Numerical Algorithms (2020) 83:565–591
https://doi.org/10.1007/s11075-019-00694-x

ORIGINAL PAPER

A Matlab software for approximate solution of 2D elliptic problems by means of the meshless Monte Carlo random walk method

Sławomir Milewski[1]

Received: 16 December 2017 / Accepted: 21 March 2019 / Published online: 11 April 2019

Abstract
This paper is devoted to the development of an innovative Matlab software, dedicated to the numerical analysis of two-dimensional elliptic problems, by means of the probabilistic approach. This approach combines features of the Monte Carlo random walk method with discretization and approximation techniques, typical for meshless methods. It allows for determination of an approximate solution of elliptic equations at the specified point (or group of points), without a necessity to generate large system of equations for the entire problem domain. While the procedure is simple and fast, the final solution may suffer from both stochastic and discretization errors. The attached Matlab software is based on several original author's concepts. It permits the use of arbitrarily irregular clouds of nodes, non-homogeneous right-hand side functions, mixed type of boundary conditions as well as variable material coefficients (of anisotropic materials). The paper is illustrated with results of analysis of selected elliptic problems, obtained by means of this software.

Keywords Monte Carlo method · Random walk technique · Meshless methods · Elliptic problems · Finite difference method · Implementation in Matlab

1 Introduction

The large class of mechanical and civil engineering stationary (time-independent) problems may be modeled by means of the partial differential equations of elliptic type (e.g., torsional deflection of a prismatic bar, stationary heat flow, distribution of

✉ Sławomir Milewski
slawek@L5.pk.edu.pl

[1] Civil Engineering Department, Cracow University of Technology, Warszawska 24,
31-155 Kraków, Poland

electrical potential or filtration through porous media). Two different types of mathematical models may be investigated, depending on the problem variables' nature, namely deterministic and probabilistic ones. In deterministic models, a particular state (either continuous or discrete) is uniquely assigned to input data and no element of randomness occurs. In other words, all input parameters are represented by their mean values and one fixed data set corresponds to one unique solution. In most cases, the specified numerical solution approach has to be incorporated, as the analytical solution is limited to problems with simple geometry and uniformly distributed subjected load. Therefore, such problems ascribed by deterministic models are usually analyzed by means of the standard numerical approaches (*hard computing methods*, [22]), which are based upon appropriate domain discretization (partition into nodes and/or elements) and unknown function approximation, built upon those geometrical quantities (e.g., finite element method (FEM, [37]), finite difference method (FDM, [31, 32]), meshless methods (MM, [29, 31, 33, 35, 45]), or boundary element method (BEM)).

On the other hand, the probabilistic class of mathematical models specifies governing differential equations that combine one or more random variables (with assigned probability distribution function) with other non-random variables. Regardless of the applied solution approach, taking advantage from the problem's stochastic nature, one deals with the analysis of multiple standard problems, varying with input parameters' values. These auxiliary problems are usually investigated by means of the standard aforementioned deterministic methods. As a consequence, the entire family of solutions is obtained, out of which the optimal solution is selected, using appropriate averaging criteria. Among the most commonly applied probabilistic approaches (*soft computing methods*), which rule the principles of a random selection of the input data and averaging techniques, one may distinguish the following:

- Fuzzy sets (FS, [36]) analysis, in which the fuzzy models are applied to selection of input data, with assigned membership function, characterizing data randomness.
- Genetic and evolutionary algorithms (GA and EA, [24]), inspired by the biological evolution and dedicated to non-gradient analysis of multidimensional optimization problems.
- Artificial neural networks (ANN, [39]), inspired by biological nervous system, tailored for analysis of variety of problems by means of a-priori prepared patterns.
- Monte Carlo stochastic approach, in which series of simulations (trials), representing the analyzed problem, with randomly selected input values, are performed. Among these trials, the specified number of properly defined successes is achieved. The ratio between the number of success trials to the number of all trials, scaled by dimensional quantity (e.g., area, function value) allows for the estimation of the unknown solution, providing the number of trials is large enough.

The first mature formulation as well as technical applications of Monte Carlo (MC) concept were derived by S. Ulam and J. von Neumann in their pioneering research [1, 21], devoted to the neutron transport in an atomic bomb framework. It was followed by a variety of works delivered by many other researchers, who applied this simple concept to analysis of algebraic and differential problems, like eigenvalues estimation [3], solution of linear systems of equations [2, 4, 5], numerical integration in multidimensional spaces [13, 26] as well as numerical solution of Laplace differential equations [6, 7, 9, 15, 17, 20, 23, 27, 28, 30], at the selected internal point of the problem domain. This application is discussed in more detailed manner in this paper.

The main idea is based upon the classic MC concept combined with the so-called fixed random walk (RW, [6, 7, 9, 42, 53]) technique. This technique allows for a random selection of the path consisting of the horizontal and vertical walks. Those walks are performed on the regular grid of lines, starting from the given internal point (thus forming a Markov chain). This path has to be terminated at the domain boundary and the entire procedure is repeated as long as all random walks are completed. Therefore, the net of connections is generated, linking the selected internal point of interest with all points located on the boundary. The total sum of all numbers of boundary indications (*boundary hits*), scaled by their boundary values, estimates the solution of the Laplace equation at this particular point. It was proved [7], that this estimation corresponds to the finite difference solution, obtained on the same regular grid of nodes. This simple solution approach was further developed by many investigators in subsequent years. Apart from the standard RW, in which both step size and move directions are fixed and preassigned, two main random walk types were derived, namely semi floating random walk (step size is fixed, though the move direction is not limited, [47, 52]) and full floating random walk (step size is not preassigned and changes at each step, whereas the move direction is not limited, [8, 42, 47, 52]). Moreover, a continuous random walk procedure, minimizing the solution error has been proposed [17] along with the self-adaptive, grid-free algorithm, with improved solution smoothness and its application to diffusion equations [20]. State-of-the-art as well as recent developments (up to date) may be found in [23]. Further reading may include the following subjects: effective reduction of random walk steps' number [28], weighted version of MC method [30], MC treatment of complex 2D geometries [38]. Within the last several years, investigated were, for instance, reduction of MC error by new probabilistic sampling [41], development of stochastic limit theory and governing equations of continuous time random walk [44, 50], improvement of the simulation efficiency for radiative transfer problems with strongly frequency-dependent opacities [51], as well as the analysis of heat conduction equations [52].

In this research, special emphasis is laid upon the development of the meshless random walk technique, combined with the Monte Carlo approach, towards analysis of wider class of 2D elliptic problems. While the original MC/RW technique (Section 2) may be applied to Laplace equation with regular grid of lines, essential boundary conditions as well as homogeneous right-hand side function only, the proposed meshless version of MC/RW technique allows for analysis of 2D elliptic equation in more general form, including the following:

- More complex geometries, discretized by means of arbitrarily irregular clouds of nodes, without any imposed structure (therefore neither mesh size or move directions are fixed)

- Non-homogeneous right-hand side functions (typical for equations of Poisson type)
- Mixed type of boundary conditions (both essential and natural types may be taken into the account)
- Non-constant material coefficients

Similar stochastic concepts were successfully applied in variety of methods based upon the reduction of the solution of multidimensional problems for partial differential equations to integration of stochastic equations [43, 49]. Especially, the well-known Feynman-Kac formula [14, 34] is worth mentioning. It may be interpreted as a method for evaluating functional integrals of a specified continuous form. However, its practical application is based upon the stochastic approximation of these integrals [49], which incorporates both Monte Carlo approach and random walk procedure. Although the entire numerical procedure is similar to the one presented here, the mathematical formulation of the considered problem is more complex, since it requires determination of the explicit closed form solution formula. Moreover, the right-hand side function of the differential equation has to be integrated, which limits the number of potential applications to cases with smooth solution only (i.e., no concentrated loads). The meshless MC/RW, developed in [53] and implemented in Matlab package in this paper, uses the original deterministic differential equation as well as the computational framework which is typical for element-based and difference methods (i.e., existence of nodes (though with no imposed structure), approximation schemes, solution smoothing). However, those aspects are incorporated into the pure stochastic approach.

All innovative ideas and concepts, proposed and introduced in [53], are briefly presented and discussed here (Section 3). They are based upon the incorporation of selected discretization and approximation schemes, typical for meshless methods (MM). Especially, features of the moving weighted least squares approximation (MWLS) are applied in order to reformulate classic random walk principles. However, the main part of this paper is devoted to the presentation of the innovative Matlab software (Section 4), based upon those concepts. It may work in several various modes, for instance, it allows either for a determination of an approximated solution of an elliptic equation at selected point for fixed number of random walks, or for an examination of a solution convergence at selected point(s) with number of random walks increasing. Finally, determination of the solution at all nodes is possible, with additional smoothing of rough MC results. Moreover, for illustration purposes, full graphical interface has been designed, including visualization of construction of single random walks, solution convergence of N-type (i.e., with respect to the number of random walks), graphs of the final MC/RW solution and its first derivatives as well as the simplified FD versus MC/RW error analysis [40]. By default, a user works with rectangular, circular or triangular domain and irregular meshes/clouds of nodes, though any other discretization may be loaded from the disc file. Several examples of clouds of nodes corresponding to more complex geometries are attached as well. Results of their analysis, conducted by means of the attached Matlab software, are presented in Section 5. The paper is briefly concluded and directions of future work are mentioned.

2 Standard Monte Carlo random walk technique

The following 2D Laplace equation is considered as follows:

$$\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2} = 0 \quad in \quad \Omega \tag{1}$$

with essential boundary conditions as follows:

$$F = \bar{F}\left(x, y\right) \quad on \quad \partial\Omega \tag{2}$$

where $\Omega = \{(x, y)\} \in R^2$ is the problem domain, $\partial\Omega$—its boundary, $F = F\left(x, y\right) : \Omega \to R\left(C^2\right)$ is the unknown scalar function, with given values $\bar{F}\left(x, y\right)$ at every boundary point. Let us assume that the grid $\Omega_h \subset \Omega$ of mutually perpendicular, n_x horizontal and n_y vertical lines, was generated, forming a mesh of $n = n_x \times n_y$ regularly spaced points (Fig. 1). Function values $F_{i,j}$ at those points $(i = 1, ..., n_x, \quad j = 1, ..., n_y)$ constitute the set of unknowns, while values at all boundary nodes (located on $\partial\Omega_h \subset \partial\Omega$) are known. Regardless of whether the solution is sought at all grid points or at selected point (or group of points) only, it is required to combine all known (boundary) and unknown (internal) function values into one system of algebraic equations, providing the standard deterministic method is applied. Its generation and solution may be time-consuming, especially in case when Ω_h is very dense (number of nodes $n \to \infty$ or mesh size $h \to 0$, e.g., due to global mesh refinement).

