Comparison of local weak and strong form meshless methods for 2-D diffusion equation

R. Trobec[1]*, G. Kosec[2], M. Šterk[1], B. Šarler[2]

[1]*Department of Communication Systems, Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia*

[2]*Laboratory for Multiphase Processes, University of Nova Gorica, Vipavska 13, SI-5000 Nova Gorica, Slovenia*

Abstract

A comparison between weak form meshless local Petrov-Galerkin method (MLPG) and strong form meshless diffuse approximate method (DAM) is performed for the diffusion equation in two dimensions. The shape functions are in both methods obtained by moving least squares (MLS) approximation with the polynomial weight function of the fourth order on the local support domain with 13 closest nodes. The weak form test functions are similar to the MLS weight functions but defined over the square quadrature domain. Implicit timestepping is used. The methods are tested in terms of average and maximum error norms on uniform and non-uniform node arrangements on a square without and with a hole for a Dirichlet jump problem and involvement of Dirichlet and Neumann boundary conditions. The results are compared also to the results of the finite difference and finite element method. It has been found that both meshless methods provide a similar accuracy and the same convergence rate. The advantage of DAM is in simpler numerical implementation and lower computational cost.

Key words: meshless methods, MLPG, DAM, weak form, strong form, diffusion equation, error comparison, regular node arrangement, non-regular node arrangement.

1 Introduction

In recent years, a number of meshless approaches have been developed for the numerical solution of partial differential equations (PDEs) to circumvent the problem of polygo-

* Corresponding author: Božidar Šarler, E-mail: bozidar.sarler@ung.si, Phone: +386 5 3315 246, Fax: +386 5 3315 359.

nisation encountered in the classical, mesh-based numerical methods, such as the finite difference (FDM) [1] and finite volume methods (FVM) [2], finite element method (FEM) [3] or boundary domain integral method (BDIM) [4]. The "mesh" denotes the connectivity between the corresponding neighboring nodes obtained by some sort of spatial discretization. For non-structured three dimensional geometries from real world the mesh construction is one of the most cumbersome step in the entire numerical solution process [5].

Instead of using a mesh, a set of geometrically unconnected nodes can be used for the global domain discretization, resulting in the meshless methods (MLM) (also referred to as mesh-free methods). Initial concepts of MLM were published in the Seventies with the smoothed particle hydrodynamics (SPH) [6], based on the combination of weak-form and collocation techniques. Many variants of the meshless methods have been developed later, based on different strong/weak formulations and approximation/interpolation techniques. Diffuse element method (DEM) [7], element-free Galerkin method (EFG) [8], reproducing kernel particle method (RKPM)[9], hp-cloud method [10], partition of unity FEM (PUFEM) [11], meshless Galerkin method using radial basis functions (MGRBF) [12–14], and others, are based on the global weak-form. These methods became widely used in different areas of application. General finite difference method (GFDM) [15,16], with arbitrary mesh, radial basis function collocation methods (RBFCM) [17] and finite point method (FPM) [18], which are based on collocation, are meshless methods devised from the global strong-form. In spite of the simple formulation, additional strategies are needed for their stabilization [19]. All above MLM methods do not need the mesh for the construction of the trial function but by weak-form methods a simple background mesh is still needed for the weighted residual integration. Detailed information on a variety of meshless methods can be found in the review papers [20,21].

An alternative formulation of the meshless methods is based on the local weak-form. The integration can now be performed in independent quadrature subdomains, therefore no background mesh is needed. Interpolation [22,23] or moving least squares (MLS) approximation of nodal parameters from subdomains can be used for the construction of shape functions. Even though the interpolation simplifies the implementation of the essential boundary conditions, there is no general method known to solve the possible singularities of the interpolation coefficient matrix in case of degenerated distribution of nodes or to ensure the continuity of the solution. On the other hand, MLS is less influenced by inappropriate nodal distribution in the support domain because it is over-determined by default. It is an essential approach that guarantees the locality of many meshless methods. The shape functions are built from the weighted contributions of a certain number of the nearest nodes [24,25]. However, the approximative nature of the MLS trial functions makes it more difficult to impose essential boundary conditions because the trial functions do not pass through the nodal values.

The meshless local Petrov-Galerkin method (MLPG) [26,27] is the earliest representative of the 'truly' meshless methods. Several variants of MLPG differ in the construction of trial functions and the use of different test functions. Note that both of them are local. We will further test the MLPG1, a variant of MLPG [28], which uses the MLS weight function as a test function over a square quadrature subdomain. New methods that combine previous knowledge from FEM and BEM with the MLPG approach have been proposed in [29]. Several convergence issues have been addressed [28,12,30], mostly in connection with the structural analysis, and often on smaller systems with a uniform distribution of nodes. The convergence of MLPG methods depends on the MLS approximation accuracy and stability and on the integration accuracy of the local weak-form, all three depending on the distribution of discretization nodes [31,32]. Further studies have confirmed that MPLG is a general concept that can be applied in various fields of application [33–37].

An alternative, much simpler local meshless approach, is Diffuse Approximate Method (DAM) [47]. Similar to the MLPG, in the DAM, the local trial functions are constructed through the MLS approach over the local support domain. In the DAM "local" refers to the locality of the MLS trial functions. The conceptual difference is in the treatment of partial differential operations. The DAM uses strong-form and thus no integration is required. All differential operations are performed by straightforward application of the differential operator on the trial function.

To our knowledge this is the first work that compares errors of two meshless local methods MLPG and DAM, with most of method-specific parameters the same, e.g. local subdomain, base functions, nodal distribution. Thus the comparison of the weak and strong solution principles remains the primary focus of the present paper. As a reference, we add also the errors of the two mesh-based methods, FDM and FEM, obtained under the same circumstances. The meshless methods are claimed to perform well in situations with complicated geometry and non-uniform node arrangement resulting in smoother solution than mesh-based methods. However, since these methods are in development, they are in the great majority of numerical simulations [39,38,14] demonstrated only on simple geometries with regular node arrangements. Also direct comparisons between different meshless methods are very rarely found in the literature. The main incitement of this paper is in contributing towards a better understanding of the meshless approaches. Respectively, we compare the well known meshless local Petrov-Galerkin method (MLPG), which is based on the week formulation, and the local diffuse approximate method (DAM), based on the strong formulation. The methods are tested in solving the dimensionless diffusion equation. We use uniform and non-uniform node arrangements and simple as well as more complicated geometry for assessment. Errors were evaluated in discretization nodes on a square domain and on a domain with a hole with Dirichlet boundary conditions. Our analysis has taken into account different spatial domain discretization ranging from nodes on a simple orthogonal mesh, through randomized nodes with an approximately even density, to nodes on a fully optimized triangular meshes.

The structure of the paper is as follows. In the next section a short background on the meshless approaches used for the solution of a diffusion equation are given. Then the test conditions are described in more details. In Section 4, the obtained results are presented and analyzed. The paper concludes with the discussion on the obtained results and future work.

2 Governing equation and discretization

2.1 *Diffusion equation*

We compare the numerical solutions, obtained by two meshless methods MLPG1 and DAM, with the standard mesh-based methods FDM and FEM. The tested PDE is a diffusion equation in its 2-D dimensionless form:

$$
\begin{aligned}
u_{,t} &= u_{,xx} + u_{,yy}, && (x,y) \in \Omega, && (1)\\
u(x,y,t) &= \overline{u}(x,y,t), && (x,y) \in \Gamma_D, && (2)\\
u(x,y,t)_{,n} &= \overline{g}(x,y,t), && (x,y) \in \Gamma_N,, && (3)\\
u(x,y,t) &= u_0, && t = 0, && (4)
\end{aligned}
$$

where Ω, and Γ_D and Γ_N stand for interior, Dirichlet boundary and Neumann boundary of the domain, (x, y) are spatial coordinates, t is time, $u(x, y, t)$ is the unknown solution, the part of subscript following the comma denotes partial derivatives, $\overline{u}$ and $\overline{g}$ are the prescribed Dirichlet and Neumann boundary conditions and $u(x, y, 0) = u_0$ the known initial condition.

We use the term 'node', denoted by $\mathbf{x}_i$, for the domain discretization point. Any other points in the domain, e.g. evaluation points for calculation of numerical integrals, are denoted by $\mathbf{x}$. The unknown solution in node $\mathbf{x}_i$ at time t is $u(\mathbf{x}_i, t)$.

The numerical solutions of the diffusion equation (1), presented in this work, are obtained by using our own Matlab [40] implementations, except FEM, which is obtained by COMSOL Multiphysics [41]. The first test domain is the unit square $\Omega = [0, 1] \times [0, 1]$, shown in the left part of Fig. 1, with initial conditions $u_0 = 1$ and essential boundary conditions $\overline{u} = 0$, for which the exact solution is known [42]. In the case of mixed boundaries the upper and right boundaries remain Dirichlet type while left and lower boundaries are changed to insulation Neumann type. For the sake of convenient data presentations the domain size remains the same. The second test domain is the unit square with a hole of radius 0.15 centered at (0.6, 0.35), shown in the right part of Fig. 1, for which the reference solution is obtained by FEM with the number of degrees of freedom DOFs = 111156, shape functions of degree two and integration order of four. The error of the reference solution is two orders of magnitude smaller than the investigated errors of our test cases.

