Solving multi-connected BVPs with uncertainly defined complex shapes

Andrzej Kużelewski[1][0000−0003−2247−2714], Eugeniusz Zieniuk[1][0000−0002−6395−5096], and Marta Czupryna[1][0000−0003−0156−9904]

Faculty of Computer Science, University of Bialystok
Ciolkowskiego 1M, 15-245 Bialystok, Poland
{a.kuzelewski,e.zieniuk,m.czupryna}@uwb.edu.pl

Abstract. The paper presents the interval fast parametric integral equations system (IFPIES) applied to solve multi-connected boundary value problems (BVPs) with uncertainly defined complex shapes of a boundary. The method is similar to the fast PIES, which uses the fast multipole method to speed up solving BVPs and reduce RAM utilization. However, modelling uncertainty in the IFPIES uses interval numbers and directed interval arithmetic. Segments created the boundary have the form of the interval Bézier curves of the third degree (curvilinear segments) or the first degree (linear segment). The curves also required some modifications connected with applied directed interval arithmetic. In the presented paper, the reliability and efficiency of the IFPIES solutions were verified on multi-connected BVPs with uncertainly defined complex linear and curvilinear domain shapes. The solutions were compared with the ones obtained by the interval PIES only due to the lack of examples of solving uncertainly defined BVPs in the literature. All presented tests confirm the high efficiency of the IFPIES method.

Keywords: Interval fast parametric integral equations system · Interval numbers · Directed interval arithmetic · Uncertainty

1 Introduction

For many years, our team has worked on developing and applying a parametric integral equations system (PIES) to solve boundary value problems (BVPs). The multidirectional research covers problems described by different equations, such as Laplace's, Helmholtz or Navier-Lamé (i.e. [1]). On the other hand, some enhancements of the method are also considered - the authors of this paper are focused on two of them: application of uncertainty of data in the PIES [2] and accelerating performance and reducing memory utilization of the PIES [3].

Traditional modelling and solving BVPs assumes that boundary conditions, the boundary's shape, and the domain's parameters must be defined by real numbers, i.e., precisely. However, the practice indicates that to obtain the mentioned data, some measurements should be carried out, which are always affected by, e.g. gauge reading error or inaccuracy of measurement instruments. Sometimes, the approximation of the model used in the analysis of measurements

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

may also cause errors. On the other hand, we should assume that the modelled components will be manufactured later with a certain margin of error, which was not considered during the modelling process. Therefore, considering data uncertainty in modelling and solving BVPs becomes a critical problem.

We must emphasize that classical mathematical models require exact input data values. Therefore, it is not possible to apply uncertainty directly. However, many authors modified known methods to consider uncertainty (e.g. [4–6]). The efficient way of using uncertain data in modelling and solving BVPs is the application of interval arithmetic and interval numbers. It has resulted in obtaining the interval finite element method (IFEM) [7], the interval boundary element method (IBEM) [8], and also the interval version of the PIES (IPIES) [9]. However, both the IFEM and the IBEM considered only the uncertainty of boundary conditions or material parameters. Only in a few papers describing 1D problems were the boundary shape parameters, such as beam length, uncertainly defined. The IPIES is more complete - it was developed to solve problems with uncertainly defined boundary shapes. The opportunity to consider all the uncertainties mentioned above in the IPIES is a significant advantage.

The IPIES has advantages inherited from the PIES, such as defining the boundary shape by curves widely used in computer graphics (small amount of input data) and approximating boundary conditions separated from the approximation of boundary shape. Unfortunately, it also has disadvantages connected with the PIES coefficient matrices, which are dense and non-symmetric. The way of creating these matrices requires to compute slightly complicated integrals. That process requires a lot of CPU time, especially for problems with a considerable number of segments describing the shape of the boundary (complex or large-scale problems). Unfortunately, applying interval arithmetic and interval numbers also negatively affects the computational speed and utilizes more memory (RAM).

Another problem is the method of solving the final system of algebraic equations. In traditional PIES, classical Gaussian elimination was first applied. The authors of this paper adapted the specialized libraries (such as LAPACK) to solve the system more efficiently. Also, parallelization of the PIES by OpenMP and CUDA to reduce the time of computations was proposed in our previous papers (e.g. [10, 11]. However, we must still store all coefficient matrices in the operating memory. Therefore, RAM consumption is at a high level.

