A Review of Some Meshless Methods to Solve Partial Differential Equations

C. Armando Duarte *

TICAM - Texas Institute for Computational and Applied Mathematics

The University of Texas at Austin

Taylor Hall 2.400

Austin, Texas, 78712, U.S.A.

TICAM Report 95-06

1 Introduction

Many problems of practical importance, like crack propagation, fragmentation and large deformations, are characterized by a continuous change in the geometry of the domain under analysis. The analysis of this class of problems by, e.g., conventional finite element and finite difference methods can be a cumbersome and expensive task. The analysis of large deformations problems by the finite element method, e.g., may require the continuous remeshing of the domain to avoid the breakdown of the calculation due to excessive mesh distortion. Even in problems where only a few meshes are needed for an analysis, mesh generation can be a far more time-consuming and expensive task than the construction and solution of the discrete set of equations. Meshless methods provide an attractive alternative for the analysis of this class of problems.

In the next sections, some of the meshless methods found in the literature are reviewed and the connection among them investigated. In this review a method is considered meshless if the basic equations governing the discrete model of a boundary-value problem do not depend upon the availability of a well defined mesh. Some meshless methods do have a weak dependence

*Research Assistant, TICAM. e-mail: armando@ticam.utexas.edu

on background meshes that may support numerical quadrature calculations. Such methods are still regarded as meshless if there are no fixed connectivities among the nodes. The underlying approximation (interpolation) techniques of all the methods for solving PDEs reviewed in this paper are also discussed. This review does not cover all existing meshless methods but only the most representatives. The interested reader should consult the references at the end of this paper for information on other existing meshless methods. The paper is organized as follows.

Following this introduction, Section 2 reviews the Moving Least Squares Method (MLSM) which is the approximation technique used in the Diffuse Element Method (DEM) and in the Element Free Galerkin Method (EFGM). These two methods are discussed and compared in Section 2. Section 3 reviews a generalized finite difference method proposed by Liszka [17]. It is shown that the Liszka's method and the MLSM are equivalent under some conditions. Section 4 discusses interpolation by kernel estimates and the Smoothed Particle Hydrodynamics Method (SPH). A comparison of the SPH method and EFGM is also presented. The construction and properties of Daubechies scaling functions and wavelet functions is the subject of Section 5. The solution of PDEs using these functions is also presented.

2 The Diffuse Element Method and The Element-Free Galerkin Method

2.1 Introduction

Moving least squares approximations [14] play a fundamental role in the diffuse element method [27] and in the element-free Galerkin method [3]. In this section, the Moving Least Squares Method (MLSM) is reviewed and some of its approximation properties are investigated.

2.2 The Moving Least Squares Method

Suppose that a continuous function $u : \bar{\Omega} \rightarrow \Re$, $\bar{\Omega} \subset \Re^d, d = 1, 2$ or 3, is to be approximated and that its values u_I at the points $\mathbf{x}_I \in \bar{\Omega}, I = 1, \ldots, N$, are given. In the MLSM, a *global* approximating function, Gu, of u is constructed by first forming, at each point $\mathbf{y} \in \bar{\Omega}$, a *local* weighted least squares approximation, $L_{\mathbf{y}}u$, defined in terms of some basis $\{P_i\}_{i=1}^n, n \leq N$. It is assumed that the basis $\{P_i\}_{i=1}^n$ has the following properties:

(i) $P_1 \equiv 1$

(ii) $P_i \in C^m(\Omega),\ i = 1, \ldots, n$

(iii) $\{P_i\}_{i=1}^n$ is linearly independent over some set n of the given N points in $\bar{\Omega}$.

The local approximation $L_{\mathbf{y}}u$, for each point $\mathbf{y} \in \bar{\Omega}$, is defined as

$$(L_{\mathbf{y}}u)(\mathbf{x}) := \sum_{i=1}^{n} a_i(\mathbf{y}) P_i(\mathbf{x}) \tag{2.1}$$

The coefficients $a_i(\mathbf{y})$ are found by solving the following problem:
Find $\mathbf{a}^*(\mathbf{y}) \in \Re^{\mathbf{n}}$ such that $\forall\, \mathbf{a}(\mathbf{y}) \in \Re^{\mathbf{n}}$.

$$\begin{aligned} J(\mathbf{a}^*) &= (u - \sum_i^n a_i^* P_i, u - \sum_i^n a_i^* P_i)_{\mathbf{y}} \\ &\leq (u - \sum_i^n a_i P_i, u - \sum_i^n a_i P_i)_{\mathbf{y}} \end{aligned} \tag{2.2}$$

where $(.,.)_{\mathbf{y}}$ is a weighted inner product that depends on the point $\mathbf{y}$ and is defined by [14]

$$(u, v)_{\mathbf{y}} := \sum_I^N u(\mathbf{x}_I) W_I(\mathbf{y}) v(\mathbf{x}_I) \tag{2.3}$$

where $W_I(\mathbf{y}), I = 1, N$, are weight functions, $W_I(\mathbf{y}) > 0, \forall\, \mathbf{y} \in \bar{\Omega}$ and $u, v \in C^0(\bar{\Omega})$.