The analyzed test case is evolving in time, which influences the solution error. The changes in the solution gradients are most pronounced at the beginning of the simulation, which contributes to the error from inaccurate representation of the development of the solution with time. Additionally, the steep gradients can not be accurately reconstructed with the limited number of shape functions, which is the second source of errors. As a consequence, the discrete approximation could contribute also to the error in the integration needed by FEM and MLPG. Based on the previous findings [31], we compare all methods at time $t = 0.005$, where all tree possible sources of errors contribute significantly to the finial solution error.

2.2 Node positioning

The numerical solution of PDE is based on spatial discretization of the solution domain Ω and its boundary Γ, which converts a PDE into a system of algebraic or ordinary differential equations. The spatial discretization significantly influences the solution accuracy. In order to test the sensitivity of the solution methods to the nodal distribution we use several spatial domain discretizations within the whole range of inter-nodal distances h down to $h = 5 \cdot 10^{-3}$.

In the regular discretization, the nodes are distributed in a square grid with inter-nodal distance $h = 1/(\sqrt{N} - 1)$.

The non-regular discretization is devised from regular discretization with random shifts in each dimension of the regularly distributed nodes by up to, e.g., $irreg = \pm\{0.1\overline{h}, 0.2\overline{h}, 0.3\overline{h}\}$, where $\overline{h}$ is the average inter-nodal distance. Such a discretization is not applicable to the FDM. The FEM mesh can be obtained from it by triangularization of nodes but such a discretization is not optimal and the results induce higher discretization errors. The meshless methods must use an appropriate data structure, such as k-d tree, to efficiently identify the support domains.

The FEM discretization is in our case a triangular mesh implemented by incrementally improving the mesh quality and using denser nodes near the boundaries of the domain. We obtained the FEM meshes used from COMSOL by using maximal possible quality of the elements. For the case of quadratic shape functions, additional internal nodes are generated automatically and added to the original nodes from the triangular mesh, which is evident from the number of DOFs. All nodes have been used as FEM discretization nodes. The FEM discretization can be used also in DAM and MLPG1.

We also test a distribution with four nodes in the corners, and a proportional number on the boundaries, but was otherwise completely random. The only method which works with such a discretization is FEM, however, the error increases for two orders of magnitude. Therefore we do not test further such a random discretization.

In the left part of Fig. 1 the regular (up) and non-regular (middle) discretization with $irreg = \pm 0.3\overline{h}$ on the unit square domain without hole for $N = 1600$ are shown. FEM discretization is shown for 1633 DOFs (bottom), i.e. nodes of FEM shape functions of degree 2. The right part of Fig. 1 shows the same discretizations for similar nodal densities but on the domain with a hole with $N = 1528$ for the regular (up) and non-regular (middle) discretization, and with 1464 DOFs for FEM discretization (below).

In the unit square domain the nodes are always far enough from boundaries in all the discretizations, e.g. by at least $(1 - irreg)\overline{h}$ in case of non-regular discretization. On

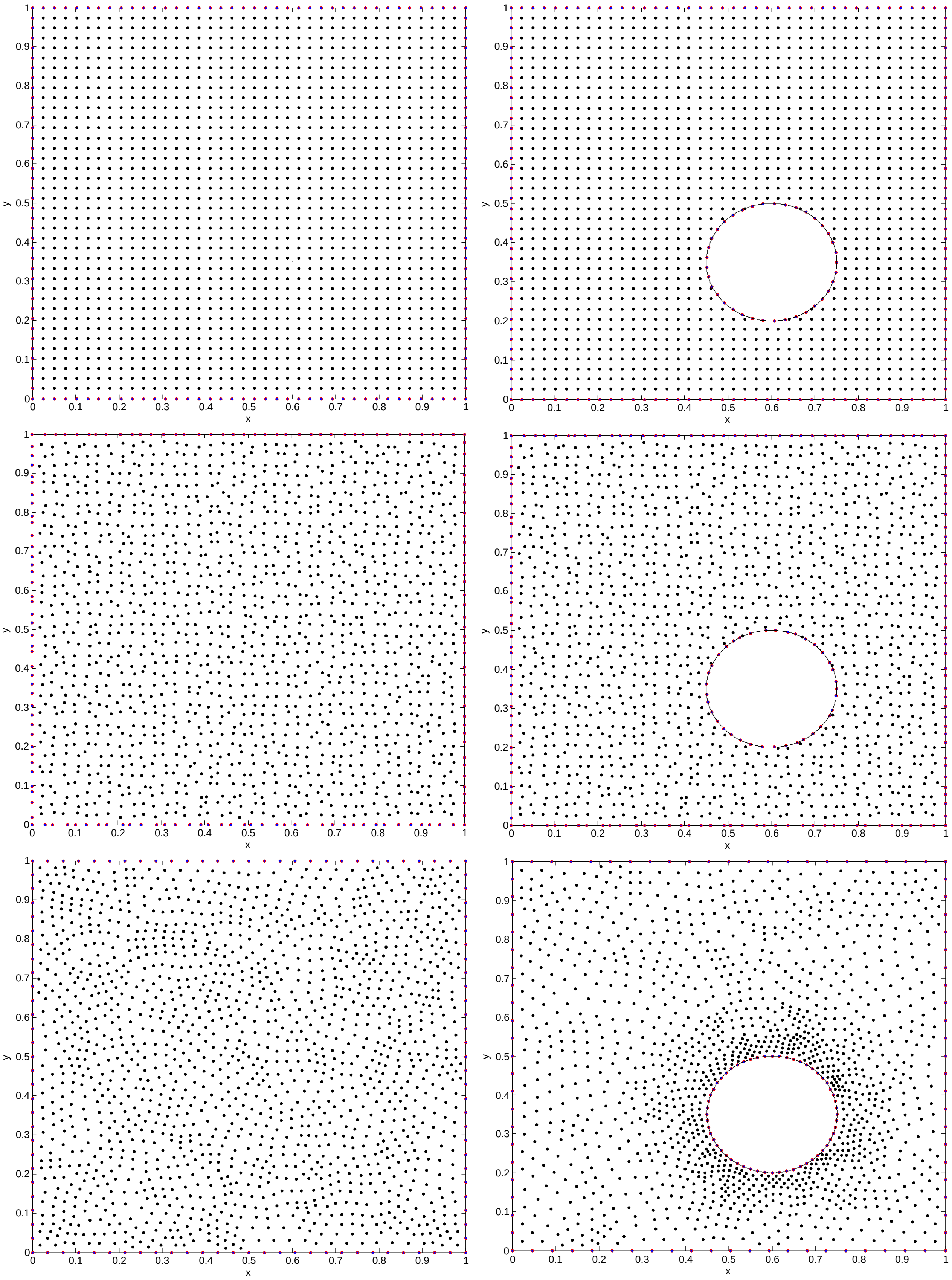

Fig. 1. Typical examples of regular discretization (upper row), non-regular discretizations for $irreg = \pm 0.3\overline{h}$ (middle row) and FEM discretization (bottom row) on a domain without hole (left column) and with a hole (right column) for the average inter-nodal distance $\overline{h} = 2.6 \cdot 10^{-2}$.

the other hand, in the domain with a hole, some of the nodes from the regular and the non-regular discretizations can fall in the vicinity of hole boundary or even very close to a node on the hole boundary, which may produce numerical instability and inaccurate integration. In case of MLPG1 and DAM, nodes are therefore removed if their distances to the hole boundary are below $10^{-3}\overline{h}$, which is the threshold that has produced the best results. For FDM, the best threshold is $0.5\overline{h}$, because in this case the errors near the hole boundary span evenly between positive and negative values. These restrictions need not to be applied in FEM discretization that manages non-orthogonal or curved boundaries by its meshing algorithm.

2.3 Time discretization

If the PDE is time-dependent, as in our case (1), the PDE is converted into a system of ordinary differential equations (ODE) with nodal parameters $u_i(t)$ as unknowns. Therefore a time discretization is also needed, which transforms the system of ODE into a system of algebraic equations. The solution procedure starts from the initial conditions $u(x, y, t_0)$ and generate approximate solution for every time-step Δt. The procedure for the generation of the final algebraic system of equations depends on the formulation, which is weak in MLPG1, and strong in DAM. We use in both meshless methods the Crank-Nicolson time discretization scheme:

$$\mathbf{u}(t+\frac{\Delta t}{2}) = \frac{\mathbf{u}(t+\Delta t)+\mathbf{u}(t)}{2}, \qquad \mathbf{u}_{,t}(t+\frac{\Delta t}{2}) = \frac{\mathbf{u}(t+\Delta t)-\mathbf{u}(t)}{\Delta t}. \tag{5}$$

This scheme is unconditionally stable and achieves second order accuracy in time [43].

Because our primary goal is the comparison of methods in terms of errors and convergence rate rather than computation time, we select a time-step $dt = 10^{-6}$ that guarantees the stability of all numerical methods for all tested systems but introduces insignificant time-discretization error, as compared with the domain-discretization error.

