CMST 22(4) 225-232 (2016) DOI:10.12921/cmst.2016.0000048

Solving the Generalized Poisson Equation in Proper and Directed Interval Arithmetic

T. Hoffmann[1], **A. Marciniak**[2,3]

[1] *Poznan Supercomputing and Networking Center*
Jana Pawla II 10, 61-139 Poznań, Poland
E-mail: tomhof@man.poznan.pl

[2] *Institute of Computing Science, Poznan University of Technology*
Piotrowo 2, 60-965 Poznań, Poland
E-mail: Andrzej.Marciniak@put.poznan.pl

[3] *Department of Computer Science, Higher Vocational State School in Kalisz,*
Poznanska 201-205, 62-800 Kalisz, Poland

Received: 26 October 2016; revised: 09 December 2016; accepted: 09 December 2016; published online: 11 December 2016

Abstract: In the paper some interval methods for solving the generalized Poisson equation (GPE) are presented. The main aim of this work is focused on providing such algorithms for solving this type of equation that are able to store information about potentially made numerical errors inside the results. In order to cope with these assumptions the floating-point interval arithmetic is used. We proposed to use interval versions of the central-difference method for two types of interval arithmetic: proper and directed. In the experimental part of this paper both arithmetics for three examples of GPE are compared.

Key words: interval arithmetic, generalized Poisson equation, central-difference method

I. INTRODUCTION

As it is well-known, there are two kinds of errors caused by floating-point arithmetic used on modern computers: representation errors and rounding errors. Representation errors (for real numbers) occur already at the beginning of computations and they are propagated further. Rounding errors occur during each floating-point operation. When we apply approximate methods to solve problems on a computer, we introduce the third kind of errors - the errors of methods. In interval algorithms errors of this kind are included in the interval solutions obtained. If such algorithms are implemented in floating-point interval arithmetic we can obtain solutions in the form of intervals which contain all possible numerical errors.

About a dozen years ago our team at Poznan University of Technology started research into interval methods for solving the initial value problem in ordinary differential equations with their implementation in floating-point interval arithmetic. On the basis of the theory initiated by Shokin [1] we studied in details explicit and implicit interval methods of Runge-Kutta type (see e.g. [2-7]), and explicit and implicit multistep interval methods, including those of Adams-Bashforth, Adams-Moulton, Nyström and Milne-Simpson types (see e.g. [8-12]). Our studies have been summarized in [13] and especially in [14], where we have also considered a problem of computational complexities of interval methods presented. Other interval methods for solving the initial value problem in ordinary differential equations include among other the Moore method [15, 16], a method of Krückeberg [17], the Shokin method [1], and a variety of the interval methods based on the high-order Taylor series (a traditional method of this kind is presented in [18]).

In partial differential equations (PDE) we have three kinds of equations: elliptic (e.g. the Poisson equation), parabolic (e.g. the diffusion equation) and hyperbolic (e.g. the wave equation), and we consider the boundary or initial-boundary conditions. A very interesting theory to evaluate and verify the numerical solution of PDE can be found in a number of papers of Nakao and others (see e.g. [19-21]).

Our approach is different and rather simple. For some PDEs we have presented it in [22-27].

This paper is devoted to interval methods for solving some kind of elliptic PDEs. The elliptic PDEs often arise in different areas of physics such as electric fields or fluid dynamics. However, the analytical solution can be found only for some forms of these equations, and in most cases the only way to solve them is to find the numerical solutions, usually in floating-point arithmetic. The classical methods of solving elliptic PDEs are presented, among others, in [28] and [29]. Although we have information about the order of method error, finding the boundaries for error term (based on Taylor series) and the examination of rounding effect are also important. Thus, we propose to use interval arithmetic (see: [30, 31]). In this paper we confine our investigation to the equations given by the following formula:

$$a(x,y)\frac{\partial^2 u}{\partial x^2}(x,y)+b(x,y)\frac{\partial^2 u}{\partial y^2}(x,y)=f(x,y), \tag{1}$$

where

$$a(x,y)\cdot b(x,y)>0. \tag{2}$$

To obtain the unique solution of (1) additional constraints must be taken into account, and we apply the Dirichlet boundary conditions:

$$\begin{gathered} u(x,y)=\varphi(x,y),\ \text{for all } (x,y)\in\Gamma,\\ \Gamma=\{(x,y):(x=\alpha_1,\alpha_2\ \wedge\ \beta_1\le y\le\beta_2)\vee\\ (\alpha_1\le x\le\alpha_2\wedge\ y=\beta_1,\beta_2)\}, \end{gathered} \tag{3}$$

and

$$u|_\Gamma=\varphi(x,y)=\begin{cases}\varphi_1(y) \text{ for } x=\alpha_1,\\ \varphi_2(x) \text{ for } y=\beta_1,\\ \varphi_3(y) \text{ for } x=\alpha_2,\\ \varphi_4(x) \text{ for } y=\beta_2.\end{cases} \tag{4}$$

Hereafter the equation (1) is called the generalized Poisson equation (GPE). The topic of solving this type of elliptic PDEs in floating-point arithmetic has been discussed e.g. in [32]. Known are also interval solutions presented in [33] and [34]. However, our approach is substantially different. We propose the interval method in which the obtained results gather the information about method errors and rounding errors. The details of our approach, based on the central-difference method, are presented in the next sections.

