CHAPTER 2 THEORY

The theories consist of the meshless method, the MLS Approximation, the MLS approximation for the Local Symmetric Weak Form (LSWF), the Local Weak Form (LWF) for Poisons Equation, the Meshless Local Petrov-Galerkin (MLPG) Method, the Divergence theorem, and the Crank-Nicolson method. The details of aforementioned topics are described as follows.

2.1 The MLS Approximation

The moving least squares (MLS) approximation was devised by mathematicians for data fitting and surface construction. It can be categorized as a method for series representation of functions. An excellent description of the MLS method can be found in a paper by Lancaster and Salkausdas (1981). The MLS Approximation is now widely used in meshless methods for constructing meshless shape functions.

The idea is to start with a weighted least squares formulation for an arbitrary fixed point in $\mathbb{R}^d$, (see weighted least squares approximation) and then move this point over the entire parameter domain, where a weighted least squares fit is computed and evaluated for each point individually.

2.1.1 Least Squares (LS) Approximation

Problem Formulation. Given N points located at positions $(\mathbf{x}_i)$ in $\mathbb{R}^d$, where $i = 1, ..., N$.

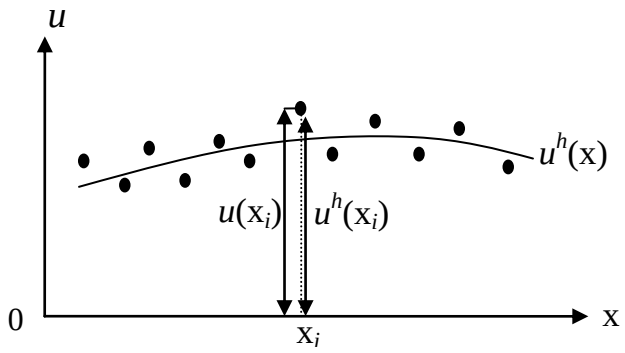

Figure 2.1 The distinction between $u(\mathbf{x}_i)$ and $u^h(\mathbf{x}_i)$.

We wish to obtain a globally defined function $u^h(\mathbf{x}_i)$ that approximates the given scalar values at points $(\mathbf{x}_i)$ in the least-squares sense with the residual function

$$J_{LS} = \sum_{i=1}^{n} \left\| u^h(\mathbf{x}_i) - u(\mathbf{x}_i) \right\|^2. \tag{2.1}$$

Thus, we want to minimize J_{Ls} in equation (2.1)

$$\min \sum_{i=1}^{n} \left\| u^h(\mathbf{x}_i) - u(\mathbf{x}_i) \right\|^2, \tag{2.2}$$

where Ω is the domain, $\mathbf{x}^{\mathrm{T}} = \{x, y\}$ for two-dimensional problem, *m* is the number of the basis functions, $u(\mathbf{x})$ is an unknown scalar function of a field variable in the domain, $u^h(\mathbf{x})$ is a trial function in the form of a linear combination of basis functions:

$$u^h(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}), \tag{2.3}$$

$\mathbf{p}(\mathbf{x})$ is the basis function of the spatial coordinates, and $\mathbf{a}(\mathbf{x})$ is a vector of coefficients given by $\mathbf{a}^{\mathrm{T}}(\mathbf{x}) = \{a_1(\mathbf{x}), a_2(\mathbf{x}), ..., a_m(\mathbf{x})\}$.

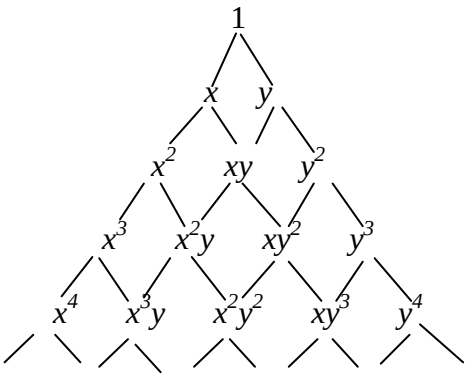

Figure 2.2 Pascal triangle of monomials for two-dimensional domains (Liu, 2005).

Figure 2.2 shows that the linear basis functions and the quadratic basis functions are given by

$$\left.\begin{aligned}
&\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \{1, x\} && m = 2\ (one-dimensional),\\
&\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \{1, x, y\} && m = 3\ (two-dimensional),\\
&\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \{1, x, x^2\} && m = 3\ (one-dimensional),\\
&\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \{1, x, y, x^2, xy, y^2\} && m = 6\ (two-dimensional).
\end{aligned}\right\} \tag{2.4}$$

Substituting equation (2.3) into equation (2.1), we obtain

$$J_{LS} = \sum_{i=1}^{n} \left\| u^h(\mathbf{x}_i) - u(\mathbf{x}_i) \right\|^2 = \sum_{i=1}^{n} \left\| \sum_{j=1}^{m} p_j(\mathbf{x}_i)\, a_j(\mathbf{x}) - u(\mathbf{x}_i) \right\|^2. \tag{2.5}$$

The minimum is determined by calculating the partial derivatives of the residual function with respect to the unknown coefficients a_1, a_2, ..., a_m and setting them to zero. We obtain a linear system of equations from which we can compute **a**:

$$\frac{\partial J}{\partial a_k} = \sum_{i=1}^{n} 2 \left\| \sum_{j=1}^{m} p_j(\mathbf{x}_i) a_j(\mathbf{x}) - u(\mathbf{x}_i) \right\| p_j(\mathbf{x}_i) = 0. \tag{2.6}$$

We divide by the constant and rearrange to

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{m} \left(p_j(\mathbf{x}_i) a_j(\mathbf{x}) p_k(\mathbf{x}_i) \right) \right) = \sum_{i=1}^{n} u(\mathbf{x}_i) p_k(\mathbf{x}_i), \tag{2.7}$$

$$\sum_{j=1}^{m} \left(\sum_{i=1}^{n} \left(p_j(\mathbf{x}_i) a_j(\mathbf{x}) p_k(\mathbf{x}_i) \right) \right) = \sum_{i=1}^{n} u(\mathbf{x}_i) p_k(\mathbf{x}_i), \tag{2.8}$$

$$\sum_{j=1}^{m} a_j(\mathbf{x}) \left(\sum_{i=1}^{n} \left(p_j(\mathbf{x}_i) p_k(\mathbf{x}_i) \right) \right) = \sum_{i=1}^{n} u(\mathbf{x}_i) p_k(\mathbf{x}_i), \tag{2.9}$$

where $k = 1, ..., m$. Substituting $j = 1, ..., m$ in equation (2.9), we obtain

$$k = 1$$

$$a_1(\mathbf{x})\sum_{i=1}^{n}(p_1(\mathbf{x}_i)p_1(\mathbf{x}_i)) + a_2(\mathbf{x})(\sum_{i=1}^{n}(p_2(\mathbf{x}_i)p_1(\mathbf{x}_i)) + \ldots \tag{2.10}$$

