Numer Algor (2013) 63:107–126
DOI 10.1007/s11075-012-9614-6

ORIGINAL PAPER

Selected computational aspects of the meshless finite difference method

Sławomir Milewski

Received: 4 December 2010 / Accepted: 20 June 2012 /
Published online: 11 July 2012

Abstract Meshless Finite Difference Method (MFDM) is nowadays a powerful engineering tool for numerical analysis of boundary value problems. Nowadays, its computational capabilities are not fully used mainly due to the lack of suitable commercial software. This paper briefly presents current state-of-the-art of the MFDM solution approach as well as deals with the selected computational aspects of the MFDM. A set of Matlab functions written by the author is attached to the paper. Techniques for generation of nodes, MFD stars, formulas, equations as well as local approximation technique and numerical integration schemes are discussed there.

Keywords Meshless methods · Meshless finite difference method · MWLS approximation · Implementation in Matlab

1 Introduction

Nowadays, world of computational mechanics is dominated by the Finite Element Method. FEM constitutes general, effective and fully mature solver for numerical analysis of the engineering and mechanical boundary value problems. However, it is not the only solution approach allowing for such numerical analysis. One may mention the Boundary Element Method (BEM), the Finite Difference Method (FDM) as well as a wide group of the meshless methods

Electronic supplementary material The online version of this article (doi:10.1007/s11075-012-9614-6) contains supplementary material, which is available to authorized users.

S. Milewski (✉)
Cracow University of Technology, Warszawska st. 24, 31-155 Cracow, Poland
e-mail: slawek@L5.pk.edu.pl

(MM), which are under extensive development. All these methods posses some advantages, which make them competitive to the FEM in some applications. For instance, in the MM, no nodes connectivity is required. In some special cases, numerical analysis performed by those methods can be even more effective than in the case of the FEM. Hereby, constant development of existing computational methods as well as search for new ones are justified and are recently conducted worldwide. Here, specially investigated is the Meshless Finite Difference Method (MFDM, [16, 18]), which belongs to the group of those meshless methods [2, 3, 18, 33], in which the local approximation is built by means of the Moving Weighted Least Squares (MWLS, [13, 16, 18]). In fact, the MFDM is one of the oldest meshless approach, and therefore possibly the most developed one. Its bases reach pre-computer times (FDM), although worldwide attention devoted to the FEM caused stagnation in generalization and automatization of the FDM. Concept of the FDM generalization for the arbitrarily irregular sets of nodes (the Generalized FDM, now called the Meshless FDM) emerged in Poland in the late 70ties [14, 15]. Afterwards, the rapid development of the methods called meshless (or meshfree) started in the early 90ties in other parts of the world. Following E. Onate et al., one may classify meshless methods as those in which the approximation can be constructed entirely in terms of nodes rather than elements.

Various MM are nowadays widely applied in many aspects of mechanics, however still not enough attention is devoted to them in the computational mechanic science. Probably, the main drawback is the lack of the commercial software, which may discourage potential users from the meshless approach. In most cases, the only possible solution is to construct own solvers. However, the amount of necessary work may be not as large as it seems, since many of the already implemented solution steps may be transformed from the FEM code, for instance. Therefore, the main goal of this paper is to encourage scientists who are familiar only with FE analysis, to prepare their own program codes for the alternative MFDM solution approach on the basis of those discussed here. All given descriptions and codes are part of the original author's contribution to the MFDM analysis. They may be useful in itself at the very first stage of investigation as well as they may be treated as the interesting starting point for their further development and extension towards more sophisticated applications. The software is available from Netlib (http://www.netlib.org/numeralgo/) as the na36 package.

2 Finite difference method

The Finite Difference Method FDM [16, 18] is one of the oldest numerical methods of analysis of the boundary value and initial value problems, used long time before the computer age. The classical FDM is a very effective tool for analysis of the boundary value problems posed in regular shape domains. Especially convenient is then generation of the mesh, FD stars, formulas and equations. Moreover, for regular meshes there are many mathematical proofs regarding the stability and convergence of the method as well as the existence, and uniqueness of the solution. However, its power and scope of applications were practically limited to the regular meshes and regular shaped domains. Lack of possibility of local mesh refinement is another drawback of the classical FDM. Moreover, its full automation was very difficult to perform. Needed was generalisation of the FDM at least for the arbitrarily irregular meshes (clouds of nodes), and for the domains with arbitrary shapes.

3 Meshless finite difference method

Though idea of irregular meshes is not new, a possibility of practical calculation was dependent on computer technique development. Evolution of irregular meshes starts from the mesh being partially regular in sub-domains, then irregular, but with restricted topology, which allows for mapping onto the regular one to arbitrarily irregular cloud of nodes. It is worth mentioning that one of the earlier methods without a mesh is smoothed particle hydrodynamics, presented in 1977 [7]. This approach constituted the basis for a more general method known as the Reproducing Kernel Particle Method (RPKM). Completely independent concept was based upon the Moving Least Squares shape function. The MLS shape function has been successfully used in both the strong (avoiding the background grid) and a weak form (Galerkin) with a background grid for the integration domain by Orkisz and Liszka et al. (Generalized MFD [15], called Meshless MFD nowadays [18]), Nayroles et al. [17] and by Belytschko et al. [3].

The basis of the MFDM was published in the early seventies. Fully arbitrary mesh, though for local formulation and the interpolation schemes only, was firstly considered by Jensen [9]. The main disadvantage of his approach was frequent singularity or ill-conditioning of a control scheme. Several authors tried to develop an automatic procedure which avoids incorrect stars and thus improving the accuracy of the FD formulas. Perrone and Kao proposed using of additional nodes in the FD stars, selected from the geometrical criterion. The approach for FD analysis of boundary value problems posed in the variational form were considered first by R.A. Nay and S. Utku. Those early formulations of the so called Generalised FMD were later extended and improved by many other researchers. The most interesting works were published by M.J. Wyatt, G. Davies, C. Snell, P. Mullord, D.G. Vesey and much later by B. Nayroles, G. Touzot and P. Villon. It is worth mentioning here a contribution of the polish authors, Z. Kczkowski, R. Tribio, M. Syczewski and J. Cendrowicz, in the early stage of this research.