II. THE CENTRAL-DIFFERENCE METHOD

The first step of the method is to define a grid of mesh points on the rectangle bounded by conditions (3). The interval $[\alpha_1,\alpha_2]$ is partitioned into m equal parts of the width $h=\frac{\alpha_2-\alpha_1}{m}$, and the interval $[\beta_1,\beta_2]$ is partitioned into n equal parts of the width $k=\frac{\beta_2-\beta_1}{n}$, respectively. Next, all mesh points are defined as $(x_i,y_j)=(ih,jk)$. Then, assuming that for each internal mesh point there exist fourth order partial derivatives of the function $u(x,y)$, we use the Taylor series for variables x and y to obtain

$$\begin{aligned} &a(x_i,y_j)\cdot\Big[\frac{u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_j)}{h^2}\\ &-\frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i,y_j)\Big]+b(x_i,y_j)\\ &\cdot\Big[\frac{u(x_i,y_{j+1})-2u(x_i,y_j)+u(x_i,y_{j-1})}{k^2}\\ &-\frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i,\eta_j)\Big]\\ &=f(x_i,y_j), \end{aligned} \tag{5}$$

where $\xi_i\in(x_{i-1},x_{i+1}),\eta_j\in(y_{j-1},y_{j+1})$. The boundary conditions are given by

$$\begin{aligned} u(\alpha_1,y_j)&=\varphi_1(y_j), \text{ for each } j=0,1\cdots,m,\\ u(x_i,\beta_1)&=\varphi_2(x_i), \text{ for each } i=1,2\cdots,n-1,\\ u(\alpha_2,y_j)&=\varphi_3(y_j), \text{ for each } j=0,1\cdots,m,\\ u(x_i,\beta_2)&=\varphi_4(x_i), \text{ for each } i=1,2\cdots,n-1. \end{aligned} \tag{6}$$

Omitting in (5) the partial derivatives results in a method with local truncation error of order $O(h^2+k^2)$. Simplifying the notation by $u(x_i,y_j)=u_{i,j}, a(x_i,y_j)=a_{i,j}$ and $b(x_i,y_j)=b_{i,j}$ the central-difference method for the GPE can be rewritten in the form

$$\begin{aligned} &a_{i,j}\cdot\left(\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}\right)\\ &+b_{i,j}\cdot\left(\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{k^2}\right)=f_{i,j}. \end{aligned} \tag{7}$$

III. INTERVAL CENTRAL-DIFFERENCE METHODS

In our approach the main goal is that the error term should be included into the solution (see: [26, 27]). Thus, we rewrite the GPE given by (5) in two following forms:

$$\begin{aligned} &a(x_i,y_j)\cdot\left[\frac{u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_j)}{h^2}\right]\\ &+b(x_i,y_j)\cdot\left[\frac{u(x_i,y_{j+1})-2u(x_i,y_j)+u(x_i,y_{j-1})}{k^2}\right]\\ &-a(x_i,y_j)\cdot\frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i,y_j)\\ &-b(x_i,y_j)\cdot\frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i,\eta_j)\\ &=f(x_i,y_j), \end{aligned} \tag{8}$$

and

$$a(x_i, y_j) \cdot \left[\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2}\right]$$
$$+b(x_i, y_j) \cdot \left[\frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2}\right] =$$
$$=f(x_i, y_j) + a(x_i, y_j) \cdot \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, y_j)$$
$$+b(x_i, y_j) \cdot \frac{k^2}{12}\frac{\partial^4 u}{\partial y^4}(x_i, \eta_j). \tag{9}$$

Let us assume that there exist constants M and N such that

$$\left|\frac{\partial^4 u}{\partial x^4}(x, y)\right| \leq M \quad \text{for all } \alpha_1 \leq x \leq \alpha_2 \ \wedge \ \beta_1 \leq y \leq \beta_2, \tag{10}$$
$$\left|\frac{\partial^4 u}{\partial y^4}(x, y)\right| \leq N \quad \text{for all } \alpha_1 \leq x \leq \alpha_2 \ \wedge \ \beta_1 \leq y \leq \beta_2.$$

In general, when constants M and N cannot be determined from the physical or technical properties of the problem, we propose a method of evaluating M and N constants, which is based on results in floating-point arithmetic. It is possible to use the following central-difference formula in order to find values of fourth order derivatives:

$$M_h =$$
$$=\max_{i,j}\left\{\frac{6u_{i,j} - 4u_{i-1,j} - 4u_{i+1,j} + u_{i-2,j} + u_{i+2,j}}{h^4}\right\},$$
$$N_k =$$
$$=\max_{i,j}\left\{\frac{6u_{i,j} - 4u_{i,j-1} - 4u_{i,j+1} + u_{i,j-2} + u_{i,j+2}}{k^4}\right\}, \tag{11}$$

where $u_{i,j}$ can be obtained from solving the system of linear equations given by (7) and using floating-point arithmetic. Then, we have the following approximation of M and N:

$$M = \lim_{h\to 0} M_h, \qquad N = \lim_{k\to 0} N_k. \tag{12}$$

It is obvious that $h \to 0$ and $k \to 0$, when $m \to \infty$ and $n \to \infty$, respectively. It means that increasing the grid size we can get M and N experimentally.

