1. Introduction

The work will encompass dynamic and large deformation problems which is always been boasted as chief characteristic of Meshfree Methods.

2. Challenges

The work be challenging in time consumed in coding

3.1 EFGM

The element free Galerkin method (EFGM) has been considered a Meshfree method because it requires only a set of nodes over the given domain along with boundary description to construct the approximation function. Therefore, there is no need for element and element connectivity data like FEM. In this method, both trial and test functions are constructed from the same space using moving least square (MLS) approximants.

The MLS approximant requires only a set of nodes for its construction and is made up of three components: a compact support weight function associated with each node, a polynomial basis function and a set of coefficients that depends on node position. The support of the weight function defines the nodal domain of influence, over which a particular node contributes to the approximation. The overlap of the node's domain of influence defines the nodal connectivity. One useful property of MLS approximation is that their continuity is governed by the continuity of weight functions. Therefore, a highly continuous approximation function can be generated by an appropriate choice of weight function.

3.2 Moving least square (MLS) approximations

In EFGM, the field variable u is approximated by MLS approximation, $u^h(\mathbf{x})$ from Belytschko *et al.* [15] which is given by:

$$u^h(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})\, a_j(\mathbf{x}) \equiv \mathbf{p}^T(\mathbf{x})\,\mathbf{a}(\mathbf{x}) \qquad (3.1)$$

Where, $\mathbf{p(x)}$ is a vector of complete basis functions (usually polynomial) is given as:

$$\mathbf{p}^T(\mathbf{x}) = [1,\ x,\ y,\ z,\ xy,\ yz,\ zx, \ldots\ x^{k'},\ y^{k'},\ z^{k'}] \qquad (3.2)$$

and $\mathbf{a(x)}$ is a vector of unknown coefficients

$$\mathbf{a}^T(\mathbf{x}) = [a_1(\mathbf{x}),\ a_2(\mathbf{x}),\ a_3(\mathbf{x}),\ \ldots\ a_m(\mathbf{x})] \tag{3.3}$$

Where, $\mathbf{x}^T = [x \quad y \quad z]$, k' is degree of the polynomial and m is the number of terms in the basis.

Some complete polynomial basis functions and corresponding coefficient vectors are given as

1-D: Linear basis

$$\mathbf{p}^T(\mathbf{x}) = [1, x], \qquad (m = 2, \text{ linear}) \tag{3.4}$$

$$\mathbf{a}^T(\mathbf{x}) = [a_1(x),\ a_2(x)] \tag{3.5}$$

Quadratic basis

$$\mathbf{p}^T(\mathbf{x}) = [1,\ x,\ x^2], \qquad (m = 3, \text{ quadratic}) \tag{3.6}$$

$$\mathbf{a}^T(\mathbf{x}) = [a_1(x),\ a_2(x),\ a_3(x)] \tag{3.7}$$

2-D: Linear basis

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y] \qquad (m = 3, \text{ linear}) \tag{3.8}$$

$$\mathbf{a}^T(\mathbf{x}) = [a_1(\mathbf{x}),\ a_2(\mathbf{x}),\ a_3(\mathbf{x})] \tag{3.9}$$

Quadratic basis

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y, xy, x^2, \mathrm{y}^2], \qquad (m = 6, \text{ quadratic}) \tag{3.10}$$

$$\mathbf{a}^T(\mathbf{x}) = [a_1(\mathbf{x}),\ a_2(\mathbf{x}),\ a_3(\mathbf{x}), a_4(\mathbf{x}),\ a_5(\mathbf{x}),\ a_6(\mathbf{x})] \tag{3.11}$$

3-D: Linear basis

$$\mathbf{p}^T(\mathbf{x}) = [1 \quad x \quad y \quad z] \qquad (m = 4, \text{ linear}) \tag{3.12}$$

$$\mathbf{a}^T(\mathbf{x}) = [a_1(\mathbf{x}),\ a_2(\mathbf{x}),\ a_3(\mathbf{x}), a_4(\mathbf{x})] \tag{3.13}$$

Quadratic basis

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y, z, xy, yz, zx, x^2, \mathrm{y}^2, \mathrm{z}^2] \qquad (m = 10, \text{ quadratic}) \tag{3.14}$$

$$\mathbf{a}^T(\mathbf{x}) = [a_1(\mathbf{x}),\ a_2(\mathbf{x}),\ a_3(\mathbf{x}), a_4(\mathbf{x}), a_5(\mathbf{x}),\ a_6(\mathbf{x}),\ a_7(\mathbf{x}), a_8(\mathbf{x}), a_9(\mathbf{x}),\ a_{10}(\mathbf{x})] \tag{3.15}$$

The unknown coefficients $\mathbf{a}(\mathbf{x})$ are obtained by minimizing a weighted least square sum of the difference between local approximation, $u^h(\mathbf{x})$ and field function nodal parameters u_I. The weighted least square sum denoted by $L(\mathbf{x})$ can be written in following quadratic form:

$$L(\mathbf{x}) = \sum_{I=1}^{n} w(\mathbf{x} - \mathbf{x}_I)[\mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) - u_I]^2 \tag{3.16}$$

where, u_I is the nodal parameter associated with node I at $\mathbf{x} = \mathbf{x}_I$ but these are not the nodal values of $u^h(\mathbf{x} = \mathbf{x}_I)$ because $u^h(\mathbf{x})$ as an approximant, not an interpolant (the difference between u_I and $u^h(\mathbf{x} = \mathbf{x}_I)$ is shown in **Figure 3.1**), $w(\mathbf{x} - \mathbf{x}_I)$ is the weight function having compact support associated with node I, and n is the number of nodes with domain of influence containing the point $\mathbf{x}$, i.e. $w(\mathbf{x} - \mathbf{x}_I) \neq 0$ as shown in **Figure. 3.4**.

