O(h), \\ \frac{\partial^p \nu}{\partial x^p}(\xi_i, y_j) = \frac{\partial^p \nu}{\partial x^p}(x_i, y_j) + O(h) = \frac{\partial^p \nu_{ij}}{\partial x^p} + O(h), \\ a(x_i, \eta_j) = a_{ij} + O(k), \quad c(x_i, \eta_j) = c_{ij} + O(k), \quad \frac{1}{b(x_i, \eta_j)} = \frac{1}{b_{ij}} + O(k), \\ \frac{\partial^p \nu}{\partial y^p}(x_i, \eta_j) = \frac{\partial^p \nu}{\partial y^p}(x_i, y_j) + O(k) = \frac{\partial^p \nu_{ij}}{\partial x^p} + O(k) \end{gathered} \tag{6.12}$$

for $p = 1, 2$ i $\nu = a, b, c$. Moreover, we have

$$\begin{gathered} \frac{\partial u}{\partial x}(\xi_i, y_j) = \frac{\partial u}{\partial x}(x_i, y_j) + O(h) = \delta_x u_{ij} + O(h), \\ \frac{\partial^2 u}{\partial x^2}(\xi_i, y_j) = \frac{\partial^2 u}{\partial x^2}(x_i, y_j) + O(h) = \delta_x^2 u_{ij} + O(h), \\ \frac{\partial^2 u}{\partial y^2}(\xi_i, y_j) = \frac{\partial^2 u}{\partial y^2}(x_i, y_j) + O(h) = \delta_y^2 u_{ij} + O(k^2) + O(h), \\ \frac{\partial u}{\partial y}(x_i, \eta_j) = \frac{\partial u}{\partial y}(x_i, y_j) + O(k) = \delta_y u_{ij} + O(k), \\ \frac{\partial^2 u}{\partial y^2}(x_i, \eta_j) = \frac{\partial^2 u}{\partial y^2}(x_i, y_j) + O(k) = \delta_y^2 u_{ij} + O(k), \\ \frac{\partial^2 u}{\partial x^2}(x_i, \eta_j) = \frac{\partial^2 u}{\partial x^2}(x_i, y_j) + O(h) = \delta_x^2 u_{ij} + O(h^2) + O(k), \end{gathered} \tag{6.13}$$

where

$$\delta_x u_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad \delta_y u_{ij} = \frac{u_{i,j+1} - u_{i,j-1}}{2k}.$$

NE3C Method. Substituting equations (6.12) and (6.13) into equations (6.10) and (6.11), and then substituting the obtained results into equation (6.6), after simple transformations we obtain

$$\begin{gathered} \left(\frac{w_{1ij}}{h^2} - \frac{w_{3ij}}{2h}\right) u_{i-1,j} + \left(\frac{w_{2ij}}{k^2} - \frac{w_{4ij}}{2k}\right) u_{i,j-1} \\ - \left(2\frac{w_{1ij}}{h^2} + 2\frac{w_{2ij}}{k^2} - w_{5ij} - w_{6ij} - c_{ij}\right) u_{ij} \\ + \left(\frac{w_{2ij}}{k^2} + \frac{w_{4ij}}{2k}\right) u_{i,j+1} + \left(\frac{w_{1ij}}{h^2} + \frac{w_{3ij}}{2h}\right) u_{i+1,j} \\ = f_{ij} + w_{7ij} + O(h^3) + O(k^3) + O(h^2 k^2), \end{gathered} \tag{6.14}$$

where

$$\begin{gathered} w_{1ij} = a_{ij} + \frac{h^2}{12}\left[\frac{\partial^2 a_{ij}}{\partial x^2} - \frac{2}{a_{ij}}\left(\frac{\partial a_{ij}}{\partial x}\right)^2 + c_{ij}\right] + \frac{k^2}{12}\left(\frac{\partial^2 a_{ij}}{\partial y^2} - \frac{2}{b_{ij}}\frac{\partial a_{ij}}{\partial y}\frac{\partial b_{ij}}{\partial y}\right), \\ w_{2ij} = b_{ij} + \frac{h^2}{12}\left(\frac{\partial^2 b_{ij}}{\partial x^2} - \frac{2}{a_{ij}}\frac{\partial a_{ij}}{\partial x}\frac{\partial b_{ij}}{\partial x}\right) + \frac{k^2}{12}\left[\frac{\partial^2 b_{ij}}{\partial y^2} - \frac{2}{b_{ij}}\left(\frac{\partial b_{ij}}{\partial y}\right)^2 + c_{ij}\right], \\ w_{3ij} = \frac{h^2}{6}\left(\frac{\partial c_{ij}}{\partial x} - \frac{c_{ij}}{a_{ij}}\frac{\partial a_{ij}}{\partial x}\right), \quad w_{4ij} = \frac{k^2}{6}\left(\frac{\partial c_{ij}}{\partial y} - \frac{c_{ij}}{b_{ij}}\frac{\partial b_{ij}}{\partial y}\right), \end{gathered}$$

$$w_{5ij} = \frac{h^2}{12}\left(\frac{\partial^2 c_{ij}}{\partial x^2} - \frac{2}{a_{ij}}\frac{\partial a_{ij}}{\partial x}\frac{\partial c_{ij}}{\partial x}\right), \quad w_{6ij} = \frac{k^2}{12}\left(\frac{\partial^2 c_{ij}}{\partial y^2} - \frac{2}{b_{ij}}\frac{\partial b_{ij}}{\partial y}\frac{\partial c_{ij}}{\partial y}\right),$$

$$\begin{aligned} w_{7ij} = \frac{h^2}{12}\Bigg[&\frac{\partial^2 f}{\partial x^2}(\xi_i, y_j) - \frac{2}{a_{ij}}\frac{\partial a_{ij}}{\partial x}\frac{\partial f}{\partial x}(\xi_i, y_j) \\ &-2\left(\frac{\partial b_{ij}}{\partial x} - \frac{b_{ij}}{a_{ij}}\frac{\partial a_{ij}}{\partial x}\right)\frac{\partial^3 u}{\partial x \partial y^2}(\xi_i, y_j) - b_{ij}\frac{\partial^4 u}{\partial x^2 \partial y^2}(\xi_i, y_j)\Bigg] \\ +\frac{k^2}{12}\Bigg[&\frac{\partial^2 f}{\partial y^2}(x_i, \eta_j) - \frac{2}{b_{ij}}\frac{\partial b_{ij}}{\partial y}\frac{\partial f}{\partial y}(x_i, \eta_j) \\ &-2\left(\frac{\partial a_{ij}}{\partial y} - \frac{a_{ij}}{b_{ij}}\frac{\partial b_{ij}}{\partial y}\right)\frac{\partial^3 u}{\partial x^2 \partial y}(x_i, \eta_j) - a_{ij}\frac{\partial^4 u}{\partial x^2 \partial y^2}(x_i, \eta_j)\Bigg]. \end{aligned}$$

6.2.2. Interval methods

The methods described in this section are an extension of the classical method presented in the previous section. They are designed for ordinary and directed interval arithmetic, and for the purpose of further references are denoted by the abbreviations INE3 and DINE3, respectively.

From equation (6.14) we can obtain the interval method. Suppose that

$$\left|\frac{\partial^4 u}{\partial x^2 \partial y^2}(x, y)\right| \le M, \quad \left|\frac{\partial^3 u}{\partial x^2 \partial y}(x, y)\right| \le P, \quad \left|\frac{\partial^3 u}{\partial x \partial y^2}(x, y)\right| \le Q$$

for all points (x, y) lying in area Ω and let $\Psi_1(X, Y)$, $\Psi_2(X, Y)$, $\Xi_1(X, Y)$, $\Xi_2(X, Y)$ denote the interval expansions of the functions $\frac{\partial f}{\partial x}(x, y)$, $\frac{\partial^2 f}{\partial x^2}(x, y)$, $\frac{\partial f}{\partial y}(x, y)$, $\frac{\partial^2 f}{\partial y^2}(x, y)$, respectively. Then

$$\frac{\partial^4 u}{\partial x^2 \partial y^2}(x, y) \in [-M, M], \quad \frac{\partial^3 u}{\partial x^2 \partial y}(x, y) \in [-P, P], \quad \frac{\partial^3 u}{\partial x \partial y^2}(x, y) \in [-Q, Q]$$

for each point (x, y) and

$$\begin{aligned} &\frac{\partial f}{\partial x}(\xi_i, y_j) \in \Psi_1\left(X_i + [-h, h], Y_j\right), \quad &&\frac{\partial^2 f}{\partial x^2} \in \Psi_2\left(X_i + [-h, h], Y_j\right), \\ &\frac{\partial f}{\partial y}(x_i, \eta_j) \in \Xi_1\left(X_i, Y_j + [-k, k]\right), \quad &&\frac{\partial^2 f}{\partial y^2} \in \Xi_2\left(X_i, Y_j + [-k, k]\right), \end{aligned}$$

since $\xi_i \in (x_i - h, x_i + h)$ and $\eta_j \in (y_j - k, y_j + k)$. Thus, we have $w_{7ij} \in W_{7ij}$, where

$$\begin{aligned} W_{7ij} = \frac{h^2}{12}\Bigg\{&\Psi_2(X_i + [-h, h], Y_j) - \frac{2}{A_{ij}} D_x A_{ij} \Psi_1(X_i + [-h, h], Y_j) \\ &-2\left(D_x B_{ij} - \frac{B_{ij}}{A_{ij}} D_x A_{ij}\right)[-P, P] - B_{ij}[-M, M]\Bigg\} \\ +\frac{k^2}{12}\Bigg\{&\Xi_2(X_i, Y_j + [-k, k]) - \frac{2}{B_{ij}} D_y B_{ij} \Xi_1(X_i, Y_j + [-k, k]) \\ &-2\left(D_y A_{ij} - \frac{A_{ij}}{B_{ij}} D_y B_{ij}\right)[-Q, Q] - A_{ij}[-M, M]\Bigg\} \end{aligned} \tag{6.15}$$

and where V_{ij} and $D_z V_{ij}$ for $V = A, B$ and $z = x, y$ denote the interval expansions of the quantities ν_{ij} and $\frac{\partial \nu_{ij}}{\partial z}$ for $\nu = a, b$, respectively.

INE3C Method. If we denote the interval expansions of the quantities c_{ij} and w_{pij} by C_{ij} and W_{pij}, $(p = 1, 2, \ldots, 6)$, then from the above considerations, as well as from

relation (6.14), an interval method of the form

$$\begin{aligned}
&\left(\frac{W_{1ij}}{h^2}-\frac{W_{3ij}}{2h}\right)U_{i-1,j}+\left(\frac{W_{2ij}}{k^2}-\frac{W_{4ij}}{2k}\right)U_{i,j-1}\\
&-\left(\frac{2W_{1ij}}{h^2}+\frac{2W_{2ij}}{k^2}-W_{5ij}-W_{6ij}-C_{ij}\right)U_{ij}\\
&+\left(\frac{W_{2ij}}{k^2}+\frac{W_{4ij}}{2k}\right)U_{i,j+1}+\left(\frac{W_{1ij}}{h^2}+\frac{W_{3ij}}{2h}\right)U_{i+1,j}\\
&=F_{ij}+W_{7ij}+[-\delta,\delta],\quad i=1,2,\ldots,n-1,\quad j=1,2,\ldots,m-1,
\end{aligned}\tag{6.16}$$

where the interval $[-\delta,\delta]$, called δ-extension, is represented by the expression $O(h^3)+$ $+O(k^3)+O(h^2k^2)$ and where

$$\begin{gathered}
U_{0j}=\Phi_1(Y_j),\quad U_{i0}=\Phi_2(X_i),\quad U_{nj}=\Phi_3(Y_j),\quad U_{im}=\Phi_4(X_i),\\
j=0,1,\ldots,m,\quad i=1,2,\ldots,n-1.
\end{gathered}\tag{6.17}$$

Here by $\Phi_1(Y),\Phi_2(Y),\Phi_3(Y)$ and $\Phi_4(Y)$ we denote the interval expansions for the functions $\varphi_1(y),\varphi_2(y),\varphi_3(y)$ and $\varphi_4(y)$, respectively. The interval system of linear equations arising from formulas (6.16) and (6.17) can be solved using ordinary (proper) interval arithmetic since all intervals are proper.

DINE3C Method. The method in directed interval arithmetic differs only in the coefficient W_{7ij}, which in this case is written using the element opposite to $[-M,M]$, i.e.:

$$\begin{aligned}
\overline{W}_{7ij}=&\frac{h^2}{12}\left\{\Psi_2(X_i+[-h,h],Y_j)-\frac{2}{A_{ij}}D_xA_{ij}\Psi_1(X_i+[-h,h],Y_j)\right.\\
&\left.-2\left(D_xB_{ij}-\frac{B_{ij}}{A_{ij}}D_xA_{ij}\right)[-P,P]+B_{ij}[M,-M]\right\}\\
+&\frac{k^2}{12}\left\{\Xi_2(X_i,Y_j+[-k,k])-\frac{2}{B_{ij}}D_yB_{ij}\Xi_1(X_i,Y_j+[-k,k])\right.\\
&\left.-2\left(D_yA_{ij}-\frac{A_{ij}}{B_{ij}}D_yB_{ij}\right)[-Q,Q]+A_{ij}[M,-M]\right\}
\end{aligned}\tag{6.18}$$

It should also be noted that if no conclusions can be drawn about the values of M, P and Q based on the physical, mechanical properties or features of the problem under consideration, we propose to find these constants taking into account that

$$\begin{aligned}
\frac{\partial^4u}{\partial x^2\partial y^2}(x_i,y_j)&=\lim_{h\to0}\lim_{k\to0}\left(\frac{u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1}}{h^2k^2}\right.\\
&\left.+\frac{4u_{ij}-2(u_{i-1,j}+u_{i,j-1}+u_{i,j+1}+u_{i+1,j})}{h^2k^2}\right),\\
\frac{\partial^3u}{\partial x^2\partial y}&=\lim_{h\to0}\lim_{k\to0}\left(\frac{u_{i-1,j+1}-u_{i-1,j-1}-2(u_{i1,j+1}-u_{i1,j-1})+u_{i+1,j+1}-u_{i+1,j-1}}{2h^2k}\right),\\
\frac{\partial^3u}{\partial x\partial y^2}&=\lim_{h\to0}\lim_{k\to0}\left(\frac{u_{i+1,j-1}-u_{i-1,j-1}-2(u_{i+1,j}-u_{i-1,j})+u_{i+1,j+1}-u_{i-1,j+1}}{2hk^2}\right).
\end{aligned}$$

We can calculate

$$M_{nm} = \frac{1}{h^2k^2} \max_{\substack{i=1,2,\ldots,n-1\\ j=1,2,\ldots,m-1}} |u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1} - 4u_{ij} - 2(u_{i-1,j} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j})|,$$

$$P_{nm} = \frac{1}{2h^2k} \max_{\substack{i=1,2,\ldots,n-1\\ j=1,2,\ldots,m-1}} |u_{i-1,j+1} - u_{i-1,j-1} - 2(u_{i,j+1} - u_{i,j-1}) + u_{i+1,j+1} - u_{i+1,j-1}|,$$

$$Q_{nm} = \frac{1}{2hk^2} \max_{\substack{i=1,2,\ldots,n-1\\ j=1,2,\ldots,m-1}} |u_{i+1,j-1} - u_{i-1,j-1} - 2(u_{i+1,j} - u_{i-1,j}) + u_{i+1,j+1} - u_{i-1,j+1}|,$$

where the quantities u_{ij} were obtained using the classical method for different values of n and m. Then, we can plot the quantities M_{nm}, P_{nm} and Q_{nm} as a function of the values of n and m. The constants M, P and Q can be easily determined from the obtained graphs, since

$$\lim_{\substack{n\to\infty\\ m\to\infty}} M_{nm} \le M, \quad \lim_{\substack{n\to\infty\\ m\to\infty}} P_{nm} \le P, \quad \lim_{\substack{n\to\infty\\ m\to\infty}} Q_{nm} \le Q.$$

6.3. Methods for equations of the form $a\Delta u + cu = f$ with a larger number of error estimating constants

This section presents alternative interval methods for finding solutions of elliptic PDEs of the form expressed by equation (6.5). The methods described earlier were distinguished by the fact that in the construction of differential schemes attention was paid to minimizing the number of constants estimating the error of the method. However, one may ask the question whether increasing the number of constants estimating the error of the method has a significant influence on the obtained interval results? Hence, it was undertaken to develop a differential scheme, where more constants were used in the interval expansion than previously. The results of comparing the methods described here with methods with fewer error estimating constants are given in Chapter 7.

