International Journal of Analysis and Applications

An Improved Finite Cloud Method With Uniformly Distributed Clouds and Enhanced Boundary Conditions

Miew Leng Oh, See Pheng Hang*

Department of Mathematical Sciences, Faculty of Science,

Universiti Teknologi Malaysia, 81300 Johor Bahru, Johor, Malaysia

*Corresponding author: sphang@utm.my

ABSTRACT. Finite cloud method (FCM) employs the fixed kernel reproducing technique to construct the interpolation function and point collocation approach is adopted for the discretization. In this study, an improved FCM is proposed such that a node of interest is approximated with its nearest cloud. This feature enables a set of uniformly distributed clouds of various densities such that all the information in the problem domain is captured and stored in the clouds. Additionally, the instability of FCM near the boundaries is treated by having the boundary nodes also satisfy the governing differential equation. Besides, a splitting mechanism is suggested for the node refinement to improve the accuracy of solution. Parameters are introduced to control the density of clouds and the singularity of the moment matrices associated with the clouds. Thus, a more controllable numerical simulation is developed. Numerical examples are presented and the results have shown that the improved FCM produces a stable and better accuracy of solution.

1. Introduction

Meshfree methods have been studied intensively by researchers in the field of engineering. Meshfree methods ([1], [2], [3]) become one of the hottest topics in researchers' eyes, owing to the fact that meshfree methods possess an attractive advantage over conventional mesh-based methods, which is that the mesh generation is not required in the formulation procedure. Hence, handling

Received: Dec. 21, 2022.

2020 Mathematics Subject Classification. 65N06, 65N38.

Key words and phrases. finite cloud method; uniformly distributed clouds; boundary conditions.

https://doi.org/10.28924/2291-8639-21-2023-12

ISSN: 2291-8639

problems such as fracture, large deformation and moving boundaries is easier for meshfree methods since adding nodes to the affected area is much simpler, compared to mesh-based methods.

There are numerous meshfree methods available in the literature. The Element-free Galerkin method (EFG) ([4], [5]) employs the moving least square (MLS) approximation to construct the interpolation function based on nodes in a local domain. For meshless local petrov-Galerkin method (MLPG) ([6], [7], [8]), a method that solves local weak form of PDEs, the weighted residual method is implemented in integral form. The integration is confined to a small local subdomain of a particular node. The random differential quadrature method (RDQ) ([9]) is a meshfree method that combined fixed reproducing kernel particle method (fixed RKPM) and differential quadrature method (DQM) ([10], [11], [12]) motivated by having DQM applied to irregular domain with randomly distributed field nodes.

Recently, the improved interpolating element-free Galerkin (IIEFG) method has emerged to be another outstanding meshfree method that has made improvements to the interpolation accuracy. IIEFG employs the improved interpolating moving least-square technique to construct the interpolation functions which possess the Kronecker delta property and the related studies ([13], [14]) have shown that this method can achieve better accuracy of solution compared to EFG method.

On the other hand, researchers and scientists also find ways to improve the existing mesh-based methods in meshfree directions. The hybrid finite element-meshfree method ([15], [16]), the meshfree finite volume method (FVM) ([17]) and the meshfree finite difference method ([18]). There is also an innovative approach being suggested to solve the mesh nodes and the meshfree points which are arbitrarily mixed in the computational domain using FVM ([19], [20]).

Meshfree methods allow a more flexible way of adding new nodes while keeping the existing field nodes because the connection information among the field nodes is not needed. Hence, this would be an added advantage for node refinement since new field nodes can be added to the critical region where the detailed analysis is required. Therefore, some researchers have devoted their effort to coupling a meshfree method with a mesh-based method ([21]) such that node refinement is carried out in the critical region by using the meshfree method whereas the mesh-based method is adopted in the smooth region ([22]).

FCM is a meshfree method that employs fixed RKPM for the construction of interpolation functions and then adopts the point collocation approach to discretize the governing differential equation. FCM has been applied in various fields, like computer-aided design ([23], [24], [25]) and

the simulation of the behavior of hydrogel ([26]). Several papers related to some improvements to FCM are found in the literature ([27], [28], [29], [30]). However, the moment matrix associated with a cloud may become singular if the cloud size is not large enough to have a sufficient number of field nodes or the cloud center is too near the boundary such that it becomes incomplete and consequently have fewer field nodes for its interpolation. In order to have a more controllable numerical simulation, parameters are introduced to cater for the cloud distribution. More importantly, the cloud density can be adjusted when necessary or even during the node refinement phase to obtain the desired accuracy of solutions.

In this paper, parameters are introduced to control the density of clouds as well as the distance of the clouds, which are adjacent to the boundaries, from the boundaries. A study is carried out to investigate the effect of the distance of the cloud centers from the boundaries to the singularity of the moment matrices associated with these clouds. Apart from that, the relationship between the density of clouds and the accuracy of solution is also examined.

2. Finite Cloud Method (FCM)

FCM is a meshfree method that employs fixed reproducing kernel technique to construct the interpolation functions using a set of overlapping clouds defined over the problem domain. Each cloud consists of a set of field nodes. As clouds are overlapping, a field node may belong to more than one cloud. With the appropriate order of polynomial basis, a moment matrix is created for each cloud defined in the domain. When the moment matrices are ready, the interpolation functions will then be constructed for the clouds.

For the discretization of the governing partial differential equations, the diffuse derivative approach is adopted to approximate the terms of the derivatives. Since the moment matrices in fixed RPKM are constants, the derivatives of the interpolation functions are straightforward, that is the differentiations involve only the polynomial basis of the interpolation functions.

2.1. Fixed Reproducing Kernel Particle Method

RKPM is inherited from smoothed particle hydrodynamics (SPH) method where an extra term, the correction function, is added to the interpolation function ([31]) to achieve a higher order of reproducibility of the field variables. The RKPM is given as

$$u^a(x,y) = \int_\Omega C(x,y,s,t)\,\psi(x-s,y-t)\,u(s,t)\,ds\,dt\,. \tag{2.1}$$

$\Omega \in \mathbb{R}^2$. $C(x,y,s,t)$ is the correction function whereas $\psi(x,y,s,t)$ is the kernel function or window function which acts as the low pass filter that reduces noise in the solution. The discrete form of RKPM is given as

$$u^a\,(x,y) = \sum_{I=1}^{NP} N_I(x,y)\,u_I\,, \tag{2.2}$$

where $N_I(x,y) = C(x,y,x_I,y_I)\,\psi(x-x_I,y-y_I)\,dV_I$, u_I is the nodal unknown for field node I and *NP* is the total number of field nodes in the domain *Ω*.

Fixed RKPM is a special case of RKPM where the kernel is fixed at the central node (x_k,y_k) and hence produces a constant moment matrix, unlike classical RPKM, moving RPKM and multiple RPKM which generate moment matrices with entries comprised of functions of *x* and *y*.