Instead, the following stochastic procedure may be adopted, in case only one function value (e.g., at domain center point) is to be determined in relatively fast manner and with reasonable accuracy:

1. Initiate the subsequent random walk starting from the given internal node $\left(x_i, y_j\right)$, randomly selecting one of the fourth equally possible direction senses

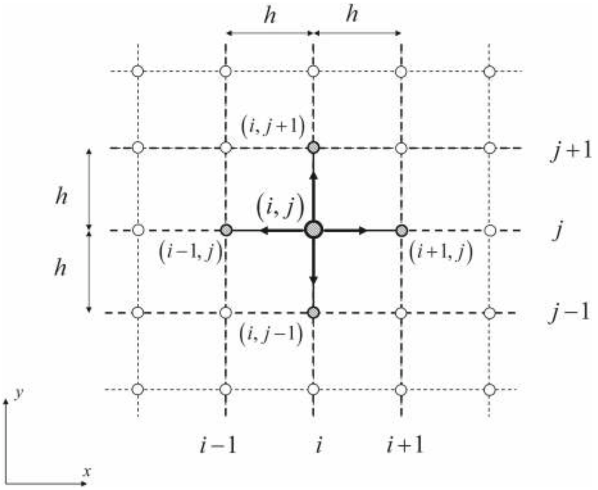


Fig. 1 Regular mesh of points with configuration of nodes (FD star) for a random walk and FD schemes

(north, east, south, or west). Move to the closest neighboring node, located in the selected direction $((i, j+1), (i-1, j), (i, j-1)$ or $(i+1, j)$, respectively (Fig. 1)), and repeat the random walk from this new node as long as the first boundary node (x_r, y_s) is reached. Evaluate the function value at this boundary node as follows:

$$\bar{F}_{r,s} = \bar{F}\left(x_r, y_s\right) \tag{3}$$

and modify the number of its indications $N_{r,s}^{(e)}$ by one (we initialize all $N_{r,s}^{(e)}$ with zeros).

2. Return to the node of interest $\left(x_i, y_j\right)$. Repeat the entire procedure described above until all N random walks are performed (N is the assumed total number of random walks).
3. Evaluate estimation of function value $F_{i,j}$, according to the Monte Carlo concept (in which reaching the boundary node is treated as a success trial), by means of the simple formula as follows:

$$F_{i,j} \approx \frac{1}{N} \sum_{r,s}^{n_b^{(e)}} \bar{F}_{r,s} N_{r,s}^{(e)} \tag{4}$$

where $n_b^{(e)}$ denotes the number of all boundary nodes with essential boundary conditions (i.e., all boundary nodes in this case).

It has been proved [7] that this stochastic formula (4) corresponds to the finite difference solution, obtained for the considered regular mesh of nodes Ω_h, namely

$$\begin{cases} F_{i,j} = \frac{1}{4}F_{i,j+1} + \frac{1}{4}F_{i-1,j} + \frac{1}{4}F_{i,j-1} + \frac{1}{4}F_{i+1,j} \quad for \quad (i,j) \in \Omega_h \\ F_{r,s} = \bar{F}_{r,s} \quad for \quad (r,s) \in \partial\Omega_h \end{cases} \tag{5}$$

The first equation in (5) is the difference equation, generated by means of the nodal collocation technique [46]. It is based upon the standard difference (FD) operator of Laplace type [31], replacing the differential operator from (1) at the central node (i, j) (Fig. 1), forming a configuration of five nodes, regularly spaced, called a FD star or stencil [31, 45]. It should be noted here that the difference coefficients assigned to four external nodal values (all equal to $\frac{1}{4}$) are the same as the probabilities of direction sense selection. On the other hand, a function value at the central node in (5) corresponds to the expected value defining the result of a random walk in (4), i.e., $F_{i,j} \sim E_{i,j} = \sum_{r,s}^{n_b^{(e)}} \bar{F}_{r,s} P_{i,j}^{r,s}$. Here, $P_{i,j}^{r,s}$ is the probability of random walk termination at boundary point (r, s), starting from the internal point (i, j). The number of random walks N has the crucial influence on the accuracy of estimation (4). Generally, the Monte Carlo solution error may be upper-bounded by the non-linear function of N, namely

$$e = ||F_{i,j} - \bar{F}_{i,j}|| < \frac{1}{\sqrt{N}} \tag{6}$$

However, this formula is valid for the simplest cases only, in which there is no curse of dimensionality (i.e., mesh size h has no influence on the solution estimation). In more complex problems, discussion on the optimal selection of N (as well as the relation between N and h) may be found in [20, 23, 28].

Although the standard Monte Carlo method with random walk technique (MC/RW) exhibits numerous advantages, for instance, an existence of an explicit formula for an approximate solution, algorithm simplicity or low computational cost, serious drawbacks may be noted either. First of all, the formula (4) works for regular meshes only, with equal and fixed mesh spacing (neither curvilinear edges nor local mesh refinements may be introduced). Moreover, it may be applied to Laplace equations only (no heterogeneity in right-hand side function), with constant material coefficients. Furthermore, boundary conditions (2) may be of essential type only. This paper, accompanied with the attached Matlab software, aims to extend the standard MC/RW method towards issues listed above. However, several aspects of the approach have to be reformulated and extended, including the following:

- Selection of potential random walk directions for the case of arbitrarily irregular cloud of nodes
- Determination of probabilities of a next move, depending on the equation type
- Significant reconstruction of the final MC/RW formula, taking all information concerning the analyzed problem into account in a stochastic manner

3 Meshless Monte Carlo random walk technique

The following 2D elliptic equation is considered as follows:

$$\frac{\partial^2 F}{\partial x^2} + a\left(x, y\right)\frac{\partial^2 F}{\partial y^2} = f\left(x, y\right) \quad in \quad \Omega \tag{7}$$

with essential (Dirichlet)

$$F = \bar{F}\left(x, y\right) \quad \text{on} \quad \partial\Omega_e \tag{8}$$

and natural (Neumann) boundary conditions

$$\frac{\partial F}{\partial n} = \bar{q}\left(x, y\right) \quad \text{on} \quad \partial\Omega_n \tag{9}$$

subjected to boundary parts $\partial\Omega_e$, $\partial\Omega_n$ ($\partial\Omega = \partial\Omega_e \cup \partial\Omega_n$). $f \in C^0$ is a scalar right-hand side function, $a = a\left(x, y\right) \in C^0$ is a scalar positive material function, whereas $\bar{q} \in C^1$ is a given scalar flux function, applied in the direction represented by the versor $\mathbf{n}\left(x, y\right) = \left[n_x\left(x, y\right), n_y\left(x, y\right)\right]^t$, normal to the boundary $\partial\Omega_n$. This equation may be a representative mathematical model for a stationary heat flow analysis [19]. Problem domain Ω may have complex geometry, therefore arbitrarily irregular cloud of nodes $\Omega_h = \left\{\left(x_i, y_i\right), i = 1, ..., n\right\} \in R^2$, without any a-priori imposed structure (like regular mesh, finite element [11], or mapping restrictions [16], has to be considered. Such discretization format is typical for the wide group of computational solution approaches, namely meshless methods (MM, [29, 31, 33, 35, 45]).

Moreover, in MM, the unknown function is approximated in terms of nodes only, by means of various techniques, like partition of unity or moving weighted least squares (MWLS, [18, 25, 31, 45, 46]). The MWLS approximation has been successfully applied in one of the oldest meshless methods, meshless finite difference method (MFDM, [10, 11, 16, 31, 45, 46]). The advantages of using MFDM and other MM may be observed in problems with large deformations, concentrated loads, moving boundary, crack development, elastic-plastic boundary, contact of deformable bodies, fluid free surface as well as in h-adaptive approach [12] since nodes may be shifted, added, or removed without any larger influence on nodes topology. In this research, we incorporate selected useful features of both the MWLS technique and the MFDM in order to extend the standard MC/RW technique towards effective stochastic analysis of (7) with (8) and (9).

3.1 Selection of random walk directions

A random selection of four mutually perpendicular direction senses, with equal probabilities, which is natural for regular mesh, does not hold in case of irregular clouds of nodes. New direction selection criteria have to be carried out, taking advantage from the nodes' irregular distribution. Let us consider the determination problem of potential walk directions, starting from the central node (i) towards selected nodes $(j(i))$, $(j(i) = 1, 2, ..., m)$, as it is depicted in Fig. 2a. Both the total number of nodes $(m + 1)$ in such configuration (denoted as the *MFD star*, or *stencil*) and their distribution should be assumed in such a manner that the resulting approximation scheme remains non-singular and non-ill-conditioned. Thus, m is usually larger than it is required from the order of differential operator (e.g., six nodes for 2D Laplace equation). The following criteria may be applied here as follows:

1. The simplest criterion is based upon the *distance* from (i) to (j) nodes only (Fig. 2b). Though very simple, it does not guarantee the well-conditioning. Therefore, the optimal nodes selection techniques are based rather on the topology information than distance between nodes only.
2. In 2D *cross criterion* [10, 16, 31, 45] (Fig. 2c), domain is divided into four zones. Each of four semi-axes is assigned to one of these zones. A specified number of nodes (usually 2), closest to the central node (point) is taken from every zone separately; thus, the number of nodes in the MFD star is always constant.
3. In more complex *Voronoi neighbous criterion* [31, 45] (Fig. 2d), only those nodes are selected to the MFD star which are the Voronoi neighbors (product of domain partitioning into a set of Voronoi polygons; each polygon is assigned to a particular node). Voronoi neighbors are those polygons which have common side (strong neighbors) or common vertex (weak neighbors). It should be stressed that the Voronoi neighbors criterion does not assure the same number of nodes in every star. Moreover, the number of nodes is variable and may not be sufficient to build full FD operator of the specified order. However, such star may be completed by additional nodes to preserve the required approximation order as well as good conditioning (e.g., the last selected ninth node in Fig. 2d, with attached dashed ray, results from the assumed distance criterion).