6.3.1. Classical method

Consider the equation

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2} + c \cdot u = f. \tag{6.19}$$

Constructing the differential scheme for the third order method requires obtaining approximations of the expressions $a\frac{\partial^4 u}{\partial x^4}$ and $b\frac{\partial^4 u}{\partial y^4}$. As in Section 5.1.1we will use the simplified notation for partial derivatives (see Equation (5.2)). If we include the error components, we can write the following formulas:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(x_i, y_j) &= \delta_x^2 - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i, y_j) - \frac{h^4}{360}\frac{\partial^6 u}{\partial x^6}(x_i, y_j) + O(h^6), \\ \frac{\partial^2 u}{\partial y^2}(x_i, y_j) &= \delta_y^2 - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, y_j) - \frac{k^4}{360}\frac{\partial^6 u}{\partial y^6}(x_i, y_j) + O(k^6). \end{aligned} \tag{6.20}$$

In order to simplify the notation in determining the differential diagram, we decided to omit the arguments (x, y) for each function. Taking into account formulas (6.20) in equation (6.5) we have

$$\begin{aligned} &a\left[\delta_x^2 - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4} - \frac{h^4}{360}\frac{\partial^6 u}{\partial x^6} + O(h^6)\right] \\ &+b\left[\delta_y^2 - \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4} - \frac{k^4}{360}\frac{\partial^6 u}{\partial y^6} + O(k^6)\right] + c\cdot u = f. \end{aligned} \tag{6.21}$$

After the first differentiation of formula (6.19) we obtain

$$\frac{\partial a}{\partial x}\cdot\frac{\partial^2 u}{\partial x^2} + a\frac{\partial^3 u}{\partial x^3} + \frac{\partial b}{\partial x}\cdot\frac{\partial^2 u}{\partial y^2} + b\frac{\partial^3 u}{\partial y^2\partial x} + \frac{\partial c}{\partial x}\cdot u + c\cdot\frac{\partial u}{\partial x} = \frac{\partial f}{\partial x}.$$

Differentiating again, we have

$$\begin{aligned} \frac{\partial^2 a}{\partial x^2}\cdot\frac{\partial^2 u}{\partial x^2} &+ \frac{\partial a}{\partial x}\cdot\frac{\partial^3 u}{\partial x^3} + \frac{\partial a}{\partial x}\frac{\partial^3 u}{\partial x^3} + a\frac{\partial^4 u}{\partial x^4} \\ &+ \frac{\partial^2 b}{\partial x^2}\cdot\frac{\partial^2 u}{\partial y^2} + \frac{\partial b}{\partial x}\cdot\frac{\partial^3 u}{\partial y^2\partial x} + \frac{\partial b}{\partial x}\frac{\partial^3 u}{\partial y^2\partial x} + b\frac{\partial^4 u}{\partial y^2\partial x^2} \\ &+ \frac{\partial^2 c}{\partial x^2}\cdot u + \frac{\partial c}{\partial x}\cdot\frac{\partial u}{\partial x} + \frac{\partial c}{\partial x}\cdot\frac{\partial u}{\partial x} + c\cdot\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 f}{\partial x^2}. \end{aligned}$$

So

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= \frac{\partial^2 a}{\partial x^2}\cdot\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial a}{\partial x}\cdot\frac{\partial^3 u}{\partial x^3} + a\frac{\partial^4 u}{\partial x^4} \\ &+ \frac{\partial^2 b}{\partial x^2}\cdot\frac{\partial^2 u}{\partial y^2} + 2\frac{\partial b}{\partial x}\cdot\frac{\partial^3 u}{\partial y^2\partial x} + b\frac{\partial^4 u}{\partial y^2\partial x^2} \\ &+ \frac{\partial^2 c}{\partial x^2}\cdot u + 2\frac{\partial c}{\partial x}\cdot\frac{\partial u}{\partial x} + c\cdot\frac{\partial^2 u}{\partial x^2} \end{aligned}$$

Proceeding similarly for the variable y we obtain

$$\begin{aligned} \frac{\partial^2 f}{\partial y^2} &= \frac{\partial^2 a}{\partial y^2}\cdot\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial a}{\partial y}\cdot\frac{\partial^3 u}{\partial x^2\partial y} + a\frac{\partial^4 u}{\partial x^2\partial y^2} \\ &+ \frac{\partial^2 b}{\partial y^2}\cdot\frac{\partial^2 u}{\partial y^2} + 2\frac{\partial b}{\partial y}\cdot\frac{\partial^3 u}{\partial y^3} + b\frac{\partial^4 u}{\partial y^4} \\ &+ \frac{\partial^2 c}{\partial y^2}\cdot u + 2\frac{\partial c}{\partial y}\cdot\frac{\partial u}{\partial y} + c\cdot\frac{\partial^2 u}{\partial y^2} \end{aligned}$$

Hence,

$$\begin{aligned} a\frac{\partial^4 u}{\partial x^4} &= \frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 a}{\partial x^2}\cdot\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial a}{\partial x}\cdot\frac{\partial^3 u}{\partial x^3} \\ &- \frac{\partial^2 b}{\partial x^2}\cdot\frac{\partial^2 u}{\partial y^2} - 2\frac{\partial b}{\partial x}\cdot\frac{\partial^3 u}{\partial y^2\partial x} - b\frac{\partial^4 u}{\partial y^2\partial x^2} \\ &- \frac{\partial^2 c}{\partial x^2}\cdot u - 2\frac{\partial c}{\partial x}\cdot\frac{\partial u}{\partial x} - c\cdot\frac{\partial^2 u}{\partial x^2}, \end{aligned} \tag{6.22}$$

$$\begin{aligned} b\frac{\partial^4 u}{\partial y^4} &= \frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 a}{\partial y^2}\cdot\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial a}{\partial y}\cdot\frac{\partial^3 u}{\partial x^2\partial y} \\ &- \frac{\partial^2 b}{\partial y^2}\cdot\frac{\partial^2 u}{\partial y^2} - 2\frac{\partial b}{\partial y}\cdot\frac{\partial^3 u}{\partial y^3} - a\frac{\partial^4 u}{\partial x^2\partial y^2} \\ &- \frac{\partial^2 c}{\partial y^2}\cdot u - 2\frac{\partial c}{\partial y}\cdot\frac{\partial u}{\partial y} - c\cdot\frac{\partial^2 u}{\partial y^2}. \end{aligned} \tag{6.23}$$

Using equations (6.22) oraz (6.23) in equation (6.21) and taking into account the upper error limits, based on relation (6.13), we obtain

$$\begin{aligned}
&a\delta_x^2 u - \frac{h^2}{12}\left[\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 a}{\partial x^2}\delta_x^2 u - 2\frac{\partial a}{\partial x}\delta_x^3 u\right.\\
&\quad - \frac{\partial^2 b}{\partial x^2}\delta_y^2 u - 2\frac{\partial b}{\partial x}\delta_y^2\delta_x u - b\delta_y^2\delta_x^2 u\\
&\quad \left. - \frac{\partial^2 c}{\partial x^2}u - 2\frac{\partial c}{\partial x}\delta_x u - c\delta_x^2 u + O(h)\right] +\\
+b\delta_y^2 u - \frac{k^2}{12}&\left[\frac{\partial^2 f}{\partial y^2} - \frac{\partial^2 a}{\partial y^2}\delta_x^2 u - 2\frac{\partial a}{\partial y}\delta_x^2\delta_y u - a\delta_x^2\delta_y^2 u\right.\\
&\quad - \frac{\partial^2 b}{\partial y^2}\delta_y^2 u - 2\frac{\partial b}{\partial y}\delta_x^3 u\\
&\quad \left. - \frac{\partial^2 c}{\partial y^2}u - 2\frac{\partial c}{\partial y}\delta_y u - c\delta_y^2 u + O(k)\right] + cu = f.
\end{aligned}$$

After further transformations we obtain the formula

$$\begin{aligned}
&\left(a\frac{h^2}{12}\frac{\partial^2 a}{\partial x^2} + \frac{h^2}{12}c + \frac{k^2}{12}\frac{\partial^2 a}{\partial y^2}\right)\delta_x^2 u\\
&+\left(b + \frac{k^2}{12}\frac{\partial^2 b}{\partial y^2} + \frac{k^2}{12}c + \frac{h^2}{12}\frac{\partial^2 b}{\partial x^2}\right)\delta_y^2 u\\
&+\left(\frac{h^2}{6}\frac{\partial a}{\partial x} + \frac{k^2}{6}\frac{\partial b}{\partial y}\right)\delta_x^3 u\\
&+\frac{h^2}{6}\frac{\partial b}{\partial x}\delta_y^2\delta_x u + \frac{k^2}{6}\frac{\partial a}{\partial y}\delta_x^2\delta_y u\\
&-\left(\frac{h^2}{12}b + \frac{k^2}{12}a\right)\delta_x^2\delta_y^2 u\\
&+\frac{k^2}{6}\frac{\partial c}{\partial x}\delta_x u + \frac{k^2}{6}\frac{\partial c}{\partial y}\delta_y u + c\cdot u\\
&=f + \frac{h^2}{12}\frac{\partial^2 f}{\partial x^2} + \frac{k^2}{12}\frac{\partial^2 f}{\partial y^2} + O(h^3) + O(k^3).
\end{aligned} \tag{6.24}$$

NE5C Method. Let us move the mixed derivative approximations to the right-hand side of equation (6.24), to include them as components of the method error. We obtain

$$\begin{aligned}
&\left(a\frac{h^2}{12}\frac{\partial^2 a}{\partial x^2} + \frac{h^2}{12}c + \frac{k^2}{12}\frac{\partial^2 a}{\partial y^2}\right)\delta_x^2 u + \left(b + \frac{k^2}{12}\frac{\partial^2 b}{\partial y^2} + \frac{k^2}{12}c + \frac{h^2}{12}\frac{\partial^2 b}{\partial x^2}\right)\delta_y^2 u\\
&+\frac{k^2}{6}\frac{\partial c}{\partial x}\delta_x u + \frac{k^2}{6}\frac{\partial x}{\partial y}\delta_y u + c\cdot u = f + \frac{h^2}{12}\frac{\partial^2 f}{\partial x^2} - 2\frac{h^2}{12}\frac{\partial a}{\partial x}\delta_x^3 u - 2\frac{h^2}{12}\frac{\partial b}{\partial x}\delta_y^2\delta_x u\\
&+\frac{h^2}{12}b\delta_x^2\delta_y^2 u + \frac{k^2}{12}\frac{\partial^2 f}{\partial y^2} - 2\frac{k^2}{12}\frac{\partial b}{\partial y}\delta_y^3 u - 2\frac{k^2}{12}\frac{\partial a}{\partial y}\delta_x^2\delta_y u\\
&+\frac{k^2}{12}a\delta_x^2\delta_y^2 u + O(h^3) + O(k^3).
\end{aligned} \tag{6.25}$$

We will use the above equation in the next section to derive an interval method that takes into account the truncation error. Let us define the following auxiliary coefficients:

$$
\begin{aligned}
w_1 &= a + \frac{h^2}{12}\frac{\partial^2 a}{\partial x^2} + \frac{h^2}{12}c + \frac{k^2}{12}\frac{\partial^2 a}{\partial y^2},\\
w_2 &= b + \frac{k^2}{12}\frac{\partial^2 b}{\partial y^2} + \frac{k^2}{12}c + \frac{h^2}{12}\frac{\partial^2 a}{\partial x^2},\\
w_3 &= \frac{h^2}{6}\frac{\partial c}{\partial x},\\
w_4 &= \frac{k^2}{6}\frac{\partial c}{\partial y},\\
w_5 &= \frac{h^2}{12}\frac{\partial^2 f}{\partial x^2} - 2\frac{h^2}{12}\frac{\partial a}{\partial x}\delta_x^3 u - 2\frac{h^2}{12}\frac{\partial b}{\partial x}\delta_y^2\delta_x u\\
&\quad + \frac{h^2}{12}b\delta_x^2\delta_y^2 u + \frac{k^2}{12}\frac{\partial^2 f}{\partial y^2} - 2\frac{k^2}{12}\frac{\partial b}{\partial y}\delta_y^3 u - 2\frac{k^2}{12}\frac{\partial a}{\partial y}\delta_x^2\delta_y u + \frac{k^2}{12}a\delta_x^2\delta_y^2 u
\end{aligned}
$$

Then equation (6.25) at the grid point (x_i, y_j) can be written in the form

$$
w_1\delta_x^2 u_{ij} + w_2\delta_y^2 u_{ij} + w_3\delta_x u_{ij} + w_4\delta_y u_{ij} + c_{ij}\cdot u_{ij} = f_{ij} + w_{5ij} + O(h^3) + O(k^3),
$$

which, considering equation (5.2), gives

$$
\begin{aligned}
&w_1\left(\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2}\right) + w_2\left(\frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{k^2}\right)\\
&+ w_3\left(\frac{u_{i+1,j} + u_{i-1,j}}{2h}\right) + w_4\left(\frac{u_{i,j+1} + u_{i,j-1}}{2k}\right)\\
&+ c_{ij}\cdot u_{ij} = f_{ij} + w_{5ij} + O(h^3) + O(k^3).
\end{aligned}
\tag{6.26}
$$

Finally, the form of the third order scheme for equation (6.19) is as follows:

$$
\begin{aligned}
&\left(c_{i,j} - \frac{2w_1}{h^2} - \frac{2w_2}{k^2}\right)u_{ij} + \left(\frac{w_1}{h^2} + \frac{w_3}{2h}\right)u_{i+1,j} + \left(\frac{w_1}{h^2} - \frac{w_3}{2h}\right)u_{i-1,j}\\
&+ \left(\frac{w_2}{k^2} + \frac{w_4}{2k}\right)u_{i,j+1} + \left(\frac{w_2}{k^2} - \frac{w_4}{2k}\right)u_{i,j-1} = f_{ij} + w_{5ij} + O(h^3) + O(k^3).
\end{aligned}
\tag{6.27}
$$

6.3.2. Interval methods

Let us introduce the following constants, which are estimates for the errors:

$$
\begin{aligned}
P &= \lim_{h\to 0}\lim_{k\to 0}|\delta_x^2\delta_y u|,\\
Q &= \lim_{k\to 0}\lim_{h\to 0}|\delta_y^2\delta_x u|,\\
R &= \lim_{h\to 0}|\delta_x^3 u|,\\
S &= \lim_{h\to 0}|\delta_y^3 u|,\\
T &= \lim_{h\to 0}\lim_{k\to 0}|\delta_x^2\delta_y^2 u|.
\end{aligned}
$$