$$+a_m(\mathbf{x})(\sum_{i=1}^{n}(p_m(\mathbf{x}_i)p_1(\mathbf{x}_i)) = \sum_{i=1}^{n}u(\mathbf{x}_i)p_1(\mathbf{x}_i),$$

$$k = 2$$

$$a_1(\mathbf{x})\sum_{i=1}^{n}(p_1(\mathbf{x}_i)p_2(\mathbf{x}_i)) + a_2(\mathbf{x})(\sum_{i=1}^{n}(p_2(\mathbf{x}_i)p_2(\mathbf{x}_i)) + \ldots \tag{2.11}$$

$$+a_m(\mathbf{x})(\sum_{i=1}^{n}(p_m(\mathbf{x}_i)p_2(\mathbf{x}_i)) = \sum_{i=1}^{n}u(\mathbf{x}_i)p_2(\mathbf{x}_i),$$

$$k = m$$

$$a_1(\mathbf{x})\sum_{i=1}^{n}(p_1(\mathbf{x}_i)p_m(\mathbf{x}_i)) + a_2(\mathbf{x})(\sum_{i=1}^{n}(p_2(\mathbf{x}_i)p_m(\mathbf{x}_i)) + \ldots \tag{2.12}$$

$$+a_m(\mathbf{x})(\sum_{i=1}^{n}(p_m(\mathbf{x}_i)p_m(\mathbf{x}_i)) = \sum_{i=1}^{n}u(\mathbf{x}_i)p_m(\mathbf{x}_i).$$

We obtain a system of linear equations:

$$\begin{bmatrix} \sum_{i=1}^{n}p_1^2(\mathbf{x}_i) & \sum_{i=1}^{n}p_2(\mathbf{x}_i)p_1(\mathbf{x}_i) & \ldots & \sum_{i=1}^{n}p_m(\mathbf{x}_i)p_1(\mathbf{x}_i) \\ \sum_{i=1}^{n}p_1(\mathbf{x}_i)p_2(\mathbf{x}_i) & \sum_{i=1}^{n}p_2^2(\mathbf{x}_i) & \ldots & \sum_{i=1}^{n}p_m(\mathbf{x}_i)p_2(\mathbf{x}_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n}p_1(\mathbf{x}_i)p_m(\mathbf{x}_i) & \sum_{i=1}^{n}p_2(\mathbf{x}_i)p_m(\mathbf{x}_i) & \cdots & \sum_{i=1}^{n}p_m^m(\mathbf{x}_i) \end{bmatrix} \begin{bmatrix} a_1(\mathbf{x}) \\ a_2(\mathbf{x}) \\ \vdots \\ a_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n}u(\mathbf{x}_i)p_1(\mathbf{x}_i) \\ \sum_{i=1}^{n}u(\mathbf{x}_i)p_2(\mathbf{x}_i) \\ \vdots \\ \sum_{i=1}^{n}u(\mathbf{x}_i)p_m(\mathbf{x}_i) \end{bmatrix} \tag{2.13}$$

or

$$\mathbf{KA} = \mathbf{U}, \tag{2.14}$$

$$\mathbf{K} = \sum_{i=1}^{n}\begin{bmatrix} p_1^2(\mathbf{x}_i) & p_2(\mathbf{x}_i)p_1(\mathbf{x}_i) & \ldots & p_m(\mathbf{x}_i)p_1(\mathbf{x}_i) \\ p_1(\mathbf{x}_i)p_2(\mathbf{x}_i) & p_2^2(\mathbf{x}_i) & \ldots & p_m(\mathbf{x}_i)p_2(\mathbf{x}_i) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_i)p_m(\mathbf{x}_i) & p_2(\mathbf{x}_i)p_m(\mathbf{x}_i) & \cdots & p_m^m(\mathbf{x}_i) \end{bmatrix}, \tag{2.15}$$

$$\mathbf{A} = \begin{bmatrix} a_1(\mathbf{x}) \\ a_2(\mathbf{x}) \\ \vdots \\ a_m(\mathbf{x}) \end{bmatrix}, \tag{2.16}$$

$$\mathbf{U} = \sum_{i=1}^{n} \begin{bmatrix} p_1(\mathbf{x}_i) \\ p_2(\mathbf{x}_i) \\ \vdots \\ p_m(\mathbf{x}_i) \end{bmatrix} u(\mathbf{x}_i), \tag{2.17}$$

which is solved by

$$\mathbf{A} = \mathbf{K}^{-1}\mathbf{U}. \tag{2.18}$$

If the square matrix **A** is nonsingular (i.e. det(**A**) $\neq$ 0), substituting equation (2.18) into equation (2.3) provides the fit function $u^h(\mathbf{x})$. For small m ($m < 5$), the matrix inversion in equation (2.18) can be carried out explicitly.

2.1.2 Weighted Least Squares (WLS) Approximation

Problem Formulation. In the weighted least squares formulation, we use the residual function

$$J_{WLS} = \sum_{i=1}^{n} w(\left\|\bar{\mathbf{x}} - \mathbf{x}_i\right\|)[\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)\mathbf{a}_j(\bar{\mathbf{x}}) - u(\mathbf{x}_i)]^2. \tag{2.19}$$

For a fixed point $\bar{\mathbf{x}} \in \mathbb{R}^d$, we minimize

$$\min_{u \in \Pi_m^d} \sum_{i=1}^{n} w(\left\|\bar{\mathbf{x}} - \mathbf{x}_i\right\|)[\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)\mathbf{a}_j(\bar{\mathbf{x}}) - u(\mathbf{x}_i)]^2, \tag{2.20}$$

similar to Least Squares Approximation, except that equation (2.19) the residual is weighted by $w(\left\|\bar{\mathbf{x}} - \mathbf{x}_i\right\|)$ where $d_i(\left\|\bar{\mathbf{x}} - \mathbf{x}_i\right\|)$, are the Euclidian distances between $\bar{\mathbf{x}}$ and the positions of data points $\mathbf{x}_i$:

$$d_i = \sqrt{(\bar{x} - x_i)^2 + (\bar{y} - y_i)^2}. \tag{2.21}$$

Analogous to Section LS, we take partial derivatives of the residual function J_{WLS} with respect to the unknown coefficients $\mathbf{a}(\bar{\mathbf{x}})$ and let $w(d_i) = w_i(\bar{\mathbf{x}})$. We obtain:

$$\sum_{i=1}^{n} w_i(\bar{\mathbf{x}}) 2\mathbf{p}(\mathbf{x}_i)[\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)\mathbf{a}_j(\bar{\mathbf{x}}) - u(\mathbf{x}_i)] = 0, \tag{2.22}$$

$$2\sum_{i=1}^{n}[w_i(\bar{\mathbf{x}})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)\mathbf{a}_j(\bar{\mathbf{x}}) - w_i(\bar{\mathbf{x}})\mathbf{p}(\mathbf{x}_i)u(\mathbf{x}_i)] = 0. \tag{2.23}$$

We divide by the constant and rearrange to

$$\sum_{i=1}^{n} w_i(\bar{\mathbf{x}})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)\mathbf{a}_j(\bar{\mathbf{x}}) = \sum_{i=1}^{n} w_i(\bar{\mathbf{x}})\mathbf{p}(\mathbf{x}_i)u(\mathbf{x}_i), \tag{2.24}$$

and solve for the coefficients:

$$\mathbf{a}(\bar{\mathbf{x}}) = [\sum_{i=1}^{n} w_i(\bar{\mathbf{x}})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)]^{-1} \cdot \sum_{i=1}^{n} w_i(\bar{\mathbf{x}})\mathbf{p}(\mathbf{x}_i)u(\mathbf{x}_i). \tag{2.25}$$

Substituting equation (2.25) into equation (2.3) provides the trial function $u^h(\mathbf{x})$.