2.4 Error norms used in comparison

The error analysis and comparison of all analyzed methods is performed by examining the error in nodes over the whole domain. Besides the error in nodes, we also evaluate the error on a dense reference grid with 220×220 points. In FEM and MLPG1 the evaluated solution values are obtained from their native shape functions. Direct evaluation of reference grid error is not possible in FDM and DAM, because the solution is known only in nodes, therefore it was interpolated to the reference grid by *griddata* Matlab function. In all test cases, the evaluated solution error was slightly higher than the error evaluated just in the nodes. With the increasing number of the nodes both errors become practically the same, because the nodal density is approaching to the density of the evaluation grid. We conclude that for the tests, presented in this paper, the error in the nodes suffices. Note that the regular FDM works only on orthogonal mesh.

We have analyzed the accuracy of the methods in terms of maximal absolute error E_∞ in the L_∞ norm and the relative error E_2 in the L_2 norm defined as:

$$E_\infty = \max_{\mathbf{x}_i \in \Omega} |\hat{u}(\mathbf{x}_i) - u(\mathbf{x}_i)| \qquad E_2 = \left(\frac{\sum_{\mathbf{x}_i \in \Omega} (\hat{u}(\mathbf{x}_i) - u(\mathbf{x}_i))^2}{\sum_{\mathbf{x}_i \in \Omega} u(\mathbf{x}_i)^2} \right)^{1/2}. \tag{6}$$

2.5 Moving least squares

The domain Ω is discretized by a set of nodes $\mathbf{x}_i$. The approximate solution $\hat{u}$ is a linear combination of shape functions ϕ_i with nodal parameters u_i as the coefficients:

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N} u_j \phi_j(\mathbf{x}) = \mathbf{u}^{\mathrm{T}} \Phi(\mathbf{x}), \qquad \mathbf{u}^{\mathrm{T}} = [u_1, \ldots, u_N]. \tag{7}$$

Both meshless methods, MLPG1 and DAM, obtain the shape functions in the same way by the moving least squares (MLS) approximation, which will be described first. The MLS approximation is an extension of the classical least squares, in that just a few nearest nodes affect the approximated value at any point $\mathbf{x}$. The locality is implemented by hat-shaped MLS weight functions w [20], which are non-zero on the circular MLS *support domain* Ω_S of the point of interest $\mathbf{x}$:

$$w(\mathbf{x}, \mathbf{x}_i) = w\left(\frac{||\mathbf{x}_i - \mathbf{x}||_2}{r_S(\mathbf{x})} \right), \quad \text{where } w(d) = \begin{cases} 1 - 6d^2 + 8d^3 - 3d^4 & d \le 1 \\ 0 & d > 1 \end{cases} \tag{8}$$

and $r_S(\mathbf{x})$ is the radius of Ω_S. The MLS support of each point contains n_S support nodes $\mathbf{x}_i$, which locally approximate the unknown function $\hat{u}(\mathbf{x})$:

$$\hat{u}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \mathbf{a}(\mathbf{x}), \tag{9}$$

where $\mathbf{p}$ is a set of m basis functions evaluated in point $\mathbf{x}$, e.g. $m = 6$ for monomials up to degree 2 in 2-D with $\mathbf{p}^{\mathrm{T}} = [1, x, y, x^2, xy, y^2]$, and $a_i(\mathbf{x})$ are unknown coefficients obtained by minimizing:

$$e_{MLS} = \sum_{\mathbf{x}_i \in \Omega_S(\mathbf{x})} w(\mathbf{x}, \mathbf{x}_i)(\hat{u}(\mathbf{x}_i) - u_i)^2. \tag{10}$$

We select $n_S = 13$ based on the experiments published in [31]. The solution of (10) results in a system of linear equations:

$$A_{(m \times m)} \mathbf{a} = B_{(m \times n_S)} \mathbf{u}_S, \tag{11}$$

where $\mathbf{a}$ stands for approximation coefficients, $\mathbf{u}_S$ for nodal parameters of nodes from Ω_S and

$$A_{i,j} = \sum_{k=1}^{n_S} w(\mathbf{x}, \mathbf{x}_k) p_i(\mathbf{x}_k) p_j(\mathbf{x}_k), \quad B_{i,j} = w(\mathbf{x}, \mathbf{x}_j) p_i(\mathbf{x}_j). \tag{12}$$

The approximated trial function $\hat{u}(\mathbf{x})$ can be reproduced by substituting the solution of (11) into (9):

$$\hat{u}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x}) A^{-1}(\mathbf{x}) B(\mathbf{x}) \mathbf{u}_S = \Phi(\mathbf{x}) \mathbf{u}_S. \tag{13}$$

In an alternative but equivalent view, a shape function ϕ_j, is equal to the approximation of the value 1 in node x_j and 0 in all other nodes. It is determined solely by the nodal positions and is independent of the nodal parameters. The derivatives of the shape functions Φ on the i-th independent variable x_i, again denoted for simplicity by a subscript after a comma, can be obtained as proposed in [20]:

$$\Phi_{,x_i} = \mathbf{g}^{\mathrm{T}}_{,x_i} B + \mathbf{g}^{\mathrm{T}} B_{,x_i} = (\mathbf{p}^{\mathrm{T}}_{,x_i} - \mathbf{g}^{\mathrm{T}} A_{,x_i}) A^{-1} B + \mathbf{g}^{\mathrm{T}} B_{,x_i}, \tag{14}$$

where $\mathbf{g}$ is an auxiliary vector $\mathbf{g} = \mathbf{p}^{\mathrm{T}} A^{-1}$. In the MLPG solution process, the shape functions Φ and their derivatives can be evaluated in any domain point, in particular in integration points. We will show in the next section that in the DAM, the trial functions and their derivatives are needed in nodes only, which simplifies the evaluation procedure.

2.6 Meshless local Petrov-Galerkin method

The MLPG method can be derived through the *weighted residual method* [44]. The MLPG equations for internal nodes can be constructed by stating that the residual $r(\mathbf{x})$ is orthogonal to a set of test functions W_i:

$$\int_{\Omega} W_i(\mathbf{x}) r(\mathbf{x}) \, d\Omega = 0. \tag{15}$$

where the residual $r(\mathbf{x})$ of Eq. (1) is obtained by replacing u with $\hat{u}$ and subtracting the right from the left side of the PDE:

$$r(\mathbf{x}) = \hat{u}_{,t} - (\hat{u}_{,xx} + \hat{u}_{,yy}). \tag{16}$$

The equations for MLPG boundary nodes are constructed by collocation using the known essential boundary conditions. The test function W_i can in principle be any nonzero function. The various versions of MLPG differ mainly in the type of test functions W_i used in the weak form (15). Test functions W_i are non-zero only in the vicinity of the node $\mathbf{x}_i$, which is called the node's *quadrature domain* Ω_{Qi}. Consequently, the integration in (15) is performed only locally over Ω_{Qi}.

The MLPG1 uses the MLS weight functions as test functions. We analyze the version of MLPG1, which uses in its 2-D version the hat-shaped test function over the square quadrature domain Ω_{Qi}, as proposed in [30]:

$$W_i(x,y) = \begin{cases} \left(1 - \left(\frac{x-x_i}{d_Q(\mathbf{x}_i)/2}\right)^2\right) \cdot \left(1 - \left(\frac{y-y_i}{d_Q(\mathbf{x}_i)/2}\right)^2\right) & x, y \in \Omega_{Qi} \\ 0 & x, y \notin \Omega_{Qi} \end{cases} \tag{17}$$

where $d_Q(\mathbf{x}_i)$ is the side of the square quadrature domain d_Q. The dimension of Ω_{Qi} should be related to the point density in order to retain the required integration accuracy in the whole Ω. We relate the size of MLS support domain $\Omega_S(\mathbf{x}_i)$ and quadrature domain Ω_{Qi} by $d_Q(\mathbf{x}_i) = \beta_Q \, r_S(\mathbf{x}_i)$, with $\beta_Q = 0.7$, which was confirmed as the optimal value for MLPG1 with MLS of degree 2 [31].

By inserting Eq. (17) and (16) in Eq. (15) and using integration by parts on the two terms containing the second spatial derivative, we obtain:

$$\int_{\Omega_{Qi}} W_i \hat{u}_{,t} \, d\Omega_{Qi} - \int_{\Gamma_{QiB}} W_i \left(\hat{u}_{,x} n_x + \hat{u}_{,y} n_y\right) d\Gamma_{QiB} \quad + \int_{\Omega_{Qi}} \left(W_{i,x} \hat{u}_{,x} + W_{i,y} \hat{u}_{,y}\right) d\Omega_{Qi} = 0, \tag{18}$$

where $\Gamma_{QiB} = \Gamma_{Qi} \cap \Gamma$ is the part of the quadrature domain boundary Γ_{Qi} that coincides with the global domain boundary Γ. To obtain ODEs with nodal parameters $u_j(t)$ as unknowns, (7) is substituted into (18). One equation per internal domain node is thus obtained after the numerical integration in each Ω_{Qi} with contributions from all the domain nodes with non-zero shape functions over Ω_{Qi}, and the remaining equations for boundary nodes are obtained by collocation, resulting in the final system of ODEs:

$$C\mathbf{u}_{,t}(t) + K\mathbf{u}(t) - \mathbf{f} = 0, \tag{19}$$

where $\mathbf{u}$ is the vector of all nodal parameters, and by tradition C and K are the damping and the stiffness matrix, respectively, and $\mathbf{f}$ the load vector.