As in Section 6.2.2, let us make the assumptions that

$$
\left|\frac{\partial^3 u}{\partial x^2\partial y}(x,y)\right| \le P, \quad \left|\frac{\partial^3 u}{\partial x\partial y^2}(x,y)\right| \le Q, \quad \left|\frac{\partial^4 u}{\partial x^2\partial y^2}(x,y)\right| \le T,
$$
$$
\left|\frac{\partial^3 u}{\partial x^3}(x,y)\right| \le R, \quad \left|\frac{\partial^3 u}{\partial y^3}(x,y)\right| \le S
$$

for all points (x, y) lying in area Ω and let $\Psi(X, Y)$, $\Xi(X, Y)$, denote the interval expansions of the functions $\frac{\partial^2 f}{\partial x^2}(x, y)$ and $\frac{\partial^2 f}{\partial y^2}(x, y)$. Then

$$\frac{\partial^3 u}{\partial x^2 \partial y}(x, y) \in [-P, P], \quad \frac{\partial^3 u}{\partial x \partial y^2}(x, y) \in [-Q, Q], \quad \frac{\partial^4 u}{\partial x^2 \partial y^2}(x, y) \in [-T, T]$$
$$\frac{\partial^3 u}{\partial x^3}(x, y) \in [-R, R], \quad \frac{\partial^3 u}{\partial y^3}(x, y) \in [-S, S]$$

for each point $(x, y) \in \Omega$ and

$$\frac{\partial^2 f}{\partial x^2} \in \Psi\left(X_i + [-h, h], Y_j\right), \quad \frac{\partial^2 f}{\partial y^2} \in \Xi\left(X_i, Y_j + [-k, k]\right),$$

since $\xi_i \in (x_i - h, x_i + h)$ oraz $\eta_j \in (y_j - k, y_j + k)$. Thus, we have $w_{5ij} \in W_{5ij}$, where

$$W_{5ij} = \frac{h^2}{12}\Psi\left(X_i + [-h, h], Y_j\right) - 2\frac{h^2}{12}D_x A_{ij}[-R, R] - 2\frac{h^2}{12}D_x B_{ij}[-Q, Q] + \frac{h^2}{6}B_{ij}[-T, T]$$
$$+\frac{k^2}{12}\Xi\left(X_i, Y_j + [-k, k]\right) - 2\frac{k^2}{12}D_y B_{ij}[-S, S] - 2\frac{k^2}{12}D_y A_{ij}[-P, P]. \tag{6.28}$$

and where V_{ij} and $D_z V_{ij}$ for $V = A, B$ and $z = x, y$ enote the interval expansions of the quantities ν_{ij} and $\frac{\partial \nu_{ij}}{\partial z}$ for $\nu = a, b$, respectively.

INE5C Method. Let us denote the interval expansions of the quantities c_{ij}, v_{zij} and w_{zij} by C_{ij}, V_{zij} and W_{pij}, $(z = x, y)$, respectively. Then, considering relation (6.14), we obtain the form of the scheme of order three for equation (6.5). It is as follows:

$$\left(C_{i,j} - \frac{2W_{1ij}}{h^2} - \frac{2W_{2ij}}{k^2}\right)U_{i,j} + \left(\frac{W_{1ij}}{h^2} + \frac{W_{3ij}}{2h}\right)U_{i+1,j} + \left(\frac{W_{1ij}}{h^2} - \frac{W_{3ij}}{2h}\right)u_{i-1,j}$$
$$+\left(\frac{W_{2ij}}{k^2} + \frac{W_{4ij}}{2k}\right)u_{i,j+1} + \left(\frac{W_{2ij}}{k^2} - \frac{W_{4ij}}{2k}\right)u_{i,j-1} = F_{ij} + W_{ij} + O(h^3) + O(k^3). \tag{6.29}$$

where

$$U_{0j} = \Phi_1(Y_j), \quad U_{i0} = \Phi_2(X_i), \quad U_{nj} = \Phi_3(Y_j), \quad U_{im} = \Phi_4(X_i),$$
$$j = 0, 1, \ldots, m, \quad i = 1, 2, \ldots, n-1. \tag{6.30}$$

Here by $\Phi_1(Y), \Phi_2(Y), \Phi_3(Y)$ and $\Phi_4(Y)$ we denote the interval expansions for the functions $\varphi_1(y), \varphi_2(y), \varphi_3(y)$ and $\varphi_4(y)$, respectively. The interval system of linear equations arising from formulas (6.16) and (6.30) can be solved by ordinary (proper) interval arithmetic, since all intervals are proper.

DINE5C Method. he FDM method in directed interval arithmetic differs only in the coefficient W_{ij}, which we replace, using the existence of opposite elements, by the coefficient

$$\overline{W}_{ij} = \frac{h^2}{12}\Psi\left(X_i + [-h, h], Y_j\right) - 2\frac{h^2}{12}D_x A_{ij}[R, -R] - 2\frac{h^2}{12}D_x B_{ij}[Q, -Q] + \frac{h^2}{6}B_{ij}[T, -T]$$
$$+\frac{k^2}{12}\Xi\left(X_i, Y_j + [-k, k]\right) - 2\frac{k^2}{12}D_y B_{ij}[S, -S] - 2\frac{k^2}{12}D_y A_{ij}[P, -P].$$

All methods described in Chapters 5 and 6 were tested and compared with one other, wherever possible, in Chapter 7.

7
Computational experiments

This chapter presents results obtained by means of the interval methods described earlier. The individual examples start from the simplest, classical form of the Poisson equation, through its generalization, up to a certain class of elliptic equations of the form $a\Delta u + cu = f$. The first and second examples show the application of interval methods of second (IPE, DIPE) and fourth (IPE4, DIPE4) order to finding estimates for solutions of the Poisson equation. The third example presents the interval methods for the generalized form of Poisson's equation (IGPE, DIGPE methods). Then, in the fourth example it is shown that for this type of equation interval methods developed for the mentioned class of equations can be used (assuming the relevant parameters). The results obtained by these methods are compared with the methods of the third example. The last two examples are intended to compare the methods described in Chapters 5 and 6 with the method proposed by Nakao - described in Chapter 4. They are referred to here for two reasons: first, to show that Nakao's method is not applicable to equations such as the Poisson equation (PE) and the generalized Poisson equation (GPE). Second, to show that the interval methods constructed in the manner described in this work make it possible to efficiently find correct estimates for the equations considered by Nakao (NE) and, what is more, that these estimates are more accurate than those we can obtain with Nakao's interval method.

Table 7.1. Running environment parameters

Parameter	Value
Operating system	Debian GNU/Linux 10 (buster)
Processor	Intel Xeon E312xx 2.0 [GHz] x 10
RAM	64 [GB]
Compiler	GCC v. 8.3.0
Version of the .boost	1.60
MPFR++ library version	3.6.8

In each example, the errors of the method were estimated experimentally using constants and their designations are summarized in Table 7.2.

All the examples given here use the implementation of interval arithmetic in C++ [64]. The exception is example 4, for which results are presented directly from the publication of the author of this paper (see [30]), and whose implementation was performed using

Table 7.2. Designation of error estimation constants for each method.

Method	Order of the method	Constants for estimating errors
PE	2	$M \geq \left\lvert \frac{\partial^4 u}{\partial x^2 \partial y^2} \right\rvert$
GPE	2	$M \geq \left\lvert \frac{\partial^4 u}{\partial x^4} \right\rvert, N \geq \left\lvert \frac{\partial^4 u}{\partial y^4} \right\rvert$
PE4	4	$P \geq \left\lvert \frac{\partial^6 u}{\partial x^4 \partial y^2} \right\rvert, Q \geq \left\lvert \frac{\partial^6 u}{\partial x^2 \partial y^4} \right\rvert$
NE3C	4	$M \geq \left\lvert \frac{\partial^4 u}{\partial x^x \partial y^2} \right\rvert, P \geq \left\lvert \frac{\partial^3 u}{\partial x^2 \partial y} \right\rvert, Q \geq \left\lvert \frac{\partial^3 u}{\partial x \partial y^2} \right\rvert$
NE5C	4	$P \geq \left\lvert \frac{\partial^3 u}{\partial x^2 \partial y} \right\rvert, Q \geq \left\lvert \frac{\partial^3 u}{\partial x \partial y^2} \right\rvert, R \geq \left\lvert \frac{\partial^3 u}{\partial x^3} \right\rvert, S \geq \left\lvert \frac{\partial^3 u}{\partial y^3} \right\rvert, T \geq \left\lvert \frac{\partial^4 u}{\partial x^x \partial y^2} \right\rvert$

the *IntervalArithmetic* module (see [62]). The parameters of the runtime environment are shown in Table 7.1.

Each of the interval methods presented in the thesis was tested according to the algorithm 7.7. Since the purpose of the experiments was only to compare the quality of the results obtained, depending on the method and type of arithmetic, therefore in the implementation of the procedure $SolveIntervalSystem(\boldsymbol{A}, \boldsymbol{b})$ only one, the same algorithm for solving the system of linear equations was used. The Gauss-Jordan elimination method with full selection of the basis element was chosen (see [66]). However, there is no obstacle to the use of other known methods for solving systems of linear equations. Nevertheless, for comparing the effectiveness of the different methods it is crucial that it is the same procedure for all of them.

Algorytm 7.7. Interval methods - test procedure

1: {the function $f = f(x, y)$ is on the right side of each equation}
2: {variable *params* stores the vector of functions that are the parameters of the equation, located on the left-hand side, depending on the example includes the functions: $a_1 = a_1(x, y)$, $a_2 = a_2(x, y)$ and $c = c(x, y)$}
3: {parameter *bc*stores the boundary conditions for the given BVP problem}
4: $f, params, bc := GetInitialDataForGivenExample(exampleId)$
5: {variable *errorContants* stores error estimation constants for the method, see Table 7.2}
6: $errorContants := GetErrorBoundsForMethod(methodId, m, f, params, bc)$
7: $results \leftarrow []$
8: **for** $m := 10, 20 \ldots, 100$ **do**
9: $\quad \boldsymbol{A}, \boldsymbol{b} := BuildIntervalSystem(methodId, f, bc, errorConstants)$
10: $\quad tmp \leftarrow SolveIntervalSystem(\boldsymbol{A}, \boldsymbol{b})$
11: $\quad results \leftarrow Append(results, tmp)$
12: **end for**
13: {the result is a set of results for a given method and different grid sizes}
14: **return** *results*

Example 1

Consider the following boundary issue:

$$\frac{\partial^2 u}{\partial x^2}(x,y)+\frac{\partial^2 u}{\partial y^2}(x,y)=0,\quad 0\le x\le 1,\ \ 0\le y\le 1,$$
$$u|_\Gamma(x,y)=\varphi(x,y)=\begin{cases}\varphi_1(y)=\cos(3y)\ \text{ dla } x=0,\\ \varphi_2(x)=\exp(3x)\ \text{ dla } y=0,\\ \varphi_3(y)=\exp(3)\cos(3y)\ \text{ dla } x=1,\\ \varphi_4(x)=\exp(3x)\cos(3)\ \text{ dla } y=1.\end{cases} \tag{7.1}$$

This is the so-called *Laplace equation* and it is the simplest example of an elliptic equation of the form (2.6), denoted in this paper by the abbreviation PE. The exact solution has the form $u(x,y)=\exp(3x)\cos(3y)$.

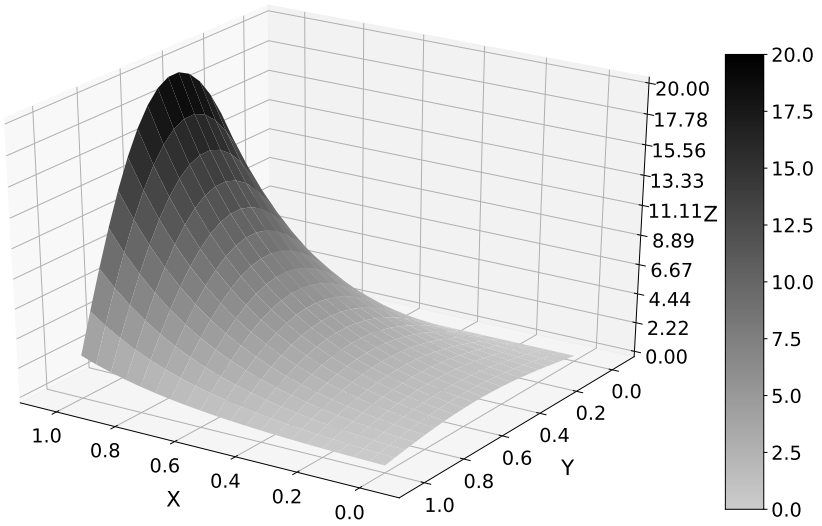

Figure 7.1. Exact solution for problem (7.1)

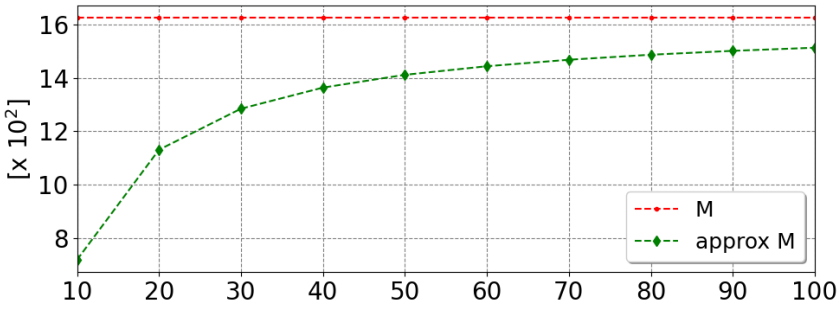

Figure 7.2. Approximations of the constant M nd its exact value in the methods of order two for problem (7.1)

Table 7.3. Interval solutions and interval widths obtained by IPE and IPE4 methods in proper (U_p) and by DIPE and DIPE4 methods in directed (U_d) interval arithmetic for problem (7.1) at point $(0.5, 0.5)$, $u_{exact}(0.5, 0.5) \approx 0.31702214358044366$

$m = n$	$U_p(0.5, 0.5)$	Szerokość	$U_d(0.5, 0.5)$	Szerokość
20 (PE)	[0.26795781801796551, 0.36764778128690462]	0.099689963	[0.26795781801796628, 0.36764778128690385]	0.099689963
20 (PE4)	[0.31687231501883790, 0.31717197371330709]	0.000299659	[0.31687231501883870, 0.31717197371330630]	0.000299659
60 (PE)	[0.31156101681974879, 0.32265704145441798]	0.011096024	[0.31156101681975913, 0.32265704145440782]	0.011096024
60 (PE4)	[0.31702029383932179, 0.31702399332372090]	0.000003700	[0.31702029383933359, 0.31702399332370710]	0.000003699
100 (PE)	[0.31505586246198510, 0.31905099073825598]	0.003995128	[0.31505586246202793, 0.31905099073821316]	0.003995128
100 (PE4)	[0.31702190385388648, 0.31702238330710142]	0.000000048	[0.31702212784104150, 0.31702215931994640]	0.000000031

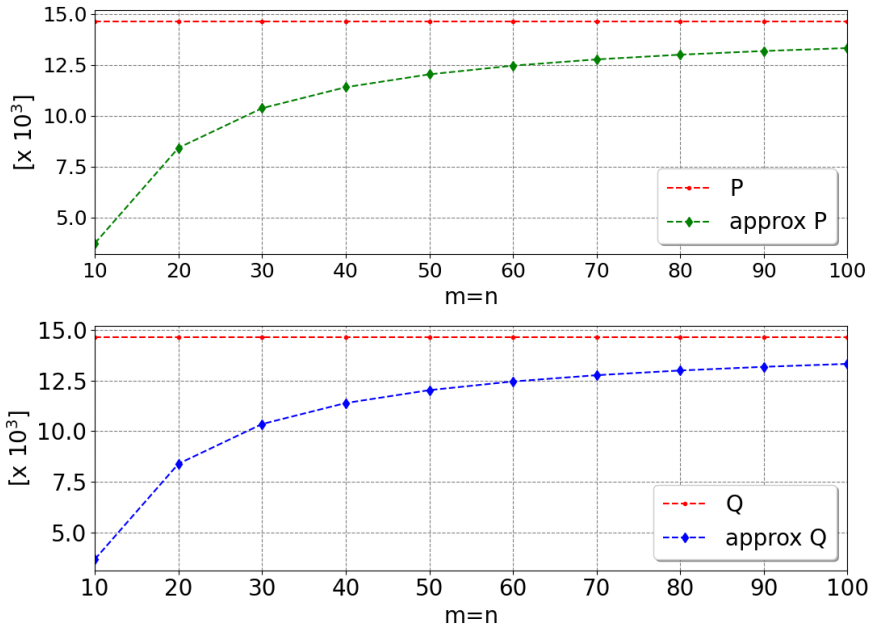

Figure 7.3. Approximations of the constant P and Q and their exact values adopted in fourth order methods for problem (7.1)

The use of adequate interval methods (IPE, IPE4, DIPE, DIPE4) to address this issue requires an initial estimate of the method errors for each method. As proposed in Section 5.1.2 this can be done experimentally. The fact that we know the exact solution, and thus can determine the exact value of the estimate, will be used to verify the approximations obtained. This is shown in Figures 7.2 and 7.3. In the second-order methods, we assumed $M = 1627$ – this value was obtained from the known exact solution, but note that a similar value (estimate) of this constant can be obtained from the graph shown in Figure 7.2. In the fourth-order methods, we assumed $P = Q = 14643$. W In this case, the experimental estimation method can also be used, as can be seen in Figure 7.3.