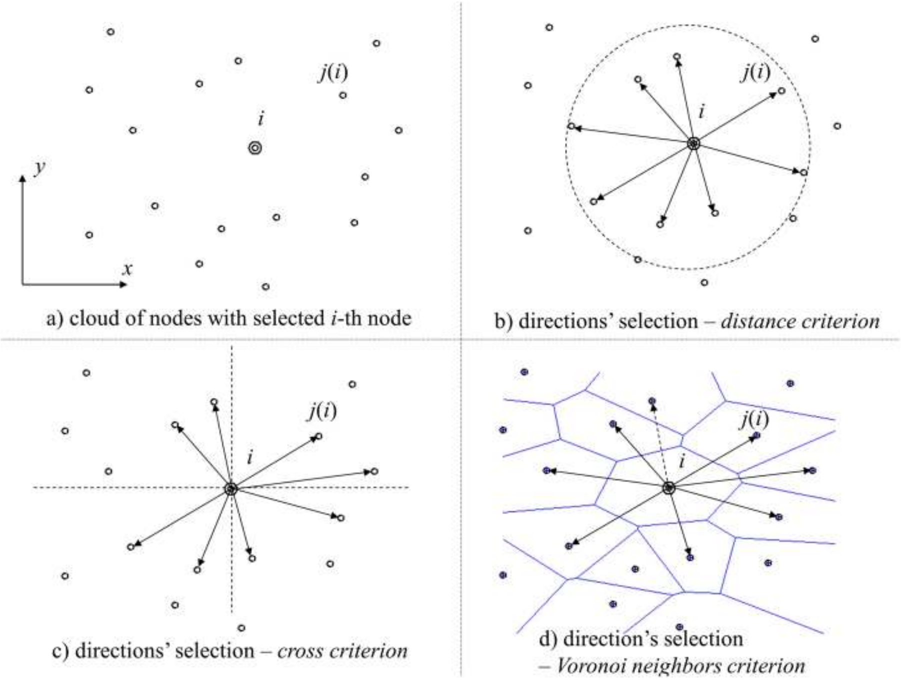

y
x
i
j(i)
a) cloud of nodes with selected i-th node
i
j(i)
b) directions' selection – distance criterion
i
j(i)
c) directions' selection – cross criterion
i
j(i)
d) direction's selection
– Voronoi neighbors criterion

Fig. 2 Cloud of nodes (**a**) and three main criteria for star selection (**b**, **c**, **d**)

Regardless of the applied selection technique, the final MFD star is ascribed by the topological $(m + 1) \times 4$ matrix **S**, namely

$$\mathbf{S} = \begin{bmatrix} 0 & 0(i) & x_0 = x_i & y_0 = y_i \\ 1 & 1(i) & x_1 & y_1 \\ 2 & 2(i) & x_2 & y_2 \\ \ldots & \ldots & \ldots & \ldots \\ j & j(i) & x_j & y_j \\ \ldots & \ldots & \ldots & \ldots \\ m & m(i) & x_m & y_m \end{bmatrix} \qquad (10)$$

containing local and global nodes' numbers as well as (x, y) coordinates of selected nodes corresponding to potential move directions, respectively, sorted in ascending order.

3.2 Determination of direction selection probabilities

Once m potential directions are assigned to the node (i), a procedure continues with the random selection of one of those directions. Respectively to the standard MC/RW method, probabilities of direction selection may be derived from the meshless FD operator, replacing the left-hand side differential operator from (7).

Moreover, the selection probability $P_j^{(i)}$, attached to the $j(i)$th star node/direction, should be proportional to the distance from node (i) to node $(j(i))$. Therefore, appropriate approximation scheme has to be determined. We use here the MWLS technique, which is based upon the Taylor series expansion of all star nodal values with respect to the central star node (i). In such a manner, resulting approximation coefficients have simple interpretation as local derivatives, up to the assumed pth order. Fundamentals and general remarks concerning the MWLS technique may be found in [18, 25, 31], whereas details of its application for the second-order differential equations are given in [45, 46, 48]. Comparison of MWLS (with singular weight functions) and MLS (with non-singular weight functions) techniques with other meshless approximation methods may be found in [29, 31, 33, 35]. Here, we focus on the final approximation formula (for $p = 2$) as follows:

$$\mathbf{D}_F = \mathbf{Mq}, \quad \mathbf{M} = (\mathbf{P}^T\mathbf{W}^2\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}^2 \tag{11}$$

which may be obtained after the minimization of the appropriate weighted error functional. $\mathbf{P}$ is the $(m+1)\times 6$ matrix of local interpolants $h_{x(j)} = x_j - x_i$, $h_{y(j)} = y_j - y_i$ (defined in $\mathbf{S}$), $\mathbf{W}$ is a diagonal $(m+1)\times(m+1)$ weight matrix, $\mathbf{M}$ is the $6\times(m+1)$ difference coefficients matrix, $\mathbf{D}_F$ is the 6×1 approximation coefficients vector, and $\mathbf{q}$ is the $(m+1)\times 1$ vector of degrees of freedom:

$$\mathbf{P} = \begin{bmatrix} 1 & h_{x(0)} & h_{y(0)} & 0.5h^2_{x(0)} & h_{x(1)}h_{y(0)} & 0.5h^2_{y(0)} \\ 1 & h_{x(1)} & h_{y(1)} & 0.5h^2_{x(1)} & h_{x(1)}h_{y(1)} & 0.5h^2_{y(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & h_{x(m)} & h_{y(m)} & 0.5h^2_{x(m)} & h_{x(m)}h_{y(m)} & 0.5h^2_{y(m)} \end{bmatrix} \tag{12}$$

$$\mathbf{W} = \begin{bmatrix} \omega_0 & 0 & \dots & 0 \\ 0 & \omega_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \omega_m \end{bmatrix}, \quad \mathbf{D}_F = \begin{bmatrix} F \\ F'_x \\ F'_y \\ F''_{xx} \\ F''_{xy} \\ F''_{yy} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} F_{0(i)} \\ F_{1(i)} \\ F_{2(i)} \\ F_{3(i)} \\ \dots \\ F_{m(i)} \end{bmatrix} \tag{13}$$

in which the singular weights are applied as follows:

$$\omega_j = \frac{1}{\left(\sqrt{h^2_{x(j)} + h^2_{y(j)}}\right)^3 + \epsilon}, \quad j = 0, 1, 2, \dots, m \tag{14}$$

with ϵ being a relatively small though non-zero (up to the machine precision) number (e.g., $\epsilon \sim 10^{-15}$ for double precision real numbers). Therefore, interpolation is forced at the central star node, in spite of using larger number of nodes than it is required from the assumed second-order differential operator order in (7) and the first-order differential operator in (9). Once $\mathbf{M}$ is determined, values of all derivatives included in $\mathbf{D}_F$ as well as value of any arbitrary differential operators (up to the second order) may be composed. By means of collocation technique at subsequent nodes with unknown function values, appropriate difference equations may be generated as follows:

$$\sum_{j=0}^{m_i} \left(M_{4,j} + a\left(x_i, y_i\right) M_{6,j} \right) F_{j(i)} = f\left(x_i, y_i\right) \tag{15}$$

$$F_k = \bar{F}\left(x_k, y_k\right) \tag{16}$$

$$\sum_{j=0}^{m_b} \left(n_x\left(x_l, y_l\right) M_{2,j} + n_y\left(x_l, y_l\right) M_{3,j} \right) F_{j(l)} = \bar{q}\left(x_l, y_l\right) \tag{17}$$

for nodes $(i) \in \Omega$, $(k) \in \partial\Omega_e$, $(l) \in \partial\Omega_n$, respectively. Here, m_i is the number of directions for the internal node, whereas m_b is the number of directions for the boundary node.

By rearranging terms in (15) and (17), one obtains the selection probabilities $P_j^{(i)}$ and $P_j^{(l)}$ of each jth potential direction of the subsequent random walk (Fig. 3), initialized from the ith or lth node, respectively

$$P_j^{(i)} = -\frac{M_{4,j} + a\left(x_i, y_i\right) M_{6,j}}{M_{4,i} + a\left(x_i, y_i\right) M_{6,i}} \quad , \quad j = 1, ..., m_i \tag{18}$$

$$P_j^{(l)} = -\frac{n_x\left(x_l, y_l\right) M_{2,j} + n_y\left(x_l, y_l\right) M_{3,j}}{n_x\left(x_l, y_l\right) M_{2,l} + n_y\left(x_l, y_l\right) M_{3,l}} \quad , \quad j = 1, ..., m_b \tag{19}$$

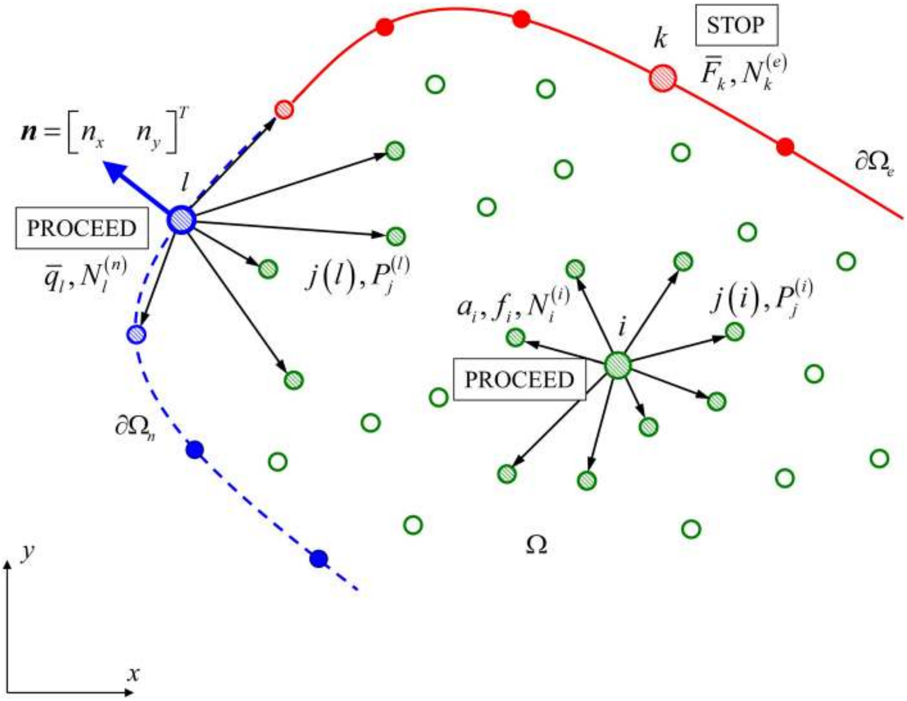


Fig. 3 Meshless random walk technique for an irregular cloud of nodes

It should be noted that $\sum_{j=1}^{m_i} P_j^{(i)} = \sum_{j=1}^{m_b} P_j^{(l)} = 1$, which is the probability of a certain event. In other words, the probability of selection of any other direction, not included within the FD star assigned to the ith or lth node, equals zero.

3.3 Modified Monte Carlo random walk formula

The principles of a random walk technique remain unmodified. We start from the specified arbitrary node located inside the domain (marked as a green circle in Fig. 3) or on its boundary with natural boundary conditions, where the function F is unknown (marked as a blue circle in Fig. 3). We proceed until the first node located on the boundary with essential boundary conditions is reached (marked as red circle in Fig. 3). Eventually, the final closed-form formula for the stochastic estimation of unknown function value F_j, by means of the meshless Monte Carlo method with random walk technique is as follows:

$$F_j \approx \frac{1}{N}\left(\sum_{k}^{n_b^{(e)}} \bar{F}_k N_k^{(e)} + \sum_{l}^{n_b^{(n)}} \frac{\bar{q}_l}{n_{x(l)} M_{2,l}^{(l)} + n_{y(l)} M_{3,l}^{(l)}} N_l^{(n)} + \right.$$
$$\left. + \sum_{i}^{n_i} \frac{f_i}{M_{4,i}^{(i)} + a_i M_{6,i}^{(i)}} N_i^{(i)} \right) \quad , \quad j \in \left[1, ..., n_i + n_b^{(n)}\right] \tag{20}$$

It takes into account series of all N random walks as well as all additional right-hand side components from (15) and (17). Here, $\bar{F}_k = \bar{F}(x_k, y_k)$, $\bar{q}_l = \bar{q}(x_l, y_l)$, $f_i = f(x_i, y_i)$, $a_i = a(x_i, y_i)$, $n_{x(l)} = n_x(x_l, y_l)$ and $n_{y(l)} = n_y(x_l, y_l)$. Various difference coefficients M correspond to both internal ($M^{(i)}$) as well as boundary nodes ($M^{(l)}$). Moreover, $n_b^{(e)}$ is the number of boundary nodes located on $\partial\Omega_e$, $n_b^{(n)}$ is the number of boundary nodes located on $\partial\Omega_n$, n_i is the number of internal nodes ($n_b^{(e)} + n_b^{(n)} + n_i = n$), whereas $N_k^{(e)}$, $N_l^{(n)}$, and $N_i^{(i)}$ are the numbers (counters) of kth, lth, and ith nodes' random walk indications, respectively. It should be emphasized here that the simplified error estimation formula (6) no longer holds, since the final MC solution (20) depends on both the number of random walks N and the physical discretization parameter(s) h, included in difference coefficients M. Therefore, new combined stochastic-deterministic-type estimation ε has to be defined, in the following general form as follows:

$$e = ||F_j - \bar{F}_j|| < \varepsilon(N, h) \tag{21}$$

Determination of ε as well as the optimal relation between N and h, will be considered in the following author's papers.