The initial concept of P.S. Jensen, which was based on considering the irregular meshes, though for local formulation and the interpolation schemes only, was mainly developed throughout last thirty years by J. Orkisz and his numerous co-workers (T. Liszka [14, 15], W. Tworzydo, J. Krok [11, 12], W. Cecot, W. Karmowski [10], J. Magiera, M. Pazdanowski, I. Jaworska [20, 21], S. Milewski [23–30]). The most complete and general version of the Generalised FDM (GFDM) (now called Meshless FDM), based on the arbitrary cloud of nodes (totally irregular meshes) and MWLS approximation appeared in the late seventies. Since then it has been widely extended towards non-linear problems, combinations with order discrete methods, differential manifold, higher order approximation as well as multigrid adaptive approach. Several general presentations of the MFDM were made in the last years, including [16, 18, 29, 30]. The MFDM falls into the category of the meshless methods (MM, [2–4, 18, 33]), introduced in the early 90ties. The characteristic feature of all MM is formulated by Idehlson and Belytschko [3], "*meshless are these methods, in which the local approximation of the unknown function is built only in terms of nodes*". Thus MM use unstructured clouds of nodes, that may be distributed totally arbitrarily, without any structure imposed a-priori, like domain division into elements or mesh regularity, or any mapping restrictions. This feature allows for inserting, shifting and deleting nodes in much easier manner than in element-based approaches. In such context, the MFDM presents nowadays the oldest (at least since 1972), and therefore, possibly the most developed as well as effective meshless method. The notion "*meshless*" still causes a lot of misunderstanding among involved scientists. According to its definition, any meshless method may use nodes and only nodes for prescribing approximation of the unknown function and its derivatives. However, it does not mean that any type of domain topology should be avoided, e.g. in order to MFD star (stencil) generation, for integration schemes or even visualization purposes. Approaches, which omit usage of nodes topology, may lead to a large waste of useful information. Nevertheless, many scientists search for "*truly meshless*" methods, which are wrongly understood as methods without any a-posteriori determined topology (e.g. triangles as integration subdomains). It should be stressed here that as long as additional topology information does not affect the local function approximation, the method remains meshless.

The basic MFDM solution approach consists of several steps [16, 18, 29], which are listed below

- Formulation of boundary value problems for MFDM analysis,
- Nodes generation and modification
  - Nodes generation
  - Domain partition (Voronoi tessellation and Delaunay triangulation)
  - Domain topology determination
- The optimal MFD star selection and classification
- Local MWLS approximation
- Mesh generation for the numerical integration (for global formulations only)
- Generation of MFD operators
- MFD discretization of boundary conditions
- Generation and solution of MFD equations
- Postprocessing by MWLS

Among many extensions of the basic MFD solution approach, developed in the past and still being under current development, one may mention here

- MFDM oriented node generator [14, 15, 22],
- A-posteriori error analysis [11, 16, 29],
- Mesh refinement and adaptive (multigrid) solution approach, [16, 18, 29],
- MWLS with generalised degrees of freedom [11, 12],
- Higher order approximation [16, 23–31],
- MFDM on the differential manifold,
- MFDM/FEM combinations and unification [12],
- Experimental and numerical data smoothing [10, 18],
- Hybrid experimental/theoretical/numerical approach [10],
- Software development,
- Engineering applications.

Many of those given above MFDM extensions were co-worked by the author of this paper throughout last ten years.

3.1 Boundary value problem formulation

In most cases, the FEM is associated with the weak global formulations, for instance the variational Galerkin principle or minimisation of the energy functional [34, 35]. FE are natural supports for the interpolation functions as well as integration weights. However, in both FEM and MFDM possible are also other boundary value problem formulations [18, 30]. Especially in the MFDM, any formulation which involves unknown function and its derivatives, may be applied. The local formulation, natural for the classical FDM, is given as a set of differential equations with appropriate boundary conditions. In the considered domain $\Omega \subset \Re^n$ with boundary $\partial\Omega$, a function $u(P)$ is sought at each point P, satisfying equations

$$Lu = f \quad for \quad P \in \Omega \tag{1}$$

$$Gu = g \quad for \quad P \in \partial\Omega \tag{2}$$

where L and G are given differential operators, inside the domain and on its boundary respectively, and f, g are known functions of point P.

Global formulations may be posed either in the form of a functional optimisation (mainly for the self-coupled problems) or more general, as variational principles (e.g. the principle of virtual work). In the first case, considered is minimisation of a functional given in the general form

$$I(u) = \frac{1}{2}B(u, u) - L(u) \tag{3}$$

satisfying boundary conditions (2).

In the second case, variational principle in the general Petrov–Galerkin form is considered

$$B(u, v) = L(v) \quad for \quad v \in V_{\mathrm{adm}} \tag{4}$$

where $u = u(P)$ is a searched trial function, and $v = v(P)$ is a test function from the admissible space V_{adm}. When the test and trial functions use the same basis functions for approximation, one has Bubnov–Galerkin approach. Moreover, the variational form (4) may have symmetric (e.g. Bubnov–Galerkin approach) or non-symmetric character, depending on the type of the bilinear form $B(u, v)$. Also global/local formulations may be considered. The whole domain is divided then into a finite number of subdomains Ω_i, usually assigned to each selected point (node) P_i. The original functional minimisation (3) or variational principle (4) is applied rather to those local subdomains than to the whole domain Ω at once. Such concept is a natural base for Meshless Local Petrov–Galerkin (MLPG) formulations [2, 30], recently becoming more and more popular in many applications.