Conclusions. Table 7.3 shows the results obtained by second and fourth order methods in ordinary and directed interval arithmetic. In the example considered, the benefits of using directed interval arithmetic were very limited, as the differences in the width of the resulting intervals obtained, for a given grid size, were insignificant, even negligible. There was a noticeable benefit from increasing the order of the method - clearly better results for IPE4 than for IPE and DIPE4 than for DIPE. As can be seen in Figs. 7.2 and 7.3 the experimental estimation technique is effective here and the approximations of the constants (i.e. $approxM$, $approxQ$, $approxP$) converge to the exact values obtained analytically. In principle, if the estimates of M, P and Q cannot be obtained from any data on the problem under consideration, we can use the technique presented here – determine the estimates from the values obtained in floating point arithmetic. Let us also note that for both methods, the exact solution is contained in the obtained interval solutions. ■

Example 2

As a second example, consider the following problem:

$$\begin{gathered}\frac{\partial^2 u}{\partial x^2}(x,y)+\frac{\partial^2 u}{\partial y^2}(x,y)=-2\pi^2\sin(\pi x)\sin(\pi y),\quad 0\le x\le 1,\ \ 0\le y\le 1,\\ u|_\Gamma(x,y)=0\end{gathered}\tag{7.2}$$

with exact solution $u(x,y)=\sin(\pi x)\sin(\pi y)$.

Note that the problem 7.2, like the previous one, belongs to elliptic equations of the form PE, i.e. equations given by the general formula (2.6). However, in comparison with the previous example, the right-hand side of the equation has changed – the function $f=f(x,y)$ constitutes here a new parameter of the equation. The resulting interval solution is shown in Table 7.4.

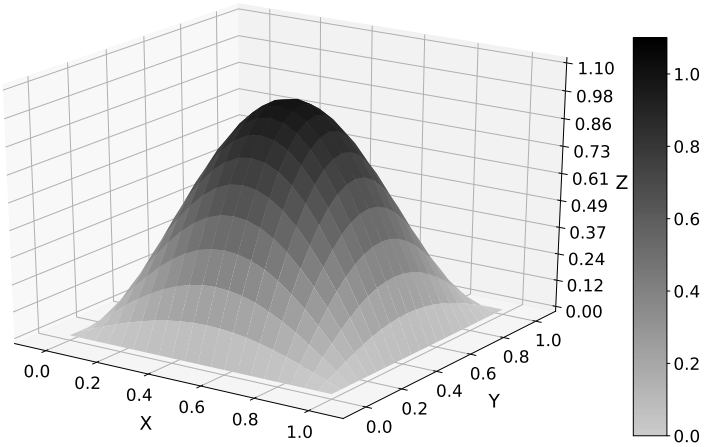

Figure 7.4. Exact solution for problem (7.2)

Table 7.4. Interval solutions and interval widths obtained by IPE and IPE4 methods in proper (U_p) and by DIPE and DIPE4 methods in directed (U_d) interval arithmetic for problem (7.2) at point (0.5, 0.5) ($u_{exact}(0.5, 0.5) = 1$)

$m = n$	$U_p(0.5, 0.5)$	Szerokość	$U_d(0.5, 0.5)$	Szerokość
20 (PE)	[0.9943031722943299, 1.0032966998827956]	0.008993528	[0.9972920186287353, 1.0003078535483902]	0.003015835
20 (PE4)	[0.9999825795708284, 1.0000059858600965]	0.000023406	[0.9999863114853876, 1.0000022539455373]	0.000015942
60 (PE)	[0.9993877757476903, 1.0001910675167757]	0.000803292	[0.9995227144656405, 1.0000561287988255]	0.000533414
60 (PE4)	[0.9999997879937084, 1.0000000498551452]	0.000000262	[0.9999998069620024, 1.0000000308868512]	0.000000217
100 (PE)	[0.9997852097215218, 1.0000562138028589]	0.000259718	[0.9998155730093303, 1.0000258505150504]	0.000210278
100 (PE4)	[0.9999999728030543, 1.0000000058410797]	0.000000033	[0.9999999743621285, 1.0000000042820054]	0.000000030

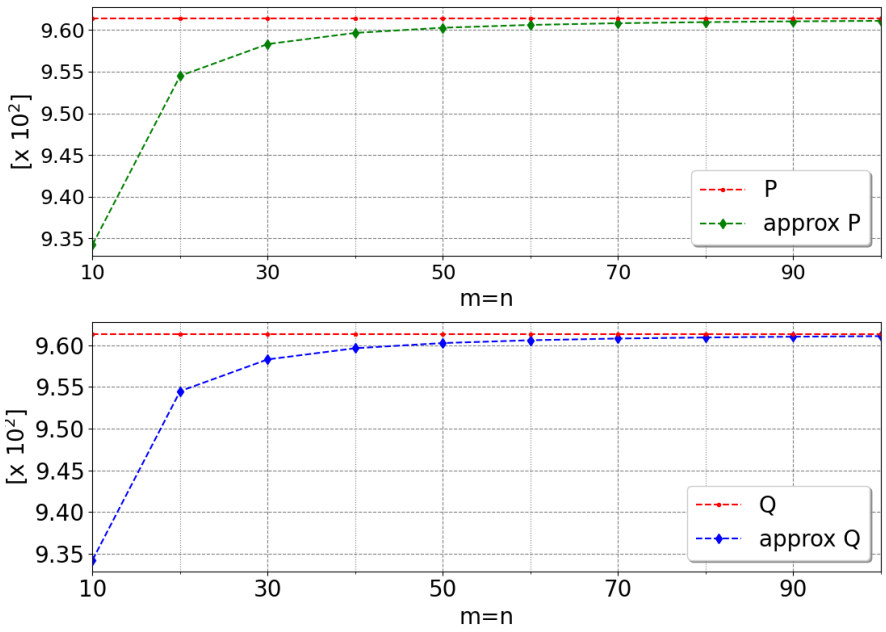

Figure 7.5. Approximations of the constants P and Q and their exact values assumed in fourth-order methods for problem (7.2)

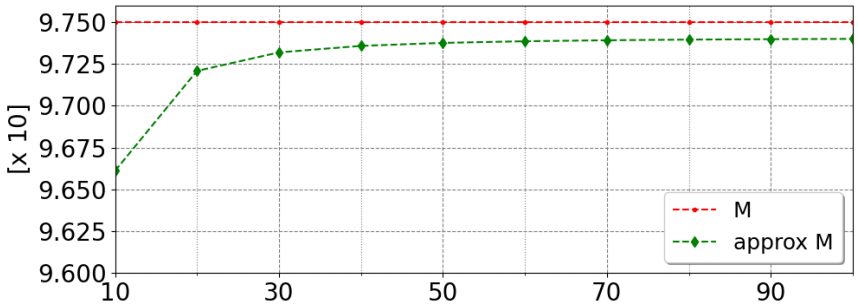

Figure 7.6. Approximations of the constant M and its exact value assumed in the methods of order two for problem (7.2)

Table 7.5. Interval solutions and interval widths obtained by IPE and IPE4 methods in proper (U_p) and by DIPE and DIPE4 methods in directed (U_d) interval arithmetic for problem (7.2) at point $(0.5, 0.5)$ for $M = 100$ i $P = Q = 1000$

$m = n$	$U_p(0.5, 0.5)$	$Width(U_p)$	$U_d(0.5, 0.5)$	$Width(U_d)$
20 (PE)	[0.9942265819722118, 1.0033732902049137]	0.009146708	[0.9972460644354644, 1.0003538077416611]	0.003107743
20 (PE4)	[0.9999821846099048, 1.0000063808210200]	0.000024196	[0.9999859165244640, 1.0000026489064608]	0.000016732
60 (PE)	[0.9993792508363238, 1.0001995924281422]	0.000820342	[0.9995175995188206, 1.0000612437456454]	0.000543644
60 (PE4)	[0.9999997831176551, 1.0000000547311985]	0.000000264	[0.9999998020859490, 1.0000000357629046]	0.000000234
100 (PE)	[0.9997821403236452, 1.0000592832007355]	0.000277143	[0.9998137313706043, 1.0000276921537746]	0.000213961
100 (PE4)	[0.9999999721711177, 1.0000000064730162]	0.000000034	[0.9999999737301920, 1.0000000049139419]	0.000000031

Conclusions. Analogously to the previous case, we had to determine the error estimates of the method, this time for problem (7.2). We assumed that $M = 97.5$ or the method of the second order (see 7.6) and $P = Q = 961.4$ for the method of the fourth order (see 7.5). Importantly, as in Example 1, the exact solution falls inside the obtained interval solution. Note that if we slightly overestimate the constants M or P and Q (which can happen if there is no data with appropriate partial derivatives or if, despite increasing the grid size, the increments of the derivatives are admittedly decreasing but are still relatively high) the interval results will change slightly. In Table 7.5 we present the results obtained by (D)IPE and (D)IPE4 methods for problem (7.2) with values $M = 100$ and $P = Q = 1000$. Let us also note that for the given problem the resulting intervals are much narrower in directed interval arithmetic, which indicates the desirability of its use. It can be assumed that it is influenced by the fact that the problem 7.2has an additional, non-zero parameter – i.e. the function $f = f(x, y)$, the inclusion of which in the calculations results in greater inaccuracies of the estimation obtained in proper interval arithmetic than in directed one.. ■

Example 3

Consider an equation that is a generalization of equation (7.2):

$$\begin{aligned} xye^y\frac{\partial^2 u}{\partial x^2}(x,y)+xye^x\frac{\partial^2 u}{\partial y^2}(x,y) = &-\pi^2 xy(e^y\sin(\pi x)\sin(\pi y)\\ &+e^x\sin(\pi x)\sin(\pi y)),\\ 0\le x\le 1,\ \ 0\le &y\le 1,\\ u|_\Gamma(x,y) &= 0 \end{aligned} \tag{7.3}$$

with the same exact solution as before $u(x,y)=\sin(\pi x)\sin(\pi y)$. Note, however, that the partial derivatives $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$ are preceded by the functions $a(x,y)=xye^y$ and $b(x,y)=xye^x$, respectively. Moreover, the function $f(x,y)$ has a more complicated form, as is shown in Figure 7.7.

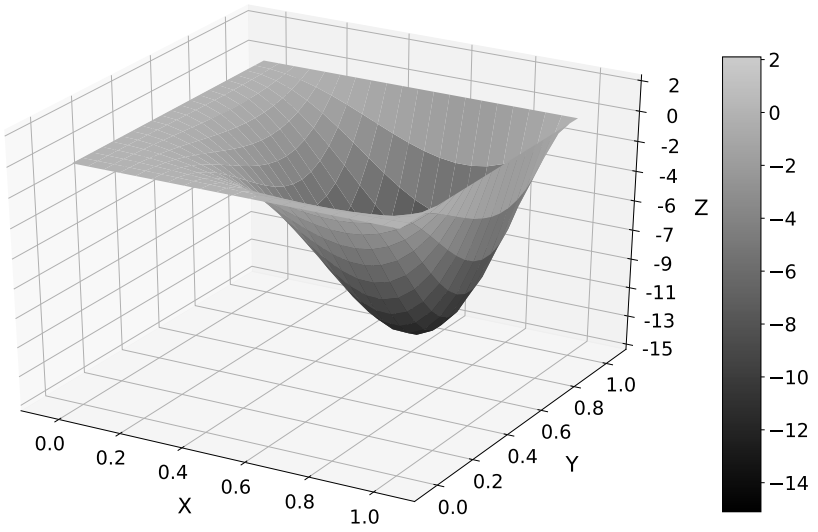

Figure 7.7. The function $f(x,y)=-\pi^2 xy(e^y\sin(\pi x)\sin(\pi y)+e^x\sin(\pi x)\sin(\pi y))$ for problem given by Equation (7.3)

Obviously, it is not possible to find solutions to this equation using methods PE and PE4, which were used in examples 1 and 2. However, three other methods, described in the previous chapters, can be used here, i.e. GPE, NE3C and NE5C methods. It is essential that only one of them is dedicated strictly to equations such as equation (7.3) i.e. of general form given by equation (2.7). The remaining methods are designed to solve an even more general class of elliptic equations (see (2.8)).

Let us determine the values of the constants estimating the errors of each of the methods used. For the GPE method, values $M=N=97.5$ were assumed, which is the same as the value of constant M from the previous example, and the graph with approximations is presented similarly as in Fig. 7.6. Similarly, also in the NE3C method, constant $M=97.5$, and constants $P=Q=32$, which is shown in Fig. 7.8. In this method, there is also the parameter σ for which a constant value of 10^{-3} was assumed. As for the NE5C method, for the example under consideration, the values of the constants $P=Q=32$, $R=S=32$ and $T=97.5$, and their approximations are not presented in the figures, as they are similar to Fig. 7.8 and 7.6 respectively. The results of the calculations are summarized in Table 7.6.