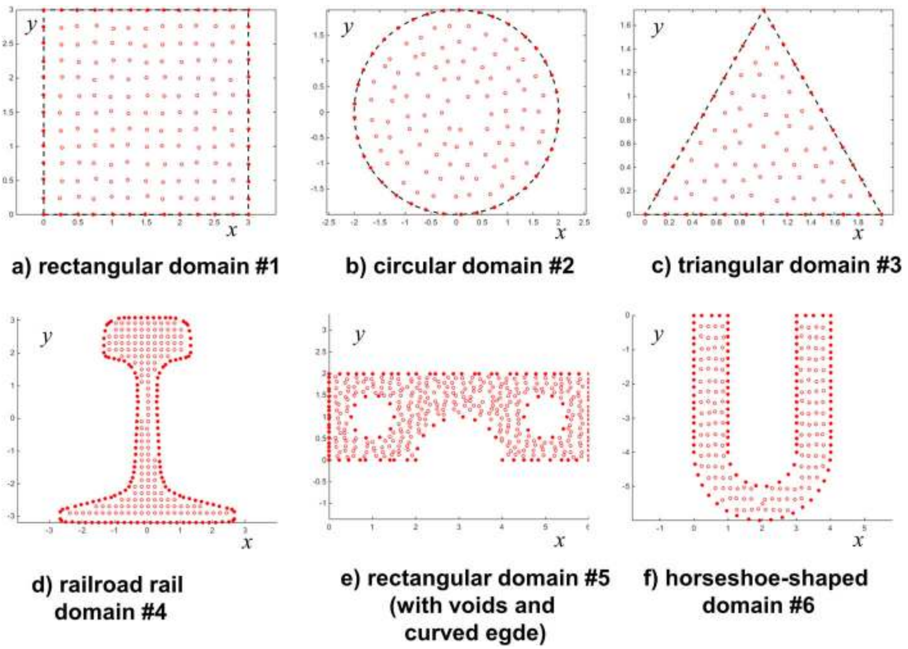
4 A Matlab software-general information

All original author's concepts and ideas, briefly presented in previous section, were successfully implemented in Matlab. The software is available from Netlib (http://www.netlib.org/numeralgo/) as the na52 package. It has been prepared and tested in Matlab R2014b; however, it does not contain any unusual functions and syntax that might cause any problems while using other Matlab versions. A user is supplied with one archive file (*meshlessMCRW_2D.zip*) which contains several script and function files. It has to be downloaded and extracted to the destination folder. The following files are included:

- *MAIN.m* file - the main script file which should be opened in Matlab editor and run—all input parameters are defined there
- *analyt_sol.m* - the function file, which formulates the true (analytical) solution $F = F(x, y)$, and its derivatives
- *fun_a.m* - the function file, which formulates the material function $a = a(x, y)$,
- *fun_f.m* - the function file, which evaluates the values of the right-hand side function $f = f(x, y)$
- *fun_q.m* - the function file, which evaluates the values of the flux function $\bar{q} = \bar{q}(x, y)$
- *mesh_generation.m* - the function file, which is responsible for generation of regular / irregular (randomly distorted) cloud of nodes, for three basic domain types, or it loads an appropriate disc text file, with mesh data
- *star_select.m* - the function file, which selects the nodes into FD star at any arbitrary point, by means of two available criteria: distance and cross ones,
- *MWLS_appro.m* - the function file, which evaluates the difference coefficients matrix **M** by means of the MWLS technique
- *meshless_FDM_solution.m* - the function file, which solves the elliptic equation by means of the meshless FDM
- *plot_domain_mesh.m* - the script file, which plots the domain contours and/or cloud of nodes

Besides the *MAIN.M* and per chance *fun_a.m* files, there is no need to modify any other files. All input parameters, included in the *MAIN.m* file, constitute the subsequent components of the *DATA* class object. Its all fields (variables) are listed and briefly described below:

- *mode* - algorithm type (1, one selected point with an unknown value, fixed n and N; 2, one selected point with an unknown value, fixed n, convergence analysis with respect to N; 3, full analysis at all nodes, for fixed N)
- *domain_type* - built-in shapes: #1, rectangle; #2, circle; #3, equilateral triangle and shapes saved in disc files (#4, railroad rail; #5, rectangle with two voids and curved edge,#6 - horseshoe-shaped domain), Fig. 4
- *x_lb,y_lb* - coordinates of the left bottom corner of rectangle and triangle
- *a,b* - sides of a rectangle
- *c* - a side of a triangle
- *r* - a radius of a circle



a) rectangular domain #1

b) circular domain #2

c) triangular domain #3

d) railroad rail domain #4

e) rectangular domain #5 (with voids and curved egde)

f) horseshoe-shaped domain #6

Fig. 4 Available domain shapes with exemplary meshless discretizations

- *x_0,y_0* - coordinates of the circle center point
- *n_x,n_y,n_h* - nodes densities for rectangular and triangular meshes (*n_x,n_y*) as well as for a circular mesh (*n_h*)
- *dist_amp* - distortion amplitude for irregular clouds generated from regular meshes ($\in [0, 0.5]$, 0 stands for perfectly regular mesh)
- *N* - fixed number of random walks (for modes = 1,3)
- *N_init* - initial number of random walks for convergence analysis (*mode* = 2)
- *dN* - random walks increment number (for *mode* = 2)
- *adap_N* - number of steps in convergence analysis with respect to the number of random walks (for *mode* = 2)
- *sol_type* - analytical solution type:
  #1: $\bar{F}(x, y) = x + y$ (Laplace equation, heterogeneous essential boundary conditions, exact FD result);
  #2: $\bar{F}(x, y) = x^2 + y^2$, (Poisson equation, heterogeneous essential boundary conditions, exact FD result);
  #3: $\bar{F}(x, y) = \dfrac{x + y}{x^2 + y^2}$, (Laplace equation, heterogeneous essential boundary conditions, discontinuous solution);
  #4: $\bar{F}(x, y) = x(x - a)y(y - b)$, (Poisson equation, homogeneous essential boundary conditions (for rectangle only));
  #5: $\bar{F}(x, y) = x^4 + y^4$, (Poisson equation, heterogeneous boundary conditions, large gradients);

#6: $\bar{F}(x, y) = ab\sin(x\pi/a + y\pi/b)$; (Poisson equation, heterogeneous boundary conditions, smooth gradients);
#7: $\bar{F}(x, y) = \exp(-\alpha(x^2 + (y - \beta x)^2)$, (Poisson equation, heterogeneous boundary conditions, large gradients),
in which a, b are the rectangle sides, while α and β are implicitly set to 5 and $9/10$, respectively.

- *m_int* - number of nodes in the FD stars inside the domain (m_i)
- *m_bound* - number of nodes in the FD stars in the boundary nodes with natural boundary conditions (m_b)
- *ord_bound* - approximation order for boundary approximation schemes (1 or 2)
- *bound_cond* - boundary conditions code; all boundary parts are with essential boundary conditions, if equal 1; if equal 2–selected parts have natural boundary conditions applied: two vertical rectangle sides, the upper half of a circle as well as the horizontal triangle side
- *select_crit* - star selection criterion: 1, distance; 2, cross (Voronoi neighbors criterion is not implemented yet, due to the requirement of the constrained Voronoi partitioning—work is in progress),
- *m_smooth* - number of nodes in the FD stars for smoothing purposes (for *mode* = 3),
- *g_smooth* - value of a smoothing parameter *g* (for *mode* = 3)
- *rw_graph* - graphical tracking of random walk paths (0, off; 1, on) (for *mode* = 1)
- *sol_graphs* - graphs of the final solutions (0, off; 1, on) (for *mode* = 3)
- *MC_sol* - meshless Monte Carlo with random walk analysis (0, off; 1, on)
- *mesh_plot* - domain and mesh plotting (0, off; 1, on)
- *font_weight* - font weight of figures' labels ("normal," "bold")
- *marker_size* - marker size of plots' points
- *font_size* - font size of figures' labels
- *graph_view* - graph view style (2, 2D with colorbars displayed; 3, 3D)

Three main modes are described below in more detailed manner.

4.1 Determination of the approximated solution at the selected point, for the fixed number of random walks (*mode*=1)

The simplest program mode allows for the determination of one unknown function value approximation at the selected point of the domain (or its boundary, assuming the natural conditions are subjected). Coordinates of this point (variables *x0,y0*) may be user-defined or default values may be proceeded. In case a regular mesh is applied (*dist_amp* = 0), this given point may not belong to the mesh itself. In that case, solution is obtained at the mesh node, closest to that point. However, in case of irregular cloud (*dist_amp* $>$ 0), the closest internal node is moved to this point (total number of nodes remains unchanged) or additional boundary node is added (the total number of nodes is increased by 1). The simplified flow chart of this algorithm mode is shown in Fig. 5. Graphical interface produces a mesh/cloud of nodes scheme, with the selected point indicated (Matlab's Fig. 1). Moreover, text results are displayed in Matlab's Command Window. Additionally, if a user sets variable *rw_graph* equal to