Thus

$$J(\mathbf{a}^*) = \sum_{I=1}^{N} W_I(\mathbf{y})[u(\mathbf{x}_I) - \sum_i^n a_i^* P_i]^2 \tag{2.4}$$

Inequality (2.2) implies that

$$(u - \sum_{i}^{n} a_i^* P_i, P_j)_{\mathbf{y}} = 0 \qquad j = 1, \ldots, n \tag{2.5}$$

Thus

$$\sum_{i}^{n} (P_i, P_j)_{\mathbf{y}} a_i^* = (u, P_j)_{\mathbf{y}} \qquad j = 1, \ldots, n \tag{2.6}$$

With the inner product defined as in (2.3) and the initial assumption that the values of the function u at $\mathbf{x}_I \in \bar{\Omega}, I = 1, N$ are known, the right hand side of (2.6) can be evaluated at each point $\mathbf{y} \in \bar{\Omega}$.

Theorem 2.1 *[14] The matrix $(P_i, P_j)_{\mathbf{y}}$ is positive definite $\forall\, \mathbf{y} \in \bar{\Omega}$ and the coefficients $a_i(\mathbf{y})$ in (2.1) are uniquely determined.*

Proof:

The matrix $\mathbf{A}(\mathbf{y}) := (P_i, P_j)_{\mathbf{y}}$ can be written as

$$\begin{bmatrix} P_1(\mathbf{x}_1) & P_1(\mathbf{x}_2) & \ldots & P_1(\mathbf{x}_N) \\ P_2(\mathbf{x}_1) & P_2(\mathbf{x}_2) & \ldots & P_2(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ P_n(\mathbf{x}_1) & P_n(\mathbf{x}_2) & \ldots & P_n(\mathbf{x}_N) \end{bmatrix} * \begin{bmatrix} W_1(\mathbf{y}) & 0 & \ldots & 0 \\ 0 & W_2(\mathbf{y}) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & W_N(\mathbf{y}) \end{bmatrix} *$$

$$* \begin{bmatrix} P_1(\mathbf{x}_1) & P_2(\mathbf{x}_1) & \ldots & P_n(\mathbf{x}_1) \\ P_1(\mathbf{x}_2) & P_2(\mathbf{x}_2) & \ldots & P_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_1(\mathbf{x}_N) & P_2(\mathbf{x}_N) & \ldots & P_n(\mathbf{x}_N) \end{bmatrix} := \mathbf{F}\mathbf{W}\mathbf{F}^T$$

The rows of $\mathbf{F}$ are linearly independent, by hypothesis (iii), thus $rank(\mathbf{F}) = n$ and $rank(\mathbf{F}^T) = n$. Also $\mathbf{W}$ is positive definite since $W_I(\mathbf{y}) > 0$. These imply that $\mathbf{A}$ is positive-definite $\Box$

For future reference, we rewrite (2.6) in the following form

$$\mathbf{A}(\mathbf{y})\mathbf{a}(\mathbf{y}) = \mathbf{B}(\mathbf{y})\mathbf{u} \tag{2.7}$$

where

$$
\begin{aligned}
A_{ij} &= (P_i, P_j)_{\mathbf{y}} \\
\mathbf{a}(\mathbf{y}) &= \left\{ \begin{array}{c} a_1(\mathbf{y}) \\ a_2(\mathbf{y}) \\ \vdots \\ a_n(\mathbf{y}) \end{array} \right\} \\
\mathbf{u} &= \left\{ \begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_N \end{array} \right\} \\
\mathbf{B}(\mathbf{y}) &= \left[\; W_1(\mathbf{y})\mathbf{P}(\mathbf{x}_1) \,\middle|\, W_2(\mathbf{y})\mathbf{P}(\mathbf{x}_2) \,\middle|\, \ldots \,\middle|\, W_N(\mathbf{y})\mathbf{P}(\mathbf{x}_N) \; \right] \\
\mathbf{P}(\mathbf{x}_I) &= \left\{ \begin{array}{c} P_1(\mathbf{x}_I) \\ P_2(\mathbf{x}_I) \\ \vdots \\ P_n(\mathbf{x}_I) \end{array} \right\}
\end{aligned}
$$

Thus

$$
a_k(\mathbf{y}) = \sum_I^N \sum_j^n A_{kj}^{-1} B_{jI}(\mathbf{y}) u_I \tag{2.8}
$$

From (2.1),

$$
\begin{aligned}
(L_{\mathbf{y}} u)(\mathbf{x}) &= \sum_k^n a_k(\mathbf{y}) P_k(\mathbf{x}) \\
&= \sum_I^N \sum_j^n \sum_k^n P_k(\mathbf{x}) A_{kj}^{-1}(\mathbf{y}) B_{jI}(\mathbf{y}) u_I \\
&= \sum_I^N \Phi_I^{\mathbf{y}}(\mathbf{x}) u_I
\end{aligned}
$$

where

$$
\Phi_I^{\mathbf{y}}(\mathbf{x}) := \sum_j^n \sum_k^n P_k(\mathbf{x}) A_{kj}^{-1}(\mathbf{y}) B_{jI}(\mathbf{y}) \tag{2.9}
$$

If $\mathbf{x} = \mathbf{y}$, we can drop the superscript $\mathbf{y}$ and write

$$\Phi_I(\mathbf{y}) := \sum_j^n \sum_k^n P_k(\mathbf{y}) A_{kj}^{-1}(\mathbf{y}) B_{jI}(\mathbf{y}) \tag{2.10}$$

Up to this point only the local approximation, $L_{\mathbf{y}}u$, of u has been constructed. The global approximating function, Gu, of u is defined by

$$(Gu)(\mathbf{y}) := (L_{\mathbf{y}}u)(\mathbf{y}) = \sum_i^n a_i(\mathbf{y}) P_i(\mathbf{y}) \qquad \forall\, \mathbf{y} \in \bar{\Omega} \tag{2.11}$$

$$(Gu)(\mathbf{y}) = \sum_I^N \Phi_I(\mathbf{y}) u_I \tag{2.12}$$

where $\Phi_I(\mathbf{y})$ was defined in (2.10).