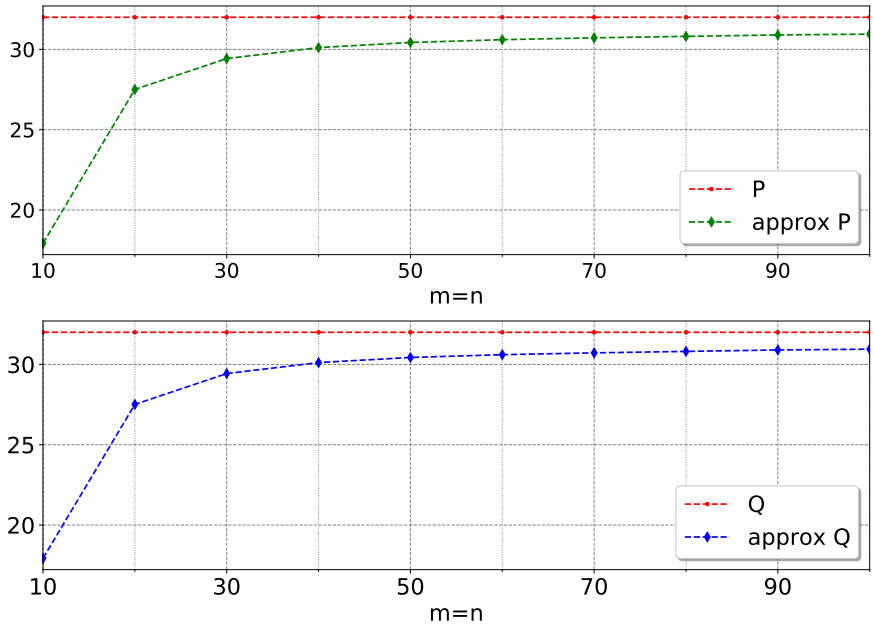

Figure 7.8. Approximations of the constants P and Q and their exact values assumed in NE3C method for problem (7.3)

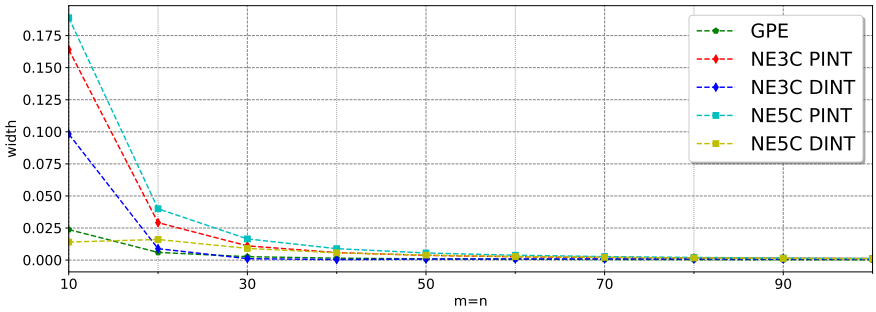

Figure 7.9. Widths of the resulting intervals at $(x, y) = (0.5, 0.5)$ obtained by the different methods for problem (7.3)

Table 7.6. Interval solutions and interval widths obtained in ordinary (U_d) and directed interval arithmetic for problem (7.3) at point $(x, y) = (0.5, 0.5)$. The exact solution $u_{exact}(0.5, 0.5) = 1.0$

$m = n$	$U_p(0.5, 0.5)$	Width	$U_d(0.5, 0.5)$	Width
20 (GPE)	[0.9990649974434728, 1.0050524160855946]	0.0059875	[0.9990649974434731, 1.0050524160855942]	0.0059875
20 (NE3C)	[0.9860739374347772, 1.0152541364845506]	0.0291802	[0.9964229264583249, 1.0053436834061202]	0.0089208
20 (NE5C)	[0.9786753667911464, 1.0186856003690083]	0.0400103	[0.9967344503643154, 1.0128046153340249]	0.0160702
40 (GPE)	[0.9997646140260656, 1.0012637869302333]	0.0014992	[0.9997646140260672, 1.0012637869302317]	0.0014992
40 (NE3C)	[0.9971061584234387, 1.0029451881631108]	0.0058391	[0.9998396275730703, 1.0002394673377627]	0.0003999
40 (NE5C)	[0.9951271381913196, 1.0040441558490130]	0.0089171	[0.9983151113625363, 1.0040082859452567]	0.0056932
60 (GPE)	[0.9998952490902022, 1.0005617396807390]	0.0006665	[0.99989524909020632, 1.0005617396807350]	0.0006665
60 (NE3C)	[0.9986813778136610, 1.0013054266639347]	0.0026241	[0.9995433062314391, 1.0004490541542588]	0.0009058
60 (NE5C)	[0.9979136690784978, 1.0016947696529887]	0.0037812	[0.9991180516182936, 1.0019120076700542]	0.0027940
80 (GPE)	[0.9999410510369446, 1.0003159897301436]	0.0003750	[0.9999410510369521, 1.0003159897301361]	0.0003760
80 (NE3C)	[0.9991601301069161, 1.0008238062494343]	0.0016637	[0.9995377341314507, 1.0004479808293076]	0.0009105
80 (NE5C)	[0.9988506397527160, 1.0009229643915976]	0.0020724	[0.9994654771167783, 1.0011140612022322]	0.0016486
100 (GPE)	[0.9999622647894478, 1.0002022367349792]	0.0002400	[0.9999622647894605, 1.0002022367349665]	0.0002391
100 (NE3C)	[0.9998944144289977, 1.0000930442322149]	0.0012635	[0.9995602349489116, 1.0004272237123009]	0.0008670
100 (NE5C)	[0.9992740680518004, 1.0005787490125610]	0.0013047	[0.9996429422202743, 1.0007281964538527]	0.0010853

Conclusions. For all investigated methods the interval solutions contained the exact solution. The most accurate estimates were obtained by the GPE method, which is dedicated only to problems of the form 2.7 to which the issue under consideration belongs. More general methods, namely NE3C and NE5C, also allowed us to obtain correct estimates, however, less accurate (wider resultant ranges). The above example also shows that minimizing the number of constants estimating the errors of the method is reason-

able – using the NE3C method we obtained more accurate estimates of exact solutions than using the NE5C method. For the NE3C method also a *rounding-off* – effect was observed - starting from grid size $m = n = 60$ the width of the resulting intervals began to increase. ■

Example 4

The main purpose of the fourth example is to further analyse the relative position of the exact solution inside the result intervals obtained by the methods proposed in this work. It was previously published in the paper [30].

Let us denote by $p(s)$ the relative position of the solution s inside the resulting interval $A = [a^-, a^+]$ defined as follows:

$$p(s) = \frac{|s - mid(A)|}{widht(A)}, \tag{7.4}$$

where $mid(A) = \frac{a^+ + a^-}{2}$ and $widht(A) = a^+ - a^-$. The value of $p(s)$ determines whether the solution s lies inside the interval A, since

$$p(s) = \begin{cases} (\frac{1}{2}, +\infty) \text{ dla } s \notin A, \\ [0, \frac{1}{2}] \text{ dla } s \in A. \end{cases}$$

Consider the problem (2.7) defined over the region $\Omega = (0,1) \times (0,1)$, in which

$$\begin{aligned} f(x,y) &= x^2y^2(3y^2 + 2x^2y^2 - 3x^2), \\ a_1(x,y) &= xy^3e^{-\frac{x^2+y^2}{2}}, \\ a_2(x,y) &= x^3ye^{-\frac{x^2-y^2}{2}}, \end{aligned} \tag{7.5}$$

with boundary conditions

$$\begin{aligned} \varphi_1(y) &= ye^{\frac{1-y^2}{2}}, \quad \varphi_2(x) = xe^{\frac{x^2-1}{2}}, \\ \varphi_3(y) &= 2ye^{\frac{4-y^2}{2}}, \quad \varphi_4(x) = 2xe^{\frac{x^2-4}{2}} \end{aligned}$$

and an exact solution

$$u(x,y) = xye^{\frac{x^2-y^2}{2}}. \tag{7.6}$$

As shown in the previous example, the most accurate method (among those described in this work) for this type of problems is the NE3C method and it was used to find interval solutions. The values of constants $M = 636.4$ and $N = 53.79$ were taken as the estimated errors of the method.

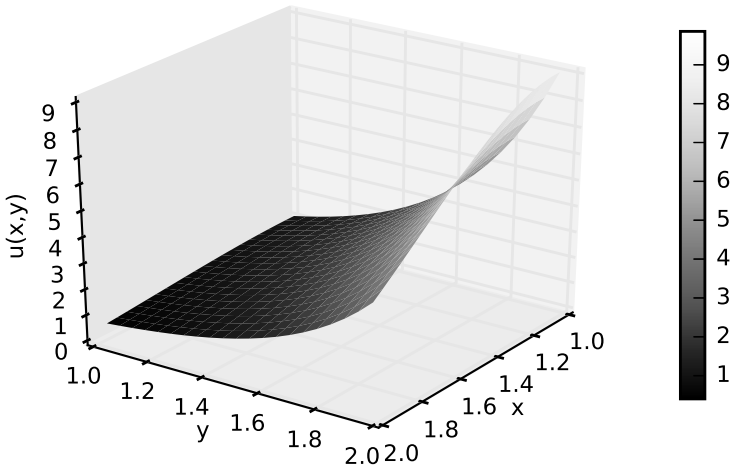

Figure 7.10. Exact solution for problem (7.6)

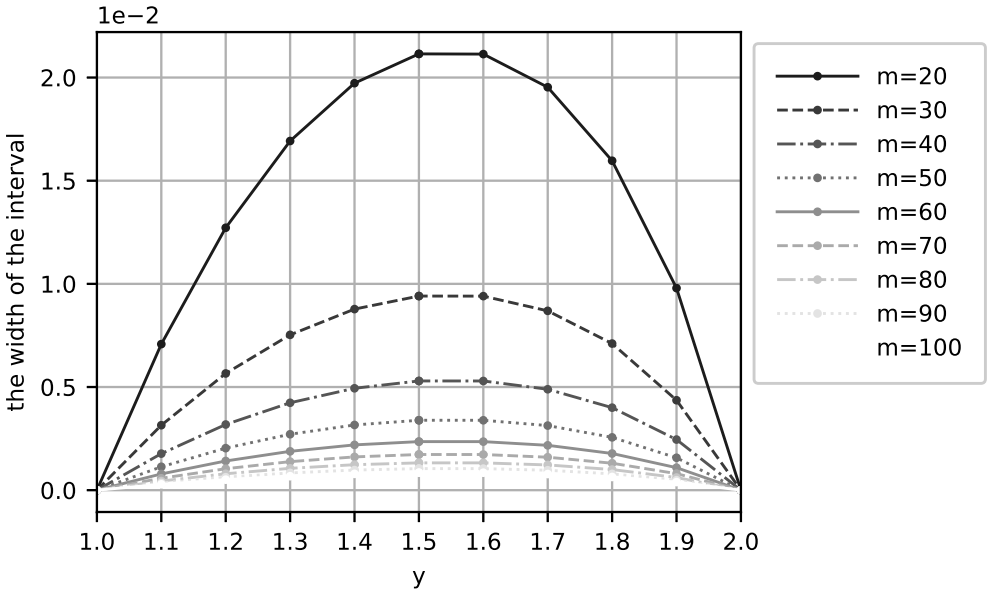

Figure 7.11. Widths of the resulting intervals obtained by ordinary interval arithmetic at $x = 1.5$ for the problem defined by (7.5).

Figure 7.11 shows that increasing the grid size results in narrower resulting intervals for both interval arithmetic. Narrower intervals imply a better estimate of the location of the exact solution.

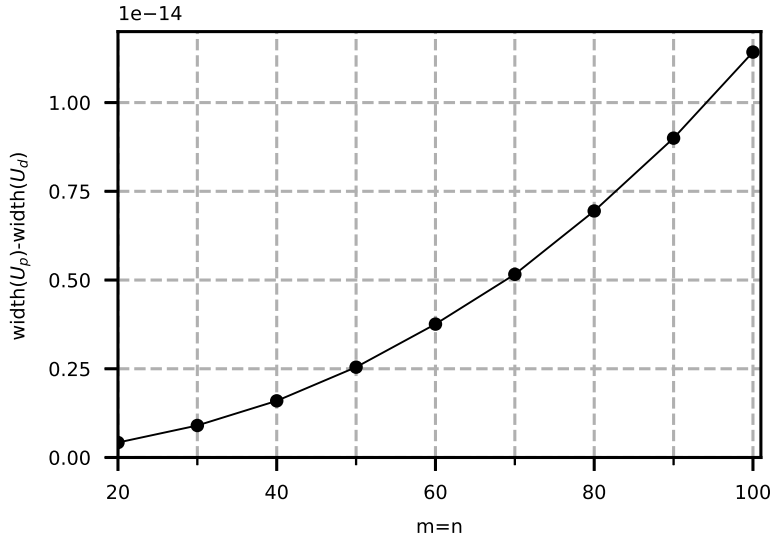

Figure 7.12. The difference in width of the resulting intervals at $(x, y) = (1.5, 1.5)$ between ordinary (U_p) and directed (U_d) interval arithmetic.

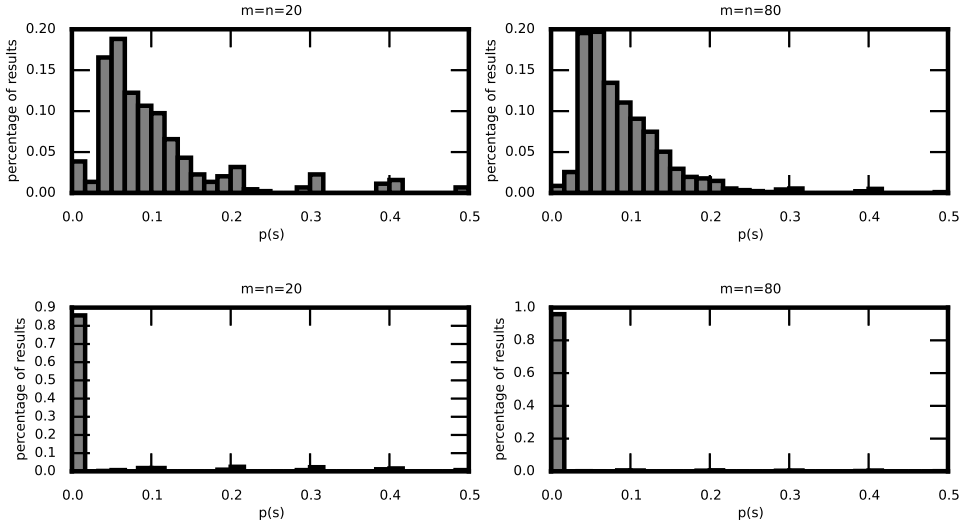

Figure 7.13. Relative position $p(s)$ (7.4) of the exact (first row) and floating-point (second row) solutions to the problem given by equation (7.5) inside the intervals obtained by ordinary interval arithmetic

Conclusions. The difference in the width of the resulting intervals between ordinary and directed interval arithmetic increases as the grid size increases, as shown in Figure 7.12. A similar effect is also observed in the following examples. Figures 7.13 show the relative position of exact and variable solutions inside the intervals obtained in ordinary interval arithmetic. These results confirm that the exact solutions are always inside the obtained interval solutions. Moreover, with the increased density of the grid we observe that the distribution of positions of exact solutions changes and the denser the grid, the more exact solutions lie relatively closer to the center of the resulting intervals. For directed interval arithmetic, analogous results were obtained. Note also that the exact solutions rarely lie in the middle of the interval solutions. On the other hand, as expected, the floating point solutions lie almost in the middle of the interval solutions and their deviation from the centers of the resulting intervals is negligibly small. ■

Example 5

In this example and the next one, let us consider the boundary problems for which Nakao presented his method in the works [76, 77, 90]. Both boundary problems were solved by both Nakao's method (MN) and the method proposed in this work in Section 6.2 (NE3C). Let us take the following boundary problem:

$$\begin{gathered}\frac{\partial^2 u}{\partial x^2}(x,y)+\frac{\partial^2 u}{\partial y^2}(x,y)+\frac{5}{4}\pi^2 u(x,y)=-\pi\sin(\frac{\pi}{2}x)\sin(\pi y),\\ u|_\Gamma(x,y)=\varphi(x,y)=0.\end{gathered} \tag{7.7}$$

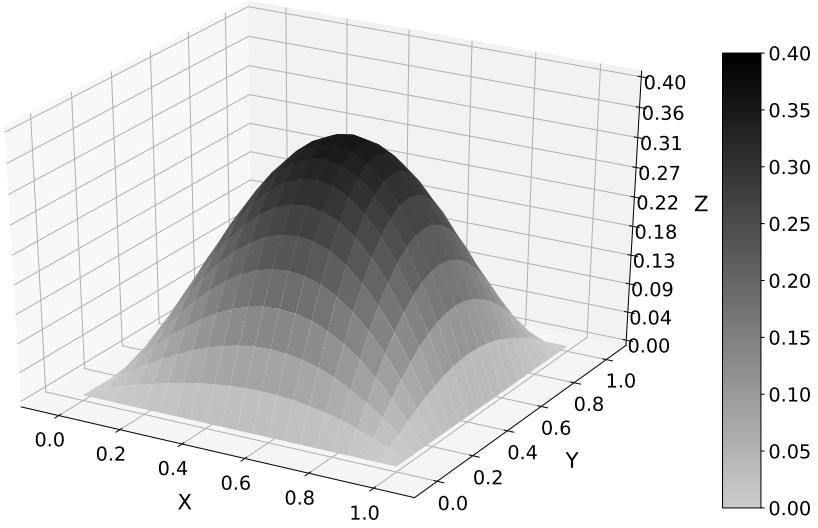

Figure 7.14. Exact solution (7.7)

The exact solution is expressed by the formula $u(x,y)=x\cos(\frac{\pi}{2})\sin(\pi y)$ and is shown in Figure 7.14. For comparison, it was also solved by the finite difference method (FDM), presented in Section 6.2, using classical and interval floating point arithmetic and by the Nakao method (NM). The results are summarized in Table 7.7.

