ADAPTIVE RBF-FD METHOD FOR POISSON'S EQUATION

JURE SLAK[1,2] & GREGOR KOSEC[1]
[1]Parallel and Distributed Systems Laboratory, Jožef Stefan Institute, Slovenia
[2]Faculty of Mathematics and Physics, University of Ljubljana, Slovenia

ABSTRACT

Solutions to many physical problems governed by partial differential equations (PDEs) often vary significantly in magnitude throughout the problem domain. Although in some special cases the areas with high error are known in advance, in general the error distribution is unknown beforehand. Adaptive techniques for solving PDEs are a standard way of dealing with this problem, where problematic regions are iteratively refined. A step further is automatic adaptivity, where problematic regions are chosen automatically using an error indicator and then refined, until a certain error threshold is reached. In this paper, we apply a recently published technique for automatic adaptivity for strong form meshless methods and solve the Poisson equation and its generalisations, using the popular RBF-FD method. Both 2D and 3D cases are considered, comparing uniform and adaptive refinement, illustrating the advantages of fully automatic adaptivity.

Keywords: adaptivity, mesh-free methods, RBF-FD, Poisson equation, Helmholtz equation, PDEs.

1 INTRODUCTION

Adaptive modification of partial differential equation (PDE) discretizations is needed in problems with variable physical behaviour or precision requirements throughout the domain. The techniques used in traditional methods, such as the Finite Element Method (FEM), are well developed [1]. Besides the procedure for the solution of the PDE, two more parts are needed for fully automatic adaptivity: an indicator of the solution quality and a refinement strategy. In the following paragraphs we offer a brief review of all three elements in a strong form meshless method context.

Local strong form methods were developed as a cheaper alternative to global collocation methods and are usually presented as generalisations of Finite Difference Method to scattered grids. The most direct generalisation is the Finite Point Method, which uses monomials to compute the stencil weights [2]. Due to stability issues, Radial Basis Functions (RBFs) have been suggested and successfully used instead of monomials [3]. To ensure consistency of the approximations and eliminate the influence of the shape parameter, Polyharmonic splines, augmented with monomials were researched and successfully applied to various problems [12], which we will also use for our solution procedure.

The most widely used error indicator was developed by Zienkiewicz and Zhu [4] for FEM and has since been generalized to meshless methods. The main idea of the indicator is to approximate the error of the solution as the difference between the original solution and the so-called recovered solution, obtained by appropriate post-computation. In strong form meshless methods, it has already been used with the Finite Point Method [5] and RBF-FD [11]. A different type of error indicator is available for least-squares based meshless methods, which use the residual of the least squares approximations to estimate the error [6]. Additionally, several ad hoc error indicators have been used, mostly using the variability of the solution to determine refinement regions, such as Davydov and Oanh [7] and Kosec and Šarler [8]. This method will be used in the present work as well, for simplicity.

After the problematic regions have been identified, the (de)refinement can be done in multiple ways. Either we can modify the approximation, resulting in p-refinement schemes, or the discretization itself can be modified. This can be done either by moving the nodes

doi:10.2495/BE420131

(r-refinement), used in, for example, Afshar et al. [6], or by placing more nodes in regions with high estimated error and removing nodes from low-error regions (h-refinement), used in, for example, Kosec and Šarler [8] and Oanh et al. [11].

Another variation in h-refinement strategies is whether the existing discretisation is only modified and partially reused, or generated completely anew. The former can cause problems with conditioning, causing the need for specially selected stencils [11], which is why we opted for the latter option.

Similar methodology to the one described above, has been recently used to solve linearly elastic contact problems [10]. In this work we apply it to the Poisson equation and its generalizations. The rest of the paper is organized as follows: in Section 2 the RBF-FD method is presented, in Section 3 the adaptive methodology is described, and in Section 4 the results of the numerical experiments are presented.