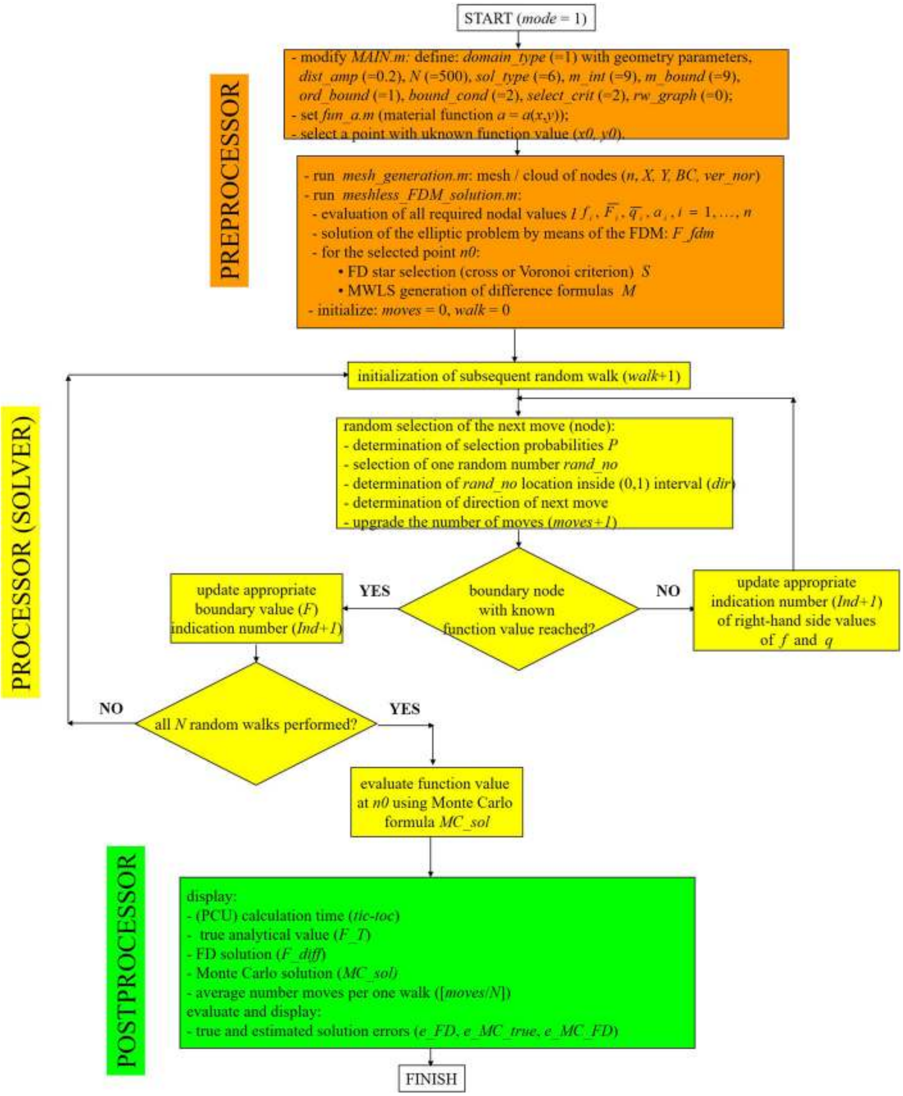

START (mode = 1)
PREPROCESSOR
- modify MAIN.m: define: domain_type (=1) with geometry parameters,
dist_amp (=0.2), N (=500), sol_type (=6), m_int (=9), m_bound (=9),
ord_bound (=1), bound_cond (=2), select_crit (=2), rw_graph (=0);
- set fun_a.m (material function a = a(x,y));
- select a point with uknown function value (x0, y0).
- run mesh_generation.m: mesh / cloud of nodes (n, X, Y, BC, ver_nor)
- run meshless_FDM_solution.m:
- evaluation of all required nodal values $I\ f_i, \overline{F}_i, \overline{q}_i, a_i, i = 1, \ldots, n$
- solution of the elliptic problem by means of the FDM: F_fdm
- for the selected point n0:
• FD star selection (cross or Voronoi criterion) S
• MWLS generation of difference formulas M
- initialize: moves = 0, walk = 0
PROCESSOR (SOLVER)
initialization of subsequent random walk (walk+1)
random selection of the next move (node):
- determination of selection probabilities P
- selection of one random number rand_no
- determination of rand_no location inside (0,1) interval (dir)
- determination of direction of next move
- upgrade the number of moves (moves+1)
boundary node with known function value reached?
YES
update appropriate boundary value (F) indication number (Ind+1)
NO
update appropriate indication number (Ind+1) of right-hand side values of f and q
all N random walks performed?
NO
YES
evaluate function value at n0 using Monte Carlo formula MC_sol
POSTPROCESSOR
display:
- (PCU) calculation time (tic-toc)
- true analytical value (F_T)
- FD solution (F_diff)
- Monte Carlo solution (MC_sol)
- average number moves per one walk ([moves/N])
evaluate and display:
- true and estimated solution errors (e_FD, e_MC_true, e_MC_FD)
FINISH

Fig. 5 The flow chart for *mode* = 1

one, all random walk paths are graphically displayed in Matlab's Fig. 1, in a form of a simple animation.

4.2 Determination of the approximated solution at the selected point, for the series of random walks' numbers (*mode*=2)

The second mode may be treated as a generalization of the first one. Instead of one fixed number of random walks (N), a user defines an initial number of random walks (variable *N_init*), increment (variable *dN*) as well as number of steps (variable *adap_N*). Compared to the flow chart for the first mode, one additional loop appears, which selects the subsequent numbers of random walks out of the series N_init, N_init $+ dN$, N_init $+ 2dN$, ..., N_init $+$ adap_$N \cdot dN$. However, this slightly modified flow chart has been omitted here. As the final result, the convergence graph is generated (Matlab's Fig. 2), in which three types of solution error are calculated for each number of random walks, namely

- The Monte Carlo exact solution error, $|F_{\mathrm{MC}} - \bar{F}|$, which varies for each number of random walks
- The Monte Carlo estimated solution error, in which the exact solution $\bar{F}$ is replaced by the FDM solution F_{FDM}, namely $|F_{\mathrm{MC}} - \bar{F}| \approx |F_{\mathrm{MC}} - F_{\mathrm{FDM}}|$, and which varies for each number of random walks
- The FDM solution error, $|F_{\mathrm{FDM}} - \bar{F}|$, which is fixed for the entire convergence analysis

Moreover, both reference solutions, $\bar{F}$ and F_{FDM}, are fixed, since the deterministic methods do not rely on the number of random walks. Additionally, the theoretical convergence curve is plotted for the sake of comparison, according to (6).

4.3 Determination of the approximated solution at all nodes, for the fixed number of random walks (*mode*=3)

Though the proposed approach is tailored to estimate the solution of elliptic equations at selected points, it would be interesting to apply the final formula (20) to all nodes with unknown function value, located inside the domain and on its boundary part with natural boundary conditions. In that case, one has to deal with different levels of the rough Monte Carlo solution accuracy, which may change from node to node, due to stochastic error, even though the same number of random walks is used each time. Therefore, the appropriate smoothing may be required to keep the reasonable balance in the solution error. It may be performed by means of the MWLS approximation (11) as well. However, this time the larger number of nodes in FD stars as well as non-singular weight functions [31] should be used, with built-in smoothing parameter g as follows:

$$\omega_j^2 = \left(h_{x(j)}^2 + h_{y(j)}^2 + \frac{g^4}{h_{x(j)}^2 + h_{y(j)}^2 + g^2 + \epsilon} \right)^{-3} \tag{22}$$

Parameter g is a positive scalar number, selected arbitrarily, by observing the smoothing effect or adaptively, in order to minimize the local curvature of the approximating function. Setting $g = 0$ in (22) leads to singular weights (14) (no smoothing applied). It should be stressed here, that this additional a-posteriori MWLS approximation may be used for recovery of nodal values of the solution derivatives F_x and F_y as well. The complete flow chart of this algorithm mode is presented in Fig. 6. In addition to the standard text results displayed in Matlab's Command Window, a user is supplied with graphs of three solutions, namely true solution, FD solution, and Monte Carlo