The approximate solution for the fixed RKPM is as follows:

$$u^a(x,y) = \int_\Omega C(x,y,s,t)\,\psi(x_k-s,y_k-t)\,u(s,t)\,ds\,dt\,. \tag{2.3}$$

(x_k,y_k) is the centre of the kernel $\psi(x_k-x_I,y_k-y_I)$ where the local interpolation takes place.

2.2. Polynomial Basis Function

Several types of basis functions are available in the literature, like polynomial basis, Fourier bases and B-spline basis function. For FCM, the relatively simpler basis functions, i.e. polynomial basis function, is employed (as shown in Equation 2.4).

$$P^T(s,t) = \{p_1,p_2,\dots,p_m\} \tag{2.4}$$

$P^T(s,t)$ is the basis function of basis degree *D*. The two-dimension basis function of order *m* can be computed by Equation 2.5.

$$m = \frac{(D+1)*(D+2)}{2} \tag{2.5}$$

If $D = 1$, then $m = 3$ and the corresponding linear basis is

$$P^T(s,t) = [1,s,t]. \tag{2.6}$$

If $D = 2$, then $m = 6$. This leads to a quadratic basis given by

$$P^T(s,t) = [1, s, t, s^2, st, t^2]. \tag{2.7}$$

3. Formulation of FCM

In FCM, the interpolation functions are constructed by fixed RKPM where its derivatives are easy to derive due to the constant moment matrix associated with each cloud ([32]). Then, Jin et al. ([28]) have investigated the FCM and proposed the use of shifted polynomial basis in the construction of interpolation function and have proven that the improved FCM produces superior convergence of solutions.

The approximate equation derived by the FCM is presented in this section. The fixed RKPM interpolation function is first constructed followed by the discretization accomplished by point collocation approach.

3.1. Formulation of Fixed RKPM Interpolation Functions With Shifted Polynomial Basis

The approximate solution for the fixed RKPM is defined as

$$u^a(x,y) = \int_\Omega C(x,y,x_k - s, y_k - t)\, \psi(x_k - s, y_k - t)\, u(s,t)\, ds\, dt\,. \tag{3.1}$$

$C(x,y,x_K - s, y_K - t)$ is the correction function as follow:

$$C(x,y,x_k - s, y_k - t) = P^T(x_k - s, y_k - t)\, c(x,y), \tag{3.2}$$

where $P^T(s,t)$ is the basis function as shown in Equation 2.4 and $c(x,y)$ is the unknown correction function coefficients. From Equation 3.2, Equation 2.3 becomes

$$u^a(x,y) = \int_\Omega P^T(x_k - s, y_k - t)\, c(x,y) \psi(x_k - s, y_k - t)\, u(s,t)\, ds\, dt\,. \tag{3.3}$$

Since every monomial of the polynomial defined in Equation 2.4 has to satisfy the consistency condition of the approximate function, the following equation is obtained:

$$p_i(x,y) = \int_\Omega P^T(x_k - s, y_k - t)\, c(x,y)\, \psi(x_k - s, y_k - t)\, p_i(s,t) ds\, dt, \quad i = 1, 2, .., m \tag{3.4}$$

and the relevant discretized form is rewritten as

$$p_i(x,y) = \sum_{I=1}^{NP} P^T\,(x_k - x_I, y_k - y_I)\; c(x,y)\; \psi(x_k - x_I, y_k - y_I)\; p_i\,(x_I, y_I)\; dV_I \quad (3.5)$$

where $i = 1,2,..,m$. *NP* denotes the total field nodes in the cloud and dV_I is the nodal volume at the I^{th}field node.

Let *M* be the *m* x *m* moment matrix given as

$$M_{ij} = \sum_{I=1}^{NP} p_j(x_k - x_I, y_k - y_I)\; \psi(x_k - x_I, y_k - y_I)\; p_i(x_I, y_I)\; dV_I\,, \quad i,j = 1,2,\dots,m. \quad (3.6)$$

Then, the Equation 3.6 can be written in a matrix form as

$$Mc(x,y) = P(x,y). \quad (3.7)$$

Note that the moment matrix M is not function of x and y. In another words, M is a constant matrix. From Equation 3.7, the unknown correction function coefficients can be determined as

$$c(c,y) = M^{-1}\,P(x,y). \quad (3.8)$$

Substituting Equation 3.8 into Equation 2.3 gives

$$u^a(x,y) = \int_{\Omega} P^T(x,y)\; M^{-T}\; P(x_k - s, y_k - t)\psi(x_k - s, y_k - t)\; u(s,t)\; ds\; dt \quad (3.9)$$

and the corresponding discrete form is

$$u^a(x,y) = \sum_{I=1}^{NP} N_I(x,y)u_I\,, \quad (3.10)$$

where the interpolation function for node *I* is

$$N_I(x,y) = P^T(x,y)\; M^{-T}\; P(x_k - x_I, y_k - y_I)\; \psi(x_k - x_I, y_k - y_I)\; \Delta V_I. \quad (3.11)$$

3.2. Derivatives of Interpolation Functions

Due to the fact that the moment matrix (as in Equation 3.6) is constant, the derivatives of the interpolation functions involve only the differentiations of the polynomial basis.

For quadratic polynomial basis in two dimensions, $m = 6$, $P^T(x,y) = [1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2]$, then the derivatives of the interpolation functions (Equation 3.11) are

$$N_{I,\,x}\,(x,y) \;\; = [0 \quad 1 \quad 0 \quad 2x \quad y \quad 0]\; M^{-1}P(x_I, y_I)\; \psi(x_k - x_I, y_k - y_I)\; \Delta V_I \quad (3.12)$$

$$N_{I,\ xx}(x,y) = [0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0]\, M^{-1}P(x_I,y_I)\,\psi(x_k - x_I, y_k - y_I)\,\Delta V_I \tag{3.13}$$

$$N_{I,\ y}(x,y) = [0 \quad 0 \quad 1 \quad 0 \quad x \quad 2y]\, M^{-1}P(x_I,y_I)\,\psi(x_k - x_I, y_k - y_I)\,\Delta V_I \tag{3.14}$$

$$N_{I,\ yy}(x,y) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2]\, M^{-1}P(x_I,y_I)\,\psi(x_k - x_I, y_k - y_I)\,\Delta V_I \tag{3.15}$$

$$N_{I,\ xy}(x,y) = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0]\, M^{-1}P(x_I,y_I)\,\psi(x_k - x_I, y_k - y_I)\,\Delta V_I. \tag{3.16}$$

3.3. Formulation of Point Collocation Method (Pcm)

In FCM, the point collocation approach is adopted to discretize the governing equation with a set of collocation nodes, then the approximate solution is determined from the conditions that the governing equation is satisfied at the collocation nodes. In addition, the point collocation technique is capable of enforcing the boundary conditions exactly.