2 RBF-GENERATED FINITE DIFFERENCES

Consider an elliptic boundary value problem:

$$\mathcal{L}u = f \text{ in } \Omega, \tag{1}$$

$$u = u_0 \text{ on } \partial\Omega, \tag{2}$$

which will be used as a sample theoretical problem throughout this paper. Domain Ω is discretized by placing N nodes p_i in and on the boundary of the domain. Nodes are positioned by algorithm presented in [13], which can distribute nodes according to any continuous spacing function h. Each node p_i is assigned a *stencil* of n nodes, $S(p_i) = (p_{j_1}, \dots, p_{j_n})$. The operator $\mathcal{L}$ is approximated for each point p_i in the domain interior. This is done by assuming an approximation of form

$$(\mathcal{L}u)(p_i) \approx \textstyle\sum_{k=1}^{n} w_{i,k} u(p_{j_k}). \tag{3}$$

Equality in the above approximation is desired for a certain set of functions: if node p_i are arranged in a grid and equality is enforced for monomials, standard finite difference approximation is obtained. To derive the “RBF-generated finite differences”, RBFs are used to compute the weights $w_{i,k}$.

Polyharmonic splines will be used in this work, which are radial functions of form $\phi(r) = r^k$, for an odd integer k. Radial basis functions are often thought of as being centred in one node, denoting $\phi_j(p) = \phi(\|p - p_j\|)$. To obtain RBF-FD weights, radial functions centred on the stencil nodes are used as the basis. Substituting ϕ_{j_ℓ} for u in eqn (3) and enforcing equality, we arrive at the following linear system:

$$\begin{bmatrix} \phi(\|p_{j_1} - p_{j_1}\|) & \cdots & \phi(\|p_{j_n} - p_{j_1}\|) \\ \vdots & \ddots & \vdots \\ \phi(\|p_{j_1} - p_{j_n}\|) & \cdots & \phi(\|p_{j_n} - p_{j_n}\|) \end{bmatrix} \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,n} \end{bmatrix} = \begin{bmatrix} \big(\mathcal{L}\phi(\|\cdot - p_{j_1}\|)\big)(p_i) \\ \vdots \\ \big(\mathcal{L}\phi(\|\cdot - p_{j_n}\|)\big)(p_i) \end{bmatrix}, \tag{4}$$

written compactly as $Aw = b$. Additional consistency constraints can be enforced to ensure convergence [12], augmenting the above matrix into

$$\begin{bmatrix} A & Q^T \\ Q & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}, \tag{5}$$

where the second equation $Q\lambda = c$ is expanded as

$$\begin{bmatrix} q_1(p_{j_1}) & \cdots & q_1(p_{j_n}) \\ \vdots & \ddots & \vdots \\ q_\ell(p_{j_1}) & \cdots & q_\ell(p_{j_n}) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_\ell \end{bmatrix} = \begin{bmatrix} (\mathcal{L}q_1)(p_i) \\ \vdots \\ (\mathcal{L}q_\ell)(p_i) \end{bmatrix}, \tag{6}$$

with q_l representing monomials up to and including some order m. Values λ are discarded after computation.

With the stencil weights $w_{i,k}$ computed for each node p_i the PDE $(\mathcal{L}u)(p_i) = f(p_i)$ can be approximated with a linear algebraic equation $\sum_{k=1}^{n} w_{i,k} u(p_{j_k}) = f(p_i)$. These equations can be assembled in a sparse matrix and after including the boundary conditions, can be solved for unknowns $u(p_i)$ to obtain the numerical solution.

3 ADAPTIVE METHODOLOGY

The adaptive methodology in this paper behaves similarly to "remeshing" used commonly in FEM and has been successfully used to solve contact problems from linear elasticity [7]. Some initial (possibly variable) nodal spacing $h^0(p)$ is chosen, as well as its lower and upper bounds $h_L(p)$ and $h_U(p)$, respectively. Domain Ω is filled with nodes, conforming to h^0 and the solution u^0 is obtained.