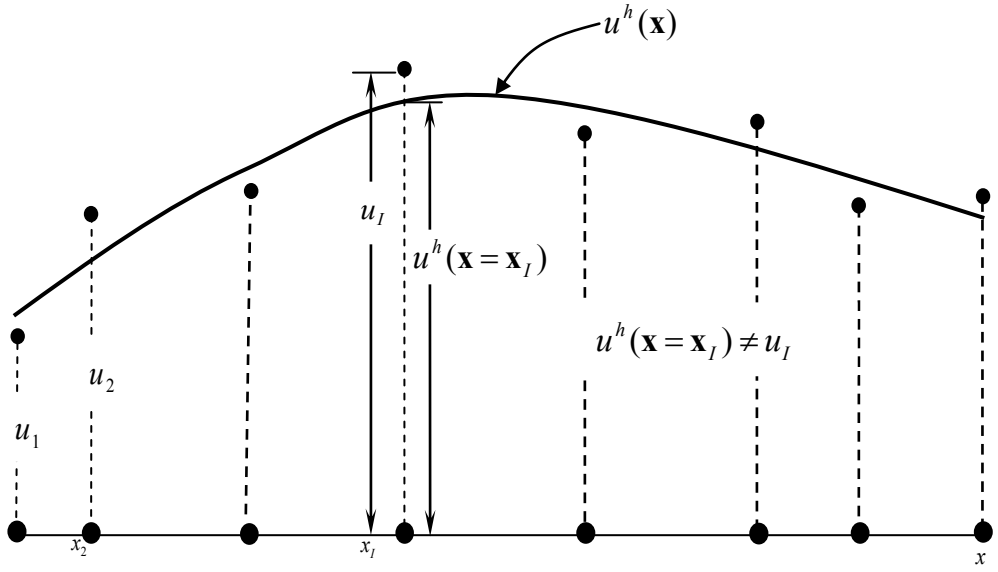


Figure. 3.4: Difference between u_I and $u^h(\mathbf{x})$

By setting $\frac{\partial L}{\partial \mathbf{a}} = 0$, a following set of linear equation is obtained

$$\mathbf{A}(\mathbf{x})\,\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\,\mathbf{u} \tag{3.17}$$

Or

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\,\mathbf{B}(\mathbf{x})\,\mathbf{u} \tag{3.18}$$

Where, $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are given as:

In 1-D

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{n} w(x - x_I)\,\mathbf{p}(x_I)\,\mathbf{p}^T(x_I) = w(x - x_1)\begin{bmatrix} 1 & x_1 \\ x_1 & {x_1}^2 \end{bmatrix} + \ldots\ldots\ldots + w(x - x_n)\begin{bmatrix} 1 & x_n \\ x_n & {x_n}^2 \end{bmatrix} \tag{3.19}$$

$$\mathbf{B(x)} = [w(x-x_1)\mathrm{p}(x_1),....., w(x-x_n)\mathrm{p}(x_n)] = \left\{ w(x-x_1)\begin{bmatrix}1\\x_1\end{bmatrix},....., w(x-x_n)\begin{bmatrix}1\\x_n\end{bmatrix}\right\} \quad (3.20)$$

In 2-D

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{n} w(\mathbf{x}-\mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I)$$

$$= w(\mathbf{x}-\mathbf{x_1})\begin{bmatrix}1 & x_1 & y_1\\ x_1 & x_1^2 & x_1y_1\\ y_1 & x_1y_1 & y_1^2\end{bmatrix} + w(\mathbf{x}-\mathbf{x_2})\begin{bmatrix}1 & x_2 & y_2\\ x_2 & x_2^2 & x_2y_2\\ y_2 & x_2y_2 & y_2^2\end{bmatrix} + ...$$
$$+ w(\mathbf{x}-\mathbf{x}_n)\begin{bmatrix}1 & x_n & y_n\\ x_n & x_n^2 & x_ny_n\\ y_n & x_ny_n & y_n^2\end{bmatrix} \quad (3.21)$$

$$\mathbf{B}(\mathbf{x}) = \{w(\mathbf{x}-\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1), w(\mathbf{x}-\mathbf{x}_2)\mathbf{p}(\mathbf{x}_2),..... ...., w(\mathbf{x}-\mathbf{x}_n)\mathbf{p}(\mathbf{x}_n)\}$$

$$= \left\{ w(\mathbf{x}-\mathbf{x}_1)\begin{bmatrix}1\\x_1\\y_1\end{bmatrix}, w(\mathbf{x}-\mathbf{x}_2)\begin{bmatrix}1\\x_2\\y_2\end{bmatrix},..... .... , w(\mathbf{x}-\mathbf{x}_n)\begin{bmatrix}1\\x_n\\y_n\end{bmatrix}\right\} \quad (3.22)$$

In 3-D

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{n} w(\mathbf{x}-\mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I)$$

$$= w(\mathbf{x}-\mathbf{x}_1)\begin{bmatrix}1 & x_1 & y_1 & z_1\\ x_1 & x_1^{\,2} & x_1y_1 & x_1z_1\\ y_1 & x_1y_1 & y_1^{\,2} & y_1z_1\\ z_1 & x_1z_1 & y_1z_1 & z_1^{\,2}\end{bmatrix} + w(\mathbf{x}-\mathbf{x_2})\begin{bmatrix}1 & x_2 & y_2 & z_2\\ x_2 & x_2^{\,2} & x_2y_2 & x_2z_2\\ y_2 & x_2y_2 & y_2^{\,2} & y_2z_2\\ z_2 & x_2z_2 & y_2z_2 & z_2^{\,2}\end{bmatrix} + ...$$
$$+ w(\mathbf{x}-\mathbf{x}_n)\begin{bmatrix}1 & x_n & y_n & z_n\\ x_n & x_n^{\,2} & x_ny_n & x_nz_n\\ y_n & x_ny_n & y_n^{\,2} & y_nz_n\\ z_n & x_nz_n & y_nz_n & z_n^{\,2}\end{bmatrix} \quad (3.23)$$

$$\mathbf{B}(\mathbf{x}) = \{w(\mathbf{x}-\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1), w(\mathbf{x}-\mathbf{x}_2)\mathbf{p}(\mathbf{x}_2),..... ...., w(\mathbf{x}-\mathbf{x}_n)\mathbf{p}(\mathbf{x}_n)\}$$

$$= \left\{ w(\mathbf{x}-\mathbf{x}_1)\begin{bmatrix}1\\x_1\\y_1\\z_1\end{bmatrix}, w(\mathbf{x}-\mathbf{x}_2)\begin{bmatrix}1\\x_2\\y_2\\z_2\end{bmatrix},..... ...., w(\mathbf{x}-\mathbf{x}_n)\begin{bmatrix}1\\x_n\\y_n\\z_n\end{bmatrix}\right\} \quad (3.24)$$