In the mid-1980s of the 20th century, Rokhlin and Greengard developed the fast multipole method (FMM) [12]. This compression technique reduces the utilization of RAM and computational time that is well-documented in solving potential BVPs (i.e. [13]), also in the BEM [14]. Applying the FMM to the PIES required a new approach to computing matrices coefficients and the iterative method for solving the system of algebraic equations. Also, we had to make additional changes in the FMM tree structure[15]. However, obtained that way, the fast PIES (FPIES) is an efficient method for solving BVPs [3].

The main goal of this paper is to present the interval fast Parametric Integral Equation System (IFPIES) applied for numerical solving of 2D uncertainly

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

defined multi-connected potential complex BVPs. The IFPIES previously proposed in [16] for single-connected polygonal domains is based on both the IPIES and the FPIES. In the presented paper, the efficiency and accuracy of the IFPIES are tested on the examples of multi-connected complex uncertainly defined domains described by linear and curvilinear segments.

2 Boundary shape and boundary conditions uncertainty

The application of classical [17] and directed [18] interval arithmetic into the PIES was troublesome, as is described in previous papers (e.g. [9]). Therefore, we proposed mapping arithmetic operators to the positive semi-axis (clearly described in [9]) while applying the directed interval arithmetic to obtain the IPIES. The same strategy was also applied in the IFPIES.

The general formula of the IFPIES [3] has the following form:

$$\begin{gathered}\frac{1}{2}u_l(\widehat{\boldsymbol{\tau}}) = \sum_{j=1}^{n} \mathbb{R}\left\{ \int\limits_{s_{j-1}}^{s_j} \widehat{\boldsymbol{U}}_{lj}^{*(c)}(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau}) \boldsymbol{p}_j(s) \boldsymbol{J}_j^{(c)}(s) ds \right\} - \\ \sum_{j=1}^{n} \mathbb{R}\left\{ \int\limits_{s_{j-1}}^{s_j} \widehat{\boldsymbol{P}}_{lj}^{*(c)}(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau}) \boldsymbol{u}_j(s) \boldsymbol{J}_j^{(c)}(s) ds \right\}, \\ l = 1, 2, ..., n,\ s_{l-1} \leq \widehat{s} \leq s_l,\ s_{j-1} \leq s \leq s_j,\end{gathered} \tag{1}$$

where: $\widehat{s}$ and s are defined in the parametric coordinate system, s_{j-1} (s_{l-1}) correspond to the beginning while s_j (s_l) to the end of interval segment $\boldsymbol{S}_j$ ($\boldsymbol{S}_l$), n is the number of parametric segments that creates a boundary of the domain in 2D, $\boldsymbol{J}_j^{(c)}(s)$ is the interval Jacobian, $\boldsymbol{u}_j(s)$ and $\boldsymbol{p}_j(s)$ are interval parametric boundary functions on individual segments $\boldsymbol{S}_j$ of the interval boundary, $\mathbb{R}$ is the real part of complex function.

The interval kernels modified to complex functions $\widehat{\boldsymbol{U}}_{lj}^{*(c)}(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau})$ and $\widehat{\boldsymbol{P}}_{lj}^{*(c)}(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau})$ have the following form [3]:

$$\begin{aligned}\widehat{\boldsymbol{U}}_{lj}^{*(c)}(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau}) &= -\frac{1}{2\pi} \ln\left(\widehat{\boldsymbol{\tau}} - \boldsymbol{\tau}\right), \\ \widehat{\boldsymbol{P}}_{lj}^{*(c)}(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau}) &= \frac{1}{2\pi} \frac{\boldsymbol{n}_j^{(c)}}{\widehat{\boldsymbol{\tau}} - \boldsymbol{\tau}}.\end{aligned} \tag{2}$$

where $\boldsymbol{n}_j^{(c)}$ is the complex notation of normal vector to the interval curve, which creates segment j. Expressions $\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau}$ are the complex version of parametric functions describing the boundary, which have the following interval form:

$$\begin{aligned}\widehat{\boldsymbol{\tau}} &= \boldsymbol{S}_l^{(1)}(\widehat{s}) + i\boldsymbol{S}_l^{(2)}(\widehat{s}), \\ \boldsymbol{\tau} &= \boldsymbol{S}_j^{(1)}(s) + i\boldsymbol{S}_j^{(2)}(s),\end{aligned} \tag{3}$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

where the interval components connected with the direction of coordinates in a 2D Cartesian reference system: $\boldsymbol{S}_j^{(1)} = [\underline{S}_j^{(1)}, \overline{S}_j^{(1)}], \boldsymbol{S}_j^{(2)} = [\underline{S}_j^{(2)}, \overline{S}_j^{(2)}],\ \boldsymbol{S}_l^{(1)} = [\underline{S}_l^{(1)}, \overline{S}_l^{(1)}]$ and $\boldsymbol{S}_l^{(2)} = [\underline{S}_l^{(2)}, \overline{S}_l^{(2)}]$. These components have the form of directed intervals [18].