Consider locally formulated boundary value problem: Find such function $u(x) \in C^2(\Omega) : \Re^2 \supset \Omega \to \Re$ that

$$\begin{cases} k\Delta u = -q & in \quad \Omega \\ u = \bar{u} & on \quad \partial\Omega_D \\ k\dfrac{\partial u}{\partial n} = \hat{g} & on \quad \partial\Omega_N \end{cases} \tag{5}$$

One may easily construct three standard variational formulations, with various continuity and integrability requirements imposed on trial (u) and test (v) functions, namely

- nonsymmetric variational form of type 1: Find such function $u \in H_0^2 + \bar{u}$, that for any test function $v \in L^2(H^0)$ satisfied is

$$-\int_\Omega k\Delta u \circ v \,\mathrm{d}\Omega = \int_\Omega vq \,\mathrm{d}\Omega \tag{6}$$

  Here, H denotes Sobolev's space, with given required order of integrability (upper index 0, 1, 2) and requirement for fulfilment of the homogeneous Dirichlet conditions (lower index 0), with the affinity shift ($\bar{u}$) applied in case of non-homogeneous ones.
- symmetric variational form: Find such function $u \in H_0^1 + \bar{u}$, that for any test function $v \in H_0^1$ satisfied is

$$\int_\Omega k\nabla u \circ \nabla v \,\mathrm{d}\Omega = \int_\Omega vq \,\mathrm{d}\Omega + \int_{\partial\Omega_N} v\hat{g} \,\mathrm{d}s \tag{7}$$

- nonsymmetric variational form of type 2: Find such function $u \in H_0^0 + \bar{u}$, that for any test function $v \in H_0^2$ satisfied is

$$-\int_\Omega ku \circ \Delta v \,\mathrm{d}\Omega = \int_\Omega vq \,\mathrm{d}\Omega + \int_{\partial\Omega_N} (v\hat{g} - \frac{\partial v}{\partial n}u) \,\mathrm{d}s \tag{8}$$

  Here, $\frac{\partial}{\partial n}$ denotes partial derivative in the normal direction to a boundary $\partial\Omega_N$.

In the case of the MLPG5 approach [2, 30], in which the test function v is defined as a Heaviside step function ($v(P) = 1$, if $P \in \Omega_i$, otherwise $v(P) = 0$), e.g. the variational symmetric form (7) may be written as below

$$-\int_{\partial\Omega_i} k\frac{\partial u}{\partial n}\,\mathrm{d}s = \int_{\Omega_i} q\,\mathrm{d}\Omega_i \tag{9}$$

with the same requirements imposed on the trial function u and with Neumann type boundary conditions from (5). Here, meshless discretization of the domain Ω is required a-priori and the (9) form is satisfied on the subsequent nodal subdomains Ω_i ($i = 1, ..., N$, N—number of nodes).

3.2 Nodes generation and topology determination

The MFDM solution approach needs generation of clouds of nodes (arbitrarily distributed irregular points, forming later on an irregular cloud, that has basically no restrictions). Any mesh generator built for the FE analysis could be used here. However, it is convenient to use nodes generator taking advantage from the features specific for the MFDM analysis. Such generator (e.g. Liszka type generator [14, 15, 18]) may be based on nodes density control, zones with the regular mesh and smooth transition between them. It is assumed here that the potential user is able to describe the domain geometry e.g. by means of set of edge points ($(x_i^{(p)}, y_i^{(p)})$, $i = 1, ..., n^{(p)}$) as well as edge lines (edge curves in more general approach) (C_{ij}, i, $j = 1, ..., n^{(p)}$). Afterwards, the whole domain is enveloped by the rectangular region

$$R = \{(x, y); \quad x_{\min} \le x \le x_{\max}, \quad y_{\min} \le y \le y_{\max}\} \tag{10}$$

where

$$\begin{aligned} xmin &= \min(x_i^{(p)}) \quad , \quad xmax = \max(x_i^{(p)}) \\ ymin &= \min(y_i^{(p)}) \quad , \quad ymax = \max(y_i^{(p)}) \end{aligned} \tag{11}$$

Then the most possible dense regular square background mesh is generated in R. Actual nodes (x_i, y_i), $i = 1, ..., N$ are selected ("sieved") out of that background mesh using prescribed local mesh density $\bar{\rho}$ (e.g. constant for the very first rough regular cloud of nodes). In such procedure, location of any potential node has to be recognized (basing on the set of edge lines, e.g. by means of the *winding algorithm*) whether it is located inside the domain or not. If it is located outside, such point has to be left. More details may be found in any monograph devoted to the computational geometry aspects, e.g. in [32]. The boundary nodes may be generated directly from the edge topology (edge points becomes nodes whereas additional nodes are generated on every edge line in accordance with the prescribed density). Such nodes density based generator allows for effective determination of points that altogether with nodes belong to the cloud of one rank denser. Moreover, these a-priori determined points may serve as potential locations of new nodes in the a-posteriori error analysis and adaptive solution approach. A new node may be inserted with only small local modifications of the nodes topology. Additionally, without restrictions imposed on the nodes structure, any node can be easily shifted or removed.

When generated, the nodes are not bounded by any type of structure, like element or mesh regularity. However, it is convenient to determine afterwards the topology information of the already generated cloud of nodes [18]. In 2D domain topology is determined by

- Voronoi tessellation (domain partition into nodal subdomains [18, 22, 32]), and list of Voronoi neighbours assigned to each node,
- Delaunay triangulation (domain partition into triangular elements), and list of triangles involving each node [18, 22, 32].

Voronoi tessellation and Delaunay triangulation of the cloud of generated nodes, followed by their topology determination, are very useful for further analysis of the boundary value problems (e.g. to MFD star selection, numerical integration, postprocessing, nodes density control, MLPG subdomains determination, visualization purposes). However, as it was already mentioned, any topology information should not be applied in order to approximate the unknown (trial) function. In other case, method can not be treated as the meshless one. For instance, in case of the local formulation, topology is not necessary to determine (no integration). In case of the variational formulation, it may be useful but not essential. An 2D example of nodes topology determination for 2-T shaped domain, is presented in Fig. 1. Irregular cloud of nodes (top left), Delaunay triangulation (top right), Voronoi tessellation (bottom left) and a-priori distribution of adaptation points (bottom right) are shown.