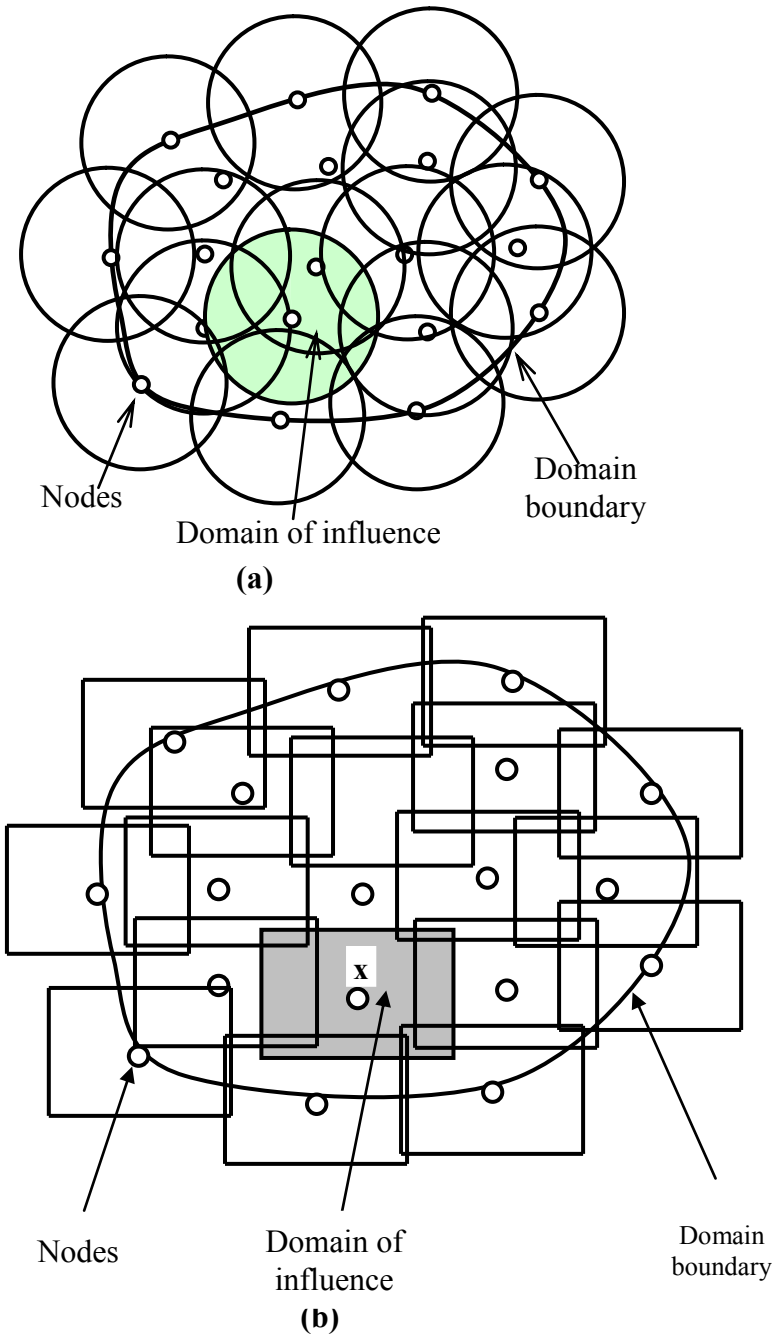


Figure. 3.5: A computational model for meshless method showing the boundary, nodes and domain of influence (a) Circular (b) Rectangular

By substituting **Eq. (3.6)** in **Eq. (3.1)**, the MLS approximation is obtained as:

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}) u_I = \mathbf{\Phi}^T(\mathbf{x})\mathbf{u} \tag{3.25}$$

Where,

$$\mathbf{\Phi}^T(\mathbf{x}) = \{\Phi_1(\mathbf{x}),\ \Phi_2(\mathbf{x}),\ \Phi_3(\mathbf{x}),\ \ldots\ \Phi_n(\mathbf{x})\} \tag{3.26}$$

$$\mathbf{u}^T = [u_1,\ u_2,\ u_3,\ \ldots\ u_n] \tag{3.27}$$

The mesh free shape function $\Phi_I(\mathbf{x})$ is defined as:

$$\Phi_I(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{jI} = \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B}_I \tag{3.28}$$

The linear consistency requirements for the shape function $\Phi_I(\mathbf{x})$ from Belytschko *et al.*[1] are given as:

$$\sum_{I=1}^{n} \Phi_I(\mathbf{x}) = 1 \tag{3.29a}$$

$$\sum_{I=1}^{n} \Phi_I(\mathbf{x})\, x_I = x \tag{3.29b}$$

$$\sum_{I=1}^{n} \Phi_I(\mathbf{x})\, y_I = y \tag{3.29c}$$

$$\sum_{I=1}^{n} \Phi_I(\mathbf{x})\, z_I = z \tag{3.29d}$$

The derivatives of MLS shape function are computed as:

$$\Phi_{I,\mathbf{x}}(\mathbf{x}) = (\mathbf{p}^T\mathbf{A}^{-1}\mathbf{B}_I)_{,\mathbf{x}} = \mathbf{p}^T{}_{,\mathbf{x}}\,\mathbf{A}^{-1}\,\mathbf{B}_I + \mathbf{p}^T(\mathbf{A}^{-1})_{,\mathbf{x}}\mathbf{B}_I + \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B}_{I,\mathbf{x}} \tag{3.30}$$

Where,

$$\mathbf{B}_{I,\mathbf{x}}(\mathbf{x}) = \frac{dw}{d\mathbf{x}}(\mathbf{x}-\mathbf{x}_I)\mathbf{p}(\mathbf{x}_I) \tag{3.31}$$

and $\mathbf{A}^{-1}{}_{,\mathbf{x}}$ is computed by

$$\mathbf{A}^{-1}{}_{,\mathbf{x}} = -\mathbf{A}^{-1}\mathbf{A}_{,\mathbf{x}}\mathbf{A}^{-1} \tag{3.32}$$

Where,

$$\mathbf{A}_{,\mathbf{x}} = \sum_{I=1}^{n} \frac{dw}{d\mathbf{x}}(\mathbf{x}-\mathbf{x}_I)\,\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I) \tag{3.33}$$

3.3 Efficient shape function calculation

To compute the mesh free shape functions Φ_I, it is necessary to calculate $\mathbf{A}^{-1}$. In 1-D problems, this operation of inverting the matrix is not very difficult but in 2-D and 3-D problems, this exercise becomes very much expensive from computational point of view. To overcome this situation, Dolbow and Belytschko [2] proposed a computationally inexpensive alternative approach. This approach involves the **LU** decomposition of the **A** matrix. The shape function is given as:

$$\Phi_I(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}_I(\mathbf{x}) = \gamma^T(\mathbf{x})\mathbf{B}_I(\mathbf{x}) \tag{3.34}$$

Where,

$$\gamma^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x}). \tag{3.35}$$

This leads to the relationship

$$\mathbf{A}(\mathbf{x})\gamma(\mathbf{x}) = \mathbf{p}(\mathbf{x}) \tag{3.36}$$

The vector $\gamma(\mathbf{x})$ is to be calculated using **LU** decomposition of the matrix $\mathbf{A}$ followed by back substitution.