Let us denote by $A(X, Y), B(X, Y)$ and $U(X, Y)$ the interval extensions (for definition see [1, 16] or [30]) of functions $a(x, y), b(x, y)$ and $u(x, y)$, respectively. Thus we can use short forms:

$$A_{i,j} = A(X_i, Y_j),$$
$$B_{i,j} = B(X_i, Y_j),$$
$$U_{i,j} = U(X_i, Y_j).$$

Therefore, using the equation (9) and the constants M and N, the following interval extension for proper interval arithmetic can be written:

$$k^2A_{i,j}U_{i+1,j} + h^2B_{i,j}U_{i,j+1} - 2(k^2A_{i,j} + h^2B_{i,j})U_{i,j}$$
$$+k^2A_{i,j}U_{i-1,j} + h^2B_{i,j}U_{i,j-1} =$$
$$=h^2k^2\left\{F_{i,j} + \frac{h^2A_{i,j}}{12}[-M, M] + \frac{k^2B_{i,j}}{12}[-N, N]\right\} \tag{13}$$

However, in the directed one we can use (8):

$$k^2A_{i,j}U_{i+1,j} + h^2B_{i,j}U_{i,j+1} - 2(k^2A_{i,j} + h^2B_{i,j})U_{i,j}$$
$$+k^2A_{i,j}U_{i-1,j} + h^2B_{i,j}U_{i,j-1} - \frac{h^4A_{i,j}}{12}[-M, M]$$
$$-\frac{k^4B_{i,j}}{12}[-N, N] = h^2k^2F_{i,j}, \tag{14}$$

and adding to both sides the opposite element (e.g. [35] [36]), we obtain:

$$k^2A_{i,j}U_{i+1,j} + h^2B_{i,j}U_{i,j+1} - 2(k^2A_{i,j} + h^2B_{i,j})U_{i,j}$$
$$+k^2A_{i,j}U_{i-1,j} + h^2B_{i,j}U_{i,j-1} =$$
$$=h^2k^2\left\{F_{i,j} + \frac{h^2A_{i,j}}{12}[M, -M] + \frac{k^2B_{i,j}}{12}[N, -N]\right\}. \tag{15}$$

In order to find solutions in proper and directed interval arithmetic for all mesh points it is necessary to solve the system of interval linear equations given by (13) and (15), respectively. It is worth noting that the existence of the opposite element is one of the most important differences between proper and directed interval arithmetic. Furthermore, the realization of the basic arithmetic operations is significantly various (see e.g. [31] and [36]). In the next section we compare the results in both arithmetics.

IV. NUMERICAL EXPERIMENTS

In this section we present three examples of the GPE. In the first two examples the exact solutions are known and the purpose of these experiments is to show experimentally that the exact solution is placed within the interval result obtained. The additional aim of presented examples is to compare the widths of intervals in proper and directed interval arithmetic. In the third experiment our goal is to present the methodology of solving the GPEs when the constants M and N are unknown, i.e. it is not possible to obtain that constants on the basis of the problem formulation. We suggest that if the experimentally obtained values of constant tends to a limit we can use that limit as the value of constant and use it in our interval versions of the central-difference method.

IV. 1. Example 1

The goal of the first example is to prove experimentally the correctness of proposed interval methods, in particular

Tab. 1. The results in proper interval arithmetic at $x = y = 1.5$. The exact solution is $u(1.5, 1.5) = 2.3714825526419476e - 01$

$m = n$	U_p	width(U_p)
20	[2.3706842889509303e-01; 2.3720354321116375e-01]	1.3511431607070813e-04
30	[2.3711274662188439e-01; 2.3717286218796079e-01]	6.0115566076390789e-05
40	[2.3712827567900367e-01; 2.3716210346652419e-01]	3.3827787520512100e-05
50	[2.3713546655808877e-01; 2.3715712013201293e-01]	2.1653573924154042e-05
60	[2.3713937354963539e-01; 2.3715441218408559e-01]	1.5038634450192429e-05
70	[2.3714172962658439e-01; 2.3715277905304529e-01]	1.1049426460891336e-05
80	[2.3714325892563301e-01; 2.3715171895776956e-01]	8.4600321365399674e-06
90	[2.3714430746011356e-01; 2.3715099210060813e-01]	6.6846404945667998e-06
100	[2.3714505749665764e-01; 2.3715047215434425e-01]	5.4146576865966704e-06

that the exact solution is included in result intervals. Let us consider the following example:

$$f(x,y) = xy(x+y)(xy-3),$$
$$a(x,y) = ye^{\frac{x^2+y^2}{2}}, \qquad (16)$$
$$b(x,y) = xe^{\frac{x^2+y^2}{2}},$$

for $x, y \in [1,2]$, with boundary conditions given by

$$\varphi_1(y) = ye^{-\frac{1+y^2}{2}} \quad \varphi_2(x) = xe^{-\frac{1+x^2}{2}},$$
$$\varphi_3(y) = 2ye^{-\frac{4+y^2}{2}} \quad \varphi_4(x) = 2xe^{-\frac{4+x^2}{2}}. \qquad (17)$$