The MLPG1 shape functions obtained by MLS are non-polynomial and the integrals in Eq. (15) can only be evaluated numerically. We use Gaussian quadrature [45] with a square mesh of $n_q = n_G \times n_G$ standard Gaussian quadrature points in Ω_{Qi}. Most of MLPG1 results presented in this paper have been obtained with $n_G = 3$. In some specific cases of nodal positions, e.g. extremely uneven nodal density, or a node very close to the domain boundary, Gaussian integration fails to provide required accuracy. In such cases the recursive adaptive Simpson quadrature rule was applied, which significantly increases the time needed for integration [46].

In Fig. 2 the domain Ω, which is the unit square with a hole, is discretized by FEM discretization with 364 DOFs/nodes (dots) obtained from COMSOL meshing utility. A circular Ω_{Si} and square Ω_{Qi} with the side $d_Q = 0.7 r_S$ are shown for nodes $i = 22, 200$ and 300. Each MLS support is composed of $n_S = 13$ nodes (asterisks). In each quadrature domain there are $n_q = 9$ Gaussian quadrature points (crosses). We can see that the dimensions of the local supports are consistent with the local nodal densities.

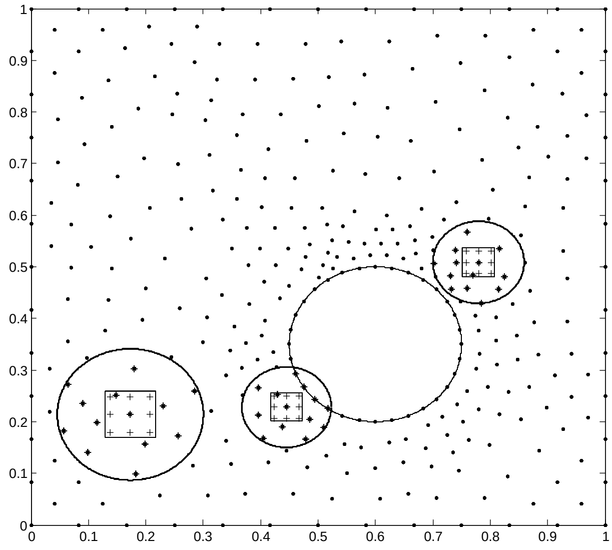

Fig. 2. Circular Ω_S and square Ω_Q for nodes 22, 200 and 300. Discretization nodes are marked with dots, MLS support nodes with asterisks, and quadrature points with crosses.

2.7 Diffuse approximate method

The strong formulation, which is the foundation of DAM, requires that the continuous spatial variables and derivatives of the PDE are replaced by their approximate values. In the original implementation of DAM [47] six monomials were used for basis functions with nine nodes in each subdomain. Various types of basis functions might appear in the calculation of the trial function. The most commonly used are multiquadrics, Gaussians, monomials, etc. If the number of basis functions m from Eq. (9) is the same as the number of support nodes n_S, the interpolation is used to determine the coefficients a_i. If the radial basis functions are employed besides monomials and collocation is used instead of MLS the method is known from the literature as Local Radial Basis Function Collocation Method (LRBFCM) [38,48–51]. The Diffuse Approximate Method (DAM), as implemented in this paper, uses the described MLS approach (9) for the local approximation of the unknown solution $\hat{u}(\mathbf{x})$.

To be comparable with the MLPG and remaining tested methods we apply for the basis functions all the monomials up to degree 2: $\mathbf{p}^{\mathrm{T}} = [1, x, y, x^2, xy, y^2]$, and the already defined hat-shaped weight functions (8) for the construction of the nodal trial functions with a constant number of MLS support nodes $n_S = 13$ in all Ω_{Si}. MLS support nodes are determined again as n_S closest nodes to a domain node $\mathbf{x}_i$.

To determine the approximation coefficients $\mathbf{a}$ the overdetermined system (11) has to be solved for each support domain Ω_{Si}. With known $\mathbf{a}$ the approximate value of the spatial derivatives in node $\mathbf{x}_k$ can be computed by applying the derivation operators x_j on the nodal trial functions (9):

$$\hat{u}(\mathbf{x}_k)_{,x_j} = \sum_{i=1}^{m} p_i(\mathbf{x}_k)_{,x_j}\, a_i \ . \tag{20}$$

The above expression can be written in a compact form by inserting explicit computation of approximation coefficients:

$$\hat{u}(\mathbf{x}_k)_{,x_j} = \sum_{i=1}^{m} \left(\sum_{j=1}^{n_S} A^{-1}_{i,j} B_{i,j}\, u_j \right) p_i(\mathbf{x}_k)_{,x_j}. \tag{21}$$

By applying the basic summation rules, can Eq. (21) be rewritten in a more convenient form:

$$\hat{u}(\mathbf{x}_k)_{,x_j} = \sum_{j=1}^{n_S} \left(\sum_{i=1}^{m} A^{-1}_{i,j} B_{i,j}\, p_i(\mathbf{x}_k)_{,x_j} \right) u_j. \tag{22}$$

Spatial derivation of basis functions is simple. Equation (22) represents in fact the relation between field values in the support domain nodes and value of arbitrary partial differential operation in the central node of the support domain. If working with static nodal distributions, the second summation in (22) can be computed in pre-processing phase and therefore only n_S operations are needed to compute arbitrary differential operator on the local support domain Ω_S.

3 Results

3.1 Discretization error

Visualized typical shapes of numerical solutions for the described test cases $\hat{u}(x, y, 0.005)$ are shown in Fig. 3. The solutions are shown for non-regular discretization with $irreg = \pm 0.3\overline{h}$ and $N = 40 \times 40 = 1600$ nodes on the domain with no hole (left) and with the same nodal density on the domain with a hole (1528 inner nodes and 193 boundary nodes), which are the spatial discretizations shown in the middle part of Fig. 1. Note that the DAM solutions are defined in nodes only therefore they are shown with dots, which are triangularized with a mesh for better visibility. On the lower part of Fig. 3 the corresponding solution errors are shown, again in nodes. Error distributions have similar shapes also for other numerical methods and for systems with a larger number of the nodes. They have peaks near corners and boundaries, and are larger in regions with steeper gradients near boundaries.

Our primary aim is to compare the local weak and strong form meshless methods, MLPG1 and DAM, on both test domains with all three discretizations and with different nodal densities. We run also FDM and FEM, both with second order convergence, as references, only on their native discretization, i.e. the regular and the triangular FEM mesh. All simulation parameters were kept constant as described in previous sections. The E_∞ is shown in Fig. 4 for the domain with no hole (left) and the domain with a hole (right), for