The partial derivatives of $\gamma(\mathbf{x})$ can be recursively calculated as:

$$\mathbf{A}(\mathbf{x})\gamma_{,x}(\mathbf{x}) = \mathbf{p}_{,x}(\mathbf{x}) - \mathbf{A}_{,x}(\mathbf{x})\gamma(\mathbf{x}) \tag{3.37}$$

$$\mathbf{A}(\mathbf{x})\gamma_{,y}(\mathbf{x}) = \mathbf{p}_{,y}(\mathbf{x}) - \mathbf{A}_{,y}(\mathbf{x})\gamma(\mathbf{x}) \tag{3.38}$$

$$\mathbf{A}(\mathbf{x})\gamma_{,z}(\mathbf{x}) = \mathbf{p}_{,z}(\mathbf{x}) - \mathbf{A}_{,z}(\mathbf{x})\gamma(\mathbf{x}) \tag{3.39}$$

The derivatives of shape function are given as:

$$\Phi_{I,x}(\mathbf{x}) = \gamma^T{}_{,x}(\mathbf{x})\mathbf{B}_I(\mathbf{x}) + \gamma^T(\mathbf{x})\mathbf{B}_{I,x}(\mathbf{x}) \tag{3.40}$$

$$\Phi_{I,y}(\mathbf{x}) = \gamma^T{}_{,y}(\mathbf{x})\mathbf{B}_I(\mathbf{x}) + \gamma^T(\mathbf{x})\mathbf{B}_{I,y}(\mathbf{x}) \tag{3.41}$$

$$\Phi_{I,z}(\mathbf{x}) = \gamma^T{}_{,z}(\mathbf{x})\mathbf{B}_I(\mathbf{x}) + \gamma^T(\mathbf{x})\mathbf{B}_{I,z}(\mathbf{x}) \tag{3.42}$$

3.4 Weight function

The choice of weight function $w(\mathbf{x}-\mathbf{x}_I)$ affects the resulting approximation $u^h(\mathbf{x}_I)$ in EFG method. Therefore, the selection of appropriate weight function is essential in these methods. The weight function is non-zero only over a small neighborhood of a node $\mathbf{x}_I$, called the support or domain of influence of node I. The smoothness and continuity of the shape function $\Phi_I(\mathbf{x})$ depends on the smoothness and continuity of the weight function $w(\mathbf{x}-\mathbf{x}_I)$. If weight function is C^1 continuous then shape function will also have C^1 continuity (This property is represented in **Figure 3.6**).

A weight function must satisfy the following conditions

- It must be positive, continuous and differentiable in the domain of influence.
- It should decrease in magnitude as the distance from $\mathbf{x}$ to $\mathbf{x}_I$ increases so that local character of MLS approximation can be maintained.
- It should be zero outside the domain of influence.

- It should have a relatively larger value for a node, which is closer to the evaluation point than those of far nodes.
- The nodes in the domain of influence should not be collinear (except 1-D) and the number of nodes must be larger than the number of terms in the basis function $(n > m)$.

The different weight functions used in the present analysis are written as a function of normalized radius $\bar{r}$ as:

The cubic spline (C.S.) weight function

$$w(\mathbf{x}-\mathbf{x}_I) = w(\bar{r}) = \begin{cases} \frac{2}{3} - 4\bar{r}^2 + 4\bar{r}^3 & \bar{r} \le \frac{1}{2} \\ \frac{4}{3} - 4\bar{r} + 4\bar{r}^2 - \frac{4}{3}\bar{r}^3 & \frac{1}{2} < \bar{r} \le 1 \\ 0 & \bar{r} > 1 \end{cases} \tag{3.43a}$$

The quadratic spline (Q.S.) weight function

$$w(\mathbf{x}-\mathbf{x}_I) = w(\bar{r}) = \begin{cases} 1 - 6\bar{r}^2 + 8\bar{r}^3 - 3\bar{r}^4 & 0 \le \bar{r} \le 1 \\ 0 & \bar{r} > 1 \end{cases} \tag{3.43b}$$

Where, $(\bar{r}) = \dfrac{\| \mathbf{x} - \mathbf{x}_I \|}{\mathrm{d}_{mI}}$

$\|\mathbf{x}-\mathbf{x}_I\|$ is the distance from a sampling point $\mathbf{x}$ to a node $\mathbf{x}_I$ and d_{mI} is the domain of influence of node I.

In 3-D

$$(\bar{r}_x) = \frac{\|\mathbf{x}-\mathbf{x}_I\|}{d_{mxI}} \tag{3.44a}$$

$$(\bar{r}_y) = \frac{\|\mathbf{x}-\mathbf{x}_I\|}{d_{myI}} \tag{3.44b}$$

$$(\bar{r}_z) = \frac{\|\mathbf{x}-\mathbf{x}_I\|}{d_{mzI}} \tag{3.44c}$$

and

$$d_{mxI} = d_{\max} c_{xI} \tag{3.45a}$$

$$d_{myI} = d_{\max} c_{yI} \tag{3.45b}$$

$$d_{mzI} = d_{\max} c_{zI} \tag{3.45c}$$

$d_{\max}$ = scaling parameter which defines size of the domain of influence and c_{xI}, c_{yI} & c_{zI} at node I are the distances to the nearest neighbors. d_{mxI} , d_{myI} and d_{mzI} are chosen such that the matrix is non-singular at every point in the domain.

The weight function at any given point is obtained as:

$$w(\mathbf{x}-\mathbf{x}_I) = w(\bar{r}_x)\,w(\bar{r}_y)\,w(\bar{r}_z) = w_x\, w_y\, w_z \tag{3.46}$$

Where, $w(\bar{r}_x)$, $w(\bar{r}_y)$ and $w(\bar{r}_z)$ can be calculated by replacing $\bar{r}$ by $\bar{r}_x$, $\bar{r}_y$ and $\bar{r}_z$ in the expression of $w(\bar{r})$.