An error indicator is employed to determine which nodes should be (de)refined and the nodal density h^0 is altered appropriately. This adaptive cycle below is repeated until the convergence criterion is met. The procedure on j-th iteration is written in more detail below:

1. Fill Ω with nodes conforming to h^j.
2. Solve the problem to obtain u^j.
3. Compute the error indicator values ε_i^j for each node p_i.
4. If the mean of ε_i^j is below some tolerance ε return u^j as the solution and stop.
5. Adapt h^j to obtain h^{j+1}.

More details on steps 3 and 5 are given in the next two sections.

3.1 Error indicator

The error indicator used in this paper is designed to detect the variability of the solution around node p_i. First, the recovered solution $\tilde{u}_i$ is computed as $\tilde{u}_i = \frac{1}{n}\sum_{k=1}^{n} u_k$ and then the indicator ε_i is computed as $\varepsilon_i = \sqrt{\frac{1}{n}\sum_{k=1}^{n} e_i^2}$, where local errors e_i are computed as $e_i = |u_i - \tilde{u}_i|$. Despite its simplicity, this error indicator turned out to work well enough for our purposes.

3.2 Nodal spacing adaptation

The existing nodal spacing function h^j is evaluated at nodes p_i to obtain values $h_{i,j} = h^j(p_i)$. These values $h_{i,j}$ are modified by a *density factor* f_i as

$$h_{i,j+1} = \min\left(max\left(h_{i,j}/f_i\,, h_L(p_i)\right), h_U(p_i)\right), \tag{7}$$

where factor f_i is computed as

$$f_i = \begin{cases} 1 + \frac{\eta - \varepsilon_i}{\eta - m}\left(\frac{1}{\beta} - 1\right); & \varepsilon_i \leq \eta, \\ 1; & \eta \leq \varepsilon_i \leq \varepsilon, \\ 1 + \frac{\varepsilon_i - \varepsilon}{M - \varepsilon}(\alpha - 1); & \varepsilon_i \geq \varepsilon, \end{cases} \tag{8}$$

and α represents the refine aggressiveness, β the derefine aggressiveness, ε the refinement threshold, η the derenfinement threshold, and $m = \min_i \varepsilon_i$ is the minimal and $M = \max_i \varepsilon_i$ is the maximal value of the error indicator. Note that setting $\alpha = 1$ or $\beta = 1$ disables refinement and derefinement, respectively.

Lower refinement bound h_L is often not necessary and can be set to ∞, while upper bound h_U is often set to the initial density h_0. Unless otherwise stated this will be the case in this paper as well.

After the values $h_{i,j+1}$ have been computed, the function h^{j+1} is obtained using modified Shepard's scattered interpolation of pairs $(p_i, h_{i,j+1})$. The 15 closest neighbouring nodes were used in this paper.

4 RESULTS

We test the fully adaptive solution procedure described in the previous section on some standard test problems, taken from Mitchell [9].

4.1 L-shaped domain

The first test case is the "L domain" test case. Laplace's equation

$$\nabla^2 u = 0 \tag{9}$$

is solved on domain $\Omega = [-1,1]^2 \setminus ([0,1] \times [-1,0])$ with a known solution u given in polar coordinates as $u = r^{\frac{2}{3}} \sin \frac{2}{3}\theta$ and Dirichlet boundary conditions obtained from u.

RBF-FD method with Polyharmonic splines augmented with monomials up to and including 2nd order was used to approximate the differential operators. The stencils for each node were chosen by simply selecting the closest $n = 15$ nodes.

The resulting sparse system was solved using the Intel® MKL Pardiso sparse solver. Both uniform and fully adaptive refinement was tested. The adaptive procedure was run with $\alpha = 3$, $\varepsilon = 10^{-2}$, $\beta = 1$ and $\eta = 0$.

The error was between the exact solution u and numerical solution $\hat{u}$ was measured in three different norms

$$e_1 = \frac{\|u - \hat{u}\|_1}{\|u\|_1}, \ \|u\|_1 = \frac{1}{N}\sum_{p \in G} |u(p)|, \tag{10}$$

$$e_2 = \frac{\|u - \hat{u}\|_2}{\|u\|_2}, \ \|u\|_2 = \sqrt{\frac{1}{N}\sum_{p \in G} |u(p)|^2}, \tag{11}$$

$$e_\infty = \frac{\|u - \hat{u}\|_\infty}{\|u\|_\infty}, \ \|u\|_\infty = \max_{p \in G} |u(p)|, \tag{12}$$

where G is a set of independent points arranged in a dense uniform grid over the whole domain.