4. The Development of the Improved FCM

The development of the improved FCM starts with the discretization of the problem domain with a set of field nodes. Then, clouds with their associated center node are also defined for the problem domain. When the two sets of nodes are ready, a moment matrix is constructed for each cloud with the field nodes located in the vicinity of the cloud. With the moment matrices, the interpolation functions for all field nodes are ready for the next stage, which is the discretization of the governing equation. In this study, the set of nodes that are used to generate the set of linear system of equations, namely collocation nodes, includes the internal nodes and the boundary nodes. By point collocation scheme with boundary conditions, a set of linear system of equations is established and the nodal parameters are computed by solving the system of equations, and thus the approximate solutions are obtained.

4.1. Discretisation of Problem Domain

Consider a simple problem model of size $Lx \times Ly$. The problem domain is divided into N_{int} cells where $N_{int} = IntX \times IntY$. $IntX$ is the number of intervals along the x-axis whereas $IntY$ is the number of intervals along the y-axis. A field node is placed at the center of each cell (Figure 1(a)). Alternatively, field nodes can be placed randomly in the domain as depicted in Figure 1(b). Note that the field nodes in this study do not include boundary nodes.

4.2. Cloud Definition and Cloud Distribution

In this work, the cloud centers are independent of collocation nodes and distributed uniformly over the problem domain as shown in Figure 1(c). These clouds are overlapping and can be of a degree of density. However, the denser the cloud distribution, the computation time required will be increased as the total number of clouds will increase too. Since a collocation node $(x_c, y_c) \neq (x_k, y_k)$ where (x_k, y_k) is a cloud center, the computation work to determine the nearest cloud center for each collocation node has to be carried out ahead of time. Under this circumstance, a moment matrix associated with a cloud may be used more than once for the approximation of solutions. Thus, the computation time required can be optimized by adjusting the number of clouds defined in the problem domain.

In Figure 1(d), the cloud, Cld_1, is centred at (x_k, y_k) and the field nodes covered by the Cld_1 are $DP_1, DP_2, DP_3, DP_4, DP_5, DP_6$ and DP_7. Note that $DP_1$is the field node that is shared among Cld_1, Cld_2 and Cld_3.

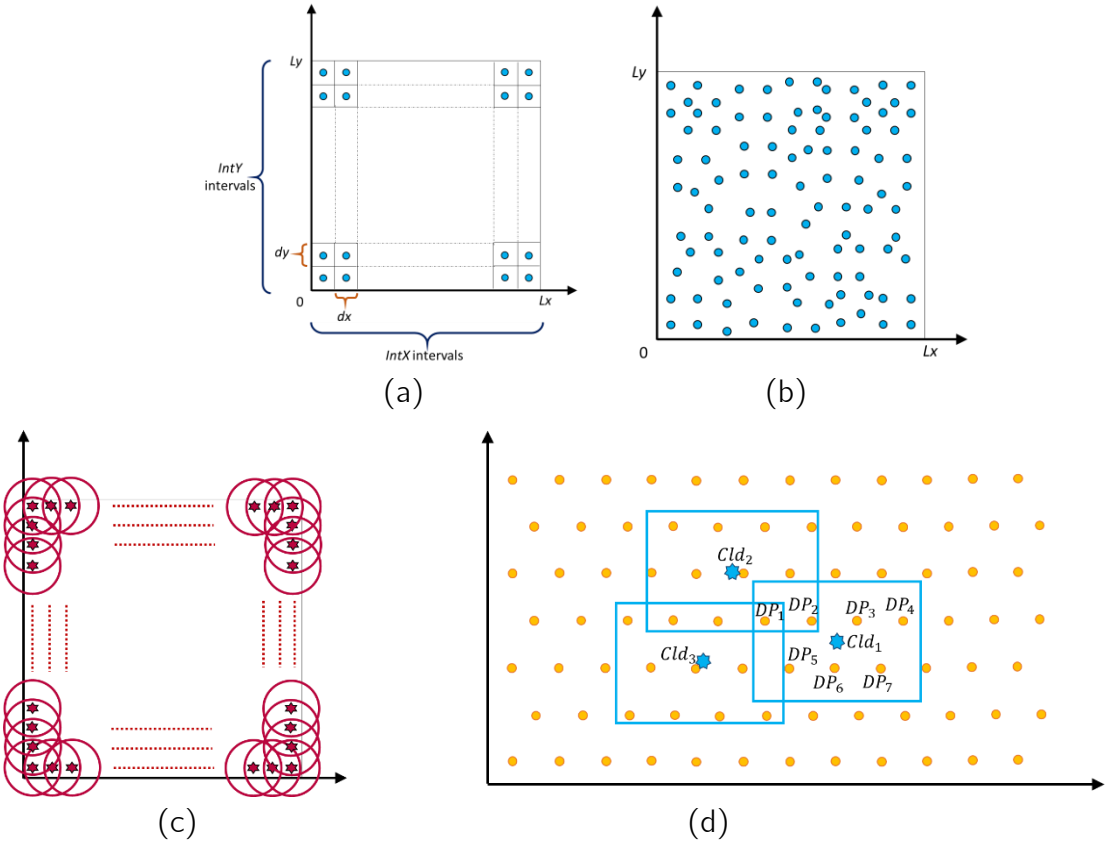


Figure 1. (a) & (b) Discretization of problem domain (c) Cloud distribution (d) Definition of cloud

Having overlapping clouds is one of the features or advantages of FCM, due to their localized interpolation domain as well as the independence of the moment matrix of one cloud from the others. Note that a moment matrix is generated for one cloud. Besides, the interpolation functions constructed by fixed reproducing kernel technique are satisfying the sum to unity property. Considering the consistency condition again. When $\ell = 1$, $p_1(x, y) = 1.0$, then substituting them into Equation 3.5 gives

$$\sum_{\mathrm{I}=1}^{NP} P^T(x_k - x_I, y_k - y_I)\mathrm{C}(x, y)\, \phi(x_k - x_I, y_k - y_I)\, \Delta V_I = 1.0\,. \tag{4.1}$$

Compare with the interpolation function defined in Equation 3.11 and we conclude that

$$\sum_{I=1}^{NP} N_I(x, y) = 1.0 \tag{4.2}$$

Theoretically, since the node of interest (x, y) can be at any point in the domain, the clouds can be centered at any location, as long as the node of interest is in the vicinity of the cloud. However, this may lead to two minor problems:

- multivalued interpolation functions
- singularity of moment matrices for the clouds which are near the boundaries

4.3. Multivalued Interpolation Functions

According to Aluru and Li ([32]), the multivalued interpolation functions can be avoided by centering the kernel at the node of interest. In another word, the node of interest (x, y) is also the cloud center node (x_k, y_k). This means that a list of nodes of interest or collocation nodes should be defined ahead of time before constructing the interpolation functions.