Theorem 2.2 *[14] If P_i, $i = 1, \ldots, n$, $\in C^m(\Omega)$ and W_I, I=1,...,N, $\in C^l(\Omega)$ then $(Gu)(\mathbf{y})$ as defined above $\in C^{min(m,l)}(\Omega)$*

Proof:

From (2.7) we have that $\mathbf{A}(\mathbf{y})\mathbf{a}(\mathbf{y}) = \mathbf{B}(\mathbf{y})\mathbf{u}$ thus

$$\mathbf{A}_{,i}\mathbf{a} + \mathbf{A}\mathbf{a}_{,i} = \mathbf{B}_{,i}\mathbf{u}$$

$$\mathbf{a}_{,i} = \mathbf{A}^{-1}[\mathbf{B}_{,i}\mathbf{u} - \mathbf{A}_{,i}\mathbf{a}]$$

The above expression makes sense since by Theorem 2.1, $\mathbf{A}^{-1}$ exists and by the assumption on the differentiability of W_I, $\mathbf{B}_{,i}$ and $\mathbf{A}_{,i}$ also exist. Higher order derivatives of $\mathbf{a}$ can be computed as above, for example

$$\mathbf{a}_{,ii} = \mathbf{A}^{-1}[\mathbf{B}_{,ii}\mathbf{u} - 2\mathbf{A}_{,i}\mathbf{a}_{,i} - \mathbf{A}_{,ii}\mathbf{a}]$$

which also makes sense if $l \geq 2$. So we conclude that $\mathbf{a} \in C^l(\Omega)$. From the definition of Gu, the assumption on the differentiability of P_i and the above we conclude that $Gu \in C^{min(m,l)}(\Omega)\square$

For future reference, we consider more closely the case $n = 1$ in (2.1). Equation (2.6) gives

$$a_1(\mathbf{y}) = \frac{(1, u)_{\mathbf{y}}}{(1, 1)_{\mathbf{y}}}$$

Using the discrete inner product defined in (2.3), the above inner product can be computed as

$$a_1(\mathbf{y}) = \frac{\sum_I^N W_I(\mathbf{y})u(\mathbf{x}_I)}{\sum_J^N W_J(\mathbf{y})} := \sum_I^N v_I(\mathbf{y})u(\mathbf{x}_I)$$

where

$$v_I(\mathbf{y}) = \frac{W_I(\mathbf{y})}{\sum_J^N W_J(\mathbf{y})} \tag{2.13}$$

The global approximating function Gu, in this case, is called the Shephard interpolant [14] and is given by

$$(Su)(\mathbf{y}) = \sum_I^N v_I(\mathbf{y})u(\mathbf{x}_I) \tag{2.14}$$

It is clear that $v_I(\mathbf{y})$ satisfy

(i) $0 < v_I(\mathbf{y}) < 1$

(ii) $\sum_I^N v_I(\mathbf{y}) = 1 \qquad \forall\, \mathbf{y} \in \Re$

Theorem 2.3 *If $W_I(\mathbf{y}) \in C^\infty(\Omega)$, then the collection of functions $v_I(\mathbf{y})$ defined in (2.13) represents a partition of unity.*

Proof: Immediate □

From the above properties of $v_I(\mathbf{y})$ it can also be noted that, $\forall\ \mathbf{y} \in \bar{\Omega}$, $(Su)(\mathbf{y})$ is just a weighted average of the given values of u.

2.2.1 General Comments

The global moving least squares approximating function Gu defined in the previous section may be computationally too expensive to be used in practice. At each point $\mathbf{y}$, a new system of equations has to be solved for N right hand sides. Equivalently, one can orthonormalize the functions P_i with respect to the weighted inner product, but, as before, the orthonormalization has to be done at each point $\mathbf{y}$. In practice, the weights W_I are defined as translations of a single function W, i.e.,

$$W_I(\mathbf{y}) := W(\mathbf{y} - \mathbf{x}_I) \tag{2.15}$$

and W is chosen to be a function satisfying the following:

(i) The support of W_I, $supp(W_I)$, is an open disc (ball) of radius ρ centered at the origin.

(ii) For any $\mathbf{y}, \in \bar{\Omega}$ there exist indices $I_1, \ldots, I_k, k \geq n$, *dependent on* $\mathbf{y}$, such that

$$\mathbf{y} \in \bigcap_{j=1}^{k} supp(W_{Ij})$$

and $\mathbf{x}_{I1}, \ldots, \mathbf{x}_{Ik}$ contains a subset of n points on which $\{P_i\}_{i=1}^n$ is linearly independent.

Condition (i) above implies that the only data points entering in the calculation of $(Gu)(\mathbf{y})$ are those satisfying $|\mathbf{x}_I - \mathbf{y}| < \rho$ and, consequently, only some columns of the matrix $\mathbf{B}$ defined in (2.7) will not be identically zero. Condition (ii) is imposed to guarantee the well poseness of the system A_{ij} defined in (2.7) for all $\mathbf{y} \in \bar{\Omega}$.

Note that a necessary condition for (ii) to be satisfied is that $k \geq n, k$ defined in (ii). Nonetheless this is *not* a sufficient condition. This fact limits in practice the use of MLS interpolants to low polynomial order basis functions $\{P_i\}_{i=1}^n$, specially in 2D and 3D.