The exact solution of (16)-(17) is known and given by (see Fig. 1)

$$u(x,y) = xye^{-\frac{x^2+y^2}{2}}. \qquad (18)$$

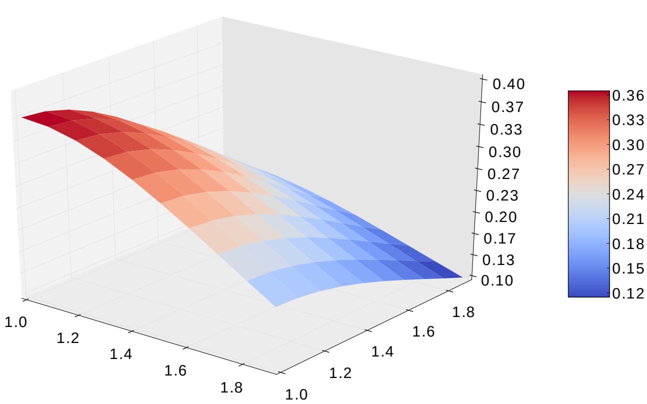

Fig. 1. The exact solution given by (18)

Using (18), from (10) it follows that

$$M = N = 2.2073.$$

The results of computations are presented in Tables 1 and 2. In both interval arithmetics the exact solutions are inside the obtained intervals. It is worth noteing that the scale of the width of the result interval allows us to determine how many decimal digits of the exact solution are obtained exactly. We can be certain that all errors we have made summarized together are not greater than the width of the interval.

Moreover, as shown in Fig. 2, the difference of the width of intervals in proper and directed interval arithmetic is increasing with the growth of grid size.

IV. 2. Example 2

The aim of the second example is closer examination of the relative position of the exact solutions inside the result intervals. We define the relative position p of the solution s in the result interval $A = [\underline{a}, \bar{a}]$ as

$$p(s) = \frac{|s - \mathrm{mid}(A)|}{\mathrm{widht}(A)}, \qquad (19)$$

where $\mathrm{mid}(A) = \frac{\bar{a}+\underline{a}}{2}$ and $\mathrm{widht}(A) = \bar{a} - \underline{a}$.

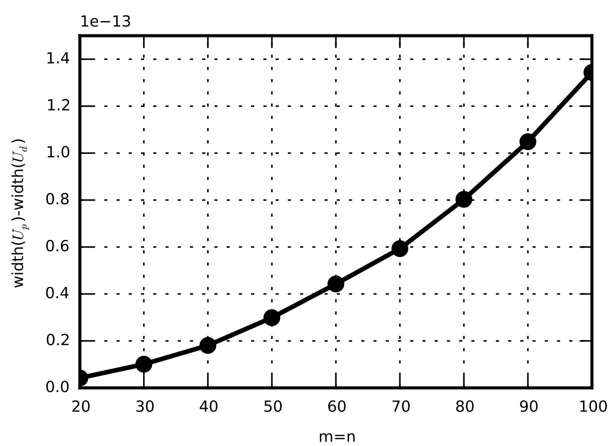

Fig. 2. The difference of the width of solutions for $(x,y) = (1.5, 1.5)$ in the proper (U_p) and directed (U_d) interval arithmetic

The value of $p(s)$ determines when the solution s lies inside the interval A. We have

$$p(s) = \begin{cases} (\frac{1}{2}, +\infty) \text{ for } s \notin A, \\ < 0, \frac{1}{2} > \text{ for } s \in A. \end{cases} \qquad (20)$$

Tab. 2. The results in directed interval arithmetic at $x = y = 1.5$. The exact solution is $u(1.5, 1.5) = 2.371482552641947578e - 01$

$m = n$	U_d	width(U_d)
20	[2.3706842889509324e-01; 2.3720354321116354e-01]	1.3511431607028675e-04
30	[2.3711274662188484e-01; 2.3717286218796034e-01]	6.0115566075489330e-05
40	[2.3712827567900447e-01; 2.3716210346652339e-01]	3.3827787518916087e-05
50	[2.3713546655809004e-01; 2.3715712013201166e-01]	2.1653573921609203e-05
60	[2.3713937354963727e-01; 2.3715441218408377e-01]	1.5038634446434287e-05
70	[2.3714172962658697e-01; 2.3715277905304277e-01]	1.1049426455730046e-05
80	[2.3714325892563648e-01; 2.3715171895776609e-01]	8.4600321295948394e-06
90	[2.3714430746011806e-01; 2.3715099210060363e-01]	6.6846404855683284e-06
100	[2.3714505749666336e-01; 2.3715047215433854e-01]	5.4146576751736519e-06