The more flexible way of defining cloud centers is by employing the improved FCM ([22]) so that with the adjustments to the interpolation functions, the cloud center nodes can be predefined and distributed uniformly over the problem domain. The approximate solution of any node of interest is computed with the adjusted interpolation functions of its nearest cloud. As a result, after the approximate solution is obtained, any new node of interest does not involve the generation of the

moment matrix and the construction of the interpolation functions for the new cloud, one only needs to find the nearest cloud to the node of interest for the construction of the corresponding interpolation functions to compute the approximate solution.

4.4. Singularity of Moment Matrices

A moment matrix may become singular if its associated cloud does not have a sufficient number of local support nodes. There are two reasons why a cloud might become singular. One of them is the size of the cloud is not large enough to enclose a sufficient number of field nodes. Another reason is the cloud is located near the boundaries where the cloud becomes incomplete. Hence, parameters are introduced to control the density of the clouds as well as the distance of clouds away from the boundaries.

For a better illustration, clouds of circle shape are used (as shown in Figure 2(a)). βR_x and βR_y are the distance between the boundaries of two adjacent overlapping clouds in x-direction and y-direction respectively, where β is the coefficient of overlapping of clouds and R_x and R_y are the radius of clouds in x and y directions. λ_x and λ_y are the distances between a boundary and the center node of the cloud which are near the boundary in x and y direction. μ_x and μ_y represent the distances between two adjacent clouds in x and y direction and are given by

$$\begin{aligned} \mu_x &= 2R_x - \beta R_x \\ \mu_y &= 2R_y - \beta R_y. \end{aligned} \tag{4.3}$$

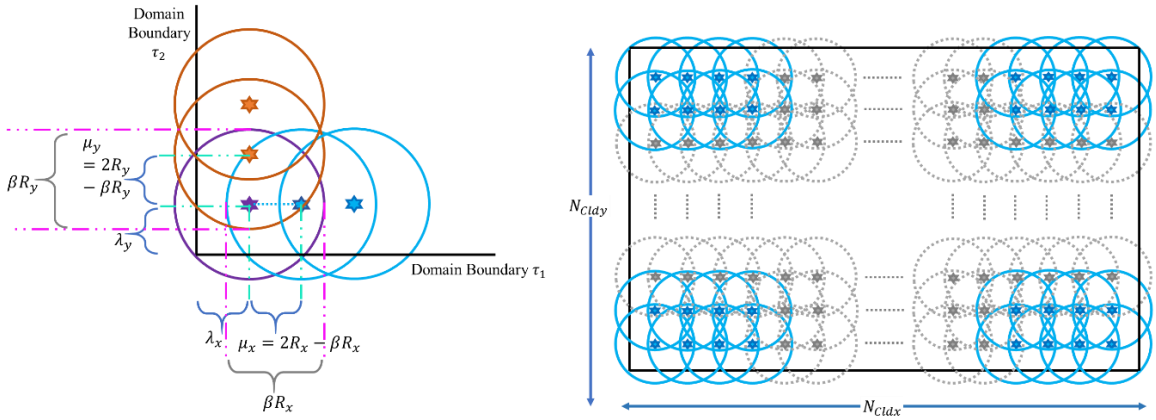
Let N_{Cldx} and N_{Cldy} be the unknown number of clouds defined in both directions. With the given λ_x, λ_y, β_x, β_y, N_{Cldx} and N_{Cldy} can be computed as follow:

$$\begin{aligned} N_{Cldx} &= \frac{L_x - 2\lambda_x}{R_x(2-\beta_x)} + 1 \\ N_{Cldy} &= \frac{L_y - 2\lambda_y}{R_y(2-\beta_y)} + 1. \end{aligned} \tag{4.4}$$

N_{Cldx} and N_{Cldy} are the smallest integers of the RHS of Equation 4.4. Thus, the total number of clouds defined in the domain is

$$N_{cld} = N_{Cldx} * N_{Cldy} \tag{4.5}$$

and the clouds are distributed as depicted in Figure 2(b).



(a) (b)

Figure 2. (a) Cloud density by parameters (b) Cloud distribution

Note that clouds, generally called local domains in other meshfree methods, are independent of nodes of interest or solution points. Since each cloud has its own set of interpolation functions, thus any node of interest, which is in the vicinity of a cloud, can utilize the corresponding interpolation functions to compute its approximate solution. However, in this study, the nearest cloud will be chosen for its interpolation for the reason of only the neighbouring field nodes are included for a better approximation of solution. Additionally, by having uniformly distributed clouds, the information carried by all field nodes is captured and stored in the clouds and these clouds are then be used to establish the set of system of equations

4.5. Discretization With Point Collocation Approach and Enhancement at the Boundary

In this study, a two-dimensional model problem is considered. The model consists of the governing differential equation that acts on the problem domain Ω surrounded by the boundary Γ_D and Γ_N which satisfy the Dirichlet boundary condition and the Neumann boundary condition respectively (as in Equation 4.6-4.8):

$$\mathcal{L}u^a(x_I, y_I) = f(x_I, y_I),\ I = 1, 2, \dots, N_{int} \quad in\ \Omega \tag{4.6}$$

$$u^a(x_I, y_I) = g(x_I, y_I),\ I = 1, 2, \dots, N_D \quad on\ \Gamma_D \tag{4.7}$$

$$\frac{\partial u^a}{\partial n}(x_I, y_I) = h(x_I, y_I),\ I = 1, 2, \dots, N_N \quad on\ \Gamma_N. \tag{4.8}$$

$\mathcal{L}$ is the differential operator and u^a is the approximate solution.

In point collocation method, the collocation nodes are comprised of N_{int} internal field nodes, N_D Dirichlet boundary nodes and N_N Neumann boundary nodes. Note that only the N_{int} internal field nodes are satisfying the governing equation. The N_D and N_N boundary nodes are satisfying their relevant boundary condition only. Hence, an approach to strengthen the ties between the governing equation and the $N_D + N_N$ boundary nodes is suggested, that is to have additional equations for boundary nodes to ensure that the boundary nodes are also satisfying the governing differential equation (Equation 4.6).

$$\mathcal{L}u^a(x_I, y_I) = f(x_I, y_I), \qquad I = 1, 2, \dots, N_D + N_N. \tag{4.9}$$

As a result, the modified system of equations established by Equation 4.6-4.9 in matrix form may consist of $N_{int} + 2(N_D + N_N)$ rows and N_{int} columns, which is an over-determined system of linear equations as shown below:

$$R'_{M\times N}\, u'_{N\times 1} = b'_{M\times 1}, \tag{4.10}$$

where M is the total number of rows and N is the total number of columns of the final stiffness matrix.

By applying the least square approach to the system of linear equations as shown in Equation 4.10, the following is obtained:

$$(R')^T_{N\times M}\, R'_{M\times N}\, u'_{N\times 1} = (R')^T_{N\times M}\, b'_{M\times 1} \tag{4.11}$$

After solving the final stiffness matrix, the unknown u' is determined. At this point, the approximate solution u^a for any node (x, y) in the problem domain can be computed with Equation 3.10 by using the moment matrix of the nearest cloud in the problem domain.