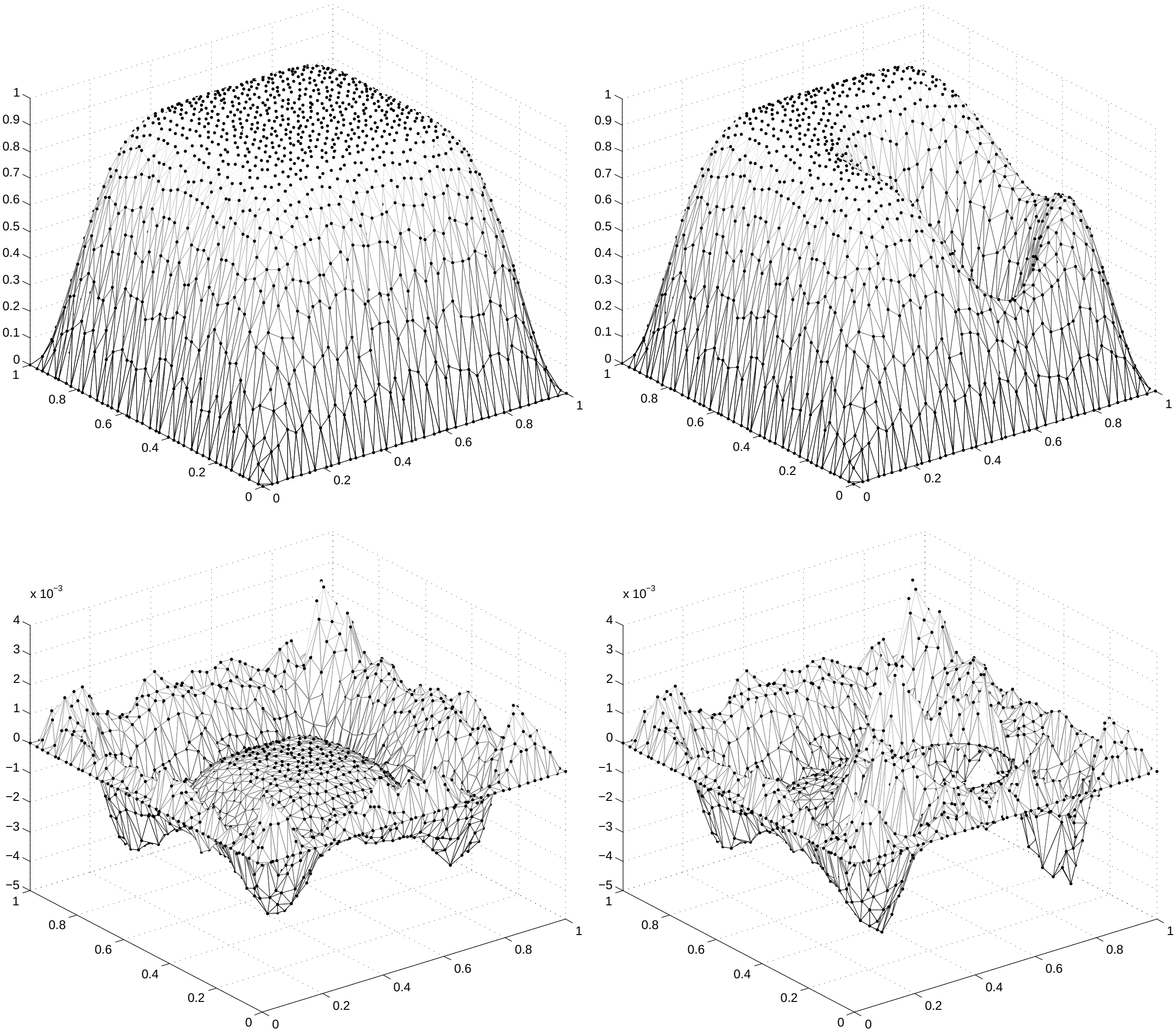

Fig. 3. Numerical solutions and errors of our test cases obtained by DAM for non-uniform discretization from the middle part of Fig.1 with $irreg = \pm 0.3\overline{h}$ and the average inter-nodal distance $\overline{h} = 2.6 \cdot 10^{-2}$.

regular (up) and non-regular discretization with $irreg = \pm 0.3\overline{h}$ (middle), and for FEM discretization (bottom).

The relative error E_2 is also calculated, as well. It behaves in the same way as E_∞ being constantly smaller for approximately a factor of two, however it can hide some single error spikes. The numerical errors for both error norms, for different number of domain nodes N and for both domains are given in Table 1 for regular discretization, in Table 2 for non-regular discretization with $irreg = \pm 0.3\overline{h}$ and in Table 3 for the FEM discretization.

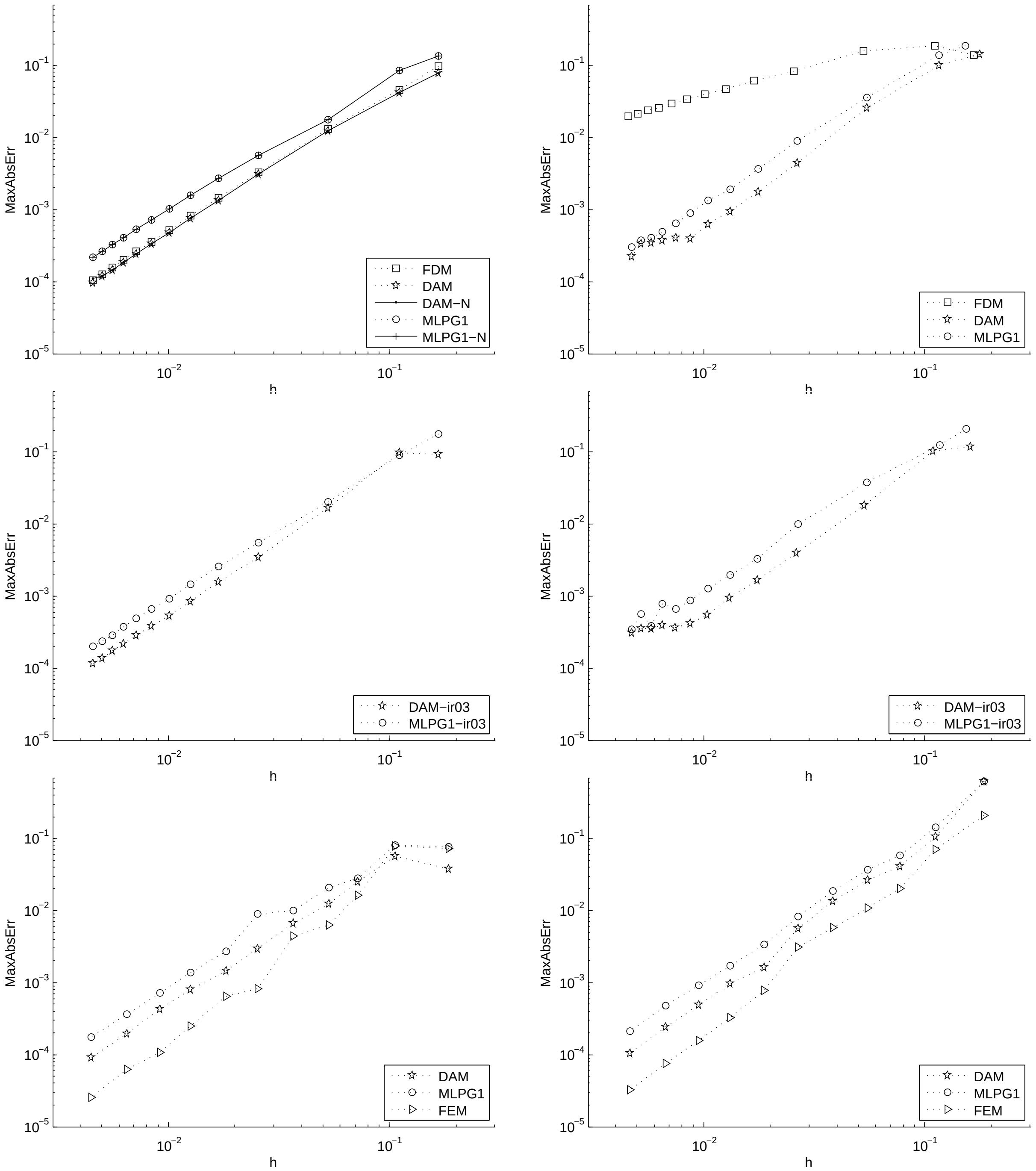

Fig. 4. Comparison of E_∞ of FDM, FEM, DAM, and MLPG1 solutions obtained on regular (upper row), non-regular (middle row) and FEM (bottom row) discretization on a domain with no hole (left column) and with a hole (right column), as a function of average inter-nodal distance $\overline{h}$.

3.2 Computation time

Both meshless techniques are compared regarding the computation time. As the methods have been implemented in MatLab and executed on the same computer (Intel Core

	MLPG1 - no hole		DAM - no hole		FDM - no hole	
$\overline{h}$	E_∞	E_2	E_∞	E_2	E_∞	E_2
1.7E-01	8.29E-02	7.03E-02	9.73E-02	6.45E-02	9.87E-02	6.50E-02
1.1E-01	4.47E-02	3.09E-02	4.51E-02	2.76E-02	4.55E-02	2.80E-02
5.3E-02	1.18E-02	7.23E-03	1.31E-02	6.73E-03	1.33E-02	6.82E-03
2.6E-02	4.05E-03	2.21E-03	3.27E-03	1.60E-03	3.33E-03	1.62E-03
1.7E-02	1.98E-03	1.06E-03	1.43E-03	6.95E-04	1.46E-03	7.07E-04
1.3E-02	1.17E-03	6.15E-04	7.97E-04	3.87E-04	8.15E-04	3.94E-04
1.0E-02	7.67E-04	4.02E-04	5.05E-04	2.45E-04	5.18E-04	2.50E-04
8.4E-03	5.43E-04	2.84E-04	3.48E-04	1.69E-04	3.58E-04	1.73E-04
7.2E-03	4.06E-04	2.11E-04	2.53E-04	1.24E-04	2.60E-04	1.27E-04
6.3E-03	3.14E-04	1.63E-04	1.92E-04	9.49E-05	1.98E-04	9.75E-05
5.6E-03	2.49E-04	1.30E-04	1.51E-04	7.52E-05	1.55E-04	7.75E-05
5.0E-03	2.03E-04	1.05E-04	1.21E-04	6.14E-05	1.25E-04	6.35E-05
4.6E-03	1.68E-04	8.74E-05	9.95E-05	5.14E-05	1.04E-04	5.33E-05
	MLPG1 - hole		DAM - hole		FDM - hole	
$\overline{h}$	E_∞	E_2	E_∞	E_2	E_∞	E_2
1.6E-01	1.77E-01	1.72E-01	1.41E-01	9.98E-02	1.38E-01	9.79E-02
1.1E-01	1.38E-01	8.61E-02	1.02E-01	5.96E-02	1.86E-01	7.57E-02
5.1E-02	3.61E-02	2.19E-02	2.58E-02	1.35E-02	1.59E-01	3.12E-02
2.5E-02	8.94E-03	5.05E-03	4.47E-03	2.74E-03	8.31E-02	1.24E-02
1.6E-02	3.60E-03	2.18E-03	1.76E-03	1.05E-03	6.14E-02	8.74E-03
1.2E-02	1.91E-03	1.22E-03	9.56E-04	5.29E-04	4.70E-02	6.23E-03
9.7E-03	1.29E-03	7.70E-04	6.20E-04	3.26E-04	3.95E-02	4.43E-03
8.1E-03	8.77E-04	5.15E-04	3.95E-04	2.15E-04	3.40E-02	3.52E-03
6.9E-03	6.33E-04	3.65E-04	4.07E-04	1.54E-04	2.93E-02	3.55E-03
6.1E-03	4.78E-04	2.73E-04	3.72E-04	1.24E-04	2.59E-02	3.09E-03
5.4E-03	3.77E-04	2.11E-04	3.46E-04	1.09E-04	2.39E-02	2.45E-03
4.8E-03	3.73E-04	1.71E-04	3.37E-04	1.00E-04	2.12E-02	1.87E-03
4.4E-03	2.57E-04	1.35E-04	2.23E-04	9.91E-05	1.96E-02	2.24E-03

Table 1
Numerical errors E_∞ and E_2 for MLPG1, DAM and FDM on a domain with no hole and with a hole, and regular discretization with different nodal densities $\overline{h}$.

i7 CPU, 2.8GHz) under the same operating system (64 bit Windows 7), the measured execution time can be compared. Most of the support functions, i.e. initial placement of nodes, finding the support domains, evaluation of errors, are equal for both methods. The solution of the final liner system with the sparse banded matrix was obtained by iterative solver (`bicgstab`). The systems are similar for both methods and can be solved in time proportional to $O(N)$. The most time consuming procedure is the creation of the system matrix. We have to determine the MLS supports and to solve an over determined system for each node, in the case of DAM, or for each integration point, in the case of MLPG1. Computation time is shown in Fig. 5. We experience a significant difference between DAM and MLPG1, in the favor of DAM, for more than an order of magnitude.