Theorem 2.4 *[14] The operator G defined in (2.11) is a transformation from $\tilde{C}^0(\bar{\Omega})$ into M, where $span\{P_i\}_{i=1}^n \subset M \subset C^0(\bar{\Omega})$, and $\tilde{C}^0(\bar{\Omega}) \subset C^0(\bar{\Omega})$. The tilde in $\tilde{C}^0(\bar{\Omega})$ indicates that two functions that do not differ on a set of given points $\{\mathbf{x}_I\}_{I=1}^N$ shall be deemed equivalent.*

Proof. The system of equations (2.7) can be written as

$$\mathbf{F}\mathbf{W}\mathbf{F}^T\mathbf{a} = \mathbf{F}\mathbf{W}\mathbf{u}$$

where $\mathbf{F}$ and $\mathbf{W}$ were defined in the proof of Theorem 2.1.

For

$$\mathbf{u} = \mathbf{P}^k = \left\{ \begin{array}{c} P_k(\mathbf{x}_1) \\ P_k(\mathbf{x}_2) \\ \vdots \\ P_k(\mathbf{x}_N) \end{array} \right\} \equiv \mathbf{F}^T_{\cdot,k} = the\ k^{th}\ column\ of\ \mathbf{F}^T$$

it can be seen that the solution is

$$\mathbf{a} = \left\{ \begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{array} \right\}$$

where $a_i = \delta_{ik}$. Thus

$$(L_\mathbf{y} P_k)(\mathbf{x}) = \sum_{i=1}^{n} a_i(\mathbf{y}) P_i(\mathbf{x}) = P_k(\mathbf{x})$$

□

2.2.2 Numerical Experiments

In this section, the global shape functions Φ_I defined in (2.10) are plotted for the one dimensional case using the following set of parameters:

- Domain: $2.5 \leq x \leq 7.5$.
- Nodes: at $x_I = 2.5 + I * 0.5,\ I = 0, \ldots, 10$.
- Weights

$$W_\rho(|\mathbf{y} - \mathbf{x}_I|) = \begin{cases} \sqrt{4/\pi} \left(1.0 - \frac{|\mathbf{y}-\mathbf{x}_I|^2}{\rho^2}\right)^4 & for\ |\mathbf{y} - \mathbf{x}_I| < \rho \\ 0 & for\ |\mathbf{y} - \mathbf{x}_I| \geq \rho \end{cases}$$

where $\rho = \alpha * d$, $d = 0.5$ and α was chosen so that the conditions (i) and (ii) of Section 2.2.1 were satisfied. [27], [3] and [21] have used weighting functions built from the exponential function. Instead, we have, chosen the above weighting function because it gives shape functions Φ_I that can be numerically integrated more precisely. This property may be important if the functions Φ_I are to be used in the Galerkin method, like in the EFGM and DEM.

- Basis P_i:

  $P_i = \{1, x\}$ for the linear case.

  $P_i = \{1, x, x^2\}$ for the quadratic case.

  $P_i = \{1, x, x^2, x^3\}$ for the cubic case.

Figure 2.1 [22] shows the functions associated with node x_5 corresponding to different values of the parameter α and linear P_i. It can be observed that as the support of the weight functions W_I approaches the dimension of the support of the standard tent function, the function Φ, built using the MLSM, approaches the shape of a tent function. Figure 2.2 shows the function associated to node x_5 for the case $\alpha = 3.0$ and linear P_i. Figures 2.3 and 2.4 show all the functions Φ_I, $I = 0, \ldots, 10$ for the cases $\alpha = 1.7$ and $\alpha = 3.0$, respectively.

Figures 2.5 and 2.6 show the function associated with node x_5 when using a quadratic basis and for the cases $\alpha = 2.0$ and $\alpha = 3.0$, respectively. Figures 2.7 and 2.8 show all the functions Φ_I, $I = 0, \ldots, 10$ for the cases $\alpha = 2.0$ and $\alpha = 3.0$, respectively, when using a quadratic basis.

Figure 2.9 show the function associated to node x_5 when using a cubic basis and $\alpha = 3.0$. Figure 2.10 shows all the functions Φ_I corresponding to this case.

2.3 The Diffuse Approximation and The Diffuse Element Method

2.3.1 The Diffuse Approximation Method

Nayroles [27] presented the Diffuse Approximation Method (DAM) as a generalization of the finite element interpolation technique. The moving least squares method is not mentioned in the paper, but the procedure described by Nayroles et al. is exactly the same as in the MLSM. In the DAM a function $u(\mathbf{x})$ is locally approximated around a point $\mathbf{y}$ by