The derivatives of the weight functions are calculated as:

$$w_{,x} = \frac{dw_x}{dx} w_y w_z \tag{3.47a}$$

$$w_{,y} = \frac{dw_y}{dy} w_x w_z \tag{3.47b}$$

$$w_{,z} = \frac{dw_z}{dz} w_x w_y \tag{3.47c}$$

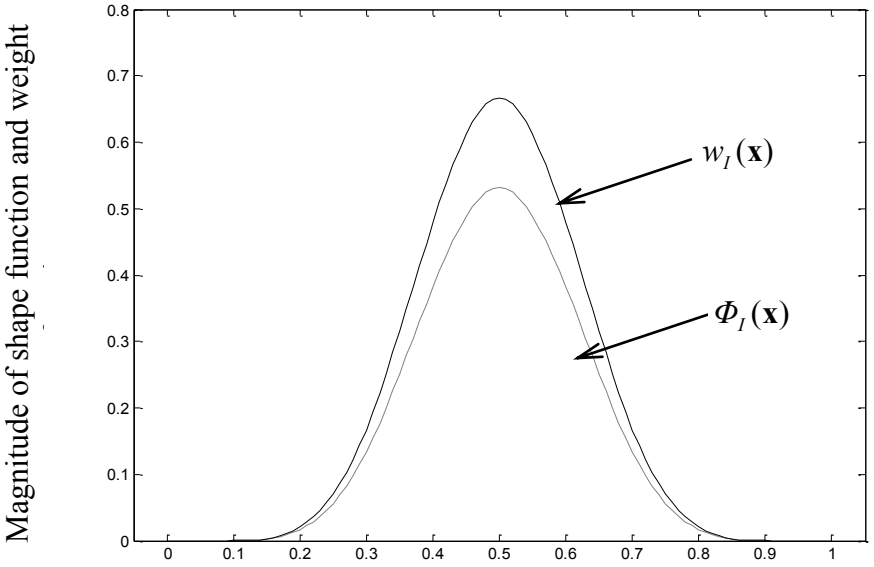


Figure. 3.6: A plot of weight function and corresponding shape function

3.5 Enforcement of essential boundary conditions

The proper imposition of essential boundary condition is quite difficult in EFGM since MLS approximation does not satisfy the Kronecker delta function property i.e. $\Phi_I(x_J) \neq \delta_{IJ}$. Many numerical techniques have been proposed to enforce the essential boundary conditions in EFGM. Lagrange multiplier method was first used by Belytschko *et al.*[3]. This technique

is quite accurate but it imposition loses the positive definite and bandedness properties of the system matrix. Lu *et al.* [4]proposed modified variational principle approach, in which Lagrange multipliers were replaced by their physical meaning. As a result, banded sets of equation were obtained but the results were not found as accurate as obtained by Lagrange multiplier approach. Another approach named as coupling with finite element method was proposed by Belytschko *et al.* [5] for the imposition of essential boundary conditions. In this approach, EFGM domain has been necklaced by FEM domain and then essential boundary conditions were applied in the same manner as in usual finite element methods. This method has simplified the enforcement of boundary conditions but the numerical integration became a more tedious job. In penalty approach, it is easy to enforce the essential boundary conditions and it gives discrete equations in a simple form similar to FEM. Although, system matrix obtained by this method is positive and possess bandedness property but improper selection of penalty parameter can lead to wrong results.

In the present work, Lagrange multiplier method will be utilized for the imposition of essential boundary condition due to its accuracy.

3.6. Elastostatics

The dissertation focuses on edge crack problems in linear elastic media. The variational form is given and numerical approximations by meshless methods are shown.

3.6.1 Governing equations

Assume a two-dimensional (2D) problem with minute displacements on the domain Ω bounded by Γ. The equilibrium equation is

$$\nabla \cdot \sigma + \mathbf{b} = 0 \qquad \text{in } \Omega \tag{3.48}$$

Where, σ is the stress tensor, which corresponds to the displacement field *u, b* is a body force vector, ∇ is the divergence operator. The boundary conditions are given as follows

$$\text{(Natural b.c.)} \qquad \sigma.\mathbf{n} = \bar{\mathbf{t}} \qquad \text{on } \Gamma_t \tag{3.49}$$

$$\text{(Essential b.c.)} \qquad \mathbf{u} = \bar{\mathbf{u}} \qquad \text{on } \Gamma_u \tag{3.50}$$

In which the superposed bar denotes prescribed boundary values, and $\mathbf{n}$ is the unit normal to the domain Ω.

The variational (or weak) form of the equilibrium **Eq. (3.48)** can be written

$$\int_\Omega \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma}\, d\Omega - \int_\Omega \delta\mathbf{u}.\mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}.\bar{\mathbf{t}}\, d\Gamma - \delta\mathbf{W}_u\left(\mathbf{u}, \nabla_s\boldsymbol{\lambda}\right) = 0 \tag{3.51}$$

Where, $\delta\boldsymbol{\varepsilon}=\nabla_s(\delta\mathbf{u})$, ∇_s is the symmetric gradient operator. The term $\delta\mathbf{W}_u$ is a term to enforce the essential boundary conditions. The term $\delta\mathbf{W}_u$ is needed, because in contrast to finite element methods since $\Phi_I(x_J)\neq\delta_{IJ}$ it is not sufficient to set the nodal displacements to enforce the essential boundary conditions. Several forms of $\mathbf{W}_u$ are possible. In the present work, Lagrange multiplier method has been considered, so

$$\mathbf{W}_u(\mathbf{u},\boldsymbol{\lambda})=\int_{\Gamma_u}\boldsymbol{\lambda}.(\mathbf{u}-\bar{\mathbf{u}})d\Gamma \tag{3.52}$$

$$\delta\mathbf{W}_u(\mathbf{u},\boldsymbol{\lambda})=\int_{\Gamma_u}\delta\boldsymbol{\lambda}.(\mathbf{u}-\bar{\mathbf{u}})d\Gamma+\int_{\Gamma_u}\delta\mathbf{u}.\boldsymbol{\lambda}\,d\Gamma \tag{3.53}$$

3.6.2 Discrete equation

We now consider linear elasticity where

$$\boldsymbol{\varepsilon}=\nabla_s\mathbf{u} \tag{3.54}$$

$$\boldsymbol{\sigma}=\mathbf{D}\boldsymbol{\varepsilon} \tag{3.55}$$

Where, $\boldsymbol{\varepsilon}$ is the strain and $\mathbf{D}$ is a matrix of material moduli. We consider the discrete equation for the weak form **Eq. (3.51)** with the essential boundary conditions in **Eqs.(3.52) and (3.53).** The Lagrange multiplier λ is expresses by

$$\begin{aligned}\lambda(\mathbf{x})&=N_I(s)\boldsymbol{\lambda}_I, &\mathbf{x}\in,u\\ \delta\lambda(\mathbf{x})&=N_I(s)\delta\lambda_I &\mathbf{x}\in,u\end{aligned} \tag{3.56}$$

Where, $N_{I(s)}$ a Lagrange interpolant & s is the arc length along the boundary; the repeated indices designate summations.