4.6. Node Refinement

The accuracy of solution for FCM can be improved further with node refinement. The node refinement mechanism adopted in this work is by splitting a field node as presented in Figure 4.

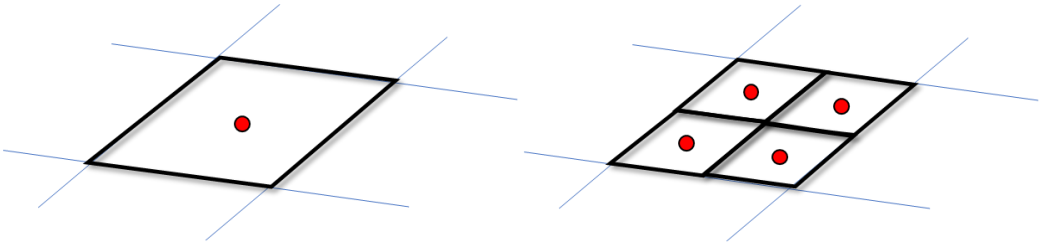


Figure 4. The splitting of a field node. (a) Before the split. (b) After the split.

The relative error for each collocation node is computed after the first approximate solution is obtained. If the relative error exceeds the given threshold, then the corresponding field nodes are marked. Each of the marked field nodes is then split into 4 field nodes. The newly added field nodes are used for the numerical analysis of FCM at the node refinement stage. By generating the affected moment matrices with the new sets of field nodes and deriving the new final stiffness matrix, the new approximate solution is obtained.

5. Numerical Examples

The FCM with proposed improvements is applied to a problem domain $\Omega = \{(x,y)|\ 0 < x < 1,\ 0 < y < 1\}$ with the Poisson equation as the governing differential equation and the given boundary conditions (as expressed in Equation 5.1-5.2).

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 4 - 2\omega\alpha^2\, e^{\left[-\alpha^2(y-c)^2\right]} + 4\omega\alpha^4(y-c)^2\, e^{-\alpha^2(y-c)^2} \tag{5.1}$$

$$\begin{cases} u|_{x=0} = y^2 + \omega e^{\left[-\alpha^2(y-c)^2\right]} \\ u|_{x=1.0} = 1.0 + y^2 + \omega e^{\left[-\alpha^2(y-c)^2\right]} \\ u_y\big|_{y=0} = 2\omega\alpha^2 c e^{-\alpha^2 c^2} \\ u_y\big|_{y=1.0} = 2.0 - 2\omega\alpha^2(1.0-c)e^{-\alpha^2(1.0-c)^2} \end{cases}. \tag{5.2}$$

Then, the cubic spline kernel function is defined as

$$\phi(x_k - x_I, y_k - y_I) = \frac{1}{d_x}\omega\left(\frac{x_k - x_I}{d_x}\right)\frac{1}{d_y}\omega\left(\frac{y_k - y_I}{d_y}\right), \tag{5.3}$$

where

$$\omega(z)=\begin{cases}0, & z<-2\\ \frac{1}{6}(z+2)^3, & -2\le z\le -1,\\ \frac{2}{3}-z^2\left(1+\frac{z}{2}\right), & -1\le z\le 0,\\ \frac{2}{3}-z^2\left(1-\frac{z}{2}\right), & 0\le z\le 1,\\ -\frac{1}{6}(z-2)^3, & 1\le z\le 2,\\ 0, & z>2\end{cases} \tag{5.4}$$

and $z_I=\frac{x_k-x_I}{d_x}$ or $\frac{y_k-y_I}{d_y}$ and d_x and d_y is the cloud size in x and y direction respectively.

The exact solution for the Poisson equation is given by

$$u(x,y)=x^2+y^2+\omega e^{\left[-\alpha^2(y-c)^2\right]} \tag{5.5}$$

and the global error ([32]) are computed as

$$\varepsilon=\frac{1}{|u^e|_{max}}\sqrt{\frac{1}{NC}\sum_{I=1}^{NC}[u_I^e-u_I^a]^2}, \tag{5.6}$$

where NC is the total number of collocation nodes, u^e is the exact solution and u^a is the approximate solution.

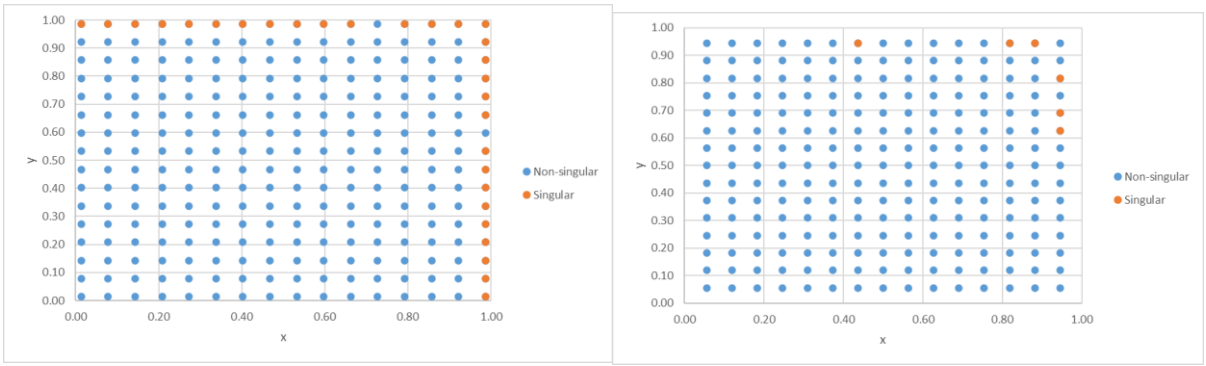
In this work, we have two models. Model A is of $\alpha=10$, $\omega=10$ and $c=0.5$ and Model B is having $\alpha=5$, $\omega=2$ and $c=0.5$. The cloud size is 2.4 Δx and 2.4 Δy. The two models use four sets of field nodes for their analysis, i.e. $NP=144,196,256$ and 324. In addition, the clouds are defined with various λ and β and the results obtained are observed.

The distance of a cloud center adjacent to a boundary from the boundary in x and y directions, λ_x and λ_y, are used to observe the singularity of moment matrices and the results are presented in Table 1.

Table 1. The relationship between λ_x or λ_y and the number of singular matrices observed with $\beta_x=\beta_y=1.5$ and the cloud size = $2.4\,\Delta x$.