Let us take into account the following equation:

$$f(x,y) = x^2y^2(3y^2 + 2x^2y^2 - 3x^2),$$
$$a(x,y) = xy^3 e^{-\frac{x^2+y^2}{2}}, \quad (21)$$
$$b(x,y) = x^3 y e^{-\frac{x^2-y^2}{2}},$$

for $x, y \in [1, 2]$ and with the boundary conditions

$$\varphi_1(y) = ye^{\frac{1-y^2}{2}} \quad \varphi_2(x) = xe^{\frac{x^2-1}{2}},$$
$$\varphi_3(y) = 2ye^{\frac{4-y^2}{2}} \quad \varphi_4(x) = 2xe^{\frac{x^2-4}{2}}, \quad (22)$$

and exact solution

$$u(x,y) = xye^{\frac{x^2-y^2}{2}}. \quad (23)$$

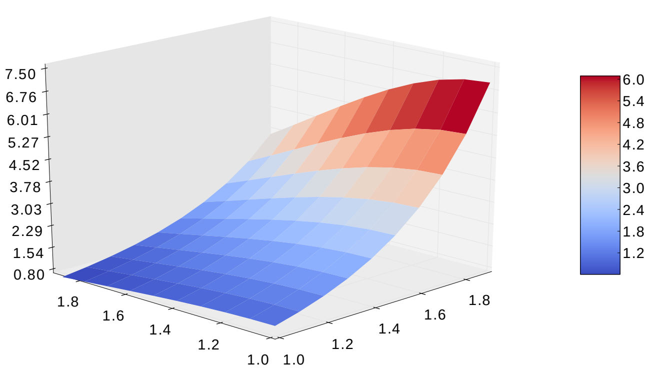

Fig. 3. The exact solution given by (23)

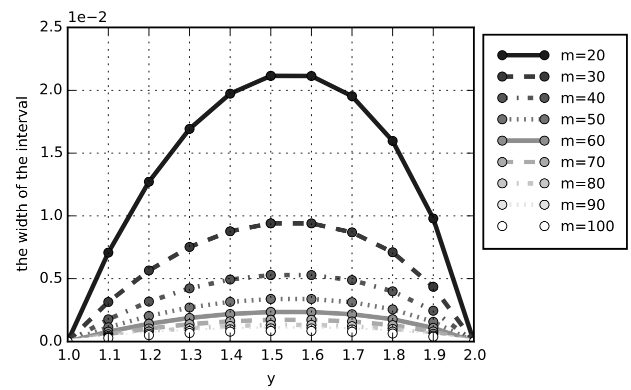

Fig. 4. The widths of intervals obtained in proper interval arithmetic at the point $x = 1.5$ for the problem given by (21)

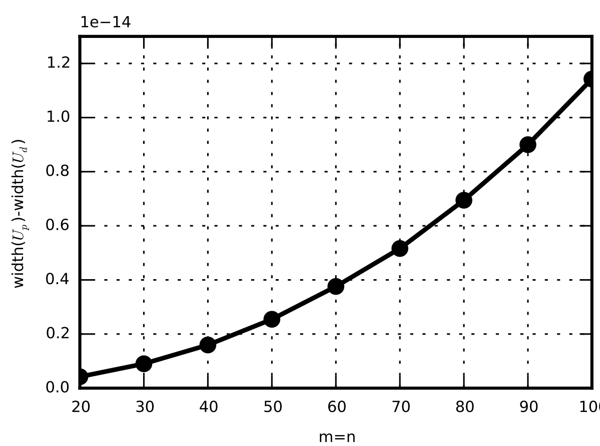

Fig. 5. The difference of the width of solutions for $(x, y) = (1.5, 1.5)$ in the proper (U_p) and directed (U_d) interval arithmetic

Fig. 4 shows that increasing the size of grid results in narrower intervals in both arithmetics. The narrower interval means the more precise estimation of the exact solution location. Thus, it is experimentally shown that increasing the size of grid results in more accurate solutions.

The difference between widths of intervals in proper and directed arithmetic is increasing, as shown in Fig. 5. We have observed the similar effect in the previous example.

In Fig. 6 we present the relative position of the exact and floating-point solutions inside obtained intervals in proper