3.3 MFD star selection and classification

A group of nodes used together as a base for a local MFD approximation is called the MFD star or stencil. Thus the MFD stars play similar role in the MFDM as the elements in the FEM, i.e. they are used for spanning a local approximation of the searched function. When dealing with irregular cloud of nodes, both MFD stars and formulas usually differ from node to node. However, configuration of stars may be common for some nodes. The most important feature of any selection criteria then is to avoid singular and ill conditioned MFD stars. Therefore, not only the distance from the central node counts, but also nodes distribution. Several MFD stars generation criteria may be distinguish [18], namely

- distance criterion—the simplest but most primitive one—a specified number of nodes from the closest distance of the central star point is taken into account,

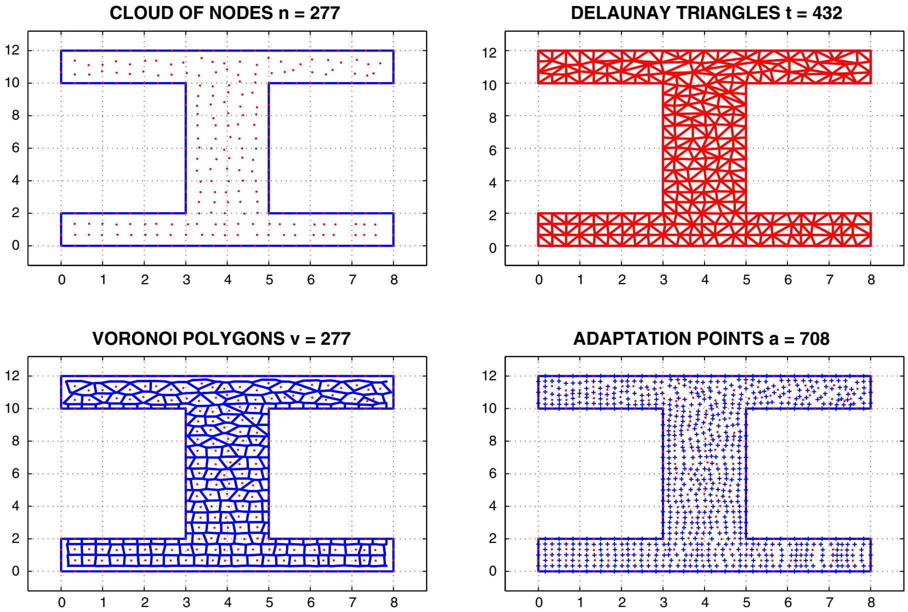


Fig. 1 Cloud of nodes topology for 2-T shaped domain: nodes, Delaunay *triangles*, Voronoi *polygons* and adaptation *points*

- improved 'cross' criterion—specified number of nodes (usually 2), closest to the central star point is taken separately from every of four quarters of the local coordination system, subjected at the central star point,
- the most complex 'Voronoi neighbours' criterion—selected to the MFD star are those nodes which are the Voronoi neighbours—that means e.g. in 2D domain that those polygons have common side (strong neighbours) or common vertex (weak neighbours) with the polygon prescribed to the central star point.

3.4 MWLS approximation and MFD schemes generation

The Moving Weighted Least Squares approximation [13, 18], spanned over local MFD stars, is widely used in the MFDM in order to generate MFD formulas as well as in the postprocessing. For any mentioned type of b.v. problem formulation, MFD formulas for prescribed approximation order p are required at every node as well as at Gauss integration point (for global type only). Consider MFD star at point $(\bar{x}, \bar{y})$, which consists of m arbitrarily distributed points. Number of these nodes, (in general: number of degrees of freedom) in the MFD star, should not be smaller than the number of coefficients s to be determined. Usually, it is greater in order to avoid dealing with ill-conditioned simultaneous algebraic equations. The local approximation $\hat{u}(x, \bar{x}, y, \bar{y})$ of p order is spanned using here Taylor series expansion, including s subsequent terms (for 2D $s = 0.5(p+1)(p+2)$). By assuming the interpolation conditions ($\hat{u}(x_i, \bar{x}, y_i, \bar{y}) = u_i$, where u_i—given nodal values), one gets overdetermined system of algebraic equations

$$\mathbf{P}^T\mathbf{D}u = \mathbf{q} \tag{12}$$

where $\mathbf{P}$—matrix of known local interpolants (h, k), $\mathbf{D}u$—unknown approximation coefficients (local derivatives) vector and $\mathbf{q}$—nodal degrees of freedom vector (e.g. function values).

$$\mathbf{P} = \begin{bmatrix} 1 & h_1 & k_1 & \frac{1}{2}h_1^2 & \dots & \frac{1}{p!}k_1^p \\ 1 & h_2 & k_2 & \frac{1}{2}h_2^2 & \dots & \frac{1}{p!}k_2^p \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & h_m & k_m & \frac{1}{2}h_m^2 & \dots & \frac{1}{p!}k_m^p \end{bmatrix}, \mathbf{D}u = \begin{bmatrix} u \\ u'_x \\ u'_y \\ u''_{xx} \\ \dots \\ u^{(p)}_{yy\dots y} \end{bmatrix}, \mathbf{q} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \dots \\ u_m \end{bmatrix} \tag{13}$$

Here, $h_i = \bar{x} - x_i$, $k_i = \bar{y} - y_i$. Unknown derivatives vector $\mathbf{Du}$ is obtained in such a manner that the weighted error functional

$$J = (\mathbf{PD}u - \mathbf{q})^T\mathbf{W}^2(\mathbf{PD}u - \mathbf{q}) \tag{14}$$

is minimised with respect to the $\mathbf{D}u$. Here, $\mathbf{W} = \text{diag}\{\omega_i\}$, $i = 1, ..., m$ is a diagonal weights matrix. The solution of the functional optimisation problem yields the MFD formulas matrix $\mathbf{M}$ for a complete set of derivatives up to and including order p

$$\mathbf{D}u = \mathbf{Mq}, \quad \mathbf{M} = (\mathbf{P}^T\mathbf{W}^2\mathbf{P})^{-1}\mathbf{P}^T\mathbf{W}^2 \tag{15}$$