The numerical values for the computation time of DAM (t_{DAM}) and MLPG1 (t_{MLPG1})

	MLPG1 - no hole		DAM - no hole	
$\overline{h}$	E_∞	E_2	E_∞	E_2
1.7E-01	1.21E-01	7.02E-02	1.13E-01	6.27E-02
1.1E-01	8.12E-02	3.53E-02	7.32E-02	2.97E-02
5.3E-02	1.82E-02	8.27E-03	1.82E-02	6.95E-03
2.6E-02	4.75E-03	2.28E-03	3.43E-03	1.58E-03
1.7E-02	2.17E-03	1.03E-03	1.63E-03	7.43E-04
1.3E-02	1.26E-03	6.19E-04	8.57E-04	3.96E-04
1.0E-02	8.10E-04	4.00E-04	5.52E-04	2.51E-04
8.4E-03	6.04E-04	2.79E-04	3.84E-04	1.69E-04
7.2E-03	4.41E-04	2.12E-04	2.59E-04	1.21E-04
6.3E-03	3.29E-04	1.64E-04	2.15E-04	9.76E-05
5.6E-03	2.58E-04	1.29E-04	1.64E-04	7.47E-05
5.0E-03	2.07E-04	1.03E-04	1.34E-04	6.36E-05
4.6E-03	1.78E-04	8.70E-05	1.13E-04	4.97E-05
	MLPG1 - hole		DAM - hole	
$\overline{h}$	E_∞	E_2	E_∞	E_2
1.6E-01	2.67E-01	1.99E-01	1.18E-01	8.71E-02
1.1E-01	1.25E-01	8.89E-02	1.04E-01	4.71E-02
5.1E-02	3.74E-02	2.00E-02	1.83E-02	1.04E-02
2.5E-02	9.83E-03	5.20E-03	4.03E-03	2.15E-03
1.6E-02	3.22E-03	1.96E-03	1.67E-03	9.43E-04
1.2E-02	2.15E-01	6.10E-03	9.54E-04	5.13E-04
9.7E-03	1.20E-03	6.78E-04	5.46E-04	3.11E-04
8.1E-03	8.26E-04	4.61E-04	4.15E-04	2.07E-04
6.9E-03	6.27E-04	3.31E-04	3.67E-04	1.56E-04
6.1E-03	4.51E-04	2.44E-04	3.90E-04	1.30E-04
5.4E-03	3.73E-04	1.89E-04	3.55E-04	1.10E-04
4.8E-03	3.64E-04	1.54E-04	3.56E-04	1.04E-04
4.4E-03	3.52E-04	1.29E-04	3.12E-04	1.04E-04

Table 2
Numerical errors E_∞ and E_2 for MLPG1 and DAM on a domain with no hole and with a hole, and non-regular discretization with $irreg = \pm 0.3\overline{h}$ and different nodal densities $\overline{h}$.

are given in Table 4. In the last column the ratio (t_{MLPG1}/t_{DAM}) is shown, which is much larger by smaller systems. The DAM requires less memory and therefore uses the computer cache memory in a more efficient way.

It was shown in [52] that the asymptotical computational complexity of the construction of the final linear system in MLPG1 is $O(Nn_q[\log N + n_S m^2])$, n_q and n_S representing respectively the number of local quadrature points and the number of nodes in the support. The construction of the final linear system in DAM is done in $O(Nn_S^3)$. We see from Table 4 that the measured computation time is in accordance with these findings because in our case $n_q = 9$. We need one additional evaluation of the MLPG1 solution for the calculation of errors. The remaining complexity of MLPG1 comes from the searching of support nodes and reconstructing trial functions for each quadrature point. Note that there are n_q times

	MLPG1 - no hole		DAM - no hole		FEM - no hole	
$\overline{h}$	E_∞	E_2	E_∞	E_2	E_∞	E_2
1.9E-01	7.71E-02	6.83E-02	3.75E-02	2.93E-02	7.25E-02	4.32E-02
1.1E-01	8.05E-02	4.01E-02	5.65E-02	2.73E-02	7.92E-02	3.69E-02
7.2E-02	2.82E-02	1.63E-02	2.49E-02	1.03E-02	1.64E-02	6.15E-03
5.3E-02	2.09E-02	1.15E-02	1.26E-02	7.51E-03	6.24E-03	2.78E-03
3.7E-02	9.96E-03	4.55E-03	6.67E-03	2.77E-03	4.38E-03	1.24E-03
2.5E-02	8.98E-03	3.03E-03	2.97E-03	1.48E-03	8.22E-04	2.81E-04
1.8E-02	2.72E-03	1.35E-03	1.47E-03	6.75E-04	6.42E-04	2.33E-04
1.3E-02	1.37E-03	7.00E-04	8.03E-04	3.57E-04	2.49E-04	8.92E-05
9.2E-03	7.12E-04	3.51E-04	4.28E-04	1.88E-04	1.06E-04	4.52E-05
6.5E-03	3.65E-04	1.79E-04	1.97E-04	9.09E-05	6.20E-05	2.65E-05
4.5E-03	1.73E-04	8.30E-05	9.12E-05	4.22E-05	2.52E-05	1.42E-05
	MLPG1 - hole		DAM - hole		FEM - hole	
$\overline{h}$	E_∞	E_2	E_∞	E_2	E_∞	E_2
1.7E-01	6.14E-01	3.58E-01	6.13E-01	2.59E-01	2.08E-01	1.36E-01
1.0E-01	1.45E-01	8.36E-02	1.06E-01	4.63E-02	7.11E-02	3.64E-02
7.1E-02	5.79E-02	3.71E-02	4.10E-02	1.65E-02	2.04E-02	1.11E-02
5.1E-02	3.71E-02	1.70E-02	2.65E-02	1.20E-02	1.08E-02	4.85E-03
3.6E-02	1.85E-02	8.34E-03	1.33E-02	5.14E-03	5.76E-03	1.88E-03
2.5E-02	8.17E-03	4.78E-03	5.67E-03	2.78E-03	3.11E-03	9.36E-04
1.7E-02	3.36E-03	2.02E-03	1.63E-03	9.74E-04	7.77E-04	2.68E-04
1.2E-02	1.73E-03	1.02E-03	9.65E-04	4.99E-04	3.24E-04	1.95E-04
8.9E-03	9.18E-04	5.57E-04	4.98E-04	2.66E-04	1.59E-04	8.65E-05
6.3E-03	4.85E-04	2.82E-04	2.42E-04	1.32E-04	7.58E-05	4.07E-05
4.3E-03	2.12E-04	1.28E-04	1.03E-04	7.00E-05	3.27E-05	1.92E-05

Table 3
Numerical errors E_∞ and E_2 for MLPG1, DAM and FEM on a domain with no hole and with a hole, and FEM discretization with different nodal densities $\overline{h}$.

N	t_{DAM}	t_{MLPG1}	t_{MLPG1}/t_{DAM}
49.00	0.05	3.70	76.72
100.00	0.06	4.93	76.20
400.00	0.26	13.38	51.80
1600.00	1.22	53.26	43.72
3600.00	3.84	136.53	35.59
6400.00	8.69	279.30	32.14
10000.00	16.98	494.07	29.10
14400.00	32.67	823.60	25.21
19600.00	42.36	1104.10	26.06
25600.00	67.34	1620.30	24.06
32400.00	106.84	2308.40	21.61
40000.00	162.13	3200.00	19.74
48400.00	237.07	4361.40	18.40

Table 4
Computation time of DAM and MLPG1 for different number of nodes.