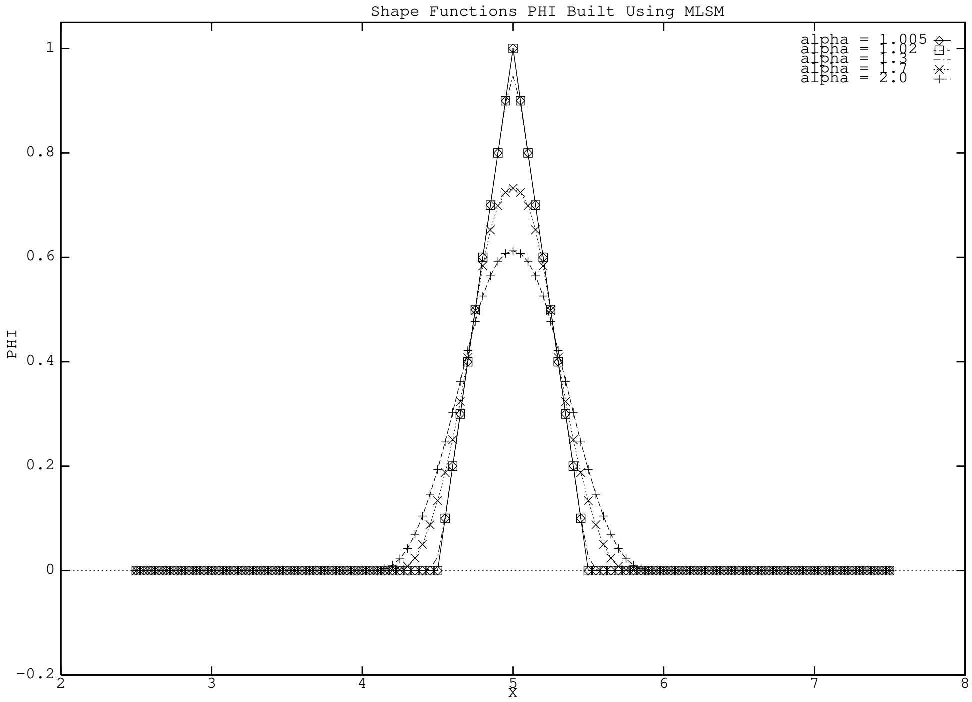

Figure 2.1: MLS functions Φ_5, corresponding to linear P_i and $1.005 \leq \alpha \leq 2.0$ [22]