Moreover, the local approximation for MFD star at point $(\bar{x}, \bar{y})$ may be written in the following form

$$\hat{u}(x, y) = \mathbf{p}^T(x, y)\mathbf{D}u = \mathbf{p}^T(x, y)\mathbf{Mq} = \Phi^T(x, y)\mathbf{q} \tag{16}$$

with the pseudo-shape functions $\Phi\ (x, y) = \{\varphi_i(x, y)\}$, $i = 1, ..., s$ and vector $\mathbf{p}$ of interpolants (h, k). For determination of the MFD formulas (15), usually the singular weights are assumed

$$\omega_i = \frac{1}{(\sqrt{h_i^2 + k_i^2})^{p+1}} \tag{17}$$

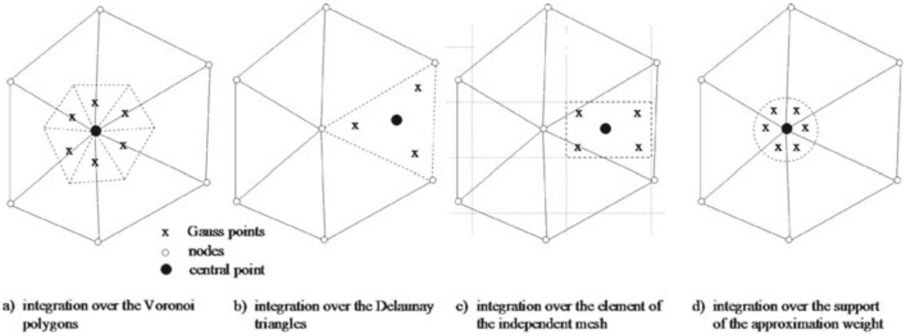
Singularity assures, in this way, the delta Kronecker property $\varphi_i(x_j, y_j) = \delta_{ij}$, and consequently enforces interpolation at the central point $(\bar{x}, \bar{y})$ of each MFD star as well as a-priori fulfilment (without any additional techniques) of the Dirichlet boundary conditions. For the postprocessing, when data smoothing is required, one may use non-singular weights and MFD stars with the greater number of nodes m than considered here. It is worth stressing that other meshless methods [3] use alternative polynomial notation for the local approximation (12). However, Taylor series based notation used here offers also information about approximation error, caused by a truncated part of the Taylor series, as well as provides a simple interpretation of the approximation coefficients considered as function derivatives (local type). In general, the MWLS approximation is powerful and effective approximation technique, however its quality depends on a number of parameters involved.

3.5 Numerical integration in the MFDM

Numerical integration plays an important role in the MFDM, and may have significant influence on the final results, when applied to boundary value problems posed in the global or mixed global-local formulations. Integration may be also required in the postprocessing of nodal results, e.g. when energy norm of the solution error has to be evaluated over a chosen subdomain. The type and values of integration parameters depend on the purpose of integration.

There are at least four basic ways of numerical integration in the MFDM

- Subdivision of the domain Ω into subdomains Ω_i, $i = 1, 2, ..., n$ assigned to each node, and integration over these subdomains (Fig. 2a). This may be performed by means of the Voronoi tessellation and integration over Voronoi polygons (in 2D) Ω_i or Voronoi polyhedrons V_i (in 3D).
- Subdivision of the domain Ω into triangular elements (in 2D) or tetrahedrons (in 3D) with nodes located at their vertices, and integration over these elements (Fig. 2b). The Delaunay triangulation (in 2D) seems to be the best choice here. Integration is performed using the same quadratures as in the finite element analysis, while values of the integrands at Gaussian points are found by means of the MWLS approximation,
- Introduction of an independent background mesh and subdivision of the domain Ω into subdomains (triangles, squares, ...) in a way independent on existing nodes, and integration over these subdomains (Fig. 2c),



a) integration over the Voronoi polygons

b) integration over the Delaunay triangles

c) integration over the element of the independent mesh

d) integration over the support of the approximation weight

Fig. 2 2D integration in the meshless FDM

- Integration over the local subdomains defined by the finite local supports (circles, ellipsis, rectangles, ...) of the weighting factors w_j, $j = 1, ..., r$ (Fig. 2d). Such approach that is typical for the other meshless methods, could be also used in the MFDM.

It is worth stressing that the background mesh consisting of the integration cells must not be confused with the cloud of nodes for function approximation. These may be completely different, or they may be dependent from each other. However, approximation of the unknown has to be based on nodes only, in each case without any influence from the integration mesh. Moreover, the selection of the integration scheme may depend on the type of the formulation of the b.v. problem applied or the derivatives order. For instance, in case of the Atluri's MLPG approach [2, 31], the integration is reduced to the nodal subdomains of simple geometrical shape, whereas the test function may be even non-zero constant function over those subdomains. Some terms in the variational principle may vanish then. This is the MLPG5 case.

3.6 Generation of the MFD equations

Having found the MFD operators for all derivatives (15), one may compose any MFD operator required either for a MFD equation, boundary conditions or for an integrand (for the global MFD formulations). If its order is greater than the assumed one p, one may use appropriate formulae composition, e.g. $u''''_{xxxx} = (u''_{xx})_{xx}$ and twice the formula (15).

Generation manner of the MFD equations depends on the type of the boundary value problem formulation [18]. In the local formulation, MFD equations are generated by collocation technique, which assumes satisfying the difference formulas at all internal nodes inside the domain. In the global formulations, numerical integration is additionally required. It is followed then by the aggregation technique, similar to the FEM. The MFDM equations are generated then

- directly from the variational principle (Galerkin type approach) or
- by means of minimisation of the appropriate functional.

Consider e.g. a class of linear differential operators of the second order

$$Lu = c_0 u + c_1 \frac{\partial u}{\partial x} + c_2 \frac{\partial u}{\partial y} + c_3 \frac{\partial^2 u}{\partial x^2} + c_4 \frac{\partial^2 u}{\partial x \partial y} + c_5 \frac{\partial^2 u}{\partial y^2} \approx \mathbf{c}^T \mathbf{D} u \tag{18}$$

where $\mathbf{c} = \{c_0, ..., c_5\}$ are known coefficients. A required MFD operator is here a linear combination of derivatives $\mathbf{D}u$.

Collocation technique, for the local formulation, leads directly to the set of algebraic equations arising node by node

$$Lu_i \approx \mathbf{c}^T \mathbf{D} u = f_i \quad , \quad P_i \in \Omega \tag{19}$$

As it was already mentioned and discussed, in the global formulations, numerical integration is additionally required. It is followed then by the aggregation technique, like in the FEM.