The boundary is modelled by interval Bézier curves of the first degree (linear):

$$\boldsymbol{S}_k(s) = \boldsymbol{a}_k^{(f)} s + \boldsymbol{b}_k^{(f)}, \quad 0 \leq s \leq 1, \quad (f) - \text{first degree}, \tag{4}$$

and the third degree (curvilinear):

$$\boldsymbol{S}_k(s) = \boldsymbol{a}_k^{(t)} s^3 + \boldsymbol{b}_k^{(t)} s^2 + \boldsymbol{c}_k^{(t)} s + \boldsymbol{d}_k^{(t)}, \quad 0 \leq s \leq 1, \quad (t) - \text{third degree} \tag{5}$$

where vector $\boldsymbol{S}_k(s) = [\boldsymbol{S}_k^{(1)}(s), \boldsymbol{S}_k^{(2)}(s)]^T,\ k = \{l, j\}$ and s is a variable in the parametric reference system. Coefficients $\boldsymbol{a}_k^{(f)}, \boldsymbol{b}_k^{(f)}, \boldsymbol{a}_k^{(t)}, \boldsymbol{b}_k^{(t)}, \boldsymbol{c}_k^{(t)}, \boldsymbol{d}_k^{(t)}$ have also form of vectors composed of two interval components (similarly to $\boldsymbol{S}_k(s)$). They are computed using interval points describing particular segments of the boundary as presented in Fig. 1 (the graphical example assumes $k = j$):

$$\boldsymbol{a}_j^{(f)} = \boldsymbol{P}_{e(j+2)} - \boldsymbol{P}_{b(j+2)}, \quad \boldsymbol{b}_j^{(f)} = \boldsymbol{P}_{b(j+2)},$$

$$\boldsymbol{a}_j^{(t)} = \boldsymbol{P}_{e(j)} - 3\boldsymbol{P}_{i2(j)} + 3\boldsymbol{P}_{i1(j)} - \boldsymbol{P}_{b(j)}, \quad \boldsymbol{b}_j^{(t)} = 3(\boldsymbol{P}_{i2(j)} - 2\boldsymbol{P}_{i1(j)} + \boldsymbol{P}_{b(j)}),$$

$$\boldsymbol{c}_j^{(t)} = 3(\boldsymbol{P}_{i1(j)} - \boldsymbol{P}_{b(j)}), \quad \boldsymbol{d}_j^{(t)} = \boldsymbol{P}_{b(j)},$$

where coordinates of all points $\boldsymbol{P}$, regardless of their subscript, have the form

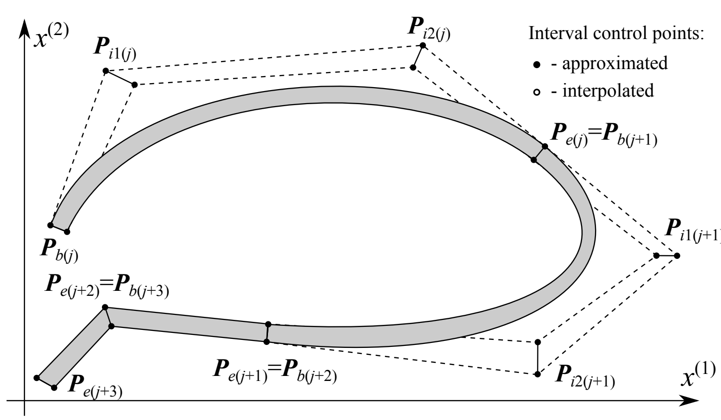

Fig. 1. The interval Bézier curves of the first and third degree used to define segments of the boundary

of a vector of intervals:

$$\boldsymbol{P} = [\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}]^T = \left[[\underline{P}^{(1)}, \overline{P}^{(1)}], [\underline{P}^{(2)}, \overline{P}^{(2)}]\right]^T.$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

The interval boundary functions $\boldsymbol{u}_j(s)$ and $\boldsymbol{p}_j(s)$ in (1) present boundary conditions and the following series approximate them:

$$\boldsymbol{u}_j(s) = \sum_{k=0}^{N} \boldsymbol{u}_j^{(k)} L_j^{(k)}(s), \quad \boldsymbol{p}_j(s) = \sum_{k=0}^{N} \boldsymbol{p}_j^{(k)} L_j^{(k)}(s), \tag{6}$$

where $\boldsymbol{u}_j^{(k)} = [\underline{u}_j^{(k)}, \overline{u}_j^{(k)}]$ and $\boldsymbol{p}_j^{(k)} = [\underline{p}_j^{(k)}, \overline{p}_j^{(k)}]$ are unknown or given interval values of boundary functions in defined points of the segment j, N - is the number of terms in approximating series (6) and $L_j^{(k)}(s)$ – the base functions (Lagrange polynomials) on segment j.