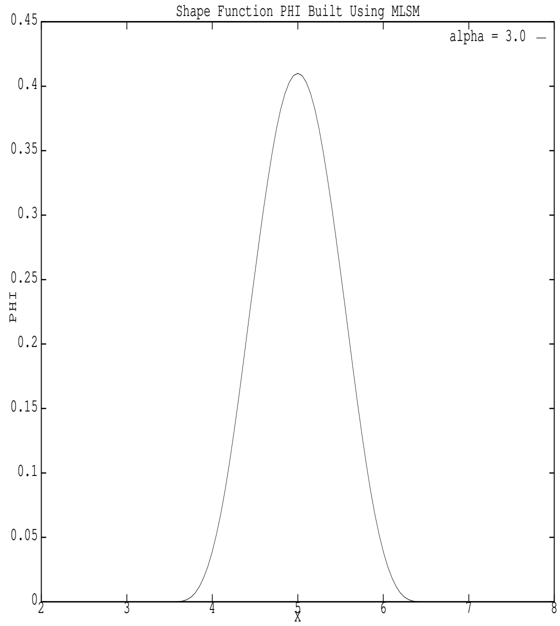

Figure 2.2: MLS function Φ_5, corresponding to linear P_i and $\alpha = 3.0$

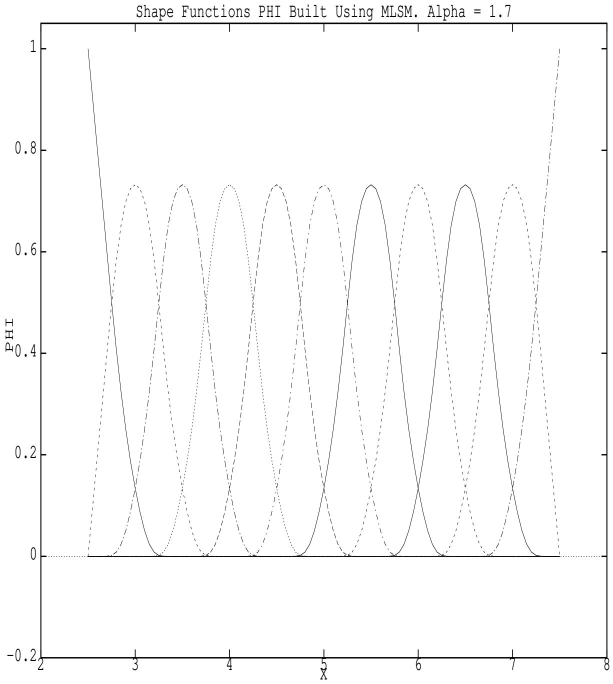

Figure 2.3: All MLS functions Φ_I corresponding to linear P_i and $\alpha = 1.7$

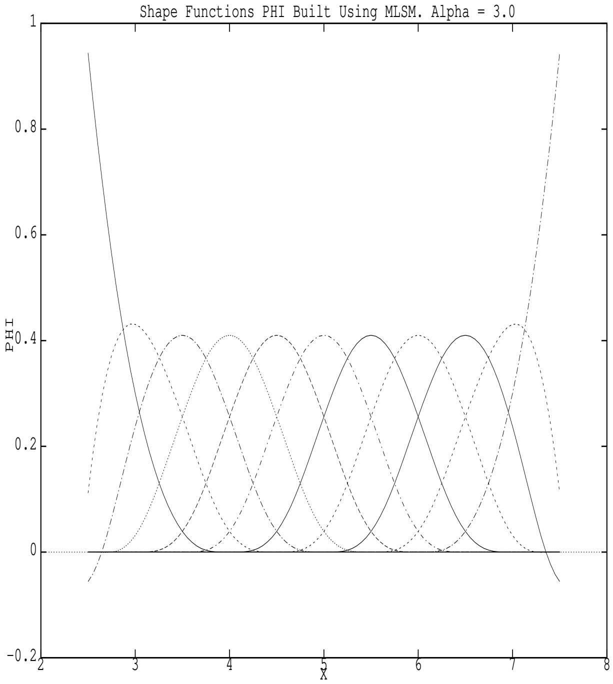

Figure 2.4: All MLS functions Φ_I corresponding to linear P_i and $\alpha = 3.0$

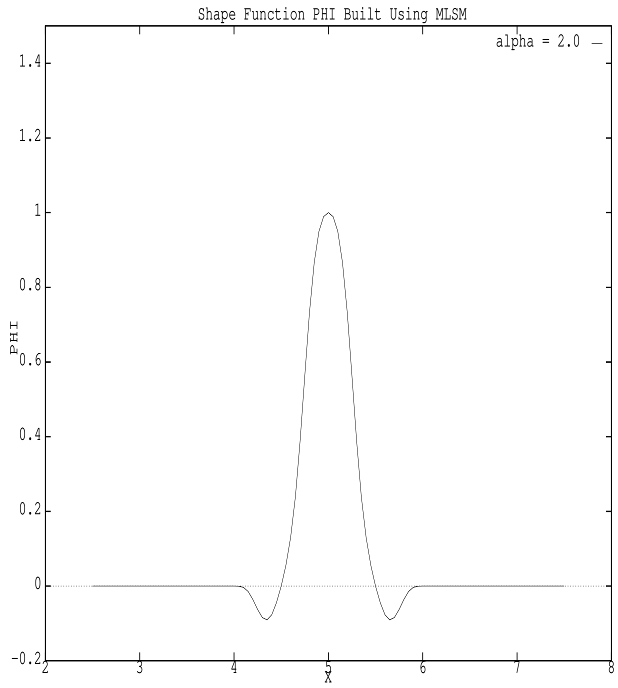

Figure 2.5: MLS function Φ_5 corresponding to quadratic P_i and $\alpha = 2.0$

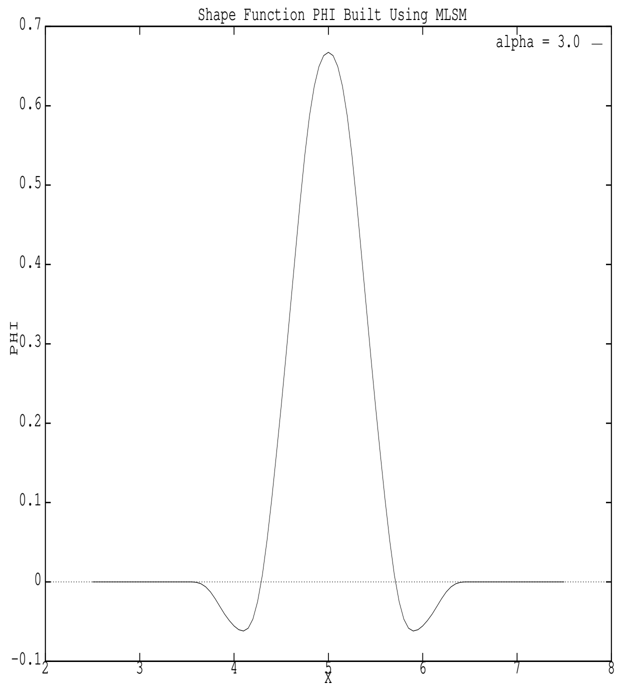

Figure 2.6: MLS function Φ_5 corresponding to quadratic P_i and $\alpha = 3.0$

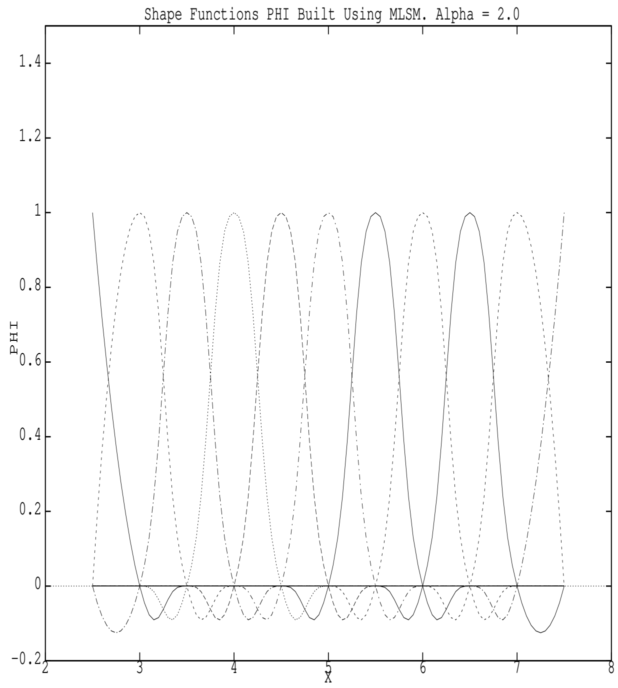

Figure 2.7: All MLS functions Φ_I corresponding to quadratic P_i and $\alpha = 2.0$