In the variational form of equilibrium **Eq.(3.51)**, replace $u(\mathbf{x})$ with the EFG approximation $u^h(\mathbf{x})$, and replace the variation $\delta u(\mathbf{x})$ with the EFG approximation $\delta u^h(\mathbf{x})$, where δu^h, by Galerkin approximation technique, is comprised of the same shape function as in u^h:

$$u^h(\mathbf{x})=\sum_{I=1}^{n}\Phi_I(\mathbf{x})u_I \tag{3.57}$$

$$\text{and }\delta u^h(\mathbf{x})=\sum_{I=1}^{n}\Phi_I(\mathbf{x})\,\delta u_I \tag{3.58}$$

The nodal test function values δu_I are arbitrary, except on Γ_u and can be eliminated from the equations.

Also substitute equations **(3.56)** into the weak form **(3.51),** which yields:

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^{\mathbf{T}} & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{q} \end{Bmatrix} \tag{3.59}$$

and

$$\mathbf{K}_{IJ} = \int_{\Omega} \mathbf{B}^{T} \mathbf{D} \mathbf{B}_{J} \, d\Omega \tag{3.60a}$$

$$\mathbf{G}_{IK} = -\int_{T_u} \Phi_I \mathbf{N}_k \, d\Gamma_u \tag{3.60b}$$

$$\mathbf{f}_I = \int_{\Gamma_t} \Phi_I \bar{\mathbf{t}} \, d\Gamma_t + \int_{\Omega} \Phi_I \mathbf{b} \, d\Omega \tag{3.60c}$$

$$\mathbf{q}_k = -\int_{\Gamma_u} \mathbf{N_k} \bar{\mathbf{u}} \, d\Gamma_u \tag{3.61}$$

Where,

$$\mathbf{B}_I = \begin{bmatrix} \Phi_{I,x} & 0 \\ 0 & \Phi_{I,y} \\ \Phi_{I,y} & \Phi_{I,x} \end{bmatrix} \tag{3.62}$$

$$\mathbf{N}_k = \begin{bmatrix} N_k & 0 \\ 0 & N_k \end{bmatrix} \tag{3.63}$$

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad \text{For plane strain} \tag{3.64}$$

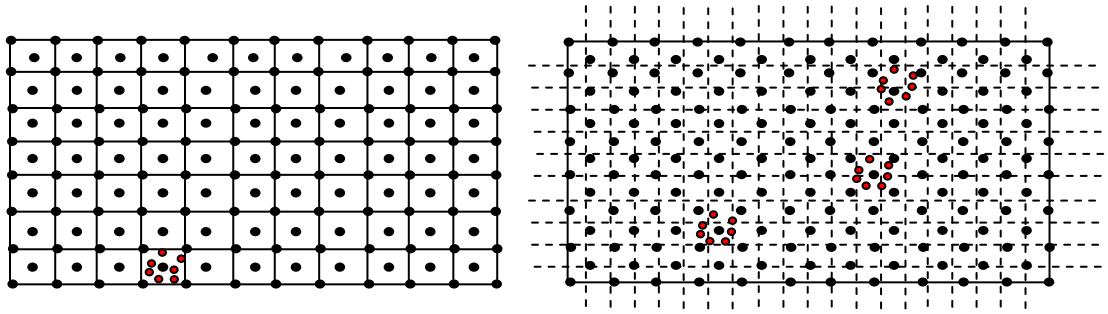
$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad \text{For plane stress} \tag{3.65}$$

In which a comma designates a partial derivative with respect to the indicated spatial variable; E and ν are Young's modulus and Poisson's ratio, respectively.

3.6.3 Integration issues

For the computation of stiffness matrix ($\mathbf{K}$), displacement matrix (U) and force vector ($\mathbf{f}$) [**Eq. (3.60)]** requires integrating the domain, which in 2D corresponds to an area integration. Integration method such as Gauss quadraure is used for integrating stiffness matrix and force vector, which in turn, requires a bifurcation of the area. Since MMs in their pristine form do not include domain bifurcation like FEM, therefore it becomes essential to bring in a subdivision scheme of the domain. One of the methods is background cell structure. Two

subdivisions are revealed in **Figure. 3.7**. The first one revealed, which is the most general makes the use of a finite element mesh generator to produce a cell structure which matches the domain; this practice is frequently called as element quadrature. The vertices of this background mesh are often used as the initial array of nodes for the EFG model; however, additional nodes may be added where desired such as the nodes at the crack tip in the model shown. The second integration technique called as cell quadrature, uses a background grid of cells which is independent of the domain. At each integration point, it is necessary to determine if it lies inside the domain before it is used for integrating **Eq. (3.60).**This technique is not widely used because it does not yield good accuracy along curved and angled boundaries. A nodal integration technique was proposed by Beissel and Belytschko [6] in an effort to make EFG a completely meshless method. However, additional stability terms must be added to make the method stable and accuracy is not as good as cell-based integration schemes.



(a) Element quadrature **(b) Cell quadrature**

Figure. 3.7 Two integration methods for integrating the weak form with a meshless method

To enhance the accuracy of Gauss quadrature, a sub-triangle technique is used [7] as it circumvents the difficulties related to discontinuities are present within a background cell. Sukumar *et al.* [8] established that a continuous increment in order of Gauss integration will not always improve the integration over a discontinuous element/cell. This numerical difficulty was surmounted by using an approach similar to one projected by Dolbow [9] for extended finite element method (XFEM). According to this technique, any background cell which intersects with a crack is subdivided on both sides into sub-triangles whose edges are adapted to crack faces, as shown in **Figure.3.8.** It is imperative to note that, while triangulation of the crack tip element considerably improves the accuracy of integration by increasing the order of Gauss quadrature, it also shuns the numerical complications of

singular fields at the crack tip because none of the Gauss points are placed on the position of the crack tip.