The errors e_1, e_2 and e_∞ for uniform refinement are shown in Fig. 1. Fig. 2 shows the errors under adaptive refinement. The adaptive refinement finished in 6 iterations. We can see that better rates and lower errors are obtained using this technique.

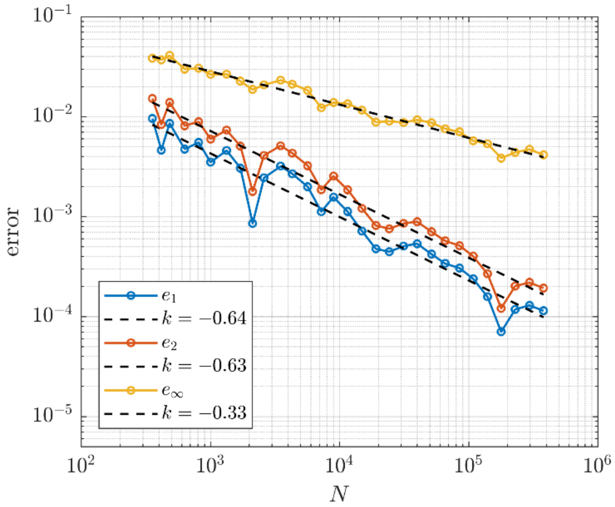

Figure 1: Errors when solving the "L domain" problem using uniform refinement.

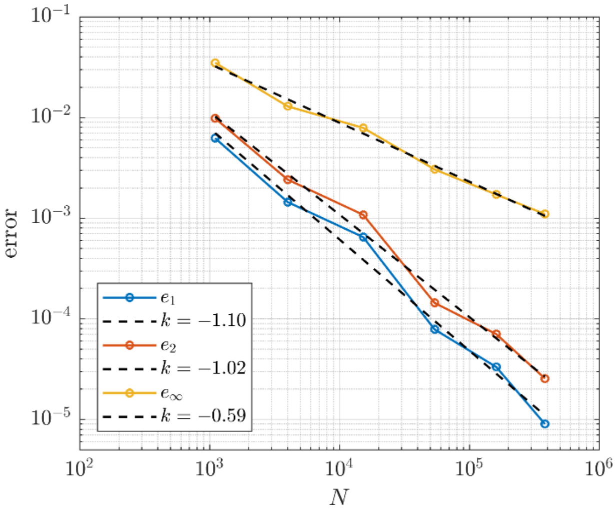

Figure 2: Errors when solving the "L domain" problem using adaptive refinement.

Additionally, local node errors and nodal densities were observed during the course of the adaptive iteration. The local errors on each iteration are computed for each node as $\epsilon_i = |\hat{u}(p_i) - u(p_i)|$. The local nodal separation distance is computed as the average distance $\bar{d}_i$ from each node to its closest neighbours, $\bar{d}_i = \frac{1}{3}(d_{i,1} + d_{i,2} + d_{i,3})$. From $\bar{d}_i$ we can compute the relative node density ρ_i as

$$\rho_i = -\log_2\left(\frac{\bar{d}_i}{\max_j \bar{d}_j}\right). \tag{13}$$

Value $\rho_i = 0$ means that node p_i has maximal separation distance (least dense) while e.g. $\rho_i = 3$ means that nodes around p_i are 8 times as dense as in the coarsest part. Values ϵ_i and ρ_i corresponding to the solutions, whose errors were computed in Fig. 2, are shown in Fig. 3. It can be seen that the node density adapts according to the error and begins to concentrate in the inner corner.