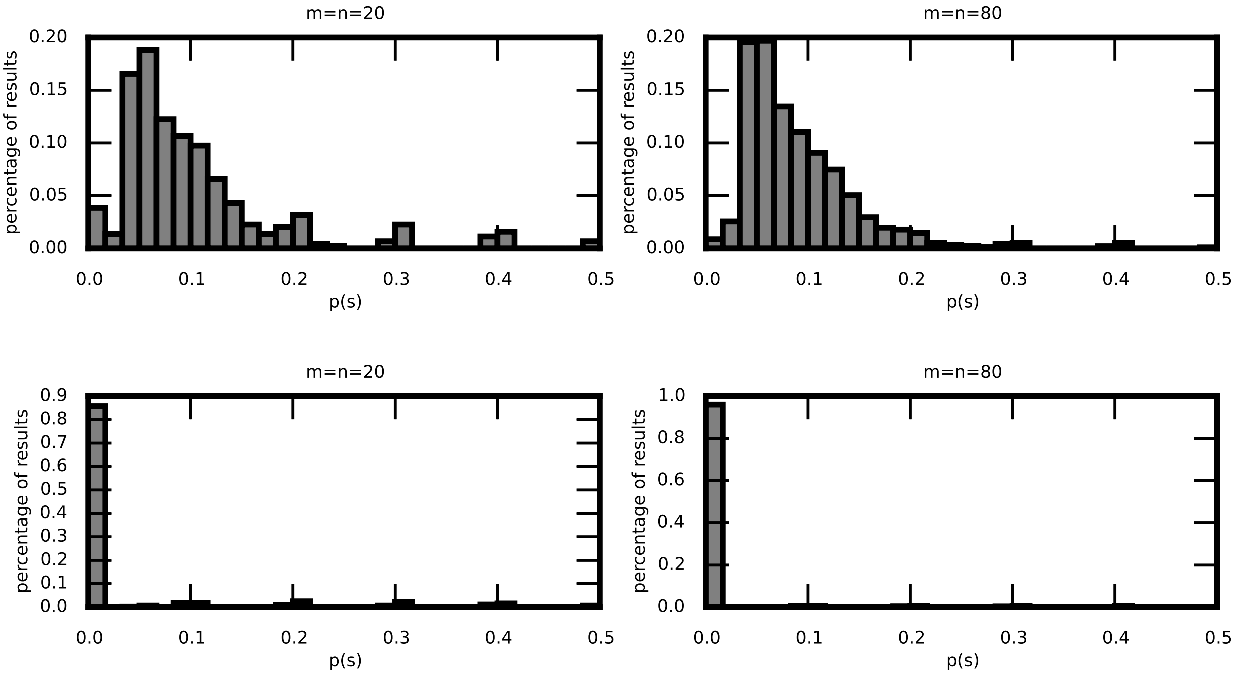

Fig. 6. The relative position of the exact (first row) and floating-point (second row) solution of the problem given by (21) in the proper interval arithmetic

interval arithmetic. The results confirmed that all exact solutions are situated inside intervals. Furthermore, with the growth of the grid size we observe that the distribution of position of exact solutions is changed, and in the greater number of intervals the exact solution lies near the middle of the interval. Also, as we can expect, the floating-point solution lies almost in the middle of intervals and the dispersion of results is negligible. For directed interval arithmetic similar results are obtained.

IV. 3. Example 3

In the last example let us consider the following problem:

$$f(x,y) = \frac{1}{xy} \cdot \cos(y\frac{\pi}{2}),$$
$$a(x,y) = e^{\cos(x\pi)-\cos(y\pi)}, \quad (24)$$
$$b(x,y) = e^{x-y}.$$

Boundary conditions are given by

$$\varphi_1(y) = \sin[(y-1)\pi],\ \varphi_2(x) = \sin[(x-1)\pi],$$
$$\varphi_3(y) = \sin[(2-y)\pi],\ \varphi_4(x) = \sin[(2-x)\pi]. \quad (25)$$

and $x, y \in [1, 2]$.

This example presents a case when the exact solution is unknown, and we have no knowledge about the problem. Therefore, we cannot determine the boundaries for the fourth order partial derivatives from their physical descriptions.

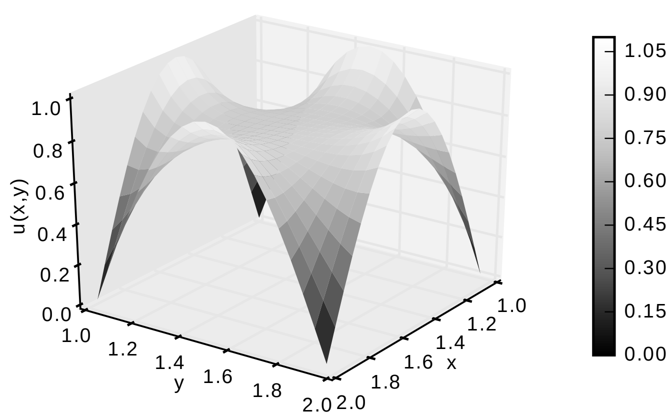

Fig. 7. The solution in floating-point arithmetic for problem given by (24)

In Fig. 7 we present the solutions obtained in floating point arithmetic. In order to find out boundaries for fourth order partial derivatives, we have to evaluate constants M and N experimentally. In our approach we propose to use the formula (11). Thus, in the first step we solve the problem in floating-point arithmetic iteratively for ascending gird sizes $m = n$. Next, based on the obtained results, we evaluate the values of M and N. The values of the constant M, obtained for different $m = n$ from (11), are presented in Fig. 8. For the constant N the graph is similar.