3 Process of solving the IFPIES

Applying the FMM into the PIES is the first step in solving the IFPIES. The FMM is based on the tree structure, which transforms interactions between the individual PIES boundary segments into interactions between some groups of segments called cells. In the IFPIES, we applied a modified version of the tree [15] (presented in Fig. 2). Unlike the classical binary tree, we had to join the beginning and end of each level to consider that a parametric 1D system describes the PIES for 2D issues. Also, the PIES's modified kernels (complex form) are expanded using the Taylor series. It allows the calculation of complex integrals to be converted into approximate sums. All process of applying the FMM into the PIES is clearly described in [15].

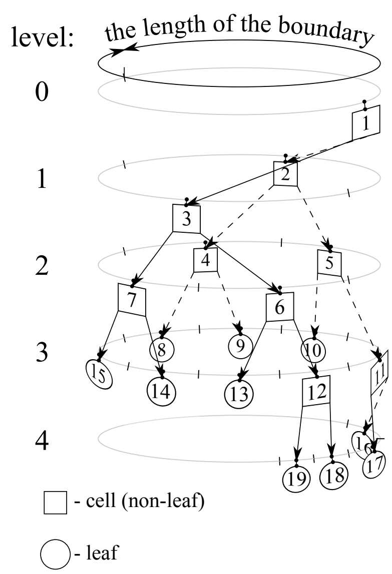

Fig. 2. The example of modified binary tree in the IFPIES for 2D problem

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

The process of applying the FMM into the IPIES is very similar, and as a result, we obtained the following form of integrals in (1) [16]:

$$\begin{gathered}
\int\limits_{s_{j-1}}^{s_j} \widehat{\boldsymbol{U}}_{lj}^{*(c)}\left(\widehat{\boldsymbol{\tau}},\boldsymbol{\tau}\right) p_j\left(s\right) \boldsymbol{J}_j^{(c)}\left(s\right) ds = \frac{1}{2\pi}\sum_{l=0}^{N_T}(-1)^l \cdot \\
\left\{\sum_{k=0}^{N_T}\sum_{m=l}^{N_T}\frac{(k+m-1)!\cdot \boldsymbol{M}_k(\boldsymbol{\tau}_c)}{\left(\boldsymbol{\tau}_{el}-\boldsymbol{\tau}_c\right)^{k+m}}\cdot\frac{\left(\boldsymbol{\tau}_{el}'-\boldsymbol{\tau}_{el}\right)^{m-l}}{(m-l)!}\right\}\frac{\left(\widehat{\boldsymbol{\tau}}-\boldsymbol{\tau}_{el}'\right)^l}{l!}, \\
\int\limits_{s_{j-1}}^{s_j} \widehat{\boldsymbol{P}}_{lj}^{*(c)}\left(\widehat{s},s\right) u_j\left(s\right) \boldsymbol{J}_j^{(c)}\left(s\right) ds = \frac{1}{2\pi}\sum_{l=0}^{N_T}(-1)^l \cdot \\
\left\{\sum_{k=1}^{N_T}\sum_{m=l}^{N_T}\frac{(k+m-1)!\cdot \boldsymbol{N}_k(\boldsymbol{\tau}_c)}{\left(\boldsymbol{\tau}_{el}-\boldsymbol{\tau}_c\right)^{k+m}}\cdot\frac{\left(\boldsymbol{\tau}_{el}'-\boldsymbol{\tau}_{el}\right)^{m-l}}{(m-l)!}\right\}\frac{\left(\widehat{\boldsymbol{\tau}}-\boldsymbol{\tau}_{el}'\right)^l}{l!}.
\end{gathered} \tag{7}$$