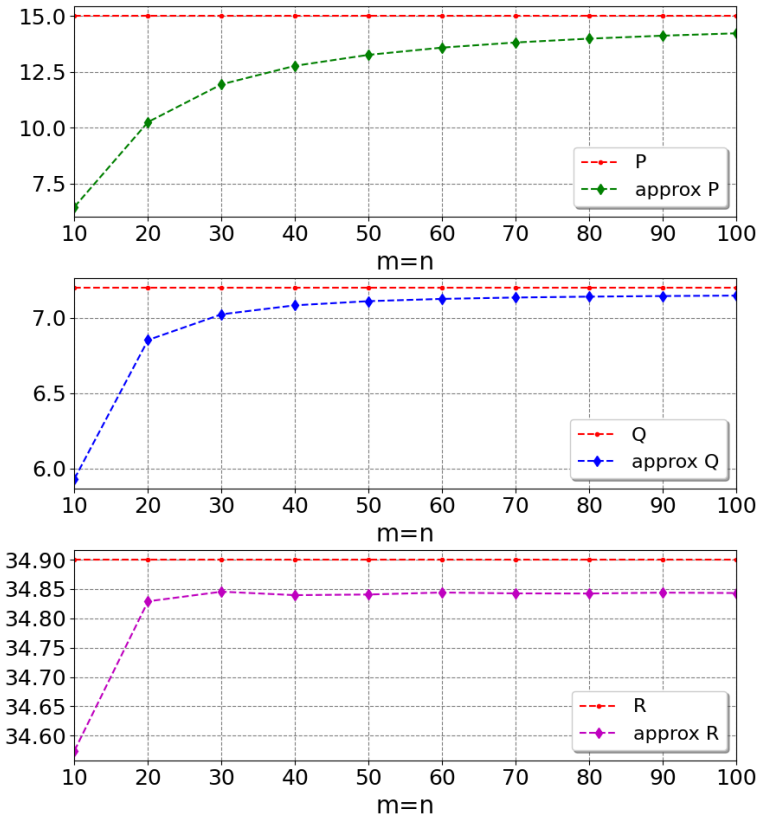

Figure 7.15. Estimation of constants P, Q and R

Figure 7.15 shows the results obtained by the experimental method of determining the error estimating constants for the NE3C method. As for Nakao's method, described by pseudo-code 4.6. all the necessary integrals defined earlier in detail in chapter 4had to be determined. Their analytical determination, though generally possible, is time – consuming, Thus, we propose, like Nakao himself, to perform a numerical integration. The GNU Scientific Library [20] and an integration method called *gsl_monte_plain_integrate*. were chosen for this task. It requires an indication of the number of internal iterations (that is, performed during each integration) – this number is specified by the parameter GSL_MC_ITER. Thus, it should be assumed that the entire computational process for the Nakao method will be characterized by high time complexity.

Table 7.7. Interval solutions and interval widths obtained in ordinary (U_p) and directed (U_d) interval arithmetic for problem (7.7) at point $(x, y) = (0.5, 0.5)$. Parameters of the NM method: $GSL\_MC\_ITER = 25000$, $\epsilon = 10^{-8}$ and $\delta = 10^{-8}$. Exact solution $u(0.5, 0.5) \approx 0.353553390593273762$

$m = n$	$U_p(0.5, 0.5)$	Szerokość	$U_d(0.5, 0.5)$	Szerokość
10 (FDM)	[0.33736480586793682, 0.36300478305332898]	0.025639	[0.33756936887156512, 0.36280022004970067]	0.025231
10 (NM)	[0.3152061118620046, 0.36132018383540089]	0.046114	———	———
20 (FDM)	[0.34941684316762481, 0.35610061725233360]	0.006684	[0.34962483373204273, 0.35589262668791567]	0.006268
20 (NM)	[0.33915297428263464, 0.36240419420738459]	0.023252	———	———
40 (FDM)	[0.35237722718097839, 0.35434723485693209]	0.001971	[0.35258608866995258, 0.35413837336795790]	0.001553
40 (NM)	[0.34825826185867063, 0.35991260298977276]	0.011655	———	———
50 (FDM)	[0.35272747688557449, 0.354136199185642]	0.000141	[0.35293644326868165, 0.35392723280253516]	0.000991
50 (NM)	[0.34982805250009871, 0.35915492449402315]	0.009327	———	———
60 (FDM)	[0.35291693874340361, 0.35402163378308488]	0.001105	[0.35312596214058223, 0.35381261038590626]	0.000687
60 (NM)	[0.35083812253001074, 0.35861146659504341]	0.007774	———	———
70 (FDM)	[0.35303084052839114, 0.35395264357629360]	0.000922	[0.35323989831501673, 0.35374358578966801]	0.000504
70 (NM)	[0.35153360919724092, 0.35819829475234529]	0.006665	———	———
80 (FDM)	[0.35310460462858216, 0.35390792521922393]	0.000804	[0.35331368474001861, 0.35369884510778749]	0.000386
80 (NM)	[0.35205135444988570, 0.35788372978196933]	0.000584	———	———
90 (FDM)	[0.35315509109784634, 0.35366820876113393]	0.000723	[0.35336418651725463, 0.35413837336795790]	0.000305
90 (NM)	[0.35245716757768397, 0.35764099607470005]	0.0051839	———	———
100 (FDM)	[0.35319115489527668, 0.35385542477058268]	0.000665	[0.35340026126546315, 0.35364631840039621]	0.000247
100 (NM)	[0.35276267037643749, 0.35742841430850891]	0.000467	———	———

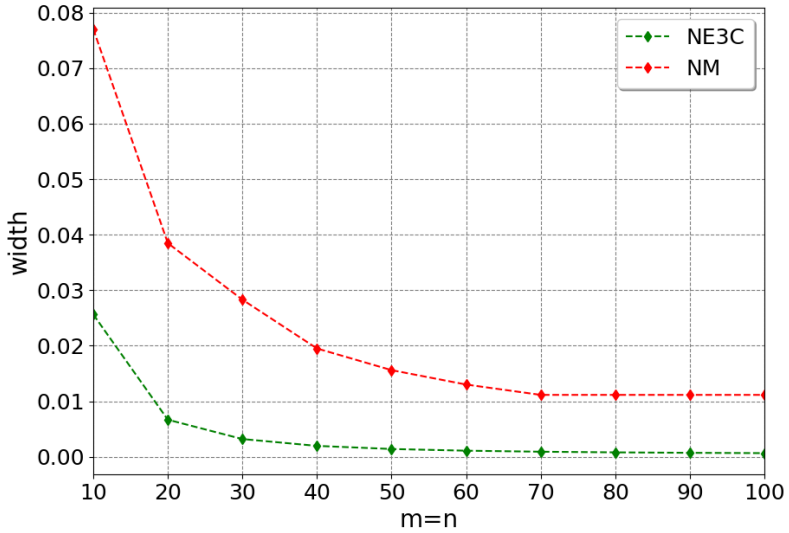

Figure 7.16. Widths of the resulting intervals at $(x, y) = (0.5, 0.5)$ obtained by NE3C and Nakao (NM) methods

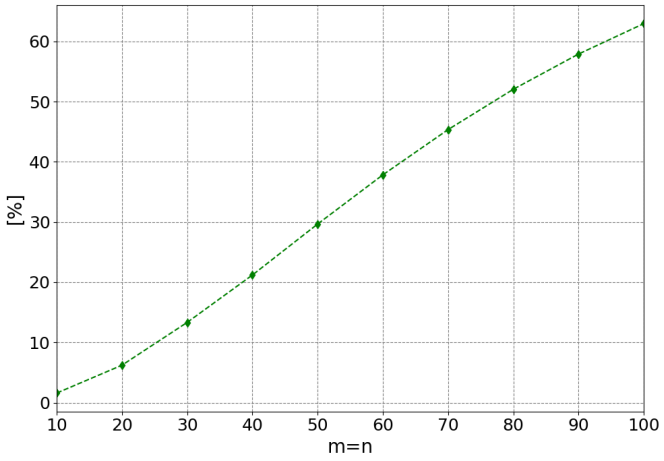

Figure 7.17. Difference in the width of the resulting intervals at $(x, y) = (0.5, 0.5)$ obtained by the N3C method in ordinary and directed floating point arithmetic (expressed as a percentage)

Conclusions. Increasing the size of the grid results in obtaining increasingly narrower intervals - solutions, and thus increasingly precise estimation of the exact solution. Note that the intervals estimating the exact solution obtained by the Nakao method are wider than those obtained by the method proposed in this work. Experiments have shown that the resulting intervals for both methods contain the exact solution, with the estimates obtained by the method in Section 6.2.2 being more accurate. It is also worth noting that the exact solution is contained in the results obtained using both interval arithmetics, with the intervals obtained in the directed arithmetic being slightly narrower. ■

Example 6

Consider an example of an elliptic equation of a more complicated form, where the parameter $c = c(x, y)$, that is, is a function of two variables rather than a constant value as in the previous example. Let us take the following boundary problem:

$$\begin{gathered}\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) + 20\sin(\pi xy)u(x,y) \\ = 20(1-x)(1-y)(1-e^{xy})\sin(\pi xy) \\ -(1-x)(1-y)y^2e^{xy} + 2(1-y)ye^{xy} - (1-x)x^2(1-y)e^{xy} + 2(1-x)xe^{xy}, \\ u|_\Gamma(x,y) = \varphi(x,y) = 0.\end{gathered} \tag{7.8}$$

As you can see, the form of the function $f = f(x, y)$ on the right side of the equation is also much more complicated. The exact solution of this problem is given by the formula

$$u(x,y) = (1-x)(1-y)(1-e^{xy}).$$

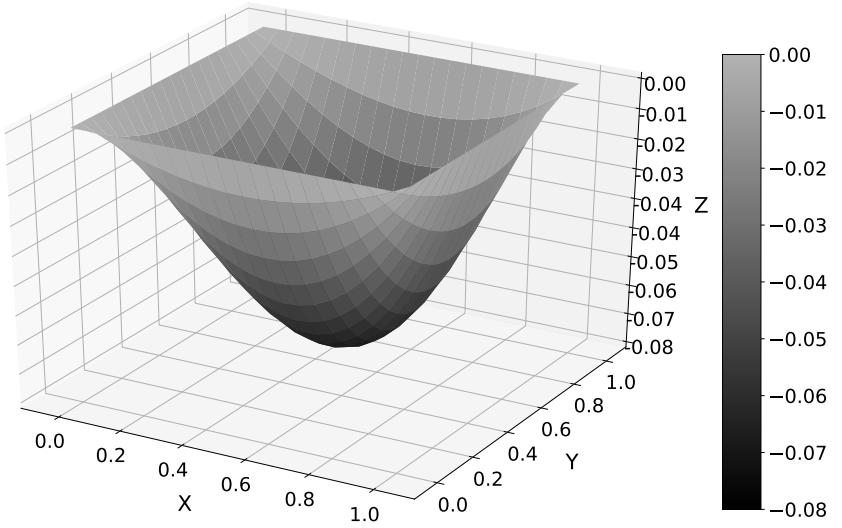

Figure 7.18. Exact solution (7.8)

The problem was solved, as in the previous example, using the NM method and the NE3C method. The estimates of constants P,Q and R for the NE3C method adopted in the calculations were 4.0, 1.0 and 18.0, respectively. They can be determined experimentally, as shown in Figure 7.19. The obtained interval results for problem 7.8 are summarized in Table 7.8.

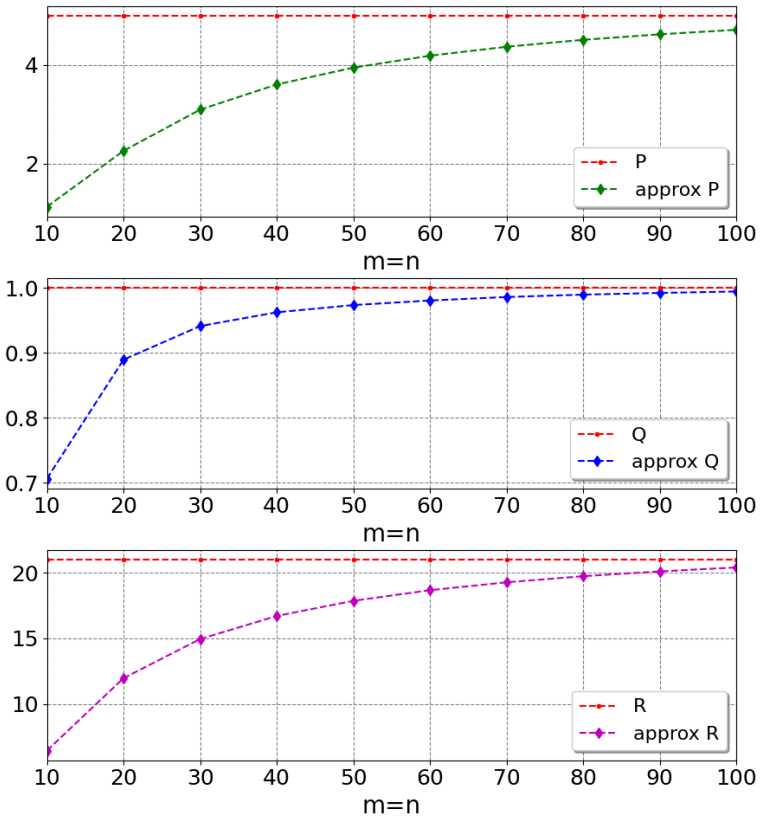

Figure 7.19. Estimation of constants P, Q and R

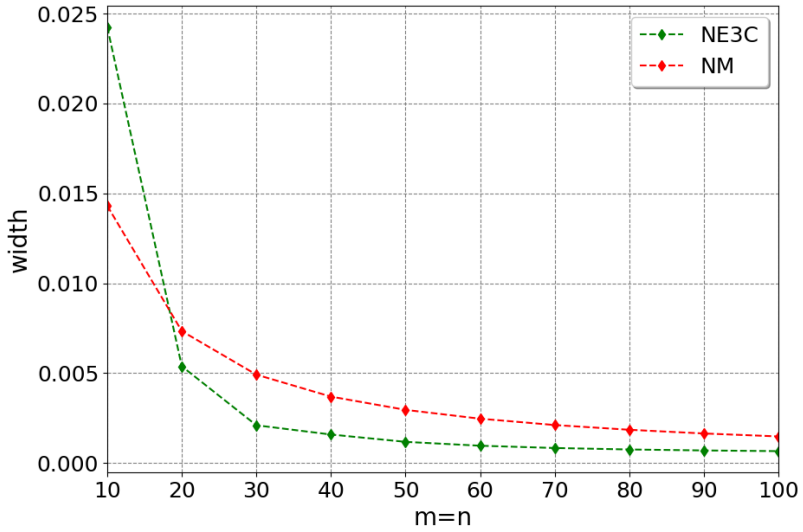

Figure 7.20. Widths of the resulting intervals at $(x, y) = (0.5, 0.5)$ obtained by NE3C and Nakao (NM) methods