Tab. 3. The results in directed interval arithmetic for $u(1.5, 1.5)$

$m = n$	U_d	width(U_d)
20	[7.6035722887164602E-1; 8.0106890385674469E-1]	4.0711674985098666e-02
30	[7.7033069149409466E-1; 7.8844461621498338E-1]	1.8113924720888708e-02
40	[7.7382667640488374E-1; 7.8401967504352799E-1]	1.0192998638644243e-02
50	[7.7544574584253236E-1; 7.8197042596930942E-1]	6.5246801267770460e-03
60	[7.7632548668777912E-1; 7.8085695269542372E-1]	4.5314660076445852e-03
70	[7.7685602651955652E-1; 7.8018546099252008E-1]	3.3294344729635576e-03
80	[7.7720040104246144E-1; 7.7974959579752853E-1]	2.5491947550670805e-03
90	[7.7743651844285919E-1; 7.7945074928755821E-1]	2.0142308446990160e-03
100	[7.7760541949720153E-1; 7.7923697677012260E-1]	1.6315572729210667e-03

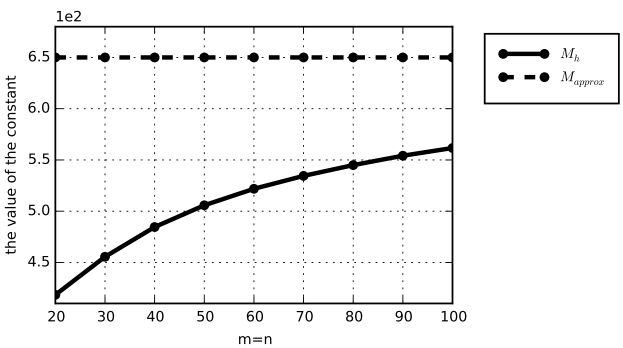

Fig. 8. The estimate of the constant M obtained in floating-point arithmetic

It is possible to estimate the values of constants M and N from the picture, and we assumed that $M_{\text{approx}} = 650$ and $N_{\text{approx}} = 675$ and such values were used in computation. The results obtained in directed interval arithmetic are presented in Tab. 3.

V. CONCLUSIONS

The most important conclusion is that the proposed method for solving the GPE gives the solutions in the form of intervals which contain all possible numerical errors, i.e. representation errors, rounding errors and error of the method. We have proposed the methodology of experimental evaluation of the method error (described by the constants M and N), which may be potentially very useful when our knowledge about the problem is restricted. Additionally, it has been experimentally confirmed that the exact solution is placed inside the resulting intervals in both tested arithmetics. Furthermore, the results pointed out that intervals in directed interval arithmetic are a little bit narrower, which means that estimation of the exact result position is more accurate.

References

[1] Yu. I. Shokin, *Interval Analysis*, Nauka, Novosibirsk, 1981.

[2] A. Marciniak, B. Szyszka, *One-and Two-Stage Implicit Interval Methods of Runge-Kutta Type*, CMST **5**, 53-65 (1999).

[3] K. Gajda, A. Marciniak, B. Szyszka, *Three-and Four-Stage Implicit Interval Methods of Runge-Kutta Type*, CMST **6**, 41-59 (2000).

[4] A. Marciniak, *Finding the Integration Interval for Interval Methods of Runge-Kutta Type in Floating-Point Interval Arithmetic*, Pro Dialog, **10**, 35-45 (2000).

[5] A. Marciniak, B. Szyszka, *On Representation of Coefficients in Implicit Interval Methods of Runge-Kutta Type*, CMST **10**(1), 57-71 (2004).

[6] A. Marciniak, *Implicit Interval Methods For Solving the Initial Value Problem*, Numerical Algorithms **37**(1-4), 241-251 (2004).

[7] A. Marciniak, *Symplectic Interval Methods for Solving Hamiltonian Problems*, Pro Dialog **22**, 27-37 (2007).

[8] M. Jankowska, A. Marciniak, *Implicit Interval Multistep Methods For Solving the Initial Value Problem*, CMST **8**(1), 17-30 (2002).

[9] M. Jankowska, A. Marciniak, *On Explicit Interval Methods of Adams-Bashforth Type*, CMST **8**(2), 46-57 (2002).

[10] M. Jankowska, A. Marciniak, *On Two Families Of Implicit Interval Methods Of Adams-Moulton Type*, CMST **12**(2), 109-113 (2006).

[11] A. Marciniak, *Multistep Interval Methods Of Nyström and Milne-Simpson Types*, CMST**13**(1), 23-40 (2007).

[12] A, Marciniak, *On Multistep Interval Methods For Solving the Initial Value Problem*, Journal of Computational and Applied Mathematics**199**(2) 229-237 (2007).

[13] K. Gajda, M. Jankowska, A. Marciniak, B. Szyszka, *A Survey Of Interval Runge-Kutta and Multistep Methods For Solving the Initial Value Problem*, in: *Parallel Processing and Applied Mathematics*, pages 1361-1371 Springer, 2008.

[14] A. Marciniak, *Selected Interval Methods for Solving the Initial Value Problem*, Publishing House of Poznań University of Technology, 2009.

[15] R.E. Moore, *The Automatic Analysis And Control Of Error In Digital Computation Based On the Use Of Interval Numbers*, Error in Digital Computation **1**, 61-130 (1965).

[16] R.E. Moore, *Interval Analysis*, volume 4, Prentice-Hall Englewood Cliffs, 1966.

[17] F. Krückeberg, *Ordinary Differential Equations*, Topics in Interval Analysis, pages 91-97, (1969).