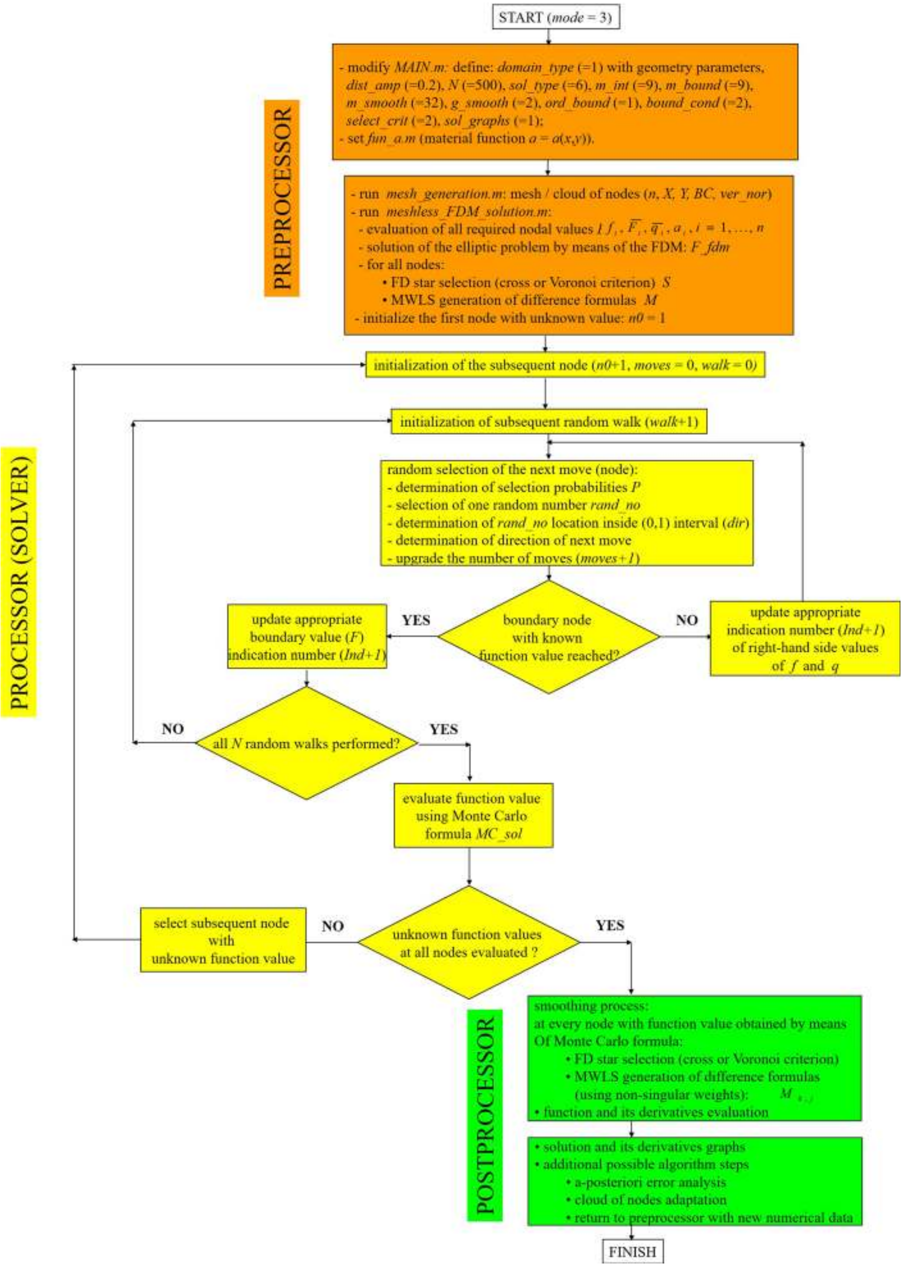

START (mode = 3)
PREPROCESSOR
- modify MAIN.m: define: domain_type (=1) with geometry parameters, dist_amp (=0.2), N (=500), sol_type (=6), m_int (=9), m_bound (=9), m_smooth (=32), g_smooth (=2), ord_bound (=1), bound_cond (=2), select_crit (=2), sol_graphs (=1);
- set fun_a.m (material function a = a(x,y)).
- run mesh_generation.m: mesh / cloud of nodes (n, X, Y, BC, ver_nor)
- run meshless_FDM_solution.m:
- evaluation of all required nodal values f_i, $\overline{F}_i$, $\overline{q}_i$, a_i, $i = 1, \ldots, n$
- solution of the elliptic problem by means of the FDM: F_fdm
- for all nodes:
• FD star selection (cross or Voronoi criterion) S
• MWLS generation of difference formulas M
- initialize the first node with unknown value: n0 = 1
PROCESSOR (SOLVER)
initialization of the subsequent node (n0+1, moves = 0, walk = 0)
initialization of subsequent random walk (walk+1)
random selection of the next move (node):
- determination of selection probabilities P
- selection of one random number rand_no
- determination of rand_no location inside (0,1) interval (dir)
- determination of direction of next move
- upgrade the number of moves (moves+1)
boundary node with known function value reached?
YES
update appropriate boundary value (F) indication number (Ind+1)
NO
update appropriate indication number (Ind+1) of right-hand side values of f and q
all N random walks performed?
NO
YES
evaluate function value using Monte Carlo formula MC_sol
unknown function values at all nodes evaluated ?
NO
select subsequent node with unknown function value
YES
POSTPROCESSOR
smoothing process:
at every node with function value obtained by means Of Monte Carlo formula:
• FD star selection (cross or Voronoi criterion)
• MWLS generation of difference formulas (using non-singular weights): $M_{s,i}$
• function and its derivatives evaluation
• solution and its derivatives graphs
• additional possible algorithm steps
• a-posteriori error analysis
• cloud of nodes adaptation
• return to preprocessor with new numerical data
FINISH

Fig. 6 The flow chart for *mode* = 3

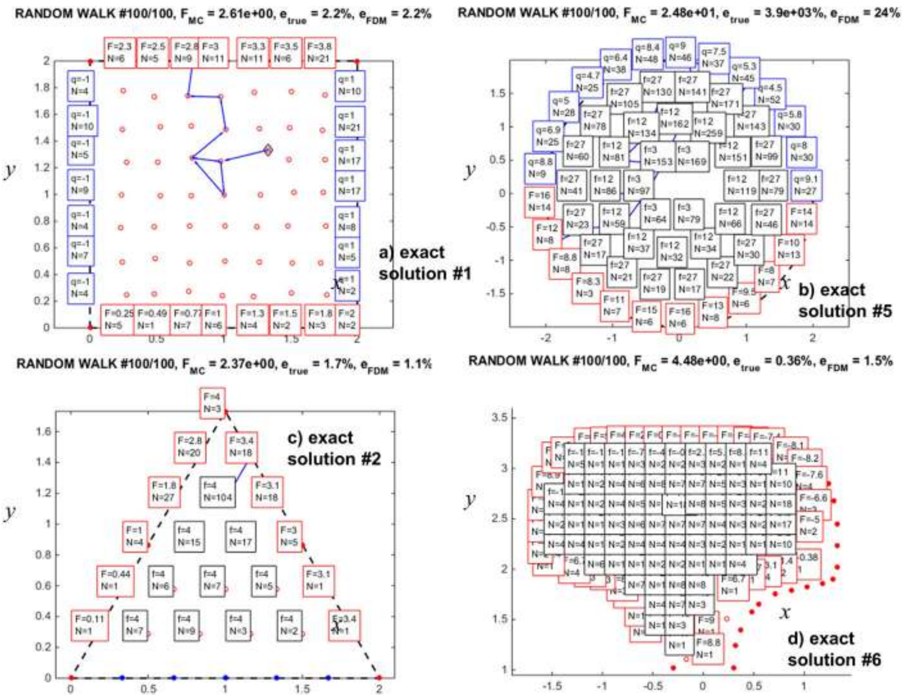
solution as well as the differences of the above (Matlab's Fig. 3). Moreover, comparison of solutions' first derivatives is presented in Matlab's Fig. 4. All graphs are plotted using the additional background plotting mesh of Delaunay triangles [16, 31].

5 Numerical experiments and software testing

A variety of tests were executed in order to examine both the proposed approach and the attached Matlab software. Selected results are presented and briefly discussed.

Illustrations of *mode* = 1 with *rw_graph* set to 1 are shown in Fig. 7, for various domain types with mixed boundary conditions as well as for various analytical solutions. $N = 100$ random walks were applied, on very coarse irregular clouds, with small numbers of nodes. All other input parameters have their default values. The following points with unknown solution were examined, namely: (1.3333, 1.3333) (Fig. 7a), (0.5, 0.86603), (Fig. 7b), (1, 1.1547) (Fig. 7c), and (0, 2.5) (Fig. 7d). The final distribution of all non-zero indication numbers ($N_k^{(e)}$, $N_l^{(n)}$, and $N_i^{(i)}$) as well as non-zero nodal values ($\bar{F}_k$, $\bar{q}_l$, and f_i), assigned to appropriate nodes (thus giving the non-zero components to the final MC formula), is shown in subsequent graphs. It may be observed how many times the particular node has been reached during the random walk process. Moreover, the exemplary random walk path is plotted (Fig. 7a). The final rough solution estimation as well as its true and numerical errors are presented in the graphs' labels.

The following tests concern the solution convergence with respect to the number of random walks (*mode* = 2). The series of 50 increasing numbers of walks, starting from $N = 50$ and with increment $dN = 200$, were performed. The same domain and


RANDOM WALK #100/100, F_{MC} = 2.61e+00, e_{true} = 2.2%, e_{FDM} = 2.2%

RANDOM WALK #100/100, F_{MC} = 2.48e+01, e_{true} = 3.9e+03%, e_{FDM} = 24%

RANDOM WALK #100/100, F_{MC} = 2.37e+00, e_{true} = 1.7%, e_{FDM} = 1.1%

RANDOM WALK #100/100, F_{MC} = 4.48e+00, e_{true} = 0.36%, e_{FDM} = 1.5%

Fig. 7 Results of random walks for *mode* = 1

solution types were applied as for *mode* = 1. Results (solution errors) are presented in Fig. 8. The MC solution error is perfectly upper-bounded by its a-priori estimation, corresponding to (6), for the simplest case only (Fig. 8a), whereas it is no longer valid for other cases. Nevertheless, in each case, very good agreement between the FD and MC/RW solutions may be observed, even though there may be enormous true errors (Fig. 8b), caused by a very coarse discretization, when compared to the exact solution.

Eventually, the full analysis of the elliptic (7) was carried out. Non-constant material function $a(x, y) = \dfrac{1}{1 + \cos^2\left((x + y)\dfrac{\pi}{3}\right)}$ was set. The meshless Monte Carlo random walk stochastic formula (20) was applied to every unknown function value at all nodes located inside the domain and on its boundary part with natural boundary conditions (if applicable). Additional a-posteriori solution smoothing was performed, according to (22). Graphs of the final solution, the solution errors as well as its first derivatives are presented in Figs. 9 and 10 (circular domain, with $n = 128$ nodes irregularly scattered, with $N = 500$ and solution #2). For this solution type, FD analysis produces the exact solution, within the second polynomial order assumed, up to the machine precision. However, the Monte Carlo solution approach yields a rough solution estimation only, as it is highly influenced by a stochastic error. Solution and

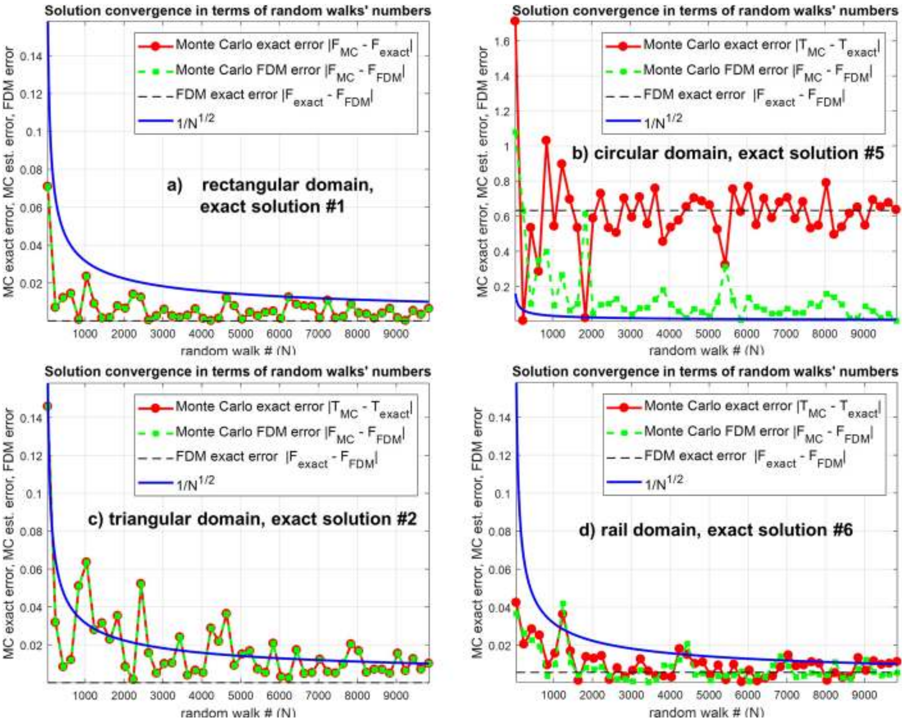


Fig. 8 Results of solution convergence for *mode* = 2

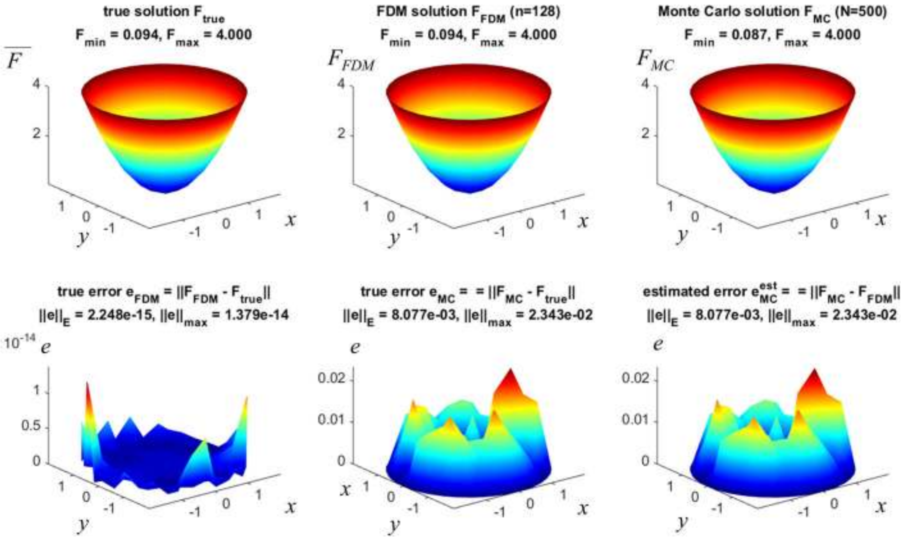


Fig. 9 Full analysis of elliptic equation—comparison of solutions and solution errors (circular domain #2, solution #2—*mode* = 3)

its derivatives comparisons for the additional domain types (#5 and #6), with data available from attached disc files, are presented in Figs. 11 and 12 (domain #5, rectangle with two voids and curved edge, with $n = 442$ nodes irregularly scattered, with $N = 1000$ and solution #6) as well as in Figs. 13 and 14 (domain #6, of a horseshoe shape, with $n = 263$ nodes irregularly scattered, with $N = 1000$ and solution