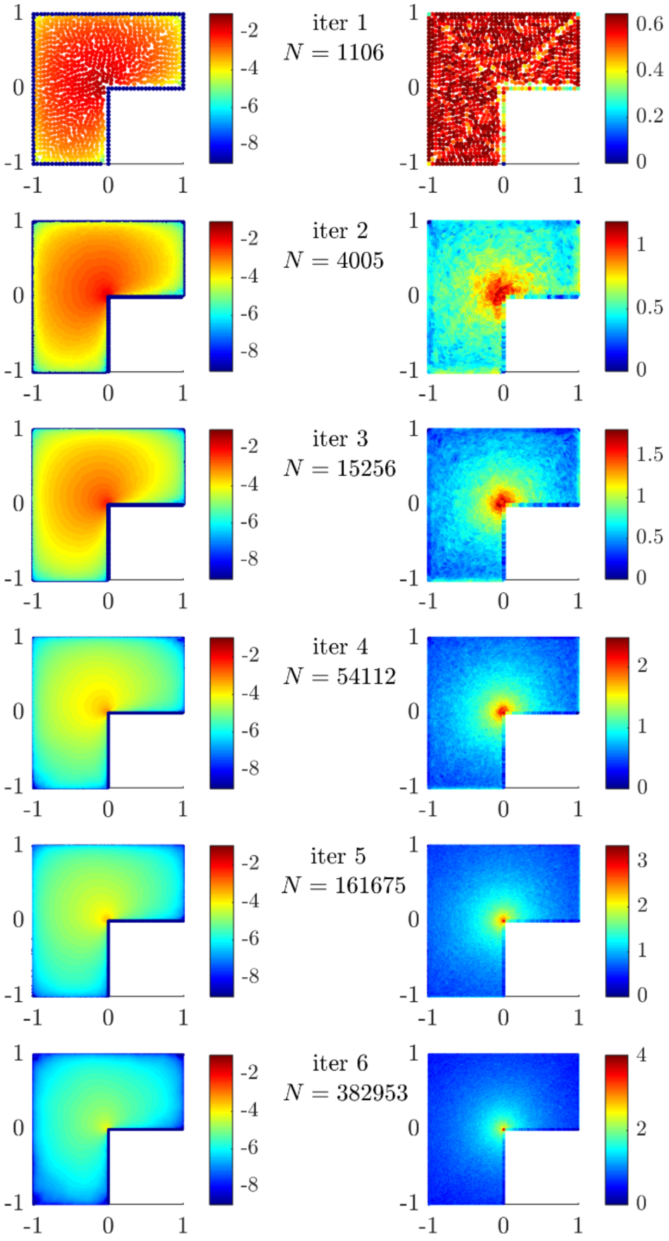

Figure 3: Plots of logarithms of errors ($\log \epsilon_i$, left column) and nodal densities (ρ_i, right column) over the course of the adaptive iteration.

4.2 2D Helmholtz equation

Next, we investigate the behaviour of the method on a Helmholtz equation:

$$-\nabla^2 u - \frac{1}{(\alpha+r)^4} u = f \tag{14}$$

with an oscillatory solution $u(r) = \sin\left(\frac{1}{\alpha+r}\right)$, where the right-hand side f is computed from the solution u.Value $\alpha = \frac{1}{5\pi}$ was used in all computations. The same numerical setup as in Section 4.1 was used to obtain the numerical results, except that the tolerance ϵ was set to 10^{-3}. Fig. 4 shows the errors during uniform and adaptive refinement. We can see that no satisfactory solution can be obtained using uniform refinement and that there is no indication of convergence. The adaptive solution procedure manages to find a satisfactory solution and exhibits better convergence properties.