2.1.3 Moving Least Squares (MLS) Approximation

Formulation of MLS shape functions. The MLS approximation of $u(\mathbf{x})$ is defined at $\mathbf{x}$ in equation (2.3), in which the coefficients $\mathbf{a}(\mathbf{x})$ can be obtained by minimizing the following weighted discrete $\mathbf{L}_2$ norm:

$$J_{MLS} = \sum_{i=1}^{n} w(\|\mathbf{x} - \mathbf{x}_i\|)[\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - u_i]^2, \tag{2.26}$$

where n is the number of nodes in the support domain of $\mathbf{x}$, u_i is the nodal parameter of u at $\mathbf{x}=\mathbf{x}_i$, $w(\|\mathbf{x}-\mathbf{x}_i\|)=w(d_i)\neq 0, d_i=\|\mathbf{x}-\mathbf{x}_i\|$, d_i is the Euclidian distances between $\mathbf{x}$ and the positions of data points $\mathbf{x}_i$. The stationarity of J with respect to $\mathbf{a}(\mathbf{x})$ gives:

$$\frac{\partial J}{\partial \mathbf{a}_j}=0, \qquad j=1,...,m \tag{2.27}$$

which leads to the following linear relation between $\mathbf{a}(\mathbf{x})$ and $\mathbf{U}$

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x})=\mathbf{B}(\mathbf{x})\mathbf{U}, \tag{2.28}$$

where $\mathbf{U}$ is the vector of field function for all the nodes

$$\mathbf{U}=\{u_1,u_2,\ldots,u_n\}^{\mathrm{T}}. \tag{2.29}$$

Here, it should be noted that $u_i,\ i=1,...,n$ in equation (2.1) and equation (2.29) are fictitious nodal values, and not the nodal values of the unknown trial function $u^h(\mathbf{x})$ in general (see Figure 2.1 for the simple one-dimensional case for the distinction between u_i and $u^h(\mathbf{x}_i)$.)

$\mathbf{A}(\mathbf{x})$ is called the weighted moment matrix, defined by:

$$\mathbf{A}(\mathbf{x})=\sum_{i=1}^{n} w(\|\mathbf{x}-\mathbf{x}_i\|)\mathbf{p}(\mathbf{x}_i)\mathbf{p}^{\mathrm{T}}(\mathbf{x}_i), \tag{2.30}$$

$$w_i(\mathbf{x})=w(\|\mathbf{x}-\mathbf{x}_i\|). \tag{2.31}$$

For a two-dimensional problem, let m = 3, $\mathbf{A}$ is a symmetric 3×3 matrix that can be explicitly written as

$$\begin{aligned}\mathbf{A}(\mathbf{x})_{3\times 3} &= \sum_{i=1}^{n} w_i(\mathbf{x})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^{\mathbf{T}}(\mathbf{x}_i) \\ &= w_1(\mathbf{x})\begin{bmatrix} 1 & x_1 & y_1 \\ x_1 & x_1^2 & x_1y_1 \\ y_1 & x_1y_1 & y_1^2 \end{bmatrix} + w_2(\mathbf{x})\begin{bmatrix} 1 & x_2 & y_2 \\ x_2 & x_2^2 & x_2y_2 \\ y_2 & x_2y_2 & y_2^2 \end{bmatrix} + \ldots + w_n(\mathbf{x})\begin{bmatrix} 1 & x_n & y_n \\ x_n & x_n^2 & x_ny_n \\ y_n & x_ny_n & y_n^2 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^{n} w_i & \sum_{i=1}^{n} x_iw_i & \sum_{i=1}^{n} y_iw_i \\ \sum_{i=1}^{n} x_iw_i & \sum_{i=1}^{n} x_i^2w_i & \sum_{i=1}^{n} x_iy_iw_i \\ \sum_{i=1}^{n} y_iw_i & \sum_{i=1}^{n} x_iy_iw_i & \sum_{i=1}^{n} y_i^2w_i \end{bmatrix}\end{aligned} \quad (2.32)$$

The matrix **B** in equation (2.28) is defined as:

$$\mathbf{B}(\mathbf{x}) = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1)\, w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2)\ldots w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)], \quad (2.33)$$

which is a $3\times n$ matrix, and can be expressed explicitly as:

$$\mathbf{B}_{3\times n}(\mathbf{x}) = \left[w_1(\mathbf{x})\begin{bmatrix} 1 \\ x_1 \\ y_1 \end{bmatrix} w_2(\mathbf{x})\begin{bmatrix} 1 \\ x_2 \\ y_2 \end{bmatrix} \ldots w_n(\mathbf{x})\begin{bmatrix} 1 \\ x_n \\ y_n \end{bmatrix}\right] = \begin{bmatrix} w_1 & w_2 & \ldots & w_n \\ x_1w_1 & x_2w_2 & \ldots & x_nw_n \\ y_1w_1 & y_2w_2 & \ldots & y_nw_n \end{bmatrix}_{3\times \mathbf{n}}. \quad (2.34)$$

Solving equation (2.28) for **a**(**x**), we multiply both sides with $\mathbf{A}^{-1}(\mathbf{x})$, obtaining:

$$\mathbf{A}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{U} \quad (2.35)$$

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{U}. \quad (2.36)$$

Substituting the above equation back into equation (2.3), we obtain:

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i(\mathbf{x})u_i = \mathbf{\Phi}^{\mathbf{T}}(\mathbf{x})\mathbf{U}, \quad (2.37)$$

where $\mathbf{\Phi}(\mathbf{x})$ is the vector of MLS shape functions corresponding to the n nodes in the support domain of the point $\mathbf{x}$, which can be written as

$$\mathbf{\Phi}^{\mathbf{T}}(\mathbf{x}) = \{\phi_1(\mathbf{x})\phi_2(\mathbf{x})\ldots\phi_n(\mathbf{x})\}_{(1\times n)} = \mathbf{P}^{\mathbf{T}}(\mathbf{x})\mathbf{A}^{(-1)}(\mathbf{x})\mathbf{B}(\mathbf{x}). \tag{2.38}$$

The shape function $\phi_i(\mathbf{x})$ for the ith node is defined by:

$$\phi_i(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})]_{ji} = \mathbf{P}^{\mathbf{T}}(\mathbf{x})(\mathbf{A}^{-1}\mathbf{B})_i. \tag{2.39}$$

$\phi_i(\mathbf{x})$ is usually called the shape function of the MLS approximation corresponding to nodal point y_i. From equation (2.33) and equation (2.39), it may be seen that $\phi_i(\mathbf{x}) = 0$, when $w_i(\mathbf{x}) = 0$. In practical applications, $w_i(\mathbf{x})$ is generally chosen such that it is non-zero over the support of nodal points y_i (see figure 2.3).