[18] N.S. Nedialkov, Interval Tools for ODEs and DAEs, In *Scientific Computing, Computer Arithmetic and Validated Numerics, 2006. SCAN 2006*, page 4, IEEE, 2006.

[19] T. Kimura M.T. Nakao, T. Kinoshita, *A Priori Error Estimates for a Full Discrete Approximation of the Heat Equation*, SIAM Journal on Numerical Analysis**51**(3), 1525–1541, (2013).

[20] M.T. Nakao, *On Verified Computations Of Solutions for Nonlinear Parabolic Problems*, Nonlinear Theory and Its Applications, IEICE **5**(3), 320-338 (2014).

[21] T. Kinoshita, T. Kimura, and M.T. Nakao, *On the A Posteriori Estimates For Inverse Operators Of Linear Parabolic Equations With Applications to the Numerical Enclosure Of Solutions For Nonlinear Problems*, Numerische Mathematik **126**(4), 679-701 (2014).

[22] A. Marciniak, *An Interval Difference Method For Solving the Poisson Equation The First Approach*, Pro Dialog **24**, 49-61 (2008).

[23] A. Marciniak, An Interval Version of the Crank-Nicolson Method-The First Approach, In *Applied Parallel and Scientific Computing*, pages 120-126 Springer, 2012.

[24] B. Szyszka, The Central Difference Interval Method for Solving the Wave Equation, In *Parallel Processing and Applied Mathematics*, volume 7204 of *Lecture Notes in Computer Science*, pages 523-532 Springer Berlin Heidelberg, 2012.

[25] A. Marciniak, B. Szyszka, A Central-Backward Difference Interval Method for Solving the Wave Equation, In *Applied Parallel and Scientific Computing*, pages 518-527 Springer, 2013.

[26] T. Hoffmann, A. Marciniak, B. Szyszka, *Interval Versions of Central-Difference Method for Solving the Poisson Equation in Proper and Directed Interval Arithmetic*, Foundations of Computing and Decision Sciences **38**(3), 193-206 (2013).

[27] T. Hoffmann and A. Marciniak, *Solving the Poisson Equation by an Interval Difference Method of the Second Order*, CMST**19**(1), 13-21 (2013).

[28] Richard L Burden and J Douglas Faires, *Numerical Analysis*, Brooks/Cole, 2001.

[29] David Ronald Kincaid and Elliott Ward Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, volume 2, American Mathematical Soc., 2002.

[30] R. Baker Kearfott, *Interval Computations: Introduction, Uses, and Resources*, Euromath Bulletin **2**(1), 95-112 (1996).

[31] R.E Moore, R. Baker Kearfott, and M.J. Cloud, *Introduction to Interval Analysis*, SIAM, 2009.

[32] J.R. Nagel, *Solving the Generalized Poisson Equation Using the Finite-Difference Method (FDM)*, Lecture Notes, Dept. of Electrical and Computer Engineering, University of Utah, (2011).

[33] H. Schwandt, *The Solution of Nonlinear Elliptic Dirichlet Problems on Rectangles by Almost Globally Convergent Interval Methods*, SIAM Journal on Scientific and Statistical Computing **6**(3), 617-638 (1985).

[34] H. Schwandt, *Almost Globally Convergent Interval Methods for Discretizations of Nonlinear Elliptic Partial Differential Equations*, SIAM Journal on Numerical Analysis **23**(2), 304-324 (1986).

[35] S. Markov, *On Directed Interval Arithmetic and Its Applications*, in: *The Journal of Universal Computer Science*, pages 514-526 Springer, 1996.

[36] E.D. Popova, *Extended Interval Arithmetic in IEEE Floating-Point Environment*, Interval Comput. **4**, 100-129 (1994).

Andrzej Marciniak was born in Poznan (Poland) in 1953. He received the MSc degree in Mathematics in 1977, the MSc degree in Astronomy in 1979 and PhD degree in Mathematics in 1981, all from the Adam Mickiewicz University in Poznan. In 1993 he received the Dr. Habil. degree in Physics from the Nicolaus Copernicus Univeristy in Torun (Poland) and in 2010 he received the Professor Title from the President of Poland. From 1977 to 1987 and from 2000 to 2011 he held a research position at the Faculty of Mathematics and Computer Science of the Adam Mickiewicz University, and since 1987 he has been an Assistant Professor at the Institute of Mathematics and then a Professor of Computer Science at the Faculty of Computing Science of Poznan University of Technology. From 2005 to 2008 he held the office of the President of Polish Information Processing Society. His research interests include computer programming and numerical methods, especially for solving ordinary and partial defferential equations with applications to dynamical problems. In these fields he wrote three monographs, more than 20 textbooks and a number of scientific articles.

Tomasz Hoffmann was born in Piotrkow Trybunalski, Poland, in 1986. He is a third year PhD student in Computer Science at Poznan University of Technology. In 2010 he received his MSc degree in Computer Science and BSc degree in Mathematics at Poznan University of Technology. His main areas of interest are: numerical methods, machine learning and information retrieval.

 DOI:10.12921/cmst.2016.0000048