Distance, λ (in Δx *or* Δy)	0.025	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3
No. of Singular moment matrix/total clouds	31/ 256	31/ 256	31/ 256	21/ 256	24/ 256	19/ 256	25/ 256	15/ 225	15/ 225	8/ 225	5/ 225	6/ 225	4/ 225	1/ 196	0/ 196

And the relationship of clouds versus the singularity of their moment matrices are depicted in Figure 5(a) and (b) as shown below:



(a) (b)

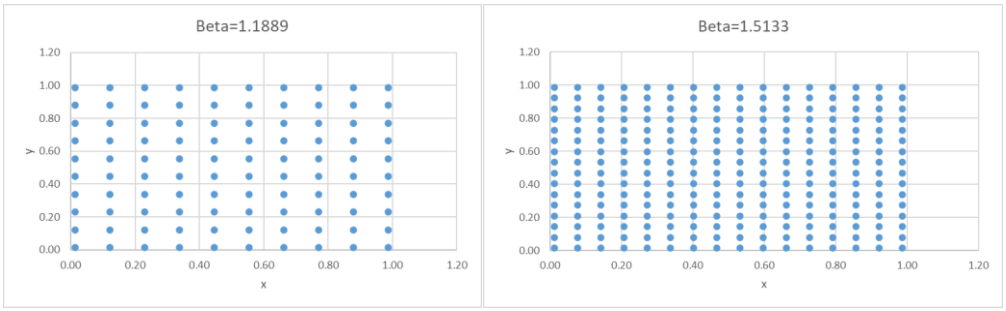
Figure 5. Clouds versus the Singularity of moment matrices. (a) $\lambda_x = 0.1\Delta x$ and $\lambda_y = 0.1\Delta y$. (b) $\lambda_x = 1.0\Delta x$ and $\lambda_y = 1.0\Delta y$.

From Figure 5, it is clearly shown that the larger the distance the cloud center, which is adjacent to the boundary, away from the boundary, its associated moment matrix is less likely to be singular.

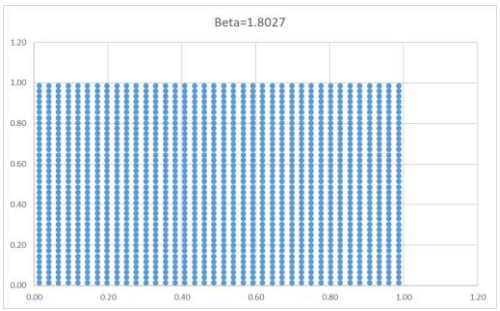
From Table 1, we can conclude that when the value of λ increases, the number of singular matrices will be reduced due to the fact that the cloud adjacent to the boundary could include more field nodes causing its associated moment matrix becomes less likely to be singular. However, if a cloud center which is adjacent to a boundary is too far away from a boundary could cause the information of field nodes which are near the boundaries are not captured into the system of equations and hence lead to less accurate solution. Therefore, it is important to choose a suitable λ for our numerical model. In this work, we have $\lambda_x = 0.1\Delta x$ and $\lambda_y = 0.1\Delta y$.

For cloud distribution, we adopted several β values and make comparison of the computed solutions. The cloud distribution of β = 1.19, 1.51 and 1.80 and the number of clouds defined are 100, 256 and 1444 (as shown in Figure 6).

Numerical results are computed for several βs and global errors, ϵ, are obtained using Equation 5.6. Graphs are plotted to describe the relationships between the global errors and the cloud density, β (as shown in Figure 7(a) for Model A and Figure 7(b) and 7(c) are for Model B).

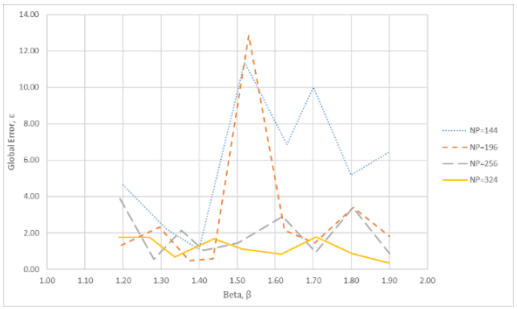


(a) (b)

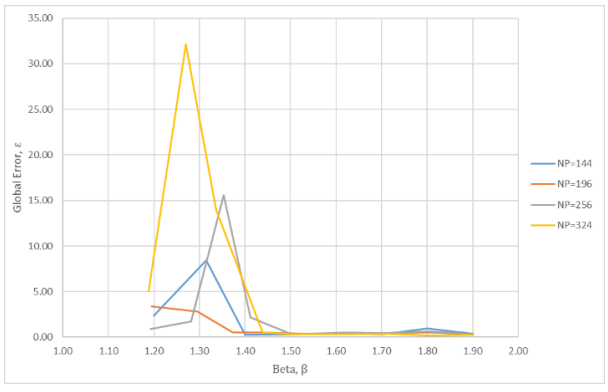


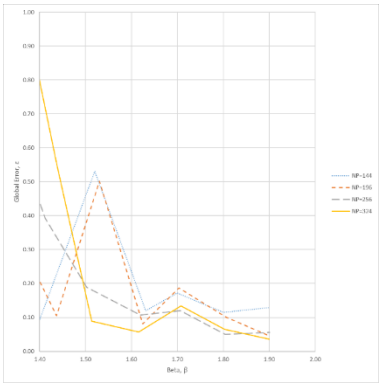
(c)

Figure 6. Cloud distribution. (a) $\beta = 1.19$ (b) $\beta = 1.51$ (c) $\beta = 1.80$



(a)





(b) (c)

Figure 7. The global error for FCM with various β. (a) Model A (b) & (c) Model B

Figure 7(c) is the enlarged version of Figure 7(b) for model B. From the numerical results, it is obvious that the computed solutions are more stable and converge slowly when β increases with more clouds being defined in the domain. For Model A which exhibits high gradient solution, the global error tends to become stable as the number of field nodes increases and the density of cloud may not play a significant role in the performance of the model. In this case, we may consider increasing the number of field nodes. Then, we may have larger size of clouds than $2.4\Delta x$ or $2.4\Delta y$ or denser clouds to capture more details from the high gradient region. On the contrary, Modal B has a smoother gradient of solution and the numerical results have shown less oscillation and converged slowly when β approaches 1.9. The numerical results prove that the more clouds are defined, the better the accuracy of solutions. This is because a collocation node could find the nearest cloud which contains the information of its nearest field nodes while retaining the reusability of moment matrices. Note that a cloud can be used more than once for approximating solutions. This feature provides the flexibility of allocating available resources while obtaining the desired accuracy of solution. In addition, the results also have shown that the shorter the distance between the cloud center and the node of interest or collocation node, the better the accuracy of solution as the longest distance between a node of interest and its associated nearest cloud center for a β cloud distribution, will decrease as β increases.

Additionally, the numerical model is modified to adopt the enhancement at the boundaries. For Model B with $\beta = 1.8$, the numerical results obtained are analyzed and presented in Figure 8. Figure 8 has shown that the numerical model with enhancement at the boundaries produces a smoother approximate solution compared with the solution computed without enhancement at the boundary. The numerical results have proven that the enhancement at the boundaries can lead to a less oscillated solution.