Consider e.g. the global formulation given by the energy functional in the particular form

$$I(u) = \int_{\Omega} F(u) \mathrm{d}\Omega \tag{20}$$

After numerical integration

$$I(u_1, u_2, ..., u_N) \approx \sum_{j=1}^{M} J_j \sum_{i=1}^{N_G} \omega_{i(j)} \; F(u)|_{P=P_{i(j)}} \tag{21}$$

where M—number of integration cells, J_j—transformation matrix determinant, N_G—number of Gauss points, $\omega_{i(j)}$—integration weight, $P_{i(j)}$—Gauss integration point. MFD equations are generated by the functional minimisation with the respect to the unknown nodal values $u_1, u_2, ..., u_n$

$$\frac{\partial I}{\partial u_i} = 0 \quad , \quad i = 1, 2, ..., n \tag{22}$$

Variational formulation, after numerical integration and aggregation, produces at once the system of FD equations.

3.7 MFD discretization of boundary conditions

Quality of the MFD solutions essentially depends on the quality of discretization of the boundary conditions. Consider b.v. problem formulated in (1). In case of essential (Dirichlet) boundary conditions, it is sufficient to substitute at boundary nodes

$$u_i = \bar{u} \quad for \quad P_i \in \partial\Omega_D \tag{23}$$

since the MFD pseudo-shape functions posses the Delta Kronecer property as long as the weights (17) applied for MWLS approximation remain singular. In that case the approach is exactly the same as in the FEM. In case of natural (Neumann) boundary conditions, the differential operators are involved in such conditions. Additional MFD approximation is required then at boundary nodes $P_i \in \partial\Omega_N$. It is obtained in the same manner as described in Section 3.4. However, the MFD approximation at boundary node which uses only internal nodes, shall be less accurate than the one inside the domain. To improve approximation accuracy, several approaches may be distinguish [18, 29]. All of them assume usage of internal nodes for approximation as well as additional

- boundary information, e.g. boundary condition or domain equation but specified on its boundary,
- generalised degrees of freedom (g.d.o.f),
- higher order terms from the Taylor series expansion of the MFD operator,
- external fictitious nodes.

Use of so called fictitious nodes, located outside the domain, introduces additional unknowns to the system of algebraic equations. Using relevant boundary formulas, they may be expressed in terms of the internal nodes values based on the appropriate boundary conditions. Thus one gets slightly better approximation, because the central node is closer to “centre of gravity” of the MFD star. However, this approach is not recommended in the hyperbolic problems (in dynamic mechanics), due to the fact, that greater number of nodes artificially increases the mass of the discretized system.

Higher order approximation, that may be provided by several mechanics including correction terms of the MFD operators, and general multipoint approach, was discussed in details in many papers co-authored by the author of the present paper [23–30]. Though interesting, it will be not considered here.

3.8 Solution of simultaneous FD equations

In the MFDM analysis, one deals with Simultaneous Algebraic Equations (SAE). They may be linear or non-linear equations, e.g. when the original boundary value problem analysed is non-linear. In the case of linear boundary value problems, appropriate SLAE may be of non-symmetric (e.g for local b.v. formulation) or symmetric form (for global formulations, with proper discretization of the boundary conditions). In the last case they might be solved by means of similar procedures like those for the FEM discretization. Non-symmetric equations may use solvers developed e.g. for the fluid mechanics. However, the best approach seems to be development of solvers specific for the MFDM, taking advantage of this method nature. Especially, the multigrid adaptive solution approach [5, 18, 22] seems to be effective then.

After obtaining solution of SLAE, postprocessing of the results is usually required. The MWLS approximation is a powerful tool for postprocessing because it may provide users with values of a considered function, and its derivatives at every required point. Approximation is based on discrete data (values of functions, their derivatives or other generalised degrees of freedom). Approximated results may be directly obtained, at each point of interest, using the approach defined by formulas discussed in Section 3.4. Thus the same MWLS approach is used as the one applied to the generation of the meshless difference operators discussed above. Though precise enough, that MWLS approach is time consuming, because solution of the local equations system is needed at each point where such approximation is required. Precision of the MWLS results significantly depends on the right choice of the set of parameters involved. They may essentially change the quality of the standard MWLS approximation.

3.9 General remarks

The basic solution MFDM approach, outlined above, has been extended in many ways so far (see beginning of Section 3), and is still under current development. Nowadays, special attention is given to a-posteriori error analysis [1, 29], higher order approximation [25, 26, 29] as well as discretization of various b.v. problem formulations (especially the MLPG ones [2, 30]). However, many problems still need to be defined and solved, some of them are under current research (mathematical foundations, multigrid adaptive nodes generator, combinations with FEM and Artificial Intelligence (AI) methods, etc). This paper focuses on one of them, namely the lack of commercial programs allowing for effective MFDM analysis of real engineering problems.

Many problems still need to be defined and solved, some of them are under current research nowadays. Among them one may distinguish

- solid mathematical bases of the MFDM, including such problems as solution existence, solution and residuum convergence, stability of the MFD schemes,
- various Petrov–Galerkin formulations and their discretization using MFDM,
- study on the influence of the numerous parameters on the quality of the MWLS approximation
- further development of the higher order approximation, based on
  - correction terms [19],
  - multipoint approach [6],
- improved, solution and residual error estimation, based on the new, higher order reference solution of high quality,
- analysis of the multigrid, full adaptive solution approach [8], based on the mesh generator, oriented on the 2D and 3D large non-linear boundary value problems,
- acceleration of the SAE solution,
- comparison and coupling of the MFDM with the other meshless methods,
- combination of the MFDM with other discrete methods, especially with the Boundary Element Method (BEM), FEM, and Artificial Intelligence (AI) methods,
- development of the engineering software,
- various engineering applications.

The present paper aims to fulfil one important topic from the list above, namely the lack of the appropriate computational software.

4 Program codes for the MFDM analysis

A general manner of designing computer codes pursuing b.v. problems using the MFDM, is presented here. Though all guidance is as much general as it may be, the exemplary codes have been prepared by the author using the Matlab technical language. Several aspects should be taken into the account while selecting the programming environment, for instance computational speed, availability of function libraries, results presentation abilities etc. In Matlab, mathematical background and graphical visualisation allow for fast and effective programming of designed numerical algorithms. Therefore, the main emphasis was laid here upon transforming fundamental aspects of the basic MFDM solution approach into the Matlab code. The code was optimized in order to reduce the usage of the "for" loop which has significant influence on the speed of program execution. However, this kind of loop appears in some procedures devoted to numerical integration or nodal collocation, in order to deliver more accessible and readable code.