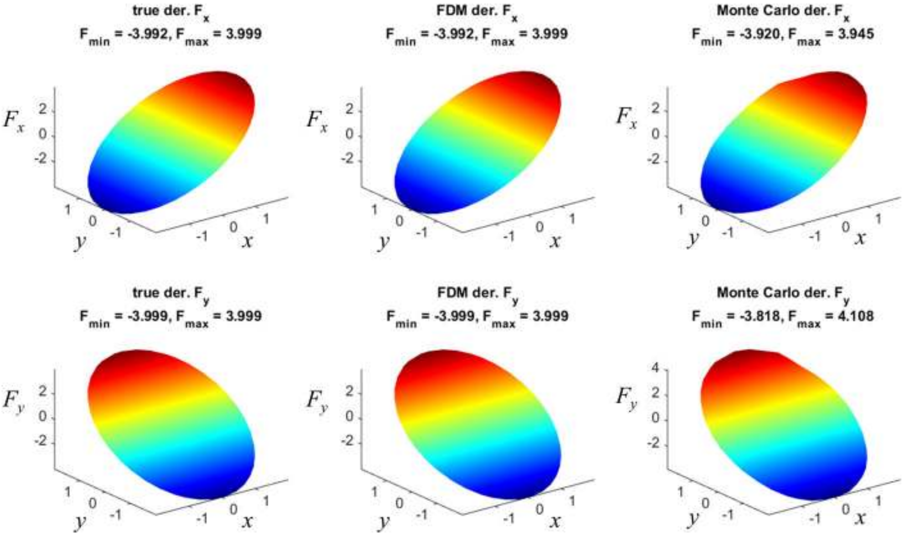


Fig. 10 Full analysis of elliptic equation - comparison of the solution's first derivatives (circular domain #2, solution #2—*mode* = 3

#6). All corresponding graphs show very good agreement between the exact solution, the FDM solution, as well as the meshless MC solution. In fact, the following corollary may be formulated: the MC solution is the stochastic approximation of the FDM solution, whereas the FDM solution is the deterministic approximation of the exact solution.

Clouds of nodes generated for the first three domain types (#1, #2, and #3) may be refined in an arbitrary manner by means of the mesh generator, being a part of the presented software (the function file *mesh_generation.m*). However, the last three, more complex domain types have fixed discretization, loaded from the external disc files, as the generation of discretization for more complex shapes would require more sophisticated software. Furthermore, it should be mentioned that the presented Matlab software may be extended by a user in many potential directions, without any difficulties. The modification of the material function may be done in *fun_a.m* function file. New exact solution requires adding new conditional branch (formulae for solution and its derivatives in explicit manner) in *analyt_sol.m*. In case the analytical solution is not known, appropriate boundary conditions as well as right-hand side functions may be formulated in *analyt_sol.m*, *fun_f.m* and *fun_q.m*. A brand new domain shape may be defined in *mesh_generation.m* or another text file with mesh data should be prepared (number of nodes, nodes coordinates and boundary codes (0, internal node; 1, boundary node with essential boundary conditions; 2, boundary node with essential boundary conditions) as well as the number of Delaunay triangles and a list of all triangles' vertices—for potential graphical purposes only).

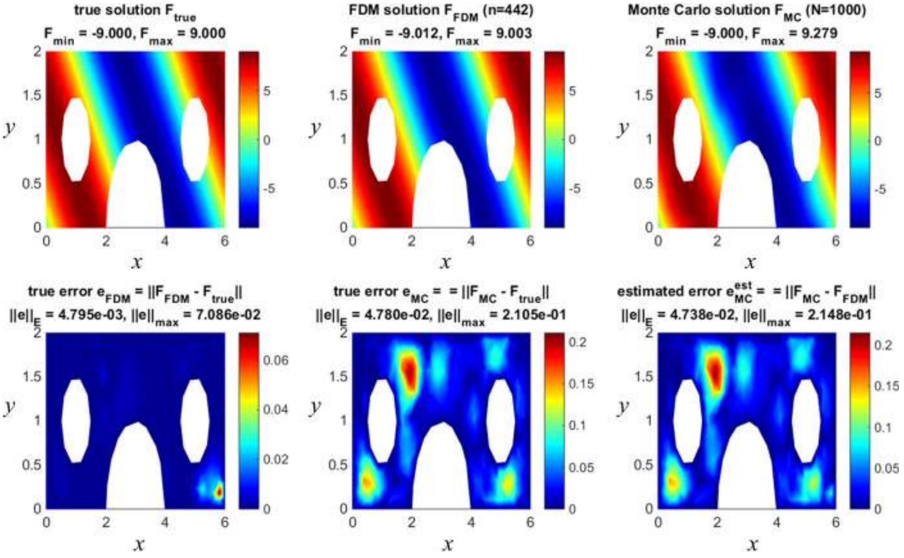


Fig. 11 Full analysis of elliptic equation - comparison of solutions and solution errors (rectangle with two voids and curved edge—domain #5, solution #6—*mode* = 3

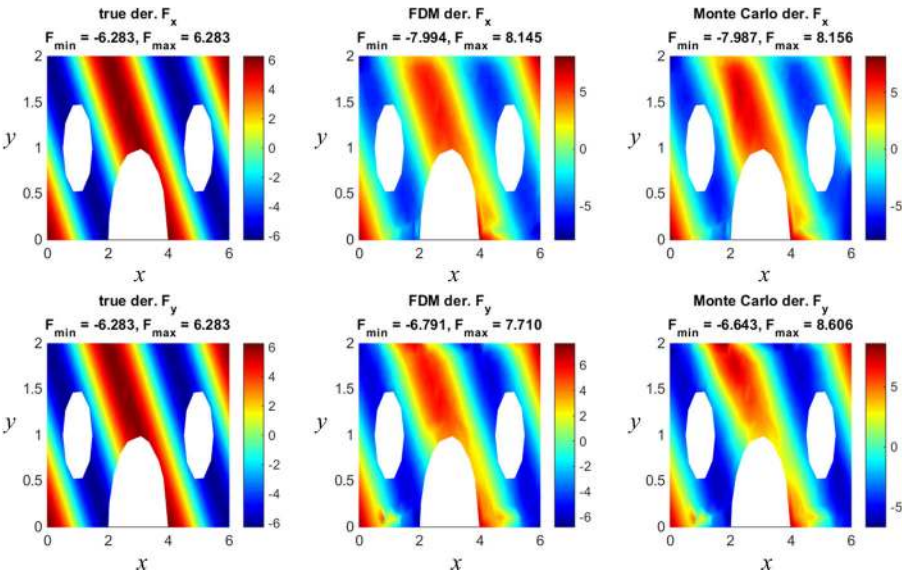


Fig. 12 Full analysis of elliptic equation—comparison of the solution first derivatives (rectangle with two voids and curved edge—domain #5, solution #6—*mode* = 3

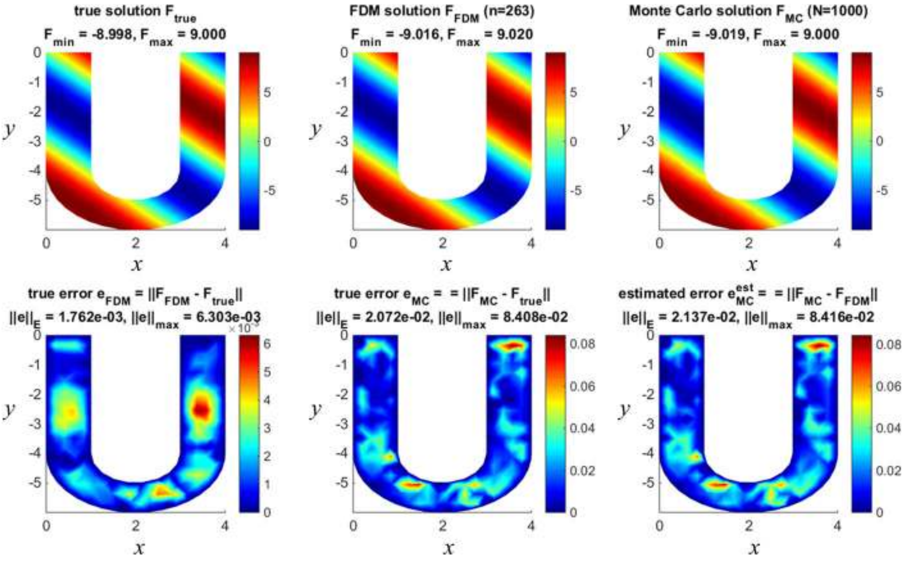


Fig. 13 Full analysis of elliptic equation—comparison of solutions and solution errors (horseshoe-shaped domain #6, solution #6—*mode* = 3

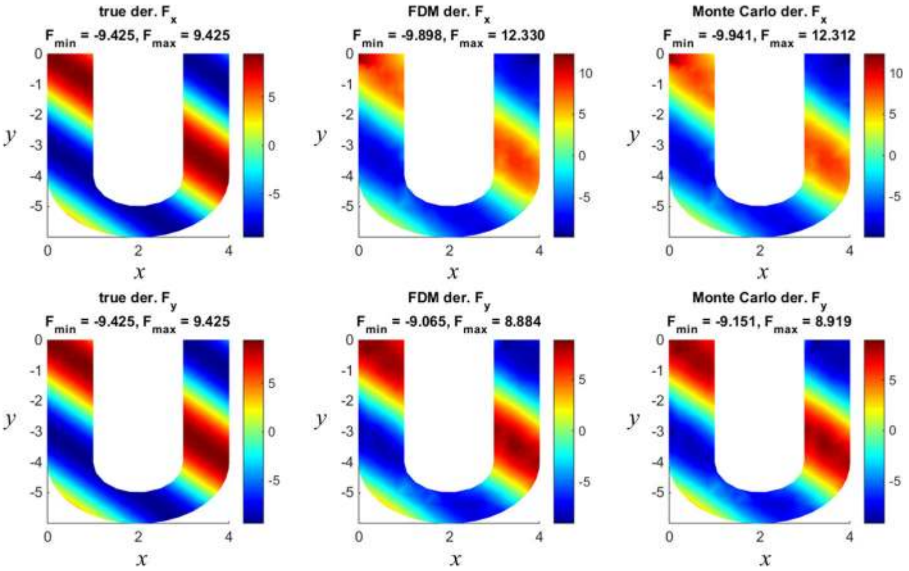


Fig. 14 Full analysis of elliptic equation—comparison of the solution first derivatives (horseshoe-shaped domain #6, solution #6—*mode* = 3

6 Final remarks

This research focuses on the modified, meshless Monte Carlo method with random walk (MC/RW) solution approach. It is based on the appropriate discretization and approximation techniques, typical for meshless modeling of boundary value problems. When compared to the standard MC/RW method, it allows for the analysis of wider class of elliptic problems, with non-homogeneous right-hand side functions, variable material coefficients as well as mixed (essential and natural) boundary conditions. Moreover, arbitrarily irregular meshes and unstructured clouds of nodes may be applied. Therefore, analysis of more complex domain shapes is possible.