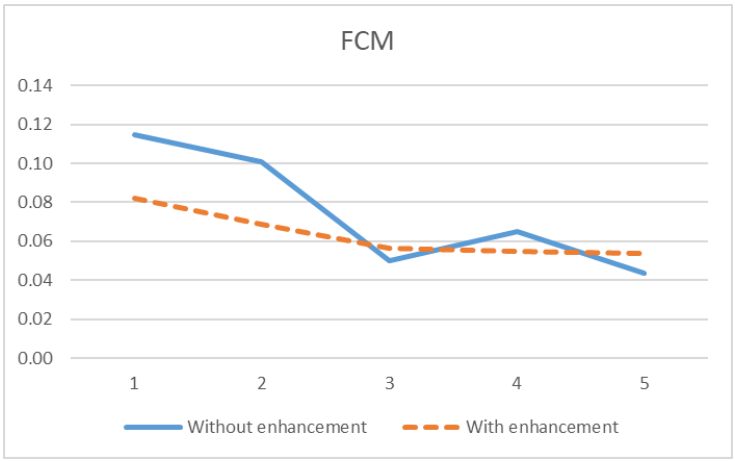
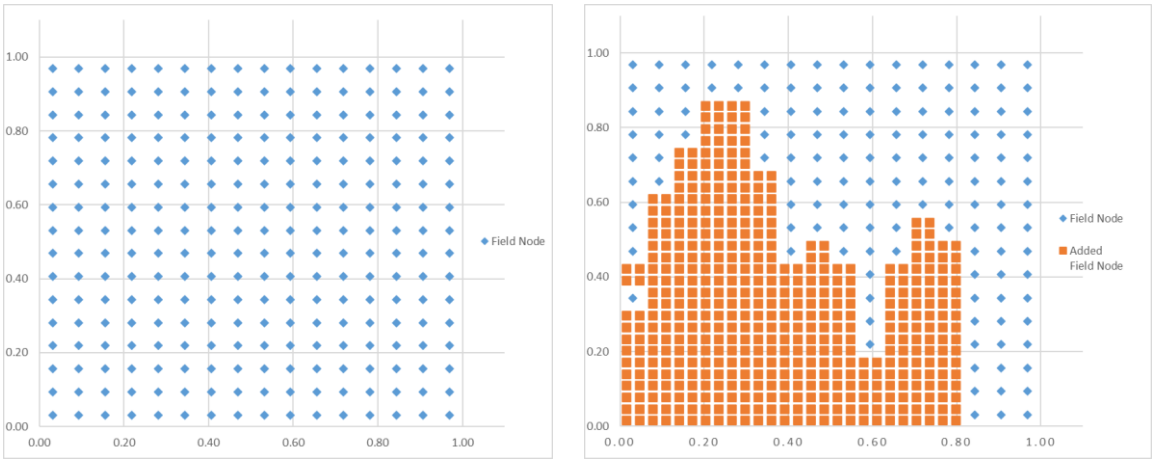


Figure 8. The global errors for FCM with and without the enhancement at the boundary (NP=144, 196, 256, 324 and 400)

Besides the enhancement at the boundaries, the approximate solution can be improved further with node refinement process. By employing the node refinement mechanism suggested in Section 4.6, after going through the splitting process at the threshold = $\frac{|u^a - u^e|}{u^e}$, where u^a is the approximate solution and u^e is the exact solution, the initial set of field nodes and the new set of field nodes during the node refinement stage are depicted in Figure 9(a) and 9(b) respectively.



(a) (b)

Figure 9. (a) Field nodes at initial stage ($NP = 256$). (b) Field nodes at node refinement stage ($NP = 604$)

The numerical model with enhancement to the boundaries proceeds to node refinement stage and the results obtained are analyzed and presented in Table 2.

Table 2. The global errors for the boundary enhancement Model B (with $\alpha = 8.0, \omega = 2.0$) before and after the node refinement process.

Global Error after the enhancement at the boundaries but before the node refinement	Global Error after the enhancement at the boundaries and node refinement
$NP = 144$, $\varepsilon = 0.3731$	$NP = 321$, $\varepsilon = 0.3570$
$NP = 196$, $\varepsilon = 0.3197$	$NP = 370$, $\varepsilon = 0.3429$
$NP = 256$, $\varepsilon = 0.3494$	$NP = 604$, $\varepsilon = 0.2320$
$NP = 324$, $\varepsilon = 0.2971$	$NP = 609$, $\varepsilon = 0.2946$
$NP = 400$, $\varepsilon = 0.2819$	$NP = 763$, $\varepsilon = 0.2499$

The numerical results have shown improvement in the accuracy of the approximate solution obtained after the node refinement stage (except for $NP = 196$). Hence, the improvements for FCM suggested in this research are effective in producing a better accuracy of solution.

6. Conclusion

In this paper, we have introduced a way of defining uniformly distributed cloud using parameters λ and β. Each collocation node is assigned to its nearest cloud for the computation of its approximate solution. We have studied the role of λ and β in detail by using a numerical example. The numerical results reveal the relationships of λ and β and the accuracy of solution. In addition, the suggested enhancement at the boundaries and the node refinement mechanism are also proven can produce a more stable and accurate of approximate solution.

Acknowledgement: The authors wish to thank the Malaysian Ministry of Education (MOE), Universiti Teknologi Malaysia (UTM) and Research Management Centre (RMC) for financial sponsorship to this work through grant funding number R.J130000.2654.17J37 UTM TIER 2.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] J.S. Chen, M. Hillman, S.W. Chi, Meshfree Methods: Progress Made After 20 Years, J. Eng. Mech. 143 (2017), 04017001. https://doi.org/10.1061/(ASCE)EM.1943-7889.0001176.

[2] S.D. Daxini, J.M. Prajapati, A Review on Recent Contribution of Meshfree Methods to Structure and Fracture Mechanics Applications, Sci. World J. 2014 (2014), 247172. https://doi.org/10.1155/2014/247172.

[3] S. Garg, M. Pant, Meshfree Methods: A Comprehensive Review of Applications, Int. J. Comput. Methods. 15 (2018), 1830001. https://doi.org/10.1142/S0219876218300015.

[4] Q. Duan, X. Gao, B. Wang, X. Li, H. Zhang, T. Belytschko, Y. Shao, Consistent Element-Free Galerkin Method, Int. J. Numer. Methods Eng. 99 (2014), 79-101. https://doi.org/10.1002/nme.4661.

[5] Q. Duan, T. Belytschko, Gradient and Dilatational Stabilizations for Stress-Point Integration in the Element-Free Galerkin Method, Int. J. Numer. Methods Eng. 77 (2009), 776-798. https://doi.org/10.1002/nme.2432.

[6] M. Najafi, V. Enjilela, Natural Convection Heat Transfer at High Rayleigh Numbers–Extended Meshless Local Petrov–Galerkin (MLPG) Primitive Variable Method, Eng. Anal. Bound. Elements. 44 (2014), 170-184. https://doi.org/10.1016/j.enganabound.2014.01.022.