Correctly designed computational package should consist of three main parts, namely

– pre-processor: domain and boundary determination, choice of b.v. problem formulation, description of material, geometry, load as well as very first (usually rough regular) cloud of nodes generation with a-posteriori determined topology (triangles, polygons, neighbours) and mesh for integration (if required),
– processor (solver): MFD star generation at nodes and Gauss points, local MWLS approximation, generation of MFD operators and equations, numerical integration (when needed), boundary condition discretization, solution of linear SAE,
– post-processor: general (MWLS) postprocessing of nodal results (e.g. calculating stresses from displacements) as well as a-posteriori solution error estimation, adaptation of cloud of nodes as well as return to solver with new adapted cloud.

Here, attention was laid upon the implementation of the solver aspects. User's task is to generate the primary cloud of nodes, which may be regular one in the simplest case or a-priori refined using e.g. geometry based criteria (concentration zones). Although the nodes generation for the meshless methods is much simpler than in the FE analysis, it remains a complex problem and it is not described here in more detailed way. However, some general ideas were already presented in the previous sections. In Matlab, the simplest topology information (if required) may be prepared by applying several Matlab functions: '*delaunay*' (or '*delaunayn*') and '*voronoi*' (or '*voronoin*'). Refer to any Matlab documentation for more details. However, it should be pointed out that those functions generate triangles and polygons without any a-priori knowledge of the domain boundary. Therefore, additional user's procedure should be designed in order to eliminate incorrect (external) triangles as well as to rebuild polygons of the boundary nodes. Eventually, one should have list of all nodes coordinates, codes of boundary nodes (internal or boundary node) as well as list of all triangles and polygons (for global formulations and for postprocessing purposes). The exemplary results for 2-T shaped domain were already presented in Fig. 1.

The following engineering problem was considered. Find total shear stress

$$\tau = \sqrt{\tau_{zx}^2 + \tau_{zy}^2} \quad , \quad \tau_{zx} = \frac{\partial F}{\partial y} \quad , \quad \tau_{zy} = -\frac{\partial F}{\partial x} \tag{24}$$

in a prismatic bar of specified cross-section (e.g. rectangular or 2-T shaped) subjected to the torsional moment. Such problem may be posed in local formulation

$$\begin{cases} \nabla^2 F = -2G\theta & in \quad \Omega \\ F = 0 & on \quad \partial\Omega \end{cases} \tag{25}$$

as well as in the weak variational (Galerkin) one: Find such (trial) function $F \in H_0^1$ that for any (test) function $v \in H_0^1$ satisfied is

$$-\int_\Omega (F'_x v'_x + F'_y v'_y)\,\mathrm{d}\Omega = -2G\theta \int_\Omega v\,\mathrm{d}\Omega \tag{26}$$

Here, $F = F(x, y)$ is a scalar Prandtl function (unknown primary solution), G is a Kirchhoff modulus (material parameter) and θ is a torsional angle (load). The local formulation (25) is a typical formulation for FDM whereas the variational one (26) is commonly applied in the FE analysis.

5 Software information

Several separate m-files and one data file were prepared by the author of the paper. The software is available from Netlib (http://www.netlib.org/numeralgo/) as the na36 package. Installation of the prepared software is not very complicated. The user is supplied with one zip file (MFDMtool.zip), which contains one directory with all necessary files. This archive file should be uncompressed to the prescribed location. The only m-file from this directory, which may be run directly, is the TEST.m file. One may add the name of this directory to the Matlab search path in order to run this file from the Matlab Command Line. In other case, it should be open through the Matlab Editor and run, after the change of the Matlab Current Directory. Although all codes have been prepared and tested by means of the Matlab 7.7.0 (R2008b), they do not contain any functions or syntax, which may be untypical for most commonly applied Matlab versions. More details of the implementation techniques of the subsequent steps of the MFDM algorithm may be found in the README.txt file as well as within the instruction manual attached to the package.

6 Conclusions

The computer implementation techniques of the basic Meshless Finite Difference Method (MFDM) solution approach were considered. MFDM belongs to the wide group of the meshless methods (MM). In MM, the approximation of the unknown function is prescribed in terms of nodes only. Therefore, no nodes connectivity is required for local approximation support. The MFDM solution approach constitutes the fully mature and effective solution tool for analysis of the boundary problem of mechanics. However, the long-time practice and enormous computational work devoted to the Finite Element Method (FEM), caused stagnation in many aspects of the MFDM. The main goal of this paper is to encourage scientists who deal with computational mechanics to design their own codes based on the MFDM solution approach. The most important aspects of this approach were described and implemented by means of the well-known Matlab package. Special attention was laid upon problems of the MFD star generation, local Moving Weighted Least Squares Approximation, MFD equations generation, integration schemes as well as the final postprocessing of the results. The exemplary codes concern the shear problem of the prismatic bar, posed in both local and variational (Galerkin) formulations. Moreover, they may be easily extended by users for a large class of mechanical and engineering applications.

However, as it may be seen, this is only an interesting starting point. Many of the aspects of the MFDM solution approach which do not appear here, are worth paying more attention and treatment. The following directions, which are under extensive development nowadays, seem to be most fascinating: higher order approximation provided by correction terms, discretization of the MLPG formulations as well as improved a-posteriori error analysis. Presentation of their computational aspects is planned as the next part of this paper.