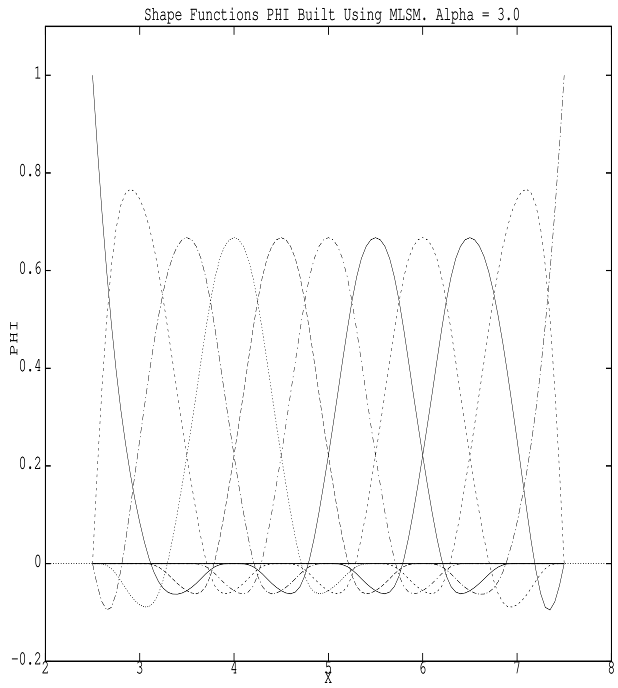

Figure 2.8: All MLS functions Φ_I corresponding to quadratic P_i and $\alpha = 3.0$

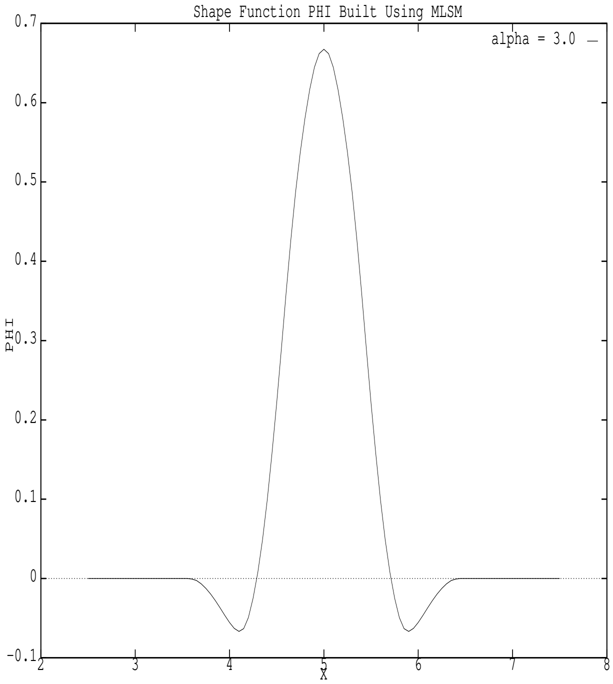

Figure 2.9: MLS function Φ_5 corresponding to cubic P_i and $\alpha = 3.0$

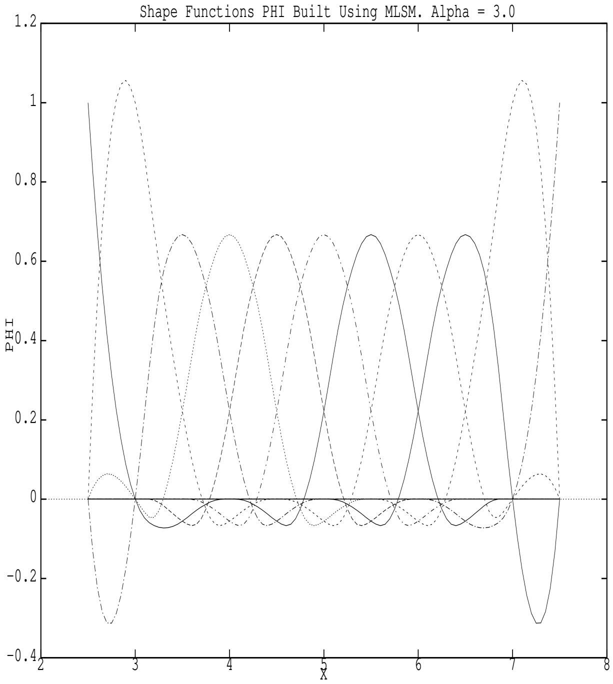

Figure 2.10: All MLS functions Φ_I corresponding to cubic P_i and $\alpha = 3.0$

$$u^{\mathbf{y}}(\mathbf{x}) = \sum_{i=1}^{n} a_i(\mathbf{y}) P_i(\mathbf{x}) \tag{2.16}$$

where $\{P_i\}_{i=1}^n$ is a polynomial basis. Note that the expression above is precisely equation (2.1).

As in the MLSM, the coefficients a_i are found by minimizing the following functional

$$J_{\mathbf{y}}(\mathbf{a}) = \sum_{I=1}^{N} W_{\mathbf{y}}(\mathbf{x}_I)[u(\mathbf{x}_I) - \sum_{i}^{n} a_i P_i]^2$$

which is the same as in (2.4), the only difference being that in (2.4) $W_I(\mathbf{y})$ is used instead of $W_{\mathbf{y}}(\mathbf{x}_I)$. But since in practice the weight functions used have the property that $W_I(\mathbf{y}) = W_{\mathbf{y}}(\mathbf{x}_I)$, the two functionals are identical.

Again, like in the MLSM, the global approximating function is obtained making $\mathbf{x} = \mathbf{y}$ in (2.16) and writing it simply as $u(\mathbf{x})$:

$$u(\mathbf{x}) = \sum_{i=1}^{n} a_i(\mathbf{x}) P_i(\mathbf{x}) \tag{2.17}$$

This is equivalent to the definition of Gu given in (2.11).