where: N_T is the number of terms in the Taylor expansion, $\widehat{\boldsymbol{\tau}} = \boldsymbol{S}_l^{(1)}(\widehat{s}) + i\boldsymbol{S}_l^{(2)}(\widehat{s})$, $\boldsymbol{\tau} = \boldsymbol{S}_j^{(1)}(s) + i\boldsymbol{S}_j^{(2)}(s)$, complex interval points $\boldsymbol{\tau}_c$, $\boldsymbol{\tau}_{el}$, $\boldsymbol{\tau}_c'$, $\boldsymbol{\tau}_{el}'$ are midpoints of leaves obtained while tracing the tree structure (see [3]). Expressions $\boldsymbol{M}_k(\boldsymbol{\tau}_c)$ and $\boldsymbol{N}_k(\boldsymbol{\tau}_c)$ are called moments (and they are computed twice only) and have the form [16]:

$$\begin{gathered}
\boldsymbol{M}_k(\boldsymbol{\tau}_c) = \int\limits_{s_{j-1}}^{s_j} \frac{\left(\boldsymbol{\tau}-\boldsymbol{\tau}_c\right)^k}{k!}\boldsymbol{p}_j(s)\boldsymbol{J}_j^{(c)}(s)ds, \\
\boldsymbol{N}_k\left(\boldsymbol{\tau}_c\right) = \int\limits_{s_{j-1}}^{s_j} \frac{\left(\boldsymbol{\tau}-\boldsymbol{\tau}_c\right)^{k-1}}{(k-1)!}\boldsymbol{n}_j^{(c)}\boldsymbol{u}_j(s)\boldsymbol{J}_j^{(c)}(s)ds.
\end{gathered} \tag{8}$$

where $\boldsymbol{n}_j^{(c)} = \boldsymbol{n}_j^{(1)} + i\boldsymbol{n}_j^{(2)}$ the complex interval normal vector to the curve created segment j.

Similarly to the original PIES, the IFPIES are written at collocation points whose number corresponds to the number of unknowns. However, during solving the IFPIES, the system of algebraic equations $\boldsymbol{A}\cdot\boldsymbol{x} = \boldsymbol{b}$ is produced implicitly, contrary to the original PIES. It means that only the result of multiplication of the matrix $\boldsymbol{A}$ by the vector of unknowns $\boldsymbol{x}$ is used by applied iterative GMRES solver [19]. The solver is modified by applying directed interval arithmetic and directly integrated with the IFPIES. The authors also applied the same GMRES solver to the IPIES to prepare a more reliable comparison.

4 Numerical results

All tests are performed on a PC based on Intel Core i5-4590S with 32 GB RAM. Application of the IPIES and the IFPIES are compiled by g++ 7.5.0 (-O2

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

optimization) on 64-bit Linux OS (kernel 6.2.0). Two multi-connected problems with linear and mixed (linear and curvilinear) segments created the shape of the boundary are considered.

4.1 L-shaped problem with randomly placed holes

The domain boundary in the first example is composed of linear segments only, as presented in Fig. 3. Laplace's equation describes the problem. Interval boundary conditions are also presented in Fig. 3 (where u - Dirichlet and p - Neumann boundary conditions).

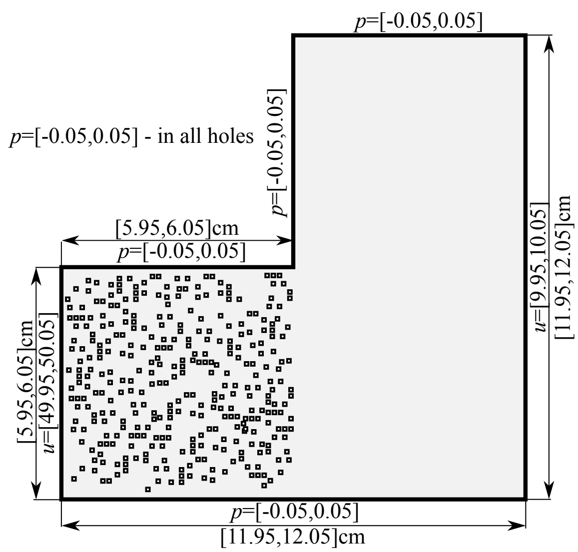

Fig. 3. L-shaped problem with randomly placed holes

The research focused on the CPU time, RAM utilization and accuracy of the IFPIES compared to the IPIES. Due to the lack of literature on solving problems with uncertainly defined boundary shapes and boundary conditions, comparison to others is impossible. The number of terms in the Taylor series is set to 25, and the value of GMRES tolerance equals 10^{-8}. The number of collocation points in all segments is constant and is changed from 2 to 8 in subsequent research.

For the L-shaped problem, we solved three examples with different numbers of holes: 400, 900 and 1600, placed in the part of the domain presented in Fig. 3. Therefore, we solved the systems from 3 212 to 51 248 equations.