References

1. Ainsworth, M., Oden, J.T.: A-posteriori error estimation in finite element analysis. Comput. Methods Appl. Mech. Eng. **142**, 1–88 (1997)
2. Atluri, S.N.: The Meshless Method (MLPG) for Domain & Bie Discretizations. Tech Science Press (2004)
3. Belytchko, T.: Meshless methods an overview and recent developments. Comput. Methods Appl. Mech. Eng. **139**, 3–47 (1996)
4. Benito, J.J., Ureña, F., Gavete, L., Alonso, B.: Application of the GFDM to improve the approximated solution of 2nd order partial differential equations. In: ICCES MM (2007)
5. Brandt, A.: Multi-level adaptive solutions to boundary value problems. Math. Comput. **31**, 333–390 (1977)
6. Collatz, L.: The Numerical Treatment of Differential Equations. Springer, Berlin (1966)
7. Gingold, R.A., Monaghan, J.J.: Smoothed particle hydrodynamics - theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. **181**, 375–389 (1977)
8. Hackbush, W.: Multi-Grid Methods and Applications. Springer, Berlin (1985)
9. Jensen, P.S.: Finite difference techniques for variable grids. Comput. Struct. **2**, 17–29 (1972)
10. Karmowski, W., Orkisz, J.: A physically based method of enhancement of experimental data - concepts, formulation and application to identification of residual stresses. In: Proc. IUTAM Symp. on Inverse Problems in Engng. Mech., Tokyo 1992. On Inverse Problems in Engineering Mechanics, pp 61–70. Springer (1993)
11. Krok, J.: A new formulation of Zienkiewicz–Zhu a posteriori error estimators without superconvergence theory. In: Proceedings Third MIT Conference on Computational Fluid and Solid Mechanics, Cambridge, MA, USA, 14–17 June 2005

12. Krok, J., Orkisz, J.: A discrete analysis of boundary-value problems with special emphasis on symbolic derivation of meshless FDM/FEM models. In: Computer Methods in Mechanics CMM, Spała, Łódź, Poland, 19–22 June 2007
13. Lancaster, P., Salkauskas, K.: Surfaces generated by moving least-squares method. Math. Comput. **155**(37), 141–158 (1981)
14. Liszka, T.: An interpolation method for an irregular net of nodes. Int. J. Numer. Methods Eng. **20**, 1599–1612 (1984)
15. Liszka, T., Orkisz, J.: The finite difference method at arbitrary irregular grids and its applications in applied mechanics. Comput. Struct. **11**, 83–95 (1980)
16. Milewski, S.: Meshless finite difference method with higher order approximation—applications in mechanics. In: Archives of Computational Methods in Engineering, vol. 19, issue 1, pp. 1–49. Springer, Berlin (2012)
17. Nayroles, B., Touzot, G., Villon, P.: Generalizing the FEM: diffuse approximation and diffuse elements. Comput. Mech. **10**, 307–318 (1992)
18. Orkisz, J.: Finite difference method (part III). In: Kleiber, M. (ed.) Handbook of Computational Solid Mechanics, pp. 336–431. Springer, Berlin (1998)
19. Orkisz, J.: Higher order meshless finite difference approach. In: 13th Inter-Institute Seminar for Young Researchers, Vienna, Austria, 26–28 Oct 2001
20. Orkisz, J., Jaworska, I.: Some concepts of 2D multipoint HO operators for the meshless FDM analysis. In: ICCES Special Symposium on Meshless Methods, Patras, Greece, 15–17 June 2007
21. Orkisz, J., Jaworska, I., Milewski, S.: Meshless finite difference method for higher order approximation, In: Third International Workshop Meshfree Methods for Partial Differential Equations, Bonn, Germany, 12–15 Sept 2005
22. Orkisz, J., Lezanski, P., Przybylski, P.: Multigrid approach to adaptive analysis of bv problems by the meshless GFDM. In: IUTAM/IACM Symposium on Discrete Methods in Structural Mechanics II, Vienna (1997)
23. Orkisz, J., Milewski, S.: On higher order approximation in the MDM method. In: Bathe, K.J. (ed.) Proceedings Third MIT Conference on Computational Fluid and Solid Mechanics, Cambridge, MA, USA, Elsevier, 14–17 June 2005
24. Orkisz, J., Milewski, S.: Higher order approximation approach in meshless finite difference analysis of boundary value problems. In: The 16th International Conference on Computer Methods in Mechanics CMM-2005, Częstochowa, Poland, 21–24 June 2005
25. Orkisz, J., Milewski, S.: On a-posteriori error analysis in higher order approximation in the meshless finite difference method. In: ICCES Special Symposium On Meshless Methods, Dubrovnik, Croatia, 14–16 June 2006
26. Orkisz, J., Milewski, S.: Recent advances in the higher order approximation in the meshless finite difference method. In: 7th World Congress on Computational Mechanics, Los Angeles, California, 16–22 July 2006
27. Orkisz, J., Milewski, S: Recent advances in a-posteriori error estimation based on the higher order correction terms in the meshless finite difference method. In: ICCES Special Symposium on Meshless Methods, Patras, Greece, 15–17 June 2007
28. Orkisz, J., Milewski, S.: Higher order approximation multigrid approach in the meshless finite difference method. In: Computer Methods in Mechanics CMM, Spała, Łódź, Poland, 19–22 June 2007
29. Orkisz, J., Milewski, S.: Higher order a-posteriori error estimation in the meshless finite difference method. In: Griebel, M., Schweitzer, M.A. (eds.) Meshfree Methods for Partial Differential Equations IV. Lecture Notes in Computational Science and Engineering, pp. 189–213. Springer (2008)
30. Orkisz, J., Milewski, S.: Global-local Petrov–Galerkin formulations in the meshless finite difference method. In: Griebel, M., Schweitzer, M.A. (eds.) Meshfree Methods for Partial Differential Equations V. Lecture Notes in Computational Science and Engineering, pp. 1–26. Springer (2011)
31. Orkisz, J., Milewski, S.: Higher order discretization of the meshless local Petrov Galerkin formulations. In: Proceedings CMM-2009—Computer Methods in Mechanics, Zielona Góra, Poland, 18–21 May 2009
32. Preparat, F.P., Shamos, M.I.: Computational Geometry. An Introduction. Springer, Berlin Heidelberg (1985)

33. Schweitzer, M.A.: Meshfree methods for partial differential equations. In: Computer Methods in Mechanics CMM, Spała, Łódź, Poland, 19–22 June 2007
34. Zienkiewicz, O.C., Taylor, R.L.: Finite Element Method its Basis and Fundamentals, 6th edn. Elsevier (2005)
35. Zienkiewicz, O.C., Zhu, J.Z.: A simple error estimator and adaptive procedure for practical engineering analysis. Int. J. Numer. Methods Eng. **24**, 337–357 (1987)