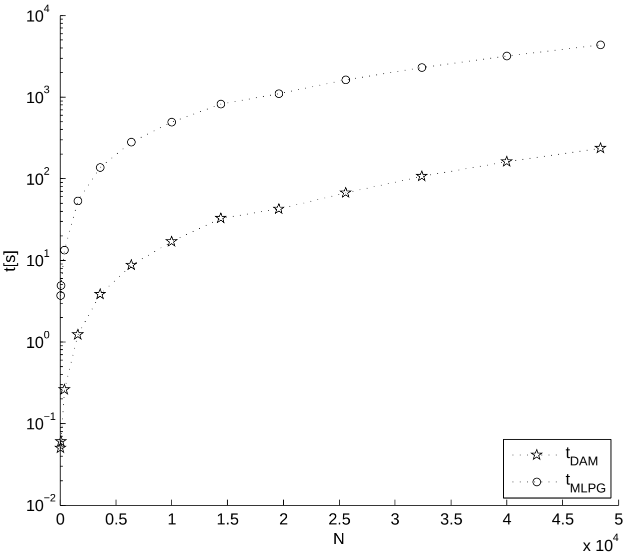

Fig. 5. Total computation time of DAM and MLPG1 as a function of the number of nodes.

more quadrature points than discretization nodes.

4 Discussion

All methods converge consistently on the unit square domain with no hole and regular discretization (see Fig. 4 upper left) for both test cases: Dirichlet boundary and mixed Dirichlet-Neumann boundaries (denoted in legend graphs by suffix N). In case of the non-regular discretization with $irreg = \pm 0.3$ and no-hole (Fig. 4 middle left) the convergence rate of the test case with only Dirichlet boundaries remains, however, the test with mixed Dirichlet-Neumann boundaries cannot provide a reasonable solution. For mixed boundary conditions, MLPG and DAM work only for smaller levels of randomization, up to $irreg = \pm 0.1$.

The number of Gaussian quadrature points in MLPG1 has been selected as $n_q = 3 \times 3$ ($n_G = 3$), based on the calculated solution error presented in Fig. 6, where E_∞ is plotted as a function of n_G for different numbers of domain nodes. The analysis has been performed on the test case with the Dirichlet boundaries and the domain with no hole. The solution error converges fast with the number of integration points. Higher number of quadrature points ($n_G > 3$) does not significantly decrease the solution error. The same conclusion has been reported in [31].

Although FEM discretization introduces uneven density of the nodes, the convergence

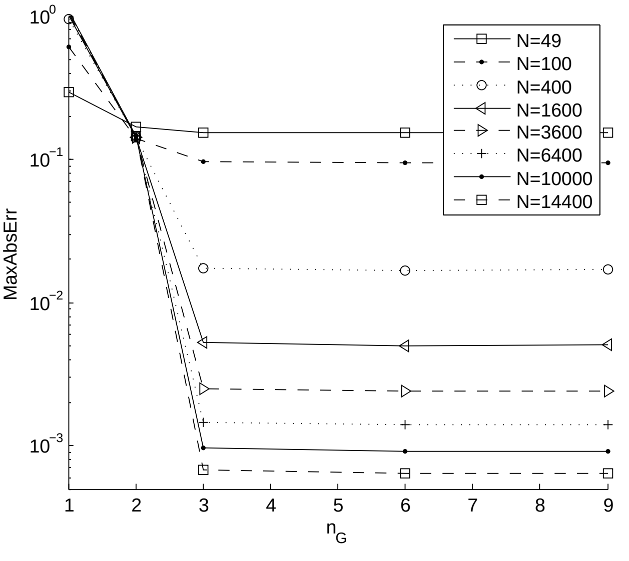

Fig. 6. E_∞ as a function of n_G for different number of domain nodes N.

rate still persists, in particular with denser nodes (Fig. 4 lower left). The most accurate method on the square domain with no hole is FEM, followed by DAM and MLPG1. On the regular discretization FDM provides the same results as DAM, as expected, with a slightly better convergence rate than MLPG1.

The behavior of methods is different on the square domain with a hole. In regular discretization (Fig. 4 upper right) FDM cannot, in most cases, use the nodes on the hole boundary and therefore significant errors appear there, regardless of the node density. The convergence rate of DAM and MLPG1 deteriorates in larger systems, even more so in case of non-regular discretization (Fig. 4 middle right). The problematic area lies also near the hole boundary. In order to retain the MLPG1 convergence rate by non-structured nodal distribution (Fig. 4 middle right) we remove all nodes that were closer to the hole boundary as $\overline{h}$ to lessen the area of the quadrature domain intersecting with the hole, which makes the integration on the hole boundary more simple.

In the case of FEM discretization (Fig. 4 lower right) the convergence rate of DAM is recovered because of denser nodes in the vicinity of the hole boundary. However, this is not the case by MLPG1. Because of significantly denser nodes near the hole, even if we remove nodes at distances less than $\overline{h}$, in some nodes the errors persisted. We have to introduce a special procedure to retain the convergence rate of MLPG1 on the FEM discretization.

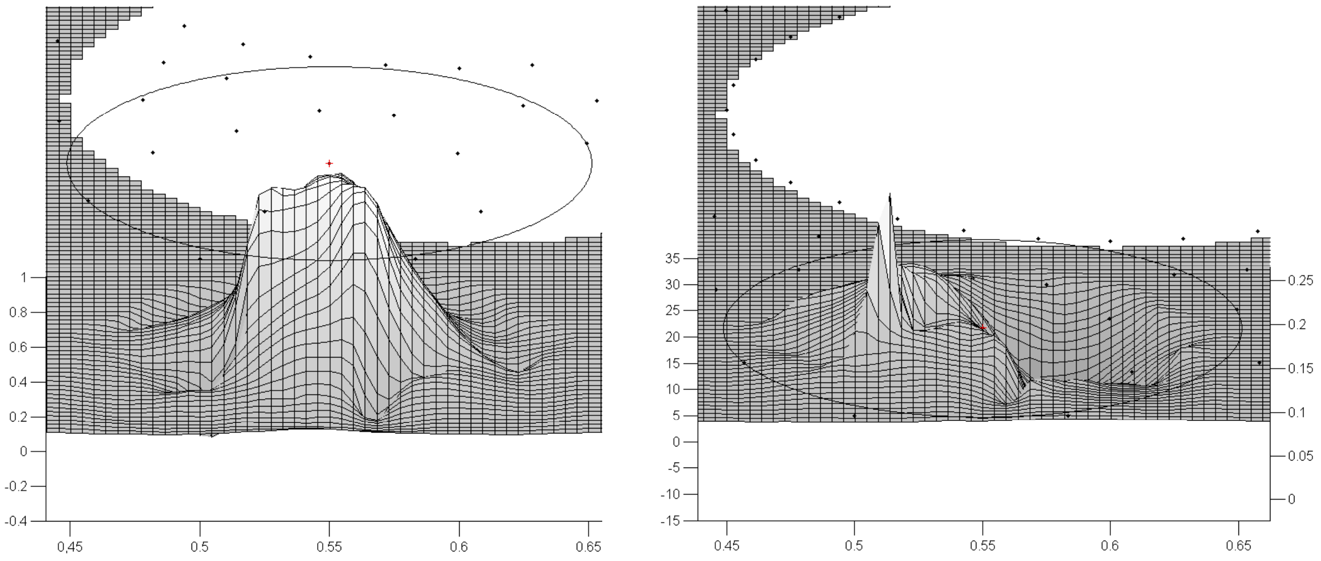

Fig. 7. Evaluated shape function of node 170 (left) and its derivative on x (right) in a part of the domain with a hole and FEM discretization with 364 DOFs.

We suspect, based on performing extensive further numerical experiments with applying adaptive integration in these near-hole-boundary nodes, that the Gaussian integration on square quadrature domains with dimension $d_Q = 0.7r_S$ is inadequate because it allows that the quadrature domain intersects in great part the hole boundary. We should know that in such cases the MLS shape functions and their derivatives are far from polynomial functions because they become zero in the hole. Therefore the weighted residual integrals (15) are hard to evaluate with any polynomial based quadrature rule.

In Fig. 7, a MLS shape function in the domain with a hole and FEM discretization with 364 DOFs (left) and its derivative on x (right) are shown, as an example. Both functions are evaluated on a 220 × 220 grid. They confirm their non-polynomial nature, which is particularly apparent in the case of non-regular nodal distribution. In order to retain the MLPG1 convergence rate for FEM discretization, shown in Fig. 4 (lower right) we use three times smaller quadrature domains in the nodes, which were closer to the hole boundary as $\overline{h}$, to lessen the area of the quadrature domain that falls in the hole.

5 Conclusions

In the present paper we compared two mehsless methods and two classical numerical approaches FDM and FEM. We have selected a weak and a strong form meshless techniques and kept all common parameters and procedures the same for both methods in order to create a reasonable comparison. The well known Dirichlet jump case has been used for the test case on a square domain. The symmetry of the problem was exploited to test the numerical methods on mixed Dirichlet-Neumann boundaries. Complicated boundary geometry is introduced with considering the domain with a hole.

We demonstrate that both meshless methods work with a similar performance with respect to the measured error, show the same spatial convergence rate and a similar behavior with respect to the randomization of the computational nodes. On the regular discretization the meshless methods perform similar as the FDM, which cannot be used on the irregular node arrangements. The FEM approach gives better results than both meshless methods on the specific FEM discretization.

Both meshless methods performed well on the domain with a hole with slight deterioration of the convergence rate by largest systems. Maximal errors have been detected near the hole, however with the FEM discretization such a behavior is diminished, because of denser nodes near the hole boundary.