As can be seen from Tab. 1, the IFPIES is as accurate as the IPIES, which is proved by the result of computation of the mean square error (MSE) between the lower and upper bound (infimum and supremum) of the IFPIES and the IPIES solutions. The MSE between both methods is very low and does not exceed 10^{-8}.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

Table 1. Comparison between the IFPIES and the IPIES for L-shaped problem

Number of		CPU time [s]		RAM utilization [MB]		MSE	
col. pts	eqs	*IFPIES*	*IPIES*	*IFPIES*	*IPIES*	inf	sup
400 holes							
2	3 212	34.13	66.52	57.14	242	$6.29 \cdot 10^{-12}$	$5.60 \cdot 10^{-11}$
4	6 424	84.95	290.74	99	962	$3.15 \cdot 10^{-14}$	$4.67 \cdot 10^{-11}$
6	9 636	148.33	706.18	113	2 158	$8.62 \cdot 10^{-10}$	$3.47 \cdot 10^{-11}$
8	12 848	202.80	1 345.22	170	3 831	$1.11 \cdot 10^{-11}$	$1.46 \cdot 10^{-10}$
900 holes							
2	7 212	98.38	333.55	140	1 221	$8.62 \cdot 10^{-10}$	$3.47 \cdot 10^{-11}$
4	14 424	286.54	1 465.37	339	4 804	$5.14 \cdot 10^{-10}$	$5.62 \cdot 10^{-11}$
6	21 636	563.77	3 552.72	585	10 700	$1.17 \cdot 10^{-10}$	$3.65 \cdot 10^{-11}$
8	28 848	896.74	6 782.20	957	19 100	$6.95 \cdot 10^{-10}$	$6.14 \cdot 10^{-11}$
1600 holes							
2	12 812	202.80	1 345.22	170	3 831	$7.18 \cdot 10^{-8}$	$3.28 \cdot 10^{-11}$
4	25 624	1 090.83	4 658.04	1 381	15 100	$9.57 \cdot 10^{-8}$	$5.31 \cdot 10^{-11}$
6	38 436	2 343.02	—	2 772	—	—	—
8	51 248	4 130.20	—	4 745	—	—	—

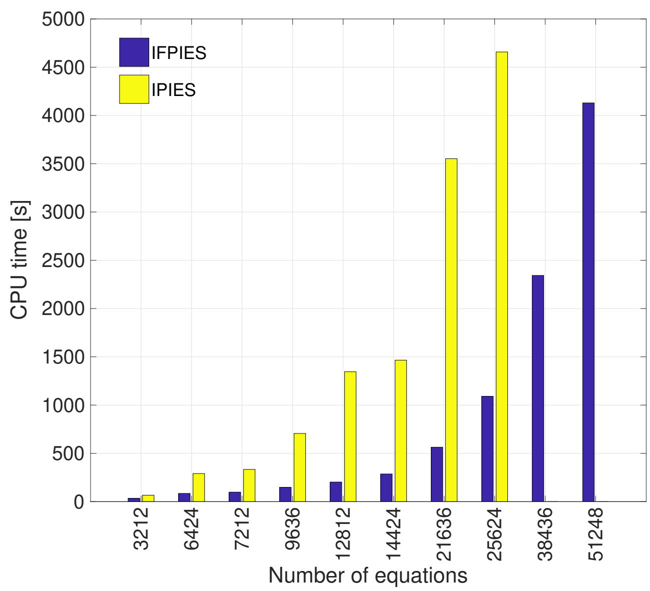

Fig. 4. Comparison of computation time of the IFPIES and the IPIES for different numbers of equations

It is worth emphasising that solving the IPIES for 38 436 and 51 248 equations is not possible due to utilization of all computer memory, while the IFPIES