The proposed approach is based upon the random generation of indications net (random walk technique) starting from the fixed point (node), at which solution estimation is required. Selection probabilities of subsequent walk directions depend on the meshless nodes configuration (equivalent to the FD star) as well as the difference coefficients corresponding to differential operators, obtained by means of the moving weighted least squares approximation. Finally, the Monte Carlo concept is incorporated, according to which the random walk terminates at boundary node located on boundary part with essential boundary conditions (with known solution). Therefore, fast and reasonably accurate solution estimation at node is possible. Moreover, the generation of a large system of equations, combining all unknown function values may be omitted. Full analysis (convenient for potential parallelization) of all internal and boundary nodes with unknown solution has been discussed. Additional post-processing includes the recovering of solution's derivatives as well as the alternative smoothing of nodal results, which leads to significant reduction of a stochastic error.

All those original author's concepts have been implemented in the attached Matlab software. It may work in three various modes, namely 1. determination of an approximated solution at specified node, 2. solution convergence analysis on the set of random walks' numbers, 3. determination of an approximated solution at all nodes. Graphical interface allows for visualization of the random walk structure and indications' numbers (*mode*=1), convergence graph (*mode*=2) as well as graphs of the true, FD and Monte Carlo solutions, corresponding errors and solution's derivatives (*mode*=3). Six types of domain geometries as well as seven types of the analytical solutions are included. Algorithm's flow charts and the selected results of software usage are presented.

Further research includes, for instance, application of the meshless Monte Carlo random walk solution approach to analysis of the 3D elliptic equations as well as non-linear and non-stationary thermo-mechanical problems. In case the reference problem has to be solved multiple times (e.g., within an incremental-iterative procedure or an implicit time integration scheme), series of random walks forming indications net, are performed only once. Afterwards, the meshless Monte Carlo formula (20) is applied to each solution increment, allowing for the significant reduction of the computational effort.

Funding information This research was supported by the National Science Centre, Poland, under the scientific project 2015/19/D/ST8/00816 (Computational coupled FEM / meshless FDM analysis dedicated to engineering non-stationary thermo-elastic and thermoplastic problems).

References

1. Metropolis, N., Ulam, S.: The Monte Carlo method. J. Am. Stat. Assoc. American Statistical Association **44**(247), 335–341 (1949)
2. Forsythe, G.E., Leibler, R.A.: Matrix inversion by a Monte Carlo method. Math. Tab. Aids Comput. **4**, 127–129 (1950)
3. Donsker, M.D., Kac, M.: A sampling method for determining the lowest eigenvalue and the principle eigenfunction of Schrödinger's equation. J. Res. Natl. Bur. Stand. **44**, 551–557 (1951)
4. Curtiss, J.H.: Monte Carlo methods for the iteration of linear operators. J. Math. Phys. **32**, 209–232 (1953)
5. Curtiss, J.H.: A theoretical comparison of the efficiencies of two classical methods and a Monte Carlo method for computing one component of the solution of a set of linear algebraic equations. In: Meyer, H.A. (ed.) Symp. Monte Carlo Methods, pp. 191–233. Wiley, New York (1956)
6. Muller, M.E.: Some continuous Monte Carlo methods for the Dirichlet problem. Ann. Math. Stat **27**, 569–589 (1956)
7. Reynolds, J.F.: A proof of the random-walk method for solving Laplace's equation in 2-D. Math. Gaz. **49**(370), 416–420 (1965). Mathematical Association
8. Haji-Sheikh, A., Sparrow, E.M.: The floating random walk and its application to Monte Carlo solutions of heat equations. J. SIAM, Appl. Math. **14**(2), 370–389 (1966)
9. Hoshino, S., Ichida, K.: Solution of partial differential equations by a modified random walk. Numer. Math **18**, 61–72 (1971). Springer
10. Perrone, N., Kao, R.: A general finite difference method for arbitrary meshes. Comput. Struct. **5**, 45–58 (1975)

11. Wyatt, M.J., Davies, G., Snell, C.: A new difference based finite element method. Instn. Eng. **59**(2), 395–409 (1975)
12. Brandt, A.: Multi-level adaptive solutions to boundary value problems. Math. Comp. **31**, 333–390 (1977)
13. Niederreiter, H.: Quasi-Monte Carlo methods and pseudo-random numbers. Bull. Amer. Math. Soc. **84**, 957–1041 (1978)
14. Baclawski, K., Donsker, M.D., Kac, M.: Probability, Number Theory, and Statistical Physics, Selected Papers, pp. 268–280. The MIT Press, Cambridge (1979)
15. Sipin, A.S.: Solving first boundary value problem for elliptic equations by Monte Carlo method. Monte Carlo Methods in Computational Mathematics and Mathematical Physics **2**, 113–119 (1979)
16. Liszka, T., Orkisz, J.: The finite difference method at arbitrary irregular grids and its application in applied mechanics. Comput. Struct. **11**, 83–95 (1980)
17. Booth, T.E.: Exact Monte Carlo solutions of elliptic partial differential equations. J. Comput. Phys. **39**, 396–404 (1981)
18. Lancaster, P., Salkauskas, K.: Surfaces generated by moving least-squares method. Math. Comput. **155**(37), 141–158 (1981)
19. Anderson, D.A., Tannenhill, J., Fletcher, R.H.: Computational Fluid Mechanics and Heat Transfer. McGraw-Hill, Washington (1984)
20. Ghoniem, A.F., Sherman, F.S.: Grid-free simulation of diffusion using random walk methods. J. Comp. Phys. **61**, 1–37 (1985)
21. Eckhardt, R., Ulam, S., von Neumann, J.: The Monte Carlo method Los Alamos Science, Special Issue (15) , pp. 131–137 (1987)
22. Atkinson, K.E.: An introduction to numerical analysis. Wiley Ed., New York (1988)
23. Ermakov, S.M., Nekrutkin, V.V., Sipin, A.S.: Random Processes for Classical Equations of Mathematical Physics. Kluwer Academic Publishers, Dordrecht (1989)
24. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Professional, Reading (1989)
25. Lancaster, P., Salkauskas, K.: Curve and Surface Fitting. Academic Press Inc, Cambridge (1990)
26. Mascagni, M.: High dimensional numerical integration and massively parallel computing. Contemp. Math. **1**, 115:53–73 (1991)
27. Sabelfeld, K.K.: Monte Carlo Methods in Boundary Value Problems. Springer, Berlin (1991)
28. Dimov, I., Tonev, O.: Random walk on distant mesh points Monte Carlo methods. J. Stat. Phys. **70**. Plenum Publishing Corporation (1993)
29. Liu, W.K., Jun, S., Zhang, Y.F.: Reproducing kernel particle methods. Int. J. Numer. Methods Eng. **20**, 1081–1106 (1995)
30. Mikhailov, G.A.: New Monte Carlo Methods with Estimating Derivatives. V. S. P. Publishers, London (1995)
31. Orkisz, J.: Finite difference method (part III), Handbook of Computational Solid Mechanics, pp. 336–431. Springer, Berlin (1998)
32. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes: the Art of Parallel Scientific Computing. Cambridge University Press, Cambridge (1999)
33. Li, S., Liu, W.K.: Meshfree and particle methods and their applications. Appl. Mech. Rev. **55**, 1–34 (2002)
34. Hwang, C.-O., Mascagni, M., Given, J.A.: A Feynman-Kac path-integral implementation for Poisson's equation using an h-conditioned Green's function. Math. Comput. Simul. **62**(3-6), 347–355 (2003)
35. Liu, G.R.: Mesh Free Methods: Moving Beyond the Finite Element Method. CRC Press, Boca Raton (2003)
36. Moller, B., Beer, M.: Fuzzy Randomness, Uncertainty in Civil Engineering and Computational Mechanics. Springer, Berlin (2004)
37. Zienkiewicz, O.C., Taylor, R.L.: Finite Element Method its Basis and Fundamentals. Elsevier, Amsterdam (2005)
38. Ramachandran, P., Ramakrishna, M., Rajan, S.C.: Efficient random walks in the presence of complex two-dimensional geometries. Computers and Mathematics with Applications **53**(2), 329–344 (2007)
39. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (2007)

40. Orkisz, J., Milewski, S.: A'posteriori error estimation based on higher order approximation in the meshless finite difference method, Lecture Notes in Computational Science and Engineering: Meshfree Methods for Partial Differential Equations IV, pp. 189–213 (2008)
41. Sabelfeld, K., Mozartova, N.: Sparsified randomization algorithms for large systems of linear equations and a new version of the random walk on boundary method. Monte Carlo Methods Appl. **15**(3), 257–284 (2009)
42. Sadiku, M.N.: Monte Carlo Methods for Electromagnetics. Taylor (2009)
43. Milstein, G., Tretyakov, M.V.: Stochastic Numerics for Mathematical Physics. Springer, Berlin (2010)
44. Jurlewicz, A., Kern, P., Meerschaert, M.M., Scheffler, H.-P.: Fractional governing equations for coupled random walks. Computers and Mathematics with Applications **64**(10), 3021–3036 (2012)
45. Milewski, S.: Meshless finite difference method with higher order approximation - applications in mechanics. Arch. Comput. Meth. Eng. **19**(1), 1–49 (2012)
46. Milewski, S.: Selected computational aspects of the meshless finite difference method. Numer. Algorithms **63**(1), 107–126 (2013)
47. Yu, W., Wang, X.: Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits. Springer, Berlin (2014)
48. Jaśkowiec, J., Milewski, S.: The effective interface approach for coupling of the FE and meshless FD methods and applying essential boundary conditions. Computers and Mathematics with Applications **70**(5), 962–979 (2015)
49. Bernal, F., Acebron, J.A.: A comparison of higher-order weak numerical schemes for stopped stochastic differential equations. Communications in Computational Physics **20**, 703–732 (2016)
50. Angstmann, C.N., Henry, B.I., Ortega-Piwonka, I.: Generalized master equations and fractional Fokker-Planck equations from continuous time random walks with arbitrary initial conditions. Computers and Mathematics with Applications **73**(6), 1315–1324 (2017)
51. Keady, K.P., Cleveland, M.A.: An improved random walk algorithm for the implicit Monte Carlo method. J. Comput. Phys. **328**, 160–176 (2017)
52. Talebi, S., Gharehbash, K., Jalali, H.R.: Study on random walk and its application to solution of heat conduction equation by Monte Carlo method. Prog. Nucl. Energy **96**, 18–35 (2017)
53. Milewski, S.: Combination of the meshless finite difference approach with the Monte Carlo random walk technique for solution of elliptic problems. Computers & Mathematics with Applications **76**(4), 854–876 (2018)