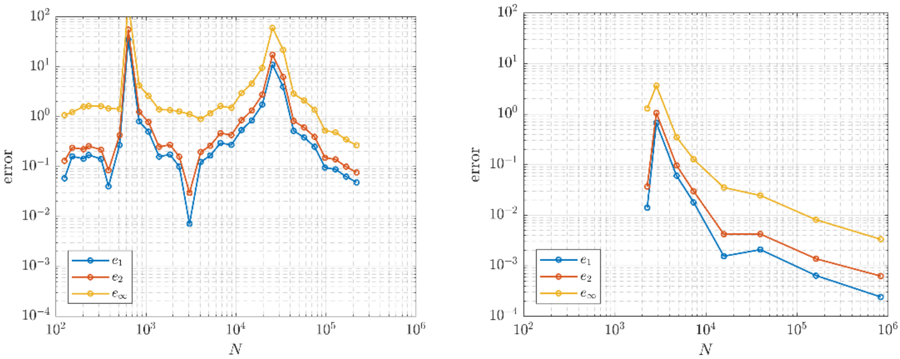

Figure 4: Errors during uniform refinement (left) and adaptive refinement (right) when solving the 2D Helmholtz equation.

The initial jump in error has been observed before [10], [11] and is usually caused by too small initial node density. The solution in the last iteration is shown in Fig. 5, along with the obtained nodal density function. Node separation distance in the densest part was around 400-times smaller than in the coarsest part.

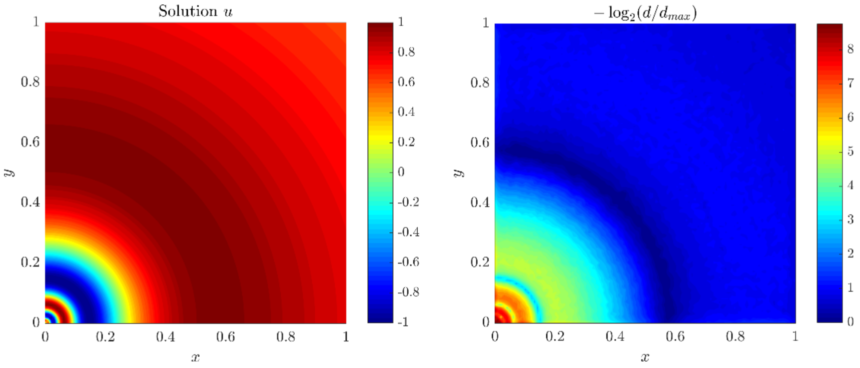

Figure 5: Solution u_i of the 2D Helmholtz equation (left) and the nodal density ρ_i (right).

4.3 3D Helmholtz equation

The same problem as in Section 4.2 is used to assess the behaviour of the method in 3D as well. The same numerical setup as in 2D was used, except that the number of nearest neighbours used to the stencil increased to $n = 25$. Uniform refinement behaves similarly to its 2D counterpart. The errors during the adaptive iteration are shown in Fig. 6. The obtained solution and nodal density in the last iteration are shown in Fig. 7.

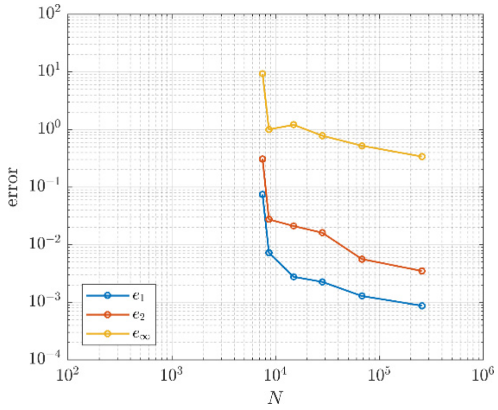

Figure 6: Errors during adaptive solution of the 3D Helmholtz equation.

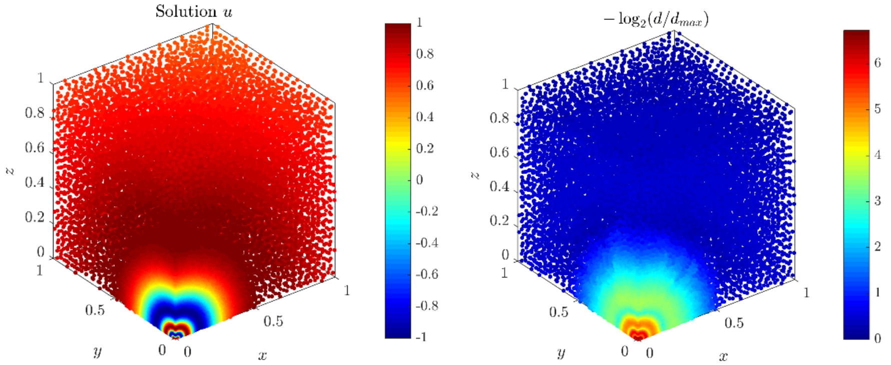

Figure 7: Solution u_i of the 3D Helmholtz equation (left) and the nodal density ρ_i (right).