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

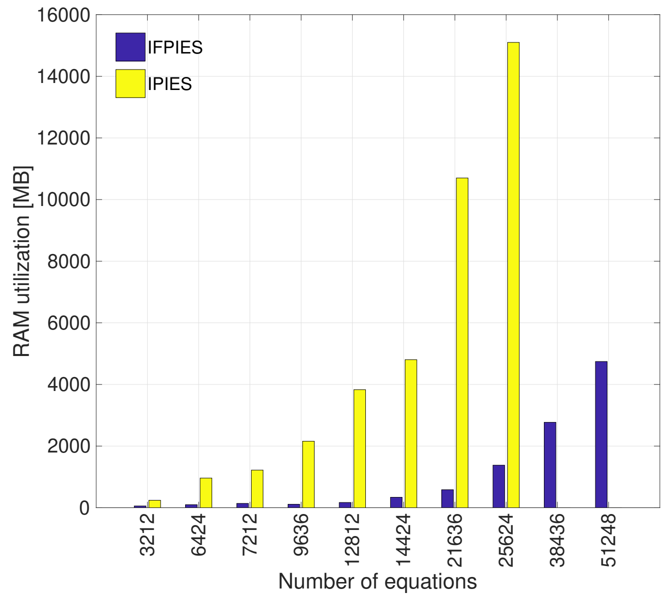

Fig. 5. Comparison of RAM utilization of the IFPIES and the IPIES for different numbers of equations

consumed about 15% of available RAM. It means that we can solve up to about 35 000 equations using the IPIES due to exhaustion of RAM, while the IFPIES uses less than 2 GB of memory. It presents limitations of the classical IPIES contrary to contemporary one.

Also, from Fig. 4 and Fig. 5, it can be seen that the speedup of the IFPIES in relation to the IPIES grows with an increasing number of equations, whilst RAM utilization of the IFPIES is smaller and grows much slower than the IPIES. Also, for a smaller number of equations, the FMM overhead gives us a considerably smaller gain in CPU time and RAM utilization.

4.2 Current flow through the plate with holes

The domain boundary in the second example is composed of mixed linear and curvilinear segments, as presented in Fig. 6. Laplace's equation also describes the problem. Interval boundary conditions are presented in Fig. 6 (where V - potential and $\frac{\partial V}{\partial n}$ - flux).

The number of terms in the Taylor series and the value of GMRES tolerance is the same as in the previous problem, equal to 25 and 10^{-8}. The number of collocation points in all segments is also constant and has changed from 2 to 8. At last, we solved the systems from 32 048 to 128 192 equations.

We are able to solve only examples with the smallest number of equations using the IPIES due to exhaustion of RAM. However, the IPIES for 32 048

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

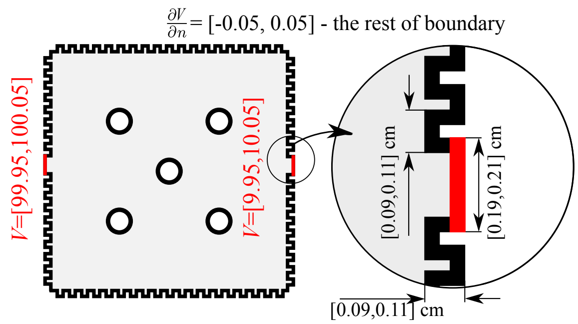

Fig. 6. Current flow through the plate with holes

Table 2. Results of solving current flow problem by the IPIES

Number of		CPU time	RAM utilization
col. pts	eqs	[s]	[MB]
2	32 048	312.30	374
4	64 096	623.43	892
6	96 144	1 151.06	1 624
8	128 192	1 860.20	2 473

equations uses 23.7 GB of RAM and requires 7 031.32 s (almost 2 hours) of CPU time. It is incomparably more than in the IPIES. The MSE between solutions of both methods is very small and equal to $2.22 \cdot 10^{-8}$ for lower and $1.89 \cdot 10^{-9}$ for upper bound.

As seen from Tab. 2, solving the example with over 128 000 equations requires less than 2.5 GB of memory and about a half hour. The obtained results confirm the very high efficiency of the IFPIES and allow us to solve large-scale examples on a standard PC in a reasonable time.

5 Conclusions

The paper presents the IFPIES in solving 2D potential curvilinear multi-connected boundary value problems with uncertainly defined boundary shapes and conditions. The method gives us the opportunity to include measurement errors (the uncertainty of measurement data) of the boundary shape and boundary conditions in calculations, which is impossible in classic practical design.

Application of the fast multipole technique in the IFPIES also allows for the highly efficient solving of complex (large-scale) engineering problems on a standard PC in a reasonable time. The real power of the IFPIES is very low RAM utilization. The IPIES is unable to cope with the solution of over 35 000 equations for 32 GB of RAM, while the IFPIES easily solves the examples with over 128 000 equations. Also, the CPU time of the IFPIES is significantly shorter than the IPIES.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

The obtained results suggest that the direction of research should be continued. Our further research should cover problems modelled by other than Laplace's equations.