Table 7.8. Interval solutions and interval widths obtained in proper (U_p) and directed (U_d) interval arithmetic for problem (7.8) at point $(x, y) = (0.5, 0.5)$. Parameters of the NM method: $GSL\_MC\_ITER = 50000$, $\epsilon = 10^{-6}$ and $\delta = 10^{-6}$. Exact solution $u(0.5, 0.5) \approx -0.07100635417193537$

$m = n$	$U_p(0.5, 0.5)$	Width	$U_d(0.5, 0.5)$	Width
10 (FDM)	[−0.0803593623373, −0.05609708833454]	0.024262	[−0.0801120392085, −0.05634441146333]	0.023767
10 (NM)	[−0.07371995397025, −0.05940775359080]	0.014312	———	———
20 (FDM)	[−0.0730226649557, −0.06764240796135]	0.005381	[−0.0727725649764, −0.06789250794075]	0.004880
20 (NM)	[−0.0736936420756, −0.06635492256125]	0.007338	———	———
40 (FDM)	[−0.0716360062762, −0.07004991689743]	0.001587	[−0.0713852159381, −0.070300707235545]	0.001085
40 (NM)	[−0.0728105668809, −0.06911503278813]	0.003696	———	———
50 (FDM)	[−0.0714915329527, −0.07031340623592]	0.001179	[−0.0713852159381, −0.07030070723554]	0.000677
50 (NM)	[−0.0725579344437, −0.06959961152060]	0.002959	———	———
60 (FDM)	[−0.07141614475551, −0.070452886154063]	0.000964	[−0.07116522674098, −0.070703804168597]	0.000462
60 (NM)	[−0.07237804343632, −0.069911177391834]	0.002467	———	———
70 (FDM)	[−0.07137198052190, −0.070535471876300]	0.000837	[−0.07112103541543, −0.070786416982778]	0.000335
70 (NM)	[−0.07224229063939, −7.012801234449167]	0.002115	———	———
80 (FDM)	[−0.07134393436792, −0.070588352794936]	0.000755	[−0.07109297167899, −0.070839315483867]	0.000253
80 (NM)	[−0.07213822483315, −0.070287571865418]	0.001851	———	———
90 (FDM)	[−0.07132503264187, −0.070624230448671]	0.000701	[−0.07107405789903, −0.070875205191503]	0.000199
90 (NM)	[−0.07205642787106 , −0.070412142956674]	0.001645	———	———
100 (FDM)	[−0.07131169771395, −0.070649680535005]	0.000662	[−0.07106071434932, −0.070900663899635]	0.000161
100 (NM)	[−0.07198821126530, −0.070507997969016]	0.001481	———	———

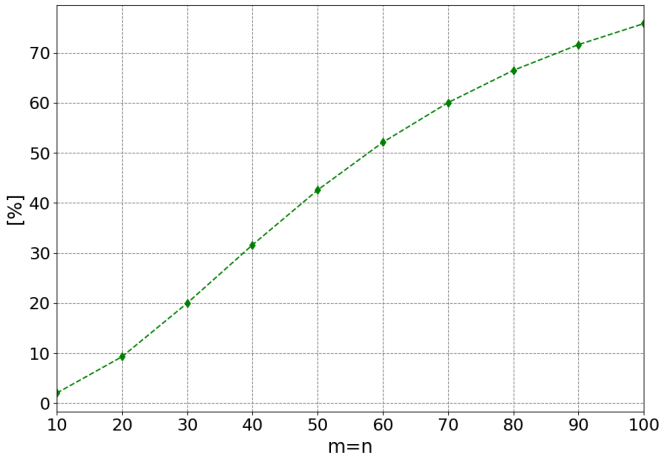

Figure 7.21. Difference in width of resultant intervals at $(x, y) = (0.5, 0.5)$ obtained by NE3C method in ordinary and directed floating point arithmetic (expressed as a percentage)

Conclusions. The Nakao method produced more accurate estimates only for the smallest grid size tested, i.e. $m = n = 10$. As the grid size increased, the width of the resulting intervals decreased much faster for the NE3C method. The difference was about an order of magnitude. This situation is probably due to the fact that in NM numerical integration is used many times – while the basis functions that occur there have not changed (they are the same pyramidal functions), but the functions that are parameters of the equation are in this case much more complex. It is interesting to note that, as in the previous example, the difference in width (expressed as a percentage) between the intervals obtained by the NE3C method in ordinary and directed arithmetic increased as the grid size increased. This suggests that the more calculations are to be performed, the greater the benefit of using directed interval arithmetic. This emphasizes the sensibility of the author's implementation of this arithmetic. ■

8 Summary

In this dissertation, interval methods of the FDM class which allow finding estimates of exact solutions for boundary problems defined for selected elliptic equations are presented. In total, methods based on five different differential schemes are presented for three types of PDEs, which are implemented in three types of arithmetic, i.e. floating-point arithmetic, ordinary interval arithmetic, and directed interval arithmetic (see Table 5.1 and Table 6.1).

The algorithms presented in this work deal with the heuristic estimation of exact solutions obtained by interval FDM methods. An attempt is also made to refer to a method that allows a rigorous verification of the existence of PDE solutions and finding their estimate supported by a mathematical proof. Such a method for elliptic equations is the Nakao method using the FEM model (described in papers [76] and [78]). The results obtained with both types of methods, i.e. the interval FDM methods proposed in this work and the interval-based (but not fully interval – as pointed out in Chapter 4) Nakao method, were compared. However, this required extending the algorithm originally developed by the author for the simplest elliptic equation, the Poisson equation, to a form designed for the more general class of elliptic equations that Nakao considered.

Therefore, the Poisson equation (PE) with Dirichlet boundary conditions was taken as a starting point. Then the problem was generalised by adding functions which are the parameters of this equation, which is called (within this dissertation) the generalised Poisson equation (GPE). Then, this equation was extended to an even more general form defining a certain class of elliptic equations considered by Nakao (NE). This was important as, from preliminary analyses of the current state of research in the world, it appeared that the method developed by Nakao uses interval arithmetic to verify the existence and to find estimates for solutions of exact elliptic equations, which, in the author's opinion, needs to be referred to in this dissertation. Hence, a separate chapter (see Chapter 4) was devoted to a detailed description of this method.

The research conducted by the author towards the verification of the main hypothesis (H1), which was published in papers [29], [32], [31] and [39], showed that we can estimate the errors of the method experimentally and then take them into account during the calculations. Furthermore, it was possible to successfully demonstrate the usefulness of interval arithmetic for selecting the optimal grid size for a given problem [39]. The properties of directed interval arithmetic are also interesting, as due to the existence of the opposite and inverse element, it allows us to perform computations in a way that enables a certain reduction in the width of the end-intervals – the solution. Experiments

which were carried out to test the hypothesis (H3), confirmed that the intervals obtained after applying this arithmetic are narrower than in the case of ordinary interval arithmetic [29, 31, 32].

Moreover, it was possible to demonstrate the utility of using interval arithmetic in solving the following problems:

a) collecting information on rounding errors,

b) experimental estimation of method errors,

c) automatic estimation of the accuracy of the obtained solutions by specifying the width of the resulting intervals, which provides complete information on this subject.

It also offers:

a) the possibility of including the analytically estimated error of the method in the calculation,

b) the possibility of choosing an optimal grid for the problem (finding a grid size above which we deal with the so-called *rounding-off* effect).

The verification of hypothesis (H2) equired the development of interval methods, prepared earlier for the Poisson equation, in such a way as to include a certain class of elliptic linear equations of order two. At the same time, the Nakao method was replicated for these equations, which was part of the verification of hypothesis (H4). As a result of this work, it was shown that it is not possible to apply Nakao's method to solve Poisson's equation and its generalized form, although they are useful for a certain, quite general class of elliptic PDEs. On the other hand, the methods proposed in this dissertation (belonging to the finite difference methods) can be effectively applied also for the equations analysed by Nakao, and their significant advantage is a simpler construction and implementation. The experiments showed, moreover, that the exact solutions belong to the intervals obtained by the methods proposed by the author, although, as mentioned earlier, finding an analytical proof of the existence of exact solutions seems to be difficult from the mathematical point of view and depends on the problem under consideration.

In the course of conducting the research and analysing the results obtained on an ongoing basis, it was found that finding the answer the following, rather important, questions could be attempted:

- Is the interval method dedicated to a given (simplified) form of equation always more effective than the methods dedicated to equations of more general form? Hence, is it worth constructing interval methods for narrower or wider classes of equations?
- How do the implementations of the above methods behave in ordinary and directed positional arithmetic? Does the use of directed interval arithmetic provide any measurable benefits?

The attempt to answer these questions influenced the choice of computational examples presented in the previous chapter. Although it cannot be said to be the rule, Example 3 showed that constructing methods with a limited number of error-estimating constants gives better results – the advantage of the NE3C method over the NE5C method. Moreover, it showed that the use of directed interval arithmetic is cost-effective, and that the more complex calculations the example requires, the more noticeable are the effects of its use.

The main conclusions of the conducted research are as follows:

- for all considered issues, the exact solution was located inside the obtained result ranges (in the case of both tested arithmetic – ordinary and directed),
- construction of the interval methods proposed in this dissertation is easy, and finding the solutions comes down, in the worst case, to solving large systems of linear equations (this depends only on the size of the grid),
- experiments demonstrated that the obtained solution estimates for PDE problems are much more accurate than those obtained using the Nakao method,
- the methods presented in this paper can be applied to a much wider class of elliptic equations than Nakao's method. It is worth noting that Nakao's methods cannot be used for Poisson's equation or for GPE equation,
- in the Nakao method (based on the FEM and Galerkin approximation), a major problem is the need to use numerical integration, which in the case of more complex equations can significantly increase the computation time - experiments showed that the Nakao method using numerical integration is much slower than the considered interval FDM methods.

Once again, it should be stressed that the interval FDM methods proposed in this dissertation can be called heuristic methods for estimating exact solutions for elliptic equations. However, it is worth noting that they can be an excellent complement to *verified computing* methods such as the Nakao method. First, to obtain a guarantee of the existence of exact solutions and their general estimation, one can apply, for example, the Nakao method. The obtained interval solutions can be treated as a preliminary approximation of the exact solutions. Then, using e.g. the means of the intervals obtained by the *verified-computing* method, we can estimate the constants necessary for the methods described in this work. As a result, using the obtained constants approximating the error of the method, we can apply one of the presented FDM interval methods for the same problem – thus obtaining its much more accurate and indirectly verified estimation. It should also be emphasised that while for ordinary differential equations (ODEs) numerical methods for finding solutions and their verification are well known [8, 83], designing such methods for PDEs is a rather complex problem [11, 33]. This is due to the fact that on the mathematical side there is no general way of proving the existence of solutions to this type of equations [12–14]. This results in the fact that even if the numerical methods developed do find a solution to the equation, we are not sure either how correct the obtained solution is or whether it exists at all for the given problem [55]. Therefore, in the author's opinion, numerical methods that allow both obtaining detailed information about the accuracy of the obtained solution, as presented in this dissertation, and methods that allow proving its existence and verifying the solution are so important [81].

List of figures

List of tables

Content of the CD

The disk attached to the thesis contains the code of programs implementing particular methods described in the thesis. Also attached is a virtual machine with the operating system Linux Ubuntu 20.04 LTS and prepared runtime environment. Its full content is described in Tab. A.1.

Table A.1. Directories on CD

Path	Content
`/thesis`	dissertation in .pdf format
`/thesis/tex`	source TeX files with the body of the work and bibliography
`/thesis/tex/figures`	graphs and graphics included in the work
`/code`	subdirectories with the code of the individual methods
`/code/pe`	source code of the PE method
`/code/gpe`	source code of the GPE method
`/code/ee`	source code of EE methods
`/code/nm`	source code of the Nakao method
`/vm`	virtual machine with ready to run environment containing compiled code of particular methods (operating system: Ubuntu 20.04)
`/instructions`	detailed instructions for running the virtual machine and the programs

Bibliography

[1] Abdulle, A. and de Souza, G. R., "A Local Discontinuous Galerkin Gradient Discretization Method for Linear and Quasilinear Elliptic Equations", *Mathematical Modelling and Numerical Analysis*, vol. 53, no. 4, pp. 1269–1303, 2019.

[2] Alefeld, G. and Herzberger, J., *Introduction to Interval Computations*. New York, Academic Press, 1983.

[3] Ames, W. F., *Numerical Methods for Partial Differential Equations*. Cambridge, Academic Press, 2014.

[4] Bauch, H., "On the Iterative Inclusion of Solutions in Initial-Value Problems for Ordinary Differential Equations", *Computing*, vol. 22, pp. 339–354, 1979.

[5] Bloor, I. M. and Wilson, M. J., "Spectral Approximations to PDE Surfaces", *Computer-Aided Design*, vol. 28, no. 2, pp. 145–152, 1996.

[6] Boost, "Boost C++ Libraries (version 1.60)", http://www.boost.org/, 2021. access 2021-06-20.

[7] Burden, R. L. and Faires, J. D., *Numerical Analysis*. Boston, USA, Brooks/Cole Publishing Company, 9 ed., 2009.

[8] Butcher, J. C., *Numerical Methods for Ordinary Differential Equations*. New York, John Wiley and Sons, 2016.

[9] Cain, G. and Meyer, G. H., *Separation of Variables for Partial Differential Equations: An Eigenfunction Approach*. Boca Raton, Florida, CRC Press, 2005.

[10] Chandrupatla, T. R., Belegundu, A. D., Ramesh, T., and Ray, C., *Introduction to Finite Elements in Engineering*, vol. 2. Upper Saddle River, NJ, USA, Prentice Hall, 2002.

[11] Duarte, C. A. and Oden, J., *A Review of Some Meshless Methods to Solve Partial Differential Equations*. Austin, TX, USA, Texas Institute for Computational and Applied Mathematics, 1995.

[12] Evans, L. C., *Partial Differential Equations*. Rhode Island, USA, ACM, 1998.

[13] Farlow, S. J., *Partial Differential Equations for Scientists and Engineers*. North Chelmsford, Massachusetts, Courier Corporation, 1993.

[14] Folland, G. B., *Introduction to Partial Differential Equations.* New Jersey, Princeton University Press, 2020.

[15] Forsythe, G. E., "Pitfalls in Computation, or Why A Math Book Isn't Enough", *The American Mathematical Monthly*, vol. 77, no. 9, pp. 931–956, 1970.

[16] Fortuna, Z., Wasowski, J., and Macukow, B., *Metody Numeryczne.* Warszawa, Wydawnictwa Naukowo-Techniczne, 2005.

[17] Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P., and Zimmermann, P., "MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding", *ACM Trans. Math. Softw.*, vol. 33, no. 2, 2007.

[18] Gennaro, R., Gentry, C., and Parno, B., "Non-interactive verifiable computing: Outsourcing computation to untrusted workers", in *Advances in Cryptology – CRYPTO 2010* (Rabin, T., ed.), (Berlin, Heidelberg), pp. 465–482, Springer Berlin Heidelberg, 2010.

[19] Goldberg, D., "What Every Computer Scientist Should Know About Floating-Point Arithmetic", *ACM Computing Surveys (CSUR)*, vol. 23, no. 1, pp. 5–48, 1991.