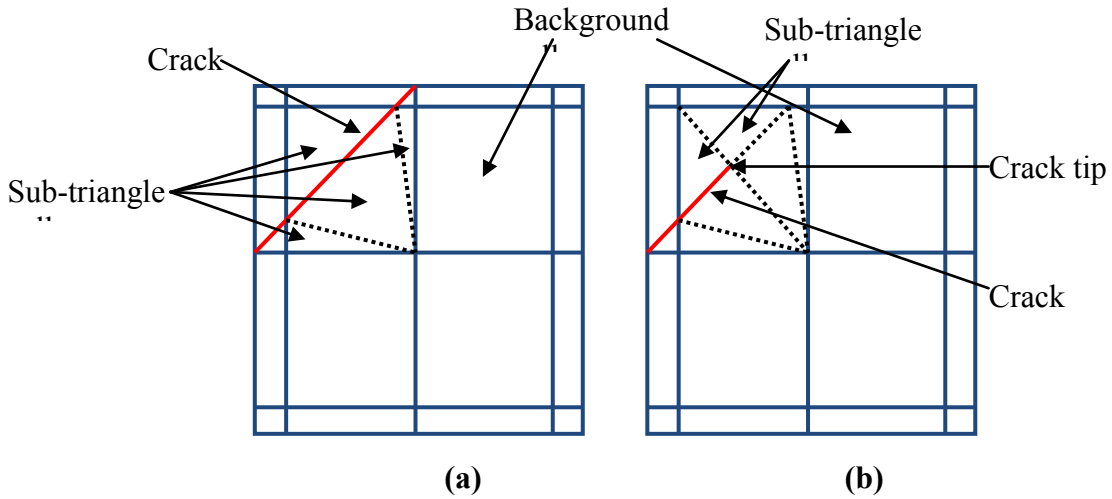


Figure.3.8 Sub-triangle technique for partitioning the cracked background cell: (a) crack edge and (b) crack tip.

3.6.4 Domain of influence

Properly choosing the domain of influence or nodal support is an important aspect of meshless methods. The size of the support should be sufficiently large so that the stiffness matrix is regular and well-conditioned. But too large domains of influence lead to a great deal of computational expense in forming the approximations as well as assembling the stiffness matrix. The domain of influence multiplier $d_{\max}$ is typically 1.5-3 for static analysis and 2-2.5 for dynamic analysis.

3.7 Computer implementation aspect

The computational implementation aspect of MMs is considerably different from the FEM procedure owing to difference in basic structure of these methods[28]. The following points in this context are worth mentioning:

- The procedure of derivation of shape functions and their derivatives.
- Assembly procedure
- Process of imposition of essential boundary conditions.
- The final post processing procedure.

The generalized meshless procedure is as follows:

- The domain is discretized into nodal points with each node having a specific value of weigh function. Along with this each nodal point following must be specified:
  - The value of radius of support domain(in case of circular domain).
  - The shape of domain of influence i.e. either circle of rectangle.
- Generation of Gaussian Quadrature points for integration of domain.
- Generating Gauss points along the traction and displacement boundaries.
- Integration of the domain. For each Gauss points x_q.
- Find nodes within the support of x_q.For each of these nodes; compute weight function, shape function and shape function derivatives.
- Assemble B matrix.
- Assemble K matrix.
- Integration on the boundaries. Integrate forces along the traction boundary to form the nodal force vector f and also on the essential boundary to impose boundary conditions.
- Resolution of the resulting system of equations.
- Solve for the nodal displacement since meshless approximation does not satisfy the Kronecker delta property and find the stress intensity factors using interaction integral.

4. Modified algorithm for radius of domain

In EFGM the radius of the support domain are determined by the following mathematical formula[10]:

$$r_i = d_{max} c_i \quad (4.1)$$

where r_i is the radius of support domain d_{max} is the scaling parameter and c_i is the distance between neighboring nodes. It is well established that the value of scaling parameter is ideal between 1.5-2.5 and the results of simulation are sensitive to it [1]. Therefore, the number of nodes in support domain (n_d) depend on the scaling parameter, not on the total amount of nodes (n_T) in the problem geometry. A varying amount of nodes in support domain are introduced using this formulation for every iteration and there occurs a loss in accuracy.

To remove this problem Sheng et al.[11] provided a new algorithm to modify the radius of support domain such that every support domain has equal number of support nodes (n_d). The method is based on calculating the distance (d_α) between Gauss points and nodes and then prescribing the desired number of nodes (n_r) in support domain by selecting the desired value of d_α. This method provided improvement in results but did not provide a concrete formulation for amount of desired nodes to be prescribed for every support domain. Another problem is that the EFGM computational time (T_{EFGM}) of analysis increases (as of more number of domain nodes are processed in the loop compared to conventional method) which

makes the method less efficient. Henceforth, the random or desired number of nodes (n_r) should be carefully selected as a low number will generate very erroneous results and too high will increase the T_{EFGM}. To remove the predicament an optimization is carried out by Taguchi method explained in ensuing sections.

To reduce the computational time (T_{EFGM}) the algorithm is extended judiciously in such a way that the area around the crack uses higher number of nodes for support domain while the area away from crack uses lower number of nodes in support domain.

Consider a domain with an edge crack as shown in Figure.4.1. The partially shaded area (Region A) represents the area in which every support domain of every Gauss point is prescribed a higher number of nodes so that the stress oscillations around the crack can be trapped efficiently while for non shaded or left over area (Region B) a fewer number of nodes can be prescribed. The area of region A is also carefully selected so as to keep the balance between accuracy and computational time. A parametric study for area ratio (Å) i.e. ratio of area of region B($Å_{Region\ B}$) and area of region A($Å_{Region\ A}$) in the domain is performed and explained further. The study establishes that the optimum distribution of areas of Region(A & B)is between 60%-80% and 20%-40% respectively. To provide clarity to the readers the flow chart of Accelerated Element Free Galerkin Method(AEFG) has been shown in Figure.4.2, this figure also shows the algorithm to modify the radius of support domain shown in green outline.