References

1. Kapturczak, M., Zieniuk, E., Kużelewski, A.: NURBS curves in parametric integral equations system for modeling and solving boundary value problems in elasticity. In: Krzhizhanovskaya, V. V., et al. (eds.) Computational Science - ICCS 2020, LNCS, vol. 12138, pp. 116–123. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50417-5_9
2. Zieniuk, E., Kużelewski, A.: Concept of the interval modelling the boundary shape using interval bézier curves in boundary problems solved by PIES. In: Simos, T.E., et al. (eds.) 12th International Conference of Numerical Analysis and Applied Mathematics ICNAAM 2014, AIP Conference Proceedings, vol. 1648, 590002. AIP Publishing LLC., Melville (2015). https://doi.org/10.1063/1.4912829
3. Kużelewski, A., Zieniuk, E.: Solving of multi-connected curvilinear boundary value problems by the fast PIES. Computer Methods in Applied Mechanics and Engineering **391**, 114618 (2022).
4. Fu, C., Zhan, Q., Liu, W.: Evidential reasoning based ensemble classifier for uncertain imbalanced data. Information Sciences **578**, 378–400 (2021).
5. Wang, C., Matthies, H.G.: Dual-stage uncertainty modeling and evaluation for transient temperature effect on structural vibration property. Computational Mechanics **63**(2), 323–333 (2019).
6. Gouyandeh, Z., Allahviranloo, T., Abbasbandy, S., Armand, A.: A fuzzy solution of heat equation under generalized Hukuhara differentiability by fuzzy Fourier transform. Fuzzy Sets and Systems **309**, 81–97 (2017).
7. Ni, B.Y., Jiang, C.: Interval field model and interval finite element analysis. Computer Methods in Applied Mechanics and Engineering **360**, 112713 (2020).
8. Zalewski, B., Mullen, R., Muhanna, R.: Interval boundary element method in the presence of uncertain boundary conditions, integration errors, and truncation errors. Engineering Analysis with Boundary Elements **33**(4), 508–513 (2009).
9. Zieniuk, E., Kapturczak, M., Kużelewski, A.: Modification of Interval Arithmetic for Modelling and Solving Uncertainly Defined Problems by Interval Parametric Integral Equations System. In: Shi, Y. et al. (eds.) Computational Science - ICCS 2018, LNCS, vol. 10862, pp. 231–240. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93713-7
10. Kużelewski, A., Zieniuk, E.: OpenMP for 3D potential boundary value problems solved by PIES. In: Simos, T.E., et al. (eds.) 13th International Conference of Numerical Analysis and Applied Mathematics ICNAAM 2015, AIP Conference Proceedings, vol. 1738, 480098. AIP Publishing LLC., Melville (2016). https://doi.org/10.1063/1.4952334
11. Kużelewski, A., Zieniuk, E., Bołtuć, A.: Application of CUDA for Acceleration of Calculations in Boundary Value Problems Solving Using PIES. In: Parallel Processing and Applied Mathematics PPAM 2013, LNCS, vol. 8385, pp. 322–331. Springer, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55195-6_30
12. Greengard, L.F., Rokhlin, V.: A fast algorithm for particle simulations. Journal of Computational Physics **73**(2) 325–348 (1987).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10

13. T. Huang, YX. Zhu, YJ. Ha, X. Wang, MK. Qiu, A Hardware Pipeline with High Energy and Resource Efficiency for FMM Acceleration, ACM Transactions on Embedded Computing Systems 17(2) (2018) Article Number:51.
14. M. Barbarino, D. Bianco, A BEM-FMM approach applied to the combined convected Helmholtz integral formulation for the solution of aeroacoustic problems, Computer Methods in Applied Mechanics and Engineering 342 (2018) 585-603.
15. Kużelewski, A., Zieniuk, E.: The FMM accelerated PIES with the modified binary tree in solving potential problems for the domains with curvilinear boundaries. Numerical Algorithms **88**(3), 1025–1050 (2021).
16. Kużelewski, A., Zieniuk, E., Czupryna, M.: Interval Modifications of the Fast PIES in Solving 2D Potential BVPs with Uncertainly Defined Polygonal Boundary Shape. In: Groen, D. et al. (eds.) Computational Science - ICCS 2022, LNCS, vol. 13351, pp. 18–25. Springer, Cham (2022). https://doi.org/10.1007/978-3-319-93713-7_19
17. Moore, R.E.: Interval Analysis. Prentice-Hall, Englewood Cliffs, New York (1966).
18. Markov, S.M.: On directed interval arithmetic and its applications. Journal of Universal Computer Science **1**(7), 514–526 (1995).
19. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM Journal on Scientific and Statistical Computing **7**, 856–869 (1986).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_10