We show that the inclusion of Neumann boundaries does not affect the performance of the meshless methods while working with regular node arrangements. The methods, however, fail to converge on irregular nodal distributions, when Neumann boundaries are present.

An important aspect of the numerical solution procedures is the computation time. We measure the computation time of both methods in the same software environment and by running on the same computer. It has been confirmed that MLPG1 is more complex to compute (order of 10 and more) in comparison to DAM. However, the DAM solution is known in nodes only. If the solution values between nodes are needed, they have to be obtained by interpolation, which introduces additional errors and computational cost.

We show that although the integration is an important issue in the MLPG1, refining it with more Gaussian quadrature points does not improve the solution error.

Our future work will be focused on the selection of the support domain as we suspect that performance of the considered meshless techniques might be improved by more sophisticated selection strategies.

Acknowledgement

The authors acknowledge the financial support from the state budget by the Slovenian Research Agency under grant No. P2-0095 and P2-0379.

References

[1] M. N. Özisik, Finite Difference Methods in Heat Transfer, CRC Press, Boca Raton, 1994.

[2] S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere, 1980.

[3] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The Finite Element Method: its basis and fundamentals, Elsevier Butterworth-Heinemann, 2005.

[4] P. K. Banerjee, The Boundary Element Methods in Engineering, McGraw-Hill College, 1994.

[5] S. J. Owen, A survey of unstructured mesh generation technology, in: Proceedings of 7th International Meshing Roundtable, Sandia National Laboratories, 1998, pp. 239–267.

[6] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. Roy. Astron. Soc. 181 (1977) 375–389.

[7] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: Diffuse approximation and diffuse elements, Computational Mechanics 10 (1992) 307–318.

[8] T. Belytschko, Y. Y. Lu, L. Gu, Element-free Galerkin methods, International Journal for Numerical Methods in Engineering 37 (1994) 229–256.

[9] W. K. Liu, J. Adee, S. Jun, Reproducing kernel and wavelet particle methods for elastic and plastic problems, in: D. J. Benson (Ed.), Advanced Computational Methods for Material Modeling, ASME, 1993, pp. 175–190.

[10] C. A. Duarte, J. T. Oden, Hp-cloud – a meshless method to solve boundary-value problems, Computer Methods in Applied Mechanics and Engineering 139 (1996) 237–262.

[11] I. Babuška, J. Melenk, The partition of unity method, International Journal of Numerical Methods in Engineering 40 (1997) 727–758.

[12] E. J. Kansa, Multiquadrics a scattered data approximation scheme with applications to computational fluid dynamics - ii. solutions to parabolic, hyperbolic and elliptic partial differential equations, Computers and Mathematics with Application 19 (1990) 147–161.

[13] H. Wendland, Error estimate for interpolation by compactly supported radial basis functions of minimal degree, Journal of Approximation Theory 93 (1998) 258–396.

[14] M. S. Ingber, C. S. Chen, J. A. Tanski, A mesh free approach using radial basis functions and parallel domain decomposition for solving three-dimensional diffusion equations, International Journal for Numerical Methods in Engineering 60 (2004) 2183–2201.

[15] V. Girault, Theory of a finite difference method on irregular networks, SIAM Journal on Numerical Analysis 11 (2) (1974) 260–282.

[16] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Computers and Structures 11 (1980) 83–95.

[17] M. D. Buhmann, Radial Basis Functions, Cambridge University Press, Cambridge, 2000.

[18] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor, A finite point method in computational mechanics. applications to convective transport and fluid flow, International Journal for Numerical Methods in Engineering 39 (1996) 3839–3866.

[19] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor, C. Sacco, A stabilized finite point method for analysis of fluid mechanics problems, Computer Methods in Applied Mechanics and Engineering 139 (1996) 315–346.

[20] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, Computer Methods in Applied Mechanics and Engineering, special issue on Meshless Methods 139 (1996) 3–47.

[21] V. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: A review and computer implementation aspects, Mathematics and Computers in Simulation 79 (2008) 763–813.

[22] Y. T. Gu, G. R. Liu, A local point interpolation method for static and dynamic analysis of thin beams, Computer Methods in Applied Mechanics and Engineering 190 (42) (2001) 5515–5528.

[23] G. R. Liu, Y. T. Gu, A local radial point interpolation method (LR-PIM) for free vibration analyses of 2-D solids, Journal of Sound and Vibration 246 (2001) 29–46.

[24] D. Shepard, Two-dimensional interpolation function for irregularly-spaced data, Proc 23rd Nat Conf (1968) 517–524.

[25] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, Mathematics of Computation 37 (1981) 141–158.

[26] S. N. Atluri, T. Zhu, A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics, Computational Mechanics 22 (2) (1998) 117–127.

[27] S. N. Atluri, H.-G. Kim, J. Y. Cho, Critical assessment of the truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) methods, Computational Mechanics 24 (5) (1999) 348–372.

[28] S. N. Atluri, S. Shen, The Meshless Local Petrov-Galerkin (MLPG) Method, Tech Science Press, 2002.

[29] G. R. Liu, Y. T. Gu, Meshless local Petrov-Galerkin (MLPG) method in combination with finite element and boundary element approaches, Computational Mechanics 26 (6) (2000) 536–546.

[30] G. R. Liu, Mesh Free Methods: Moving beyond the Finite Element Method, CRC Press, Boca Raton, 2003.

[31] M. Šterk, R. Trobec, Meshless solution of a diffusion equation with parameter optimization and error analysis, Engineering Analysis with Boundary Elements 32 (7) (2008) 567–577.

[32] A. Mazzia, M. Ferronato, G. Pini, G. Gambolati, A comparison of numerical integration rules for the meshless local Petrov-Galerkin method, Numerical Algorithms 45 (1-4) (2007) 61–74.

[33] S. N. Atluri, J. Y. Cho, H.-G. Kim, Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations, Computational Mechanics 24 (5) (1999) 334–347.

[34] J. Sorić, Q. Li, T. Jarak, S. N. Atluri, Meshless local Petrov-Galerkin (MLPG) formulation for analysis of thick plates, CMES - Computer Modeling in Engineering and Sciences 6 (4) (2004) 349–357.

[35] J. Sladek, V. Sladek, C. Zhang, Application of meshless local Petrov-Galerkin (MLPG) method to elastodynamic problems in continuously nonhomogeneous solids, CMES - Computer Modeling in Engineering and Sciences 4 (6) (2003) 637–647.

[36] Y. L. Wu, G. R. Liu, Y. T. Gu, Application of meshless local Petrov-Galerkin (MLPG) approach to simulation of incompressible flow, Numerical Heat Transfer, Part B: Fundamentals 48 (5) (2005) 459–475.

[37] H. K. Ching, R. C. Batra, Determination of crack tip fields in linear elastostatics by the meshless local Petrov-Galerkin (MLPG) method, CMES - Computer Modeling in Engineering and Sciences 2 (2) (2001) 273–289.

[38] B. Šarler, R. Vertnik, Meshfree explicit local radial basis function collocation method for diffusion problems, Computers and Mathematics with Applications 51 (8) (2006) 1269–1282.

[39] H. Lin, S. N. Atluri, Meshless local Petrov-Galerkin (MLPG) method for convection-diffusion problems, Computer Modeling in Engineering & Sciences 1 (2) (2000) 45–60.

[40] MathWorks MATLAB, http://www.mathworks.com/.

[41] COMSOL Multiphysics, http://www.comsol.com/products/multiphysics/.

[42] H. S. Carslaw, J. C. Jaeger, Conduction of Heat in Solids, Oxford University Press, London, 1959.

[43] M. T. Heath, Scientific Computing: An Introductory Survey, 2nd Edition, WCB/McGraw-Hill, New York, 2002.

[44] G. B. Pollard, Lectures on Partial Differential Equations, Wiley, New York, 1964.

[45] M. Abramowitz, I. A. Stegun, Handbook of mathematical functions with formulas, graphs, and mathematical tables, National Bureau of Standards, AMS55:New York, 1976.

[46] W. Gander, W. Gautschi, Adaptive quadrature - revisited, BIT 40 (2003) 84–101.

[47] C. Prax, H. Sadat, P. Salagnac, Diffuse approximation method for solving natural convection in porous media, Transport in Porous Media 22 2 (1996) 215–223.

[48] G. Kosec, M. Zaloznik, B. Šarler, H. Combeau, A meshless approach towards solution of macrosegregation phenomena, CMC: Computers, Materials and Continua 44 (2011) 65–95.

[49] G. Kosec, B. Šarler, Solution of thermo-fluid problems by collocation with local pressure correction, International Journal of Numerical Methods for Heat and Fluid Flow 18 (2008) 868–882.

[50] R. Vertnik, B. Šarler, Solution of incompressible turbulent flow by a mesh-free method, CMES: Computer Modeling in Engineering and Sciences 580 (2009) 1–27.

[51] R. Vertnik, B. Šarler, Local collocation approach for solving turbulent combined forced and natural convection problems, Adv. Appl. Math. Mech 3 (2011) 259–279.

[52] R. Trobec, M. Šterk, B. Robič, Computational complexity and parallelization of the meshless local Petrov-Galerkin method, Computers and Structures 87 (1-2) (2009) 81–90.