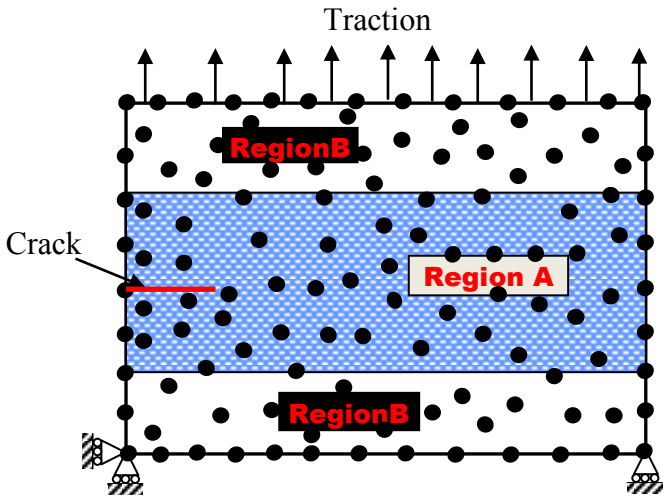
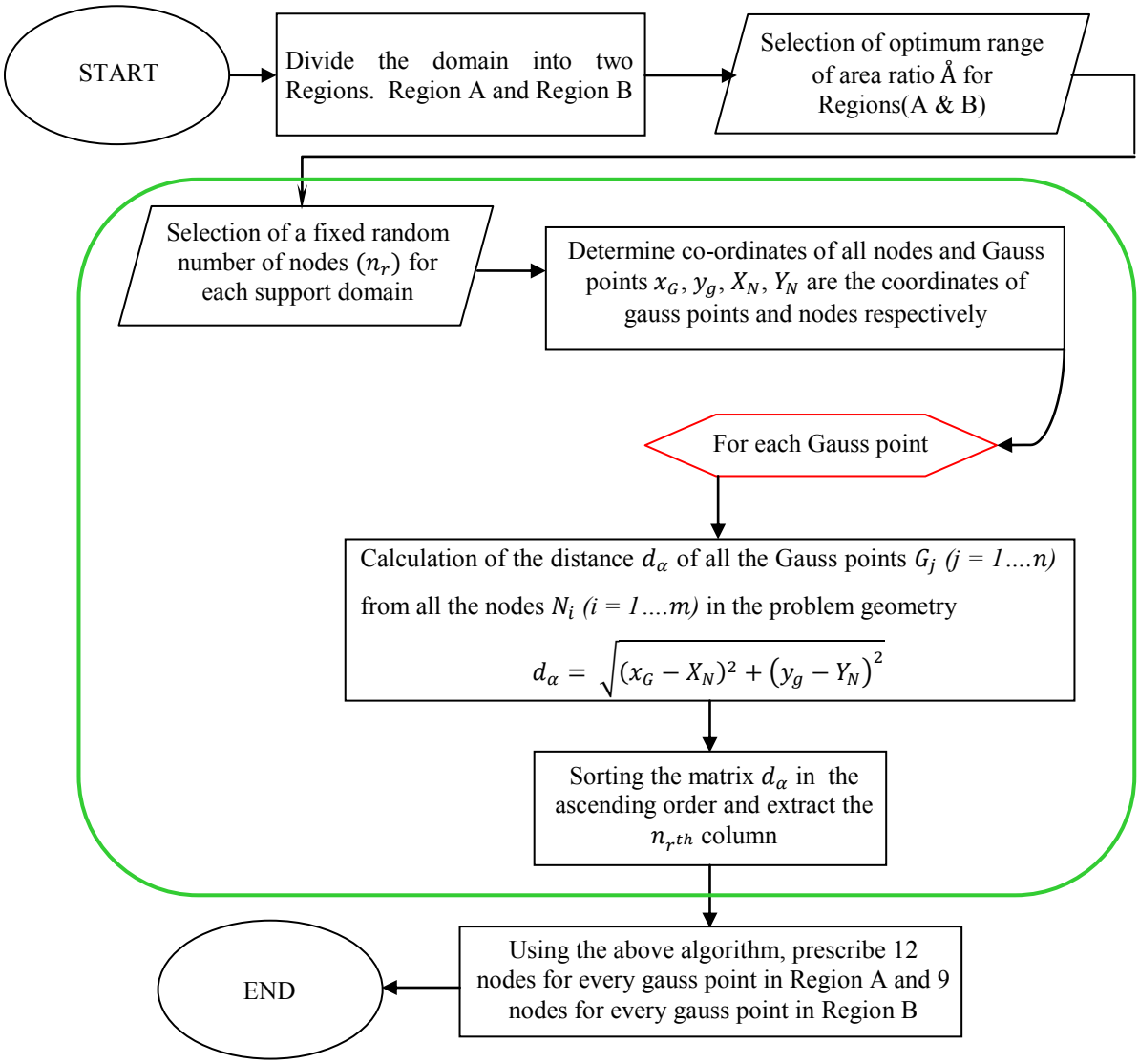


Figure.4.1 Schematic representation of modified method



Firure: 4.1 Llow chart is used to eliminate Domain of Influence

[1] T. Belytschko, Y. Y. Lu, and L. Gu, "Element free Galerkin methods," *Int. J. Numer. Methods Eng.*, vol. 37, no. April 1993, pp. 229–256, 1994.

[2] J. Dolbow and T. Belytschko, "An introduction to programming the meshless element f reeGalerkin method," *Arch. Comput. Methods Eng.*, vol. 5, no. 3, pp. 207–241, 1998.

[3] T. Belytschko, Y. Y. Lu, and L. Gu, "Crack propagation by element-free Galerkin methods," *Eng. Fract. Mech.*, vol. 51, no. 2, pp. 295–315, 1995.

[4] Y. Y. Lu, T. Belytschko, and L. Gu, "A new implementation of the element free Galerkin method," *Comput. Methods Appl. Mech. Eng.*, vol. 113, no. 2, pp. 397–414, 1994.

[5] T. Belytschko, D. Organ, and Y. Krongauz, "A coupled finite element - Element-free Galerkin method," *Comput. Mech.*, vol. 17, no. 3, pp. 186–195, 1995.

[6] S. Beissel and T. Belytschko, "Nodal integration of the element-free Galerkin method," *Comput. Methods Appl. Mech. Eng.*, vol. 139, pp. 49–74, 1996.

[7] S. S. Ghorashi, S. Mohammadi, and S.-R. Sabbagh-Yazdi, "Orthotropic enriched element free Galerkin method for fracture analysis of composites," *Eng. Fract. Mech.*, vol. 78, no. 9, pp. 1906–1927, 2011.

[8] N. Sukumar, N. Moës, B. Moran, and T. Belytschko, “Extended finite element method for three-dimensional crack modelling,” *Int. J. Numer. Methods Eng.*, vol. 48, no. 11, pp. 1549–1570, 2000.
[9] J. Dolbow, N. Moës, and T. Belytschko, “Discontinuous enrichment in finite elements with a partition of unity method,” *Finite Elem. Anal. Des.*, vol. 36, no. 3, pp. 235–260, 2000.
[10] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot, “Meshless methods: A review and computer implementation aspects,” *Math. Comput. Simul.*, vol. 79, no. 3, pp. 763–813, 2008.
[11] M. Sheng, G. Li, and S. Shah, “A modified method to determine the radius of influence domain in element-free Galerkin method,” *J. Mech. Eng. Sci. Part-c*, vol. 0, no. 0, pp. 1–11, 2015.