[20] Gough, B., *GNU Scientific Library Reference Manual.* Network Theory Ltd., 2009.

[21] Greathouse, J. L. and Daga, M., "Efficient Sparse Matrix-Vector Multiplication on GPUs Using the CSR Storage Format", in *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 769–780, IEEE, 2014.

[22] Griffiths, D. F., Dold, J. W., and Silvester, D. J., *Essential Partial Differential Equations.* Hidelberg, Berlin, Springer, 2015.

[23] Hammer, R., Hocks, M., Kulisch, U., and Ratz, D., *C++ Toolbox for Verified Computing I: Basic Numerical Problems Theory, Algorithms, and Programs.* Berlin-Heidelberg, Springer Science & Business Media, 2012.

[24] Harfash, A. J. and Huda, A. J., "Sixth and Fourth Order Compact Finite Difference Schemes for Two and Three Dimension Poisson Equation with Two Methods to Derive These Schemes", *Basrah Journal of Scienec (A)*, vol. 24, no. 2, pp. 1–20, 2006.

[25] Harris, C. R., Millman, K. J., Walt, S. J., and others, "Array Programming with NumPy", *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[26] Hickey, T., Ju, Q., van Emden, and H, M., "Interval Arithmetic: From Principles to Implementation", *Journal of the ACM (JACM)*, vol. 48, no. 5, pp. 1038–1068, 2001.

[27] Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S., *Optimization with PDE Constraints.* Heidelberg, Springer Science & Business Media, 2008.

[28] Hladik, M., "Ae Solutions and ae Solvability to General Interval Linear Systems", *Linear Algebra and its Applications*, vol. 465, pp. 221–238, 2015.

[29] Hoffmann, T. and Marciniak, A., "Solving the Poisson Equation by an Interval Difference Method of the Second Order", *Computational Methods in Science and Technology*, vol. 19, no. 1, pp. 13–21, 2013.

[30] Hoffmann, T. and Marciniak, A., "Solving the Generalized Poisson Equation in Proper and Directed Interval Arithmetic", *Computational Methods in Science and Technology*, vol. 22, no. 4, pp. 225–232, 2016.

[31] Hoffmann, T. and Marciniak, A., "Finding Optimal Numerical Solution in Interval Version of Central Difference Method for Solving the Poisson Equation", in *Data Analysis – Selected Problems* (Łatuszyńska, M. and Nermend, K., eds.), chapter 5, pp. 79–88, Szczecin, Warszawa, Polish Information Processing Society, 2013.

[32] Hoffmann, T., Marciniak, A., and Szyszka, B., "Interval Versions of Central Difference Method for Solving the Poisson Equation in Proper and Directed Interval Arithmetic", *Foundations of Computing and Decision Sciences*, vol. 38, no. 3, pp. 193–206, 2013.

[33] Houstis, E. N., "The Complexity of Numerical Methods for Elliptic Partial Differential Equations", *Journal of Computational and Applied Mathematics*, vol. 4, no. 3, pp. 191–197, 1978.

[34] "IEEE Standard for Floating-Point Arithmetic", https://standards.ieee.org/content/ieee-standards/en/standard/754-2019.html, 2019.

[35] "IEEE Standard for Floating-Point Arithmetic", https://standards.ieee.org/content/ieee-standards/en/standard/754-1985.html, 1985.

[36] "IEEE Standard for Floating-Point Arithmetic", https://standards.ieee.org/content/ieee-standards/en/standard/754-2008.html, 2008.

[37] Jacob, F. and Ted, B., *A First Course in Finite Elements*. Hoboken, New Jersey, Wiley, 2007.

[38] Jankowska, M., Jankowski, J., and Dryja, M., *Przegląd metod i algorytmów numerycznych, cz. 2*. Warszawa, Wydawnictwa Naukowo-Techniczne, 1988.

[39] Jankowska, M., Marciniak, A., and Hoffmann, T., "On an Application of an Interval Backward Finite Difference Method for Solving the One-Dimensional Heat Conduction Problem", *Control and Cybernetics*, vol. 44, 2015.

[40] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E., "Interval Analysis", in *Applied Interval Analysis*, pp. 11–43, Springer, 2001.

[41] Johansson, F. and others, *MPMATH: A Python Library for Arbitrary-Precision Floating-Point Arithmetic*, 2013. http://mpmath.org/.

[42] Kahan, W., "Ieee Standard 754 for Binary Floating-Point Arithmetic", *Lecture Notes on the Status of IEEE*, vol. 754, no. 94720-1776, p. 11, 1996.

[43] Kaucher, E., "Interval Analysis in the Extended Interval Space IR", in *Fundamentals of Numerical Computation (Computer-Oriented Numerical Analysis)*, pp. 33–49, Hidelberg, Berlin, Springer, 1980.

[44] Kikuchi, F. and Saito, H., "Remarks on A Posteriori Error Estimation for Finite Element Solutions", *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 329–336, 2007.

[45] Kincaid, D. R. and Cheney, E. W., *Numerical Analysis*, vol. 2. Providence, Rhode Island, American Mathematical Society, 2009.

[46] Kinoshita, T., Hashimoto, K., and Nakao, M. T., "On the L2 A Priori Error Estimates to the Finite Element Solution of Elliptic Problems with Singular Adjoint Operator", *Numerical Functional Analysis and Optimization*, vol. 30, no. 3-4, pp. 289–305, 2009.

[47] Kostrikin, A. I., *Wstęp do algebry, cz.1,2,3.* Warszawa, Wydawnictwo Naukowe PWN, 2008.

[48] Kreinovich, V. and Lakayev, A. V., "Solving Linear Interval Systems Is NP-Hard", *Reliable Computing*, vol. 4, pp. 383–388, 1998.

[49] Kreinovich, V., Lakeyev, A. V., Rohn, J., and Kahl, P., *Computational Complexity and Feasibility of Data Processing and Interval Computations*, vol. 10. Springer Science & Business Media, 1998.

[50] Kulisch, U., "Up-to-Date Interval Arithmetic: From Closed Intervals to Connected Sets of Real Numbers", in *Parallel Processing and Applied Mathematics*, pp. 413–434, Hidelberg, Berlin, Springer, 2016.

[51] Kulisch, U., "Mathematics and Speed for Interval Arithmetic: A Complement to IEEE 1788", *ACM Transactions on Mathematical Software (TOMS)*, vol. 45, no. 1, pp. 1–22, 2019.

[52] Kulisch, U. W., "Complete Interval Arithmetic and Its Implementation on the Computer", in *Numerical Validation in Current Hardware Architectures*, pp. 7–26, Hidelberg, Berlin, Springer, 2009.

[53] Lax, P. D. and Milgram, A. N., "Parabolic Equations. Contributions to The Theory of Partial Differential Equations", *Annals of Mathematics Studies*, pp. 167–190, 1954.

[54] LeVeque, R. J., *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems.* Philadelphia, SIAM, 2007.

[55] Lewy, H., "An Example of a Smooth Linear Partial Differential Equation Without Solution", *Annals of Mathematics*, pp. 155–158, 1957.

[56] Liu, W. and Vinter, B., "CSR5: An Efficient Storage Format for Cross-Platform Sparse Matrix-Vector Multiplication", in *Proceedings of the 29th ACM on International Conference on Supercomputing*, pp. 339–350, 2015.

[57] Liu, X., Nakao, M. T., Oishi, S., and others, "Explicit A Posteriori and A Priori Error Estimation for the Finite Element Solution of Stokes Equations", *Japan Journal of Industrial and Applied Mathematics*, pp. 1–15, 2021.

[58] Luo, Z.-H., "Direct Strain Feedback Control of Flexible Robot Arms: New Theoretical and Experimental Results", *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1610–1622, 1993.

[59] Marciniak, A., "Implicit Interval Methods for Solving the Initial Value Problem", *Numerical Algorithms*, vol. 37, no. 1–4, pp. 241–251, 2004.

[60] Marciniak, A., "On Multistep Interval Methods for Solving the Initial Value Problem", *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 229–237, 2007.

[61] Marciniak, A., *Selected Interval Methods for Solving the Initial Value Problem.* Poznań, Publishing House of Poznan University of Technology, 2009.

[62] Marciniak, A., "Interval Arithmetic Module.", http://www.cs.put.poznan.pl/amarciniak/IAUnits/IntervalArithmetic32and64.pas, 2016. access 2021-06-20.

[63] Marciniak, A., "Nakao's Method and An Interval Difference Scheme of Second Order for Solving The Elliptic BVP", *Computational Methods in Science and Technology*, vol. 25, no. 2, pp. 81–97, 2019.

[64] Marciniak, A. and Hoffmann, T., "IntervalArithmetic", http://www.cs.put.poznan.pl/amarciniak/EMN-wyklady/Interval.h, 2020.

[65] Marciniak, A. and Szyszka, B., "One-and Two-Stage Implicit Interval Methods of Runge-Kutta Type", *Computational Methods in Science and Technology*, vol. 5, no. 1, pp. 53–65, 1999.

[66] Marciniak, A., Gregulec, D., and Kaczmarek, J., *Podstawowe procedury numeryczne w języku Turbo Pascal*. Poznań, Wydawnictwo Nakom, 2000.

[67] Marciniak, A. and Hoffmann, T., "Interval Difference Methods for Solving the Poisson Equation", in *International Conference on Differential & Difference Equations and Applications*, pp. 259–270, Springer, 2017.

[68] Marciniak, A., Jankowska, M. A., and Hoffmann, T., "An Interval Difference Method of Second Order for Solving an Elliptical BVP", in *International Conference on Parallel Processing and Applied Mathematics*, pp. 407–417, Springer, 2019.

[69] Markov, S., "On Directed Interval Arithmetic and Its Applications", in *J. UCS The Journal of Universal Computer Science*, pp. 514–526, Springer, 1996.

[70] Maćkiewicz, A., *Algorytmy algebry liniowej. Metody bezposrednie.* Poznań, Wydawnictwo Politechniki Poznańskiej, 2002.

[71] Minguzzi, E., "The Equality of Mixed Partial Derivatives Under Weak Differentiability Conditions", *Real Analysis Exchange*, vol. 40, no. 1, pp. 81–98, 2015.

[72] Moore, R., *Interval Analysis*. Englewood Cliffs, Prentice-Hall, 1966.

[73] Moore, R., Kearfott, R. B., and Cloud, M. J., *Introduction to Interval Analysis*. Philadelphia, Society of Industrial and Applied Mathematics, 2003.

[74] Muller, J.-M., Brisebarre, N., De Dinechin, F., and others, *Handbook of Floating-Point Arithmetic*. Berlin-Heidelberg, Springer Science & Business Media, 2009.

[75] Nagel, J. R., "Solving the Generalized Poisson Equation Using the Finite-Difference method (FDM)", *Lecture Notes, Dept. of Electrical and Computer Engineering, University of Utah*, 2011.

[76] Nakao, M. T., "A Numerical Approach to the Proof of Existence of Solutions for Elliptic Problems", *Japan Journal of Applied Mathematics*, vol. 5, no. 2, p. 313, 1988.

[77] Nakao, M. T., "A Numerical Verification Method for The Existence of Weak Solutions for Nonlinear Boundary Value Problems", *Journal of Mathematical Analysis and Applications*, vol. 164, no. 2, pp. 489–507, 1992.

[78] Nakao, M. T., "Solving Nonlinear Elliptic Problems with Result Verification Using An H-1 Type Residual iteration", in *Validation Numerics*, pp. 161–173, Springer, 1993.

[79] Nakao, M. T., "On Verified Computations of Solutions for Nonlinear Parabolic Problems", *Nonlinear Theory and Its Applications, IEICE*, vol. 5, no. 3, pp. 320–338, 2014.

[80] Nakao, M. T., Kimura, T., and Kinoshita, T., "Constructive A Priori Error Estimates for A Full Discrete Approximation of The Heat Equation", *SIAM Journal on Numerical Analysis*, vol. 51, no. 3, pp. 1525–1541, 2013.

[81] Nakao, M. T., Plum, M., and Watanabe, Y., *Numerical Verification Methods and Computer-Assisted Proofs for Partial Differential Equations.* Singapore, Springer Singapore, 2019.

[82] Naldi, G., Pareschi, L., and Toscani, G., *Mathematical modeling of collective behavior in socio-economic and life sciences.* Basel, Switzerland, Springer Science & Business Media, 2010.

[83] Neher, M., Jackson, K. R., and Nedialkov, N. S., "On Taylor Model Based Integration of ODEs", *SIAM Journal on Numerical Analysis*, vol. 45, no. 1, pp. 236–262, 2007.

[84] Ozaki, K. and Ogita, T., "The Essentials of Verified Numerical Computations, Rounding Error Analyses, Interval Arithmetic, and Error-Free Transformations", *Nonlinear Theory and Its Applications, IEICE*, vol. 11, no. 3, pp. 279–302, 2020.

[85] Pivato, M., *Linear Partial Differential Equations and Fourier Theory.* Cambridge, England, Cambridge University Press, 2010.

[86] Popova, E. D., "Extended Interval Arithmetic in IEEE Floating-Point Environment", *Interval Computations*, vol. 4, pp. 100–129, 1994.

[87] Popova, E. D., "Solvability of Parametric Interval Linear Systems of Equations and Inequalities", *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 615–633, 2015.

[88] Rump, S. M., "On the Solution of Interval Linear Systems", *Computing*, vol. 47, no. 3-4, pp. 337–353, 1992.

[89] Rump, S. M., "Verification Methods: Rigorous Results Using Floating-Point Arithmetic", in *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pp. 3–4, 2010.

[90] Sekine, K., Nakao, M. T., and Oishi, S., "A New Formulation Using the Schur Complement for the Numerical Existence Proof of Solutions to Elliptic Problems: Without Direct Estimation for an Inverse of the Linearized Operator", *Numerische Mathematik*, vol. 146, no. 4, pp. 907–926, 2020.

[91] Shokin, Y. I. and Kalmykov, S., *Interval Analysis.* Novosibirsk, Nauka, 1981.

[92] Smailbegovic, F., Gaydadjiev, G. N., and Vassiliadis, S., "Sparse Matrix Storage Format", in *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing*, (Utrecht), pp. 445–448, Dutch Technology Foundation, 2005.

[93] Stakgold, I., *Boundary Value Problems of Mathematical Physics: Volume 1.* Newark, Delaware, SIAM, 2000.

[94] Strikwerda, J. C., *Finite Difference Schemes and Partial Differential Equations.* Madison, Wisconsin, SIAM, 2004.

[95] Sunaga, T., "Theory of An Interval Algebra and Its Application To Numerical Analysis", *RAAG memoirs*, vol. 2, no. 29-46, p. 209, 1958.

[96] Tapaswini, S. and Chakraverty, S., "New Midpoint-Based Approach for the Solution of N-th Order Interval Differential Equations", *Reliable Computing*, vol. 2, pp. 25–44, 2014.

[97] Turing, A. M., "Rounding-off Errors in Matrix Processes", *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 1, no. 1, pp. 287–308, 1948.

[98] Werschulz, A. G., *The Computational Complexity of Differential and Integral Equations: An Information-Based Approach.* Oxford, England, Oxford University Press, Inc., 1991.

[99] Wilkinson, J. H., *Rounding Errors in Algebraic Processes.* Mineola, New York, Dover Publications, 1994.

[100] Yan, B., "Introduction to Variational Methods in Partial Differential Equations and Applications", *A summer course at Michigan State University (Math 890, Summer 2008)*, 2008.

[101] Zhang, J., "Multigrid Method and Fourth-Order Compact Scheme for 2D Poisson Equation with Unequal Mesh-Size Discretization", *Journal of Computational Physics*, vol. 179, no. 1, pp. 170–179, 2002.