As in Section 2.2, we have that

$$\begin{aligned} u^{\mathbf{y}}(\mathbf{x}) &= \sum_{I}^{N} \sum_{j}^{n} \sum_{k}^{n} P_k(\mathbf{x}) A_{kj}^{-1}(\mathbf{y}) B_{jI}(\mathbf{y}) u_I \\ &= \sum_{I}^{N} \Phi_I^{\mathbf{y}}(\mathbf{x}) u_I \end{aligned}$$

If $\mathbf{x} = \mathbf{y}$ we get the global approximation, $u(\mathbf{x})$, written in terms of the functions Φ_I

$$u(\mathbf{x}) = \sum_{I}^{N} \Phi_I(\mathbf{x}) u_I \tag{2.18}$$

2.3.2 The Diffuse Element Method

Nayroles et al. pointed out many uses for the diffuse approximation. Among them was the solution of ordinary or partial differential equations leading to what they called the Diffuse Element Method (DEM) . The idea is to use the shape functions Φ_I as the basis functions of a finite dimensional subspace of, e.g., $H^1(\Omega)$ and them use a Galerkin method to find an approximate solution of the ODE or PDE in this subspace. Note that by Theorem 2.2, $\Phi_I \in H^1(\Omega)$ and we can build a subspace of $H^1(\Omega)$ using these functions. Some applications of the DEM for the solution of the Poisson equation and an elasticity problem in two dimensions can be found in the paper of Nayroles [27].

Among the stated advantages of the DEM over the FEM is that the diffuse approximation method provides a C^1 approximation of the solution while the FEM, in general, provides only a C^0 approximation. Nayroles et al. did not discuss the approximating properties of the functions Φ_I.

2.4 The Element-Free Galerkin Method

Belytschko [3] presented in 1994 a meshless method similar to the diffuse element method. The essential feature of the method, named Element-Free Galerkin Method (EFGM), is the use of an auxiliary grid, composed only of square elements, that covers the domain of the problem completely. This background grid is used in the EFGM to support numerical quadrature calculations [2].

The global functions Φ_I built using the MLSM do *not* satisfy the condition

$$\Phi_I(\mathbf{x}_J) = \delta_{IJ} \qquad I, J = 1, N$$

This fact has to be taken into account when imposing non-homogeneous Dirichlet boundary conditions. Nayroles [27] assumed that the above condition is satisfied. Belytschko et al. [3] used Lagrange multipliers to impose essential boundary conditions and showed through numerical experiments that only in this case the boundary conditions are exactly satisfied.

Another important difference between the DEM and the EFGM is that in the former the derivatives of the global functions Φ_I are computed as

$$\Phi_{I,i}(\mathbf{y}) = \sum_j^n \sum_k^n P_{k,i}(\mathbf{y}) A_{kj}^{-1}(\mathbf{y}) B_{jI}(\mathbf{y})$$

Note that the dependence of the coefficients of the approximation, a_i, defined in (2.16) with respect to $\mathbf{y}$ is neglected. In the EFGM this dependence is taken into account and the derivatives of Φ_I are computed as

$$\begin{aligned}\Phi_{I,i}(\mathbf{y}) &= \sum_j^n \sum_k^n \{P_{k,i}(\mathbf{y}) A_{kj}^{-1}(\mathbf{y}) B_{jI}(\mathbf{y})\} \\ &+ \sum_j^n \sum_k^n \{P_k(\mathbf{y}) A_{kj}^{-1}(\mathbf{y}) [B_{jI,i}(\mathbf{y}) - \sum_l^n \sum_m^n A_{jl,i}(\mathbf{y}) A_{lm}^{-1}(\mathbf{y}) B_{mI}(\mathbf{y})]\}\end{aligned}$$

Belytschko and colleges have used with success the EFGM to solve many problems in solid mechanics. The method seems particularly attractive for the analysis of crack problems because progressively growing cracks can be easily modeled without the burden of continuous remeshing as it is necessary when using the FEM [2].

The main drawback of the EFGM (and DEM) is the cost, since, as shown in Section 2.2, it is necessary to solve a system of equations at each point the interpolant is needed (or, equivalently, to orthonormalize the polynomial basis). Another practical problem is the satisfaction of condition (ii) of Section 2.2.1. This condition is difficult to satisfy when the density of nodes on the domain is very irregular or the local order of approximation is higher than quadratic.

3 Generalized Finite Difference Formulas for Irregular Grids

Liszka [17] developed an interesting interpolation technique for arbitrary meshes of nodes based on generalized finite difference formulas. In this section Liszka's method is reviewed and it is shown that under some conditions his method and the MLSM are equivalent.

We restrict, without loss of generality, the discussion of the Liszka's method to two dimensional manifolds. The extension to higher dimensions is immediate.

Assume as in Section 2.2 that a sufficiently smooth function $u : \bar{\Omega} \rightarrow \Re$, $\bar{\Omega} \subset \Re^2$, is to be approximated and that its values u_I at the points $\mathbf{x}_I \in \bar{\Omega}, I = 1, \ldots, N$, are given. Suppose that u is sufficiently smooth so that we can write for any point $\mathbf{x} = (x, y) \in \bar{\Omega}$

$$
\begin{aligned}
u(\mathbf{x}) &= u(\mathbf{0}) + \frac{\partial u}{\partial x}(\mathbf{0})(x) + \frac{\partial u}{\partial y}(\mathbf{0})(y) + \frac{\partial^2 u}{\partial x^2}(\mathbf{0})\frac{