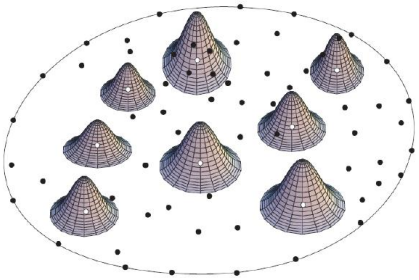

Figure 2.3 The shapes of weight functions $w_i(\mathbf{x})$ (Atluri, 1998a).

The support of the nodal point y_i is usually taken to be a circle of radius r_i, centered at y_i. The fact that $\phi_i(\mathbf{x}) = 0$, for $\mathbf{x}$ not in the support of nodal point y_i, preserves the local character of the Moving Least Squares approximation.

The smoothness of the shape functions $\phi_i(\mathbf{x})$ is determined by that of the basis functions and of the weight functions. Let $C^k(\Omega)$ be the space of $k-th$ continuously differentiable functions. If $w_i(\mathbf{x}) \in C^k(\Omega)$ and $p_j(\mathbf{x}) \in C^l(\Omega), i = 1,2,...,n; j = 1,2,...,m,$ then

$\phi_i(\mathbf{x}) \in C^r(\Omega)$ with $r = min(k,l)$. The partial derivatives of $\phi_i(\mathbf{x})$ are obtained as (Belytschko, 1996b):

$$\phi_{i,k} = \sum_{j=1}^{m} [\, p_{j,k}(\mathbf{A}^{-1}\mathbf{B})_{ji} + p_j(\mathbf{A}^{-1}\mathbf{B}_{,k} + \mathbf{A}^{-1}_{,k}\mathbf{B})_{ji}\,] \tag{2.40}$$

in which $\mathbf{A}^{-1}_{,k} = (\mathbf{A}^{-1})_{,k}$ represents the derivative of the inverse of $\mathbf{A}$ with respect to x^k, which is given by:

$$\mathbf{A}^{-1}_{,k} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}, \tag{2.41}$$

where $(\quad)_{,i}$ denotes $\dfrac{\partial(\quad)}{\partial x^i}$.

Note that the WLS formulation mentioned in section 2.1.2 is very similar to the MLS formulation. In the MLS, the coefficient **a** is the function of **x** that makes the approximation of weighted least squares move continuously. Therefore, the MLS shape function will be continuous in the entire global domain, as long as the weight functions are chosen properly. This global continuity feature is preferred in meshless global weak-form methods. In WLS, however, because the coefficient **a** in equation (2.32) is the constant, the WLS shape functions are piecewise continuous, as discussed in section 2.1.1. The WLS approximation can be viewed as a special form of the MLS approximation.

In the MLS approximation, a support domain can be formed for any point of interest. Field nodes included in this support domain are used to perform the MLS approximation for the unknown function at this point. The number of nodes, *n*, chosen in the support domain, should be sufficient to ensure that the matrix **A** in equation (2.36) is invertible, so as to provide the interpolation with stability. The choice of *n* depends on the nodal distribution and the number of basis functions, *m*. In order to ensure the existence of $\mathbf{A}^{-1}$ and a well-conditioned **A**, we usually let $n \gg m$. Unfortunately, there is no theoretical best value of *n*, and it has to be determined by numerical experiments.

2.1.4 Choices of Weighted Function

Many choices for the weighting function *w* have been proposed in the literature. The exponential function and spline functions are often used in practice. Among them, the most commonly used weight functions are listed below.

− The Gaussian weight function is a classical MLS method, which has the following form of:

$$w_i(\mathbf{x}) = \begin{cases} \dfrac{\exp[-(\frac{d_i}{c_i})^2] - \exp[-(\frac{r_i}{c_i})^2]}{1 - \exp[-(\frac{r_i}{c_i})^2]} & 0 \leqslant d_i \leqslant r_{i,} \\ 0 & d_i \geqslant r_{i,} \end{cases} \tag{2.42}$$

− The cubic spline function, which has the following form of:

$$w_i(\mathbf{x}) = \begin{cases} \dfrac{2}{3} - 4(\frac{d_i}{r_i})^2 + 4(\frac{d_i}{r_i})^3 & \frac{d_i}{r_i} \leqslant 0.5, \\ \dfrac{4}{3} - 4(\frac{d_i}{r_i}) + 4(\frac{d_i}{r_i})^2 - \dfrac{4}{3}(\frac{d_i}{r_i})^3 & 0.5 \leqslant \frac{d_i}{r_i} \leqslant 1, \\ 0 & d_i \geqslant r_i, \end{cases} \tag{2.43}$$

− The quartic spline function, given by:

$$w_i(\mathbf{x}) = \begin{cases} 1 - 6(\frac{d_i}{r_i})^2 + 8(\frac{d_i}{r_i})^3 - 3(\frac{d_i}{r_i})^4 & 0 \leqslant d_i \leqslant r_i, \\ 0 & d_i \geqslant r_i, \end{cases} \tag{2.44}$$

− The exponential function, expressed as:

$$w_i(\mathbf{x}) = \begin{cases} e^{-(\frac{d_i}{c_i r_i})^2} & \frac{d_i}{r_i} \leqslant 1, \\ 0 & \frac{d_i}{r_i} > 1, \end{cases} \tag{2.45}$$

in which $w_i(\mathbf{x}) = w(\|\mathbf{x}-\mathbf{x}_i\|), d_i = \|\mathbf{x}-\mathbf{x}_i\|$ is the Euclidian distances between $\mathbf{x}$ and the positions of data points $\mathbf{x}_i$, c_i is a constant controlling the shape of the weight function and r_i is the size of the support domain. r_i , of the weight function $w_i(\mathbf{x})$ associated with node i , should be chosen such that r_i should be large enough to have sufficient number of nodes covered in the domain of definition of every sample point $(n \geq m)$, in order to ensure the regularity of $\mathbf{A}$. A very small r_i may result in a relatively large numerical error in using Gauss numerical quadrature to calculate the entries in the system matrix. On the other hand, r_i should also be small enough to maintain the local character of the MLS approximation.

− The new weight function presented by Most and Bucher, with the following form:

$$w_i(\mathbf{x}) = \frac{\tilde{w}_i(\mathbf{x})}{\sum_{j=1}^{k} \tilde{w}_i(\mathbf{x}_j)}, \tag{2.46}$$

where

$$\tilde{w}_i(\mathbf{x}) = \begin{cases} \dfrac{(s_i^2+\varepsilon)^{-2} - (1+\varepsilon)^{-2}}{\varepsilon^{-2} - (1+\varepsilon)^{-2}} & 0 \leqslant s_i \leqslant 1 \\ 0 & s_i \geq 1. \end{cases} \tag{2.47}$$

The variable k belongs to the number of supporting points influencing $\mathbf{x}$; s is the normalized distance between the interpolation point and the considered supporting

point, $s_i = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{r_i}$ and r_i is the influence radius. Most and Bucher (2005) recommended the regularization parameter ε, which should be very small, such as

$$\varepsilon = 10^{-5}. \tag{2.48}$$

Properties of MLS Shape Functions

1. Consistency

By the definition, the consistency of the meshless shape functions is the ability of the shape functions to reproduce the complete order of polynomials. The consistency of the MLS approximation depends on the complete order of the monomial employed in the polynomial basis. If the complete order of monomial is *k*, the shape function will possess C^k consistency. This can be easily demonstrated (Belytschko, 1996; Liu, 2002) as follows:

Consider a field given by

$$u(\mathbf{x}) = \sum_{j=1}^{k} p_j(\mathbf{x})\alpha_j(\mathbf{x}), \quad k \leqslant m. \tag{2.49}$$

Such a given field can always be written in the form of

$$u(\mathbf{x}) = \sum_{j=1}^{k} p_j(\mathbf{x})\alpha_j(\mathbf{x}) + \sum_{l=k+1}^{m} p_l(\mathbf{x}) \cdot 0. \tag{2.50}$$

If we let $a_i(\mathbf{x}) = \alpha_j(\mathbf{x})$, j = 1, 2, ..., k, J in equation (2.19) will vanish and it will necessarily be a minimum, which leads to:

$$u^h(\mathbf{x}) = \sum_{j=1}^{k} p_j(\mathbf{x})\alpha_j(\mathbf{x}) = u(\mathbf{x}). \tag{2.51}$$

This proves that any monomial included in the basis of MLS will be exactly reproduced by the MLS approximation.

2. Reproduction

In the meshless method, the concept of reproduction is separated from that of consistency (Liu, 2002). This is because different types of basis functions can be used in constructing meshless shape functions. Reproduction is the ability of the shape function to reproduce functions that are among the basis functions used to construct the shape functions. The function may not be a polynomial, for example, the radial basis function (RBF) in the radial point interpolation method (RPIM). However, consistency emphasizes the reproducibility of the complete order of polynomials. This is the main difference between consistency and reproduction. Similar to the demonstration of consistency, it can be proven that the MLS approximation can reproduce any function that appears in the basis exactly. This property will be very useful in practical application. For example, we know that there is a singular stress field near the tip of a crack. If only the normal polynomial basis is used, the computational error will be certainly very large. If we can enrich the basis by including a singular function in the basis, the reproduction property of MLS will ensure the reproduction of the singular field. As a result, the solution accuracy can be significantly improved without too much additional cost (Belytschko, 1995a; Belytschko, 1995b). Of course, one has to ensure that the weighted moment matrix computed using equation (2.30) is still invertible and well-conditioned when these enriched basis functions are included, which can otherwise be a problem sometimes.

3. Partitions of Unity

If the constant is included in the basis, the MLS shape function $\phi_i(\mathbf{x})$, shown in Figure 2.4, is of the partition of unity shown in Figure 2.5, i.e.,

$$\sum_{i=1}^{n} \phi_i(\mathbf{x}) = 1. \tag{2.52}$$

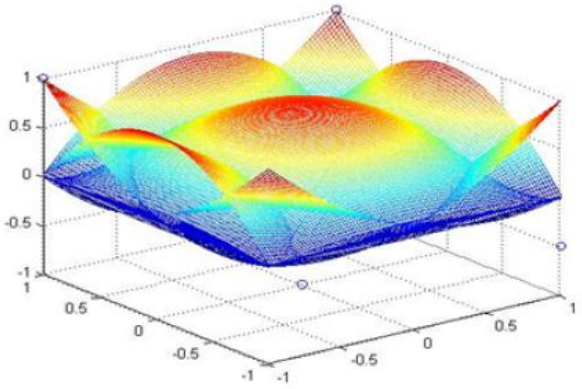

Figure 2.4 Shape function at each node.

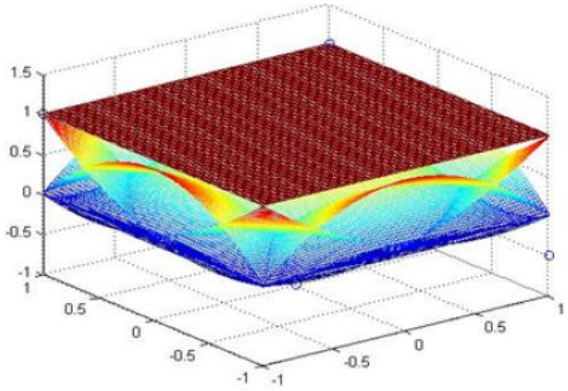

Figure 2.5 Partitions of unity.

4. Lack of Kronecker delta function property

The MLS approximation is obtained by a special least squares method. As shown in Figure 2.1, the function obtained by the MLS approximation is a smooth curve (or surface) and it does not pass through the nodal values. Therefore, the MLS shape functions given in equation (2.39) do not, in general, satisfy the Kronecker delta condition. Thus,

$$\phi_i(\mathbf{x}_j) \neq \delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases} \tag{2.53}$$

Example: Using MLS Approximation to find fitting function.

Data points live in $\mathbb{R}^2$ and we wish to fit a quadratic, $c_i = 1$, $r_i = 4$, using the Gaussian weight function in equation (2.22). Consider the set of nine data points $\mathbf{x}_i = \{(1,1),(1,-1),(-1,1),(-1,-1),(0,0),(1,0),(-1,0),(0,1),(0,-1)\}$ shown in Figure 2.6,

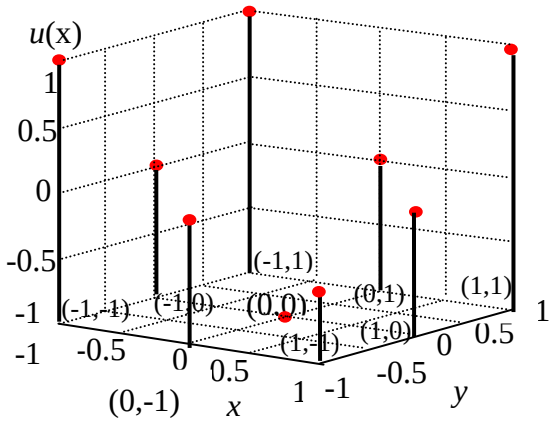

Figure 2.6 Plot of nine data points with set of associated function values.

where $\mathbf{x}_i \in \Omega = [-1,1]\times[-1,1]$ with set of associated function values $u(\mathbf{x}_i) = \{1.0, -0.5, 1.0, 1.0, -1.0, 0.0, 0.0, 0.0, 0.0\}$. The fitting linear basis function, to two-dimensional scalar fields is shown in Figure 2.7 and the fitting quadratic function, to two-dimensional scalar fields is shown in Figure 2.8.

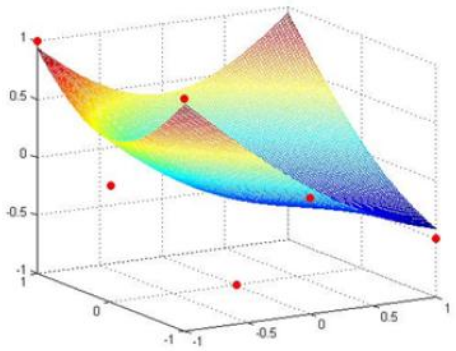

Figure 2.7 Fitting basis function, to two-dimensional scalar fields.

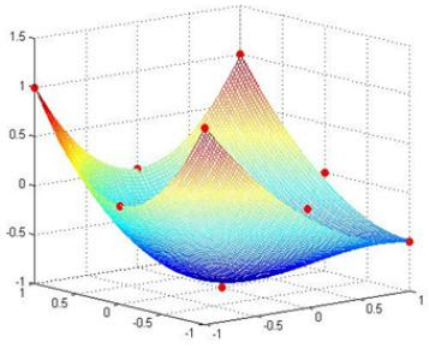

Figure 2.8 Fitting function, quadratic function, to two-dimensional scalar fields.

2.2 The MLS approximation for the Local Symmetric Weak Form (LSWF)

In implementing the MLS approximation for the present local symmetric weak form, the basis functions and weight functions should be chosen at first. Both Gaussian and spline weight functions with compact supports can be considered in the present work. The Gaussian weight function corresponding to node *i* may be written as

$$w_i(\mathbf{x}) = \begin{cases} \dfrac{\exp[-(\frac{d_i}{c_i})^{2k}] - \exp[-(\frac{r_i}{c_i})^{2k}]}{1-\exp[-(\frac{r_i}{c_i})^{2k}]} & 0 \le d_i \le r_i, \\ 0 & d_i \geqslant r_i, \end{cases} \tag{2.54}$$

where $d_i = |\mathbf{x} - \mathbf{x}_i|$ is the distance from node $\mathbf{x}_i$ to point $\mathbf{x}$; c_i is controlling the shape of the weight function, denoted by w_i, and therefore the relative weights; and r_i is the size of the support for the weight function, determining the support of node $\mathbf{x}_i$. In the present computation, k = 1 was chosen. It can be easily seen that the Gaussian weight function is C^0 continuous over the entire domain Ω. Therefore, the shape functions and trial function are also C^0 continuous over the entire domain. Even though the definition of all constants c_i; *is* more or less arbitrary, they do affect the computational results significantly. Lu, et al. (1994) recommended a method to choose constants c_i, however, better methods to choose these constants should be explored. A spline weight function is defined in equation (2.44). It can also be easily seen that the spline weight function is

C^1 continuous over the entire domain Ω. Therefore, the shape functions and the trial function are also C^1 continuous over the entire domain.

Here, the distinction between "the domain of definition of an MLS approximation for the trial function at a point" (herein after simplified as "the domain of definition of a point") and the support of a node should be noted. The domain of definition of a point **x** is a domain in which $w_i(\mathbf{x}) \neq 0, i = 1, 2, ..., n$ and the support of node *j* is a circle (sphere) of radius r_j , centered at $\mathbf{x}_j$, in which $w_i(\mathbf{x}) \neq 0$ (see Figure 2.9).

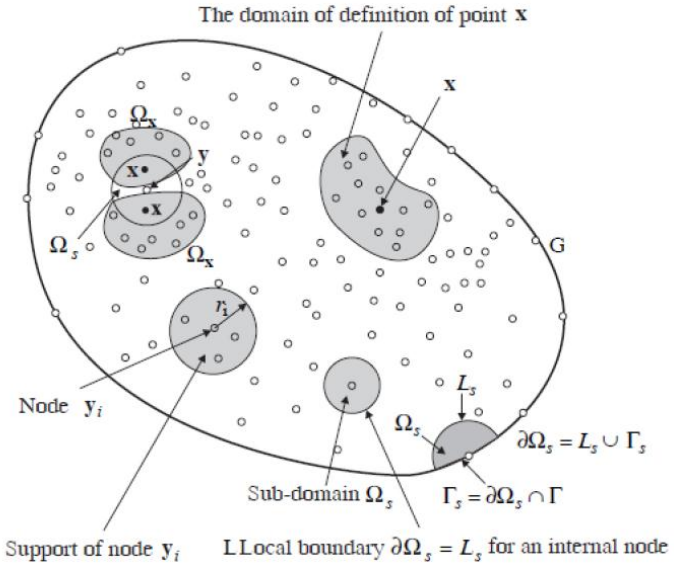

Figure 2.9 Schematics of the MLPG method (Atluri, 1998a).

In Figure 2.9, the local domains, the supports of nodes, the domain of definition of the MLS approximation for the trial function at a point, and the domain of influence of a source point (node) are defined as follows:

1. The domain of definition of the MLS approximation, Ω_x , for the trial function at any point x is the domain over which the MLS is defined, i.e., Ω_x covers all the nodes whose weight functions do not vanish at **x**.

2. The domain of influence for source point **y** is the union of all $\Omega_x, \forall \mathbf{x} \in \Omega_s$, (taken to be a circular domain of radius r_0 in this paper).

3. The support of source point $\mathbf{y}_i$ is a sub-domain (taken to be a circle of radius r_i for

convenience) in which the weight function w_i corresponding to this node is nonzero. Note that the "support" of $\mathbf{y}_i$ is distinct and different from the "domain" of influence of $\mathbf{y}_i$.

It is easy for the moving least squares approximation to attain a higher order of continuity for the shape functions and the trial function by constructing a more continuous weight function. A simple way is to use higher order spline functions.

The size of support, r_i, of the weight function w_i associated with node *i* should be chosen such that r_i should be large enough to have sufficient number of nodes covered in the domain of definition of every sample point $(n \geq m)$ in order to ensure the regularity of **A**. A very small r_i may result in a relatively large numerical error in using Gauss numerical quadrature to calculate the entries in the system matrix. On the other hand, r_i should also be small enough to maintain the local character of the MLS approximation.

2.3 The Local Weak Form (LWF) for Poison's Equation

The linear Poisson's equation is used to demonstrate the formulation. The Poisson's equation can be written as

$$\nabla^2 u(\mathbf{x}) = p(\mathbf{x}) \qquad \mathbf{x} \in \Omega \tag{2.55}$$

where $p(\mathbf{x})$ is a given source function, and the domain Ω is enclosed by $\Gamma = \Gamma_u \cup \Gamma_q$, with boundary conditions**:**

$$u = \bar{u} \qquad on \qquad \Gamma_u, \tag{2.56}$$

$$\frac{\partial u}{\partial n} \equiv q = \bar{q} \qquad on \qquad \Gamma_q, \tag{2.57}$$

where u and q are the prescribed potential and normal flux, respectively, on the boundary Γ_u, and on the boundary Γ_q, and ***n*** is the outward normal direction to the boundary Γ.

In the Galerkin finite element, and element free Galerkin methods, which are based on the global Galerkin formulation, one uses the global weak form over the entire domain Ω to solve the problem numerically. In the present LWF, we start from a weak form over a local sub-domain Ω_s, and use the MLS approximation to develop a true meshless method, where the local sub-domain Ω_s is located entirely inside the global domain Ω. The local sub-domain Ω_s is conveniently taken to be a sphere (in three-dimensions), or a circle (in two-dimensions) centered at a point $\mathbf{x}_i$ in question. A generalized local weak form of the differential equation (2.55) and the boundary conditions equation (2.56), equation (2.57), for $\mathbf{x}_i = (x^i, y^i) \in \Omega_s^i$ can be written as follows:

$$\int_{\Omega_s^i} \left(\nabla^2 u - p\right) v_i d\Omega - \alpha \int_{\Gamma_{su}^i} (u - \bar{u}) v_i d\Gamma = 0, \tag{2.58}$$

where $i = 1, 2, ..., \mathrm{N}$, u is the trial function, v is the test function, and Γ_{su} is a part of the boundary $\partial\Omega_s$ of Ω_s, over which the essential boundary conditions are specified. In general, $\partial\Omega_s = \Gamma_s \cup L_s$, with Γ_s being a part of the local boundary located on the global boundary and L_s being the other part of the local boundary over which no boundary condition is specified, i.e., $\Gamma_s = \partial\Omega_s \cup \Gamma$ and $L_s = \partial\Omega_s - \Gamma_s$ (see Figure 2.9). If the sub-domain Ω_s is located entirely within the global domain Ω, and there is no intersection between the local boundary $\partial\Omega_s$ and the global boundary Γ, the boundary integral over Γ_{su} vanishes. In equation (2.58), a penalty parameter $\alpha \gg 1$ is used to impose the essential boundary conditions, as the MLS approximation will be used to approximate the trial function, and it is not easy to directly impose the essential boundary conditions, *a priori*, in the MLS approximation. Using $(\nabla^2 u)v = u_{,ii} v = (u_{,i} v)_{,i} - u_{,i} v_{,i}$, and the divergence theorem, yields the following expression:

$$\int_{\partial\Omega_s^i} u_{,i} n_i v_i d\Gamma - \int_{\Omega_s^i} (u_{,i} v_{,i} + p v_i) d\Omega - \alpha \int_{\Gamma_{su}^i} (u - \bar{u}) v_i d\Gamma = 0, \tag{2.59}$$

in which $\partial\Omega_s$ is the boundary of the sub-domain Ω_s and n is the outward unit normal to the boundary $\partial\Omega_s$. It should be noted that equation (2.59) holds, irrespective of the size

and shape of $\partial\Omega_s$. This is an important observation which forms the basis for the following development. We now deliberately choose a simple regular shape for Ω_s and thus for $\partial\Omega_s$. The most regular shape of a sub-domain should be an *n*-dimensional sphere for a boundary value problem defined on an *n*-dimensional space. Thus, an *n*-dimensional sphere (or a part of an *n*-dimensional sphere for a boundary node), is chosen in our implementation (see Figure 2.9). In the following implementation, the Petrov-Galerkin method is used. Unlike in the conventional Galerkin method in which the trial and test functions are chosen from the same space, the Petrov-Galerkin method uses the trial functions and the test functions from different spaces.

In particular, the test functions need not vanish on the boundary where the essential boundary conditions are specified. Imposing the natural boundary condition, $q = \bar{q}$, and noticing that $u_{,i} n_i = \dfrac{\partial u}{\partial n} \equiv q$ in equation (2.59), we obtain:

$$\int_{L_s^i} q v_i d\Gamma + \int_{\Gamma_{su}^i} q v_i d\Gamma + \alpha \int_{\Gamma_{sq}^i} \bar{q} v_i d\Gamma - \int_{\Omega_s^i} (u_{,i} v_{,i} + p v_i) d\Omega - \alpha \int_{\Gamma_{su}^i} (u - \bar{u}) v_i d\Gamma = 0, \quad (2.60)$$

in which Γ_{sq} is a part of $\partial\Omega_s$, over which the natural boundary condition, $q = \bar{q}$, is specified. For a sub-domain located entirely within the global domain, there is no intersection between $\partial\Omega_s$ and Γ, $L_s = \partial\Omega_s$ and the integrals over Γ_{su} and Γ_{sq} vanish.

In order, we deliberately select a test function *v*, such that it vanishes over L_s, the circle (for an internal node) or the circular arc (for a node on the global boundary Γ). This can be easily accomplished by using the weight function in the MLS approximation as the test function as well, with the radius r_i of the support of the weight function being replaced by the radius r_0 of the local domain Ω_s, such that the test function vanishes on a circle of radius r_0. Using this test function and rearranging equation (2.60), we obtain the following local symmetric weak form (LSWF):

$$\int_{\Omega_s^i} u_{,i} v_{,i} d\Omega + \alpha \int_{\Gamma_{su}^i} u v_i d\Gamma - \int_{\Gamma_{su}^i} q v_i d\Gamma = \int_{\Gamma_{sq}^i} \bar{q} v_i d\Gamma + \alpha \int_{\Gamma_{su}^i} \bar{u} v_i d\Gamma - \int_{\Omega_s^i} p v_i d\Omega. \quad (2.61)$$

With equation (2.61) for any point **x**, the problem becomes as if we were dealing with a localized boundary value problem over an *n*-dimensional sphere Ω_s. The radius of the sphere will affect the solution. In the present formulation, the equilibrium equation and the boundary conditions are satisfied, *a posteriori*, in all local sub-domains and on their Γ_s, respectively. Theoretically, as long as the union of all local domains covers the global domain, i.e., $\cup\Omega_s \supset \Omega$, the equilibrium equation and the boundary conditions will be satisfied, *a posteriori*, in the global domain Ω and on its boundary Γ, respectively. However, from our computation, the present formulation yields a very satisfactory result, even when the union of all local domains does not cover the global domain.

2.4 The Meshless Local Petrov-Galerkin (MLPG) Method

The MLPG approach was first proposed by Atluri and Zhu (1998a; 1998b) for solving linear potential problems. The MLPG approach uses either a local symmetric weak form (LSWF), or a local unsymmetric weak form (LUSWF) as in the case of the Local Boundary Integral Equation (LBIE) method. The generality of the MLPG, based on either symmetric or unsymmetric weak forms, and a variety of meshless trial and test functions, is discussed comprehensively by Atluri and Shen (2002b; 2002c). The MLPG is a truly meshless method, which involves not only a meshless interpolation for the trial functions, but also a meshless integration of the weak-form. In the conventional Galerkin method, the trial and test functions are chosen from the same function-space. In MLPG, the nodal trial and test functions can be different: the nodal trial function may correspond to any one of MLS, PU, Shepard function, or RBF types of interpolations; and the test function may be totally different, and may correspond to any one of MLS, PU, Shepard function, RBF, a Heaviside step function, a Dirac delta function, the Gaussian weight function of MLS, a special form of the fundamental solution to the differential equation, or any other convenient function, in the support domain, Ω_{te}, of the test function.

The concept of the MLPG method:

The mainly idea,

1. Use mainly Petrov-Galerkin type approximations, i.e. the nodal trial and test functions are decidedly different.

2. Simplify the integrand in the weak-form over the local, nodal-based test function domain.

3. Avoid the domain integral in the weak-form, if possible, over the local, nodal based test-function domain.

The second idea implies that we should choose a suitable interpolation that makes the nodal (trial function) shape function simple.

The third idea means that we should choose a suitable local nodal-based test function over a regular-shaped local domain, in order to make the domain integral in the weak-form disappear.

Six different MLPG methods.

Six different nodal-based local test functions, shown in Table 2.1, are also selected, leading to six different MLPG methods as MLPG1, MLPG2, MLPG3, MLPG4, MLPG5, and MLPG6, respectively:

(1) the test function over Ω_s is the same as the weight function in the MLS approximation. The resulting Meshless Local Petrov-Galerkin Method is denoted as MLPG1;

(2) the test function over Ω_s is the collocation Dirac's Delta function (collocation method). The resulting Meshless Local Petrov-Galerkin Method is denoted as MLPG2;

(3) the test function over Ω_s is the same as the error function in the differential equation, using discrete least squares. The resulting Meshless Local Petrov-Galerkin Method is denoted as MLPG3;

(4) the test function over Ω_s is the modified fundamental solution to the differential equation (LBIE). The resulting Meshless Local Petrov-Galerkin Method is denoted as MLPG4; thus, MLPG4 is synonymous with the Local Boundary Integral Equation (LBIE) method (Atluri, 1998a).

(5) the test function over Ω_s is the Heaviside step function (constant over each local sub-domain Ω_s). The resulting Meshless Local Petrov-Galerkin Method is denoted as MLPG5;

(6) the test function over Ω_s is identical to the trial function (Galerkin method). The resulting Meshless Local Petrov-Galerkin Method is denoted as MLPG6.

Table 2.1 Meshless local Petrov-Galerkin (MLPG) methods: a summary of the variety of MLPG methods (Atluri, 2002a).

Methods	**Test function in Ω_{te}**	**Local weak form over each Ω_{te}**	**Relation between Ω_{te} and Ω_{tr}**	**Integral to evaluate the weak-form**
MLPG1	MLS weight function	LSWF	$\Omega_{te} < \Omega_{tr}$	Domain integral
MLPG2	Kronecker Delta $\delta(\mathbf{x}, \mathbf{x}_I)$	LUSWS1	Ω_{te} can be arbitrary	None
MLPG3	Least square $\phi^I_{,ii}(\mathbf{x})$	LUSWS1	$\Omega_{te} = \Omega_{tr}$ for the modified MLPG3: $\Omega_{te} < \Omega_{tr}$	Domain integral
MLPG4	Fundamental solution u^*	LUSWS2	$\Omega_{te} < \Omega_{tr}$	Singular boundary integral
MLPG5	Heaviside step	LSWF	$\Omega_{te} < \Omega_{tr}$	Regular boundary integral
MLPG6	Same as the trial function	LSWF	$\Omega_{te} = \Omega_{tr}$	Domain integral

As a known test function is used in the local weak form (LWF), the use of the LWF for one point and here for one domain Ω_s will yield only one algebraic equation. It is noted that the trial function u within the sub-domain Ω_s, in the interpolations without Kronecker Delta properties, is determined by the fictitious nodal values $\hat{u}^I$, within the domain of definition for all points $\mathbf{x}$ falling within Ω_s. One can obtain as many equations as the number of nodes. Hence, we need as many local domains Ω_s as the number of nodes in the global domain, in order to obtain as many equations as the number of unknowns.

2.5 The Divergence Theorem

Another form of the Fundamental Theorem of Calculus is the Divergence Theorem, often called Gauss's Theorem (Karl Friedrich Gauss (1777-1855)). It is exactly analogous to Green's Theorem in the plane: The surface integral of a vector field **F** over the boundary of a solid region Ω in three-dimensional space is equal to the triple integral over Ω of the divergence of **F**.

Theorem 1 (The Divergence Theorem) Let $D \in \mathbb{R}^3$ be an open connected region, and let $\mathbf{F}: D \to \mathbb{R}^3$ be a smooth vector field. For any solid region Ω contained in D whose boundary $\partial\Omega$ is piecewise smooth,

$$\int_{\Omega} \nabla \cdot \mathbf{F} d\Omega = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} d\Gamma. \tag{2.62}$$

where

n is the outward unit normal direction to the boundary $\partial\Omega$,

Γ is the whole surface that bounds the volume Ω.

2.6 The Crank-Nicolson Method

The Crank-Nicolson Method provides an alternative implicit scheme that is second-order-accurate in both space and time. To provide this accuracy, difference approximations are developed at the midpoint of the time increment, as seen in Figure 2.10.

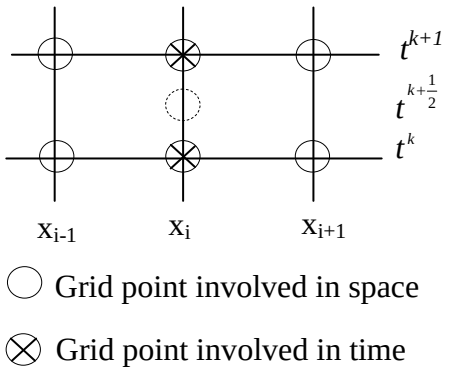

Figure 2.10 A computational molecule for the Crank-Nicolson method.

To do this, the temporal first derivative can be approximated at $t^{k+\frac{1}{2}}$ by

$$\left.\frac{\partial U}{\partial t}\right|^{k+\frac{1}{2}} \approx \frac{U_i^{k+1} - U_i^k}{\Delta t}. \tag{2.63}$$

The second derivative in space can be determined at the midpoint by averaging the difference approximations at the beginning (t^k) and at the end (t^{k+1}) of the time increment:

$$\left.\frac{\partial^2 U}{\partial x^2}\right|^{k+\frac{1}{2}} \approx \frac{1}{2(\Delta x)^2}((U_{i+1}^{k+1} - 2U_i^{k+1} + U_{i-1}^{k+1}) + (U_{i+1}^k - 2U_i^k + U_{i-1}^k)), \tag{2.64}$$

and transforming each component into the following:

$$\left.\frac{\partial U}{\partial x}\right|^{k+\frac{1}{2}} \approx \frac{1}{2}\left(\frac{(U_{i+1}^{k+1} - U_{i-1}^{k+1})}{2(\Delta x)} + \frac{(U_{i+1}^k - U_{i-1}^k)}{2(\Delta x)}\right) \tag{2.65}$$

$$U_i^{k+\frac{1}{2}} \approx \frac{1}{2}(U_i^{k+1} + U_i^k). \tag{2.66}$$