5 CONCLUSIONS

We presented a fully automatic adaptive procedure for elliptic problems. RBF-FD method along with a node positioning algorithm and an ad hoc error indicator was used to iteratively refine initially uniform discretizations to obtain solution to Poisson's and Helmholtz equations in 2D and 3D, which would not have been feasible using only uniform discretizations. All computations were done using the Medusa library [14] for meshless solutions of PDEs. Future work will be focused on the development of better error indicators and more efficient refinement strategies.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support of the Research Foundation Flanders (FWO), The Luxembourg National Research Fund (FNR) and Slovenian Research Agency (ARRS) in the framework of the FWO Lead Agency project: G018916N Multi-analysis of fretting fatigue using physical and virtual experiments, the ARRS research core funding No. P2-0095 and the Young Researcher program PR-08346.

REFERENCES

[1] Bangerth, W. & Rannacher, R., *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser, 2013.

[2] Oñate, E., Perazzo, F. & Miquel, J., A finite point method for elasticity problems. *Computers & Structures*, **79**(22–25), pp. 2151–2163, 2001.

[3] Tolstykh, A.I. & Shirobokov, D.A., On using radial basis functions in a "finite difference mode" with applications to elasticity problems. *Computational Mechanics*, **33**(1), pp. 68–79, 2003.

[4] Zienkiewicz, O.C. & Zhu, J.Z., A simple error estimator and adaptive procedure for practical engineering analysis. *International journal for Numerical Methods in Engineering*, **24**(2), pp. 337–357, 1987.

[5] Angulo, A., Pozo, L.P. & Perazzo, F., A posteriori error estimator and an adaptive technique in meshless finite points method. *Engineering Analysis with Boundary Elements*, **33**(11), pp. 1322–1338, 2009.

[6] Afshar, M.H., Naisipour, M. & Amani, J., Node moving adaptive refinement strategy for planar elasticity problems using discrete least squares meshless method. *Finite Elements in Analysis and Design*, **47**(12), pp. 1315–1325, 2011.

[7] Davydov, O. & Oanh, D.T., Adaptive meshless centres and RBF stencils for Poisson equation. *Journal of Computational Physics*, **230**(2), pp. 287–304, 2011.

[8] Kosec, G. & Šarler, B., H-adaptive local radial basis function collocation meshless method. *Computers Materials and Continua*, **26**(3), p. 227, 2011.

[9] Mitchell, W.F., A collection of 2D elliptic problems for testing adaptive grid refinement algorithms. *Applied Mathematics and Computation*, **220**, pp. 350–364, 2013.

[10] Slak, J. & Kosec, G., Adaptive radial basis function-generated finite differences method for contact problems. *International Journal for Numerical Methods in Engineering*, 2019. DOI: 10.1002/nme.6067.

[11] Oanh, D.T., Davydov, O. & Phu, H.X., Adaptive RBF-FD method for elliptic problems with point singularities in 2D. *Applied Mathematics and Computation*, **313**, pp. 474–497, 2017.

[12] Bayona, V., Flyer, N., Fornberg, B. & Barnett, G.A., On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. *Journal of Computational Physics*, **332**, pp. 257–273, 2017.

[13] Slak, J. & Kosec, G., On generation of node distributions for meshless PDE discretizations. arXiv preprint arXiv:1812.03160, 2018.

[14] Medusa. Medusa: coordinate free implementation of meshless methods, http://e6.ijs.si/medusa/. Accessed on: 10 Jun. 2019.