[7] R. Singh, K.M. Singh, Interpolating Meshless Local Petrov-Galerkin Method for Steady State Heat Conduction Problem, Eng. Anal. Bound. Elements. 101 (2019), 56-66. https://doi.org/10.1016/j.enganabound.2018.12.012.

[8] R. Divya, V. Sriram, K. Murali, Wave-Vegetation Interaction Using Improved Meshless Local Petrov Galerkin Method, Appl. Ocean Res. 101 (2020), 102116. https://doi.org/10.1016/j.apor.2020.102116.

[9] S.S. Mulay, H. Li, S. See, On the Development of Adaptive Random Differential Quadrature Method with an Error Recovery Technique and its Application in the Locally High Gradient Problems, Comput. Mech. 45 (2010), 467-493. https://doi.org/10.1007/s00466-010-0468-2.

[10] C.W. Bert, M. Malik, Differential Quadrature Method in Computational Mechanics: A Review, Appl. Mech. Rev. 49 (1996), 1–28. https://doi.org/10.1115/1.3101882.

[11] X. Liang, T. Wang, D. Huang, Z. Liu, R. Zhu, C. Wang, An Improved RBF Based Differential Quadrature Method, Eng. Anal. Bound. Elements. 135 (2022), 299–314. https://doi.org/10.1016/j.enganabound.2021.11.023.

[12] T. Liu, J. Yu Ding, X. Yu Xu, Differential Quadrature Method for Partial Differential Dynamic Equations of Beam–Ring Structure, AIAA J. 60 (2022), 2542-2554. https://doi.org/10.2514/1.J061113.

[13] F.X. Sun, J.F. Wang, Y.M. Cheng, An Improved Interpolating Element-Free Galerkin Method for Elasticity, Chin. Phys. B. 22 (2013), 43–50. https://doi.org/10.1088/1674-1056/22/12/120203.

[14] S. Wu, Y. Xiang, B. Liu, G. Li, A Weak-Form Interpolation Meshfree Method for Computing Underwater Acoustic Radiation, Ocean Eng. 233 (2021), 109105. https://doi.org/10.1016/j.oceaneng.2021.109105.

[15] S. R. Idelsohn, E. Onate, N. Calvo, F. Del Pin, The Meshless Finite Element Method, Int. J. Numer. Methods Eng. 58 (2003), 893–912. https://doi.org/10.1002/nme.798.

[16] Y. Chai, C. Cheng, W. Li, Y. Huang, A Hybrid Finite Element-Meshfree Method Based on Partition of Unity for Transient Wave Propagation Problems in Homogeneous and Inhomogeneous Media, Appl. Math. Model. 85 (2020), 192–209. https://doi.org/10.1016/j.apm.2020.03.026.

[17] H. Li, Q. Zhang, A Meshfree Finite Volume Method With Optimal Numerical Integration and Direct Imposition of Essential Boundary Conditions, Appl. Numer. Math. 153 (2020), 98-113. https://doi.org/10.1016/j.apnum.2020.02.005.

[18] I. Jaworska and S. Milewski, On Two-Scale Analysis of Heterogeneous Materials by Means of the Meshless Finite Difference Method, Int. J. Multiscale Comput. Eng. 14 (2016), 113–134. https://doi.org/10.1615/IntJMultCompEng.2016014435.

[19] Y. Jiang, Algebraic-Volume Meshfree Method for Application in Finite Volume Solver, Comput. Methods Appl. Mech. Eng. 355 (2019), 44–66. https://doi.org/10.1016/j.cma.2019.05.048.

[20] Y. Jiang, General Mesh Method: A Unified Numerical Scheme, Comput. Methods Appl. Mech. Eng. 369 (2020), 1–28. https://doi.org/10.1016/j.cma.2020.113049.

[21] Y. Gu, L.C. Zhang, Coupling of the Meshfree and Finite Element Methods for Determination of the Crack Tip Fields, Eng. Fracture Mech. 75 (2008), 986-1004. https://doi.org/10.1016/j.engfracmech.2007.05.003.

[22] M. L. Oh and S. H. Yeak, A Hybrid Multiscale Finite Cloud Method and Finite Volume Method in Solving High Gradient Problem, Int. J. Comput. Methods, 19 (2022), 2250002. https://doi.org/10.1142/S0219876222500025.

[23] D.R. Burke, T.J. Smy, Optical Mode Solving for Complex Waveguides Using a Finite Cloud Method, Optics Express. 20 (2012), 17783-17796. https://doi.org/10.1364/OE.20.017783.

[24] D.R. Burke, T.J. Smy, Thermal Models for Optical Circuit Simulation Using a Finite Cloud Method and Model Reduction Techniques, IEEE Trans. Computer-Aided Design Integrated Circuits Syst. 32 (2013), 1177-1186. https://doi.org/10.1109/TCAD.2013.2253835.

[25] D.R. Burke, A Meshless Approach to Solving Partial Differential Equations Using the Finite Cloud Method for the Purposes of Computer Aided Design, Doctoral Dissertation, Carleton University, 2013.

[26] S.K. De, N.R. Aluru, A Chemo-Electro-Mechanical Mathematical Model for Simulation of pH Sensitive Hydrogels, Mech. Mater. 36 (2004), 395–410. https://doi.org/10.1016/S0167-6636(03)00067-X.

[27] X.Z. Jin, G. Li, N.R. Aluru, On the Equivalence between Least-Squares and Kernel Approximations in Meshless Methods, Computer Model. Eng. Sci. 2 (2001), 447-462.

[28] X.Z. Jin, G. Li, N.R. Aluru, Positivity Conditions in Meshless Collocation Methods, Computer Methods Appl. Mech. Eng. 193 (2004), 1171-1202. https://doi.org/10.1016/j.cma.2003.12.013.

[29] X.Z. Jin, G. Li, N.R. Aluru, New Approximations and Collocation Schemes in the Finite Cloud Method, Comput. Struct. 83 (2005), 1366-85. https://doi.org/10.1016/j.compstruc.2004.08.030.

[30] W.X. Chan, H. Son, Y.J. Yoon, Computational Efficiency of Meshfree Methods With Local-Coordinates Algorithm, Int. J. Precis. Eng. Manuf. 16 (2015), 547–556. https://doi.org/10.1007/s12541-015-0074-5.

[31] G.R. Liu, Y.T. Gu, A Point Interpolation Method, In: Proceedings of the Fourth Asia-Pacific Conference on Computational Mechanics, Singapore. (1999), 1009-1014.

[32] N.R. Aluru, G. Li, Finite Cloud Method: A True Meshless Technique Based on a Fixed Reproducing Kernel Approximation, Int. J. Numer. Meth. Eng. 50 (2001), 2373-2410. https://doi.org/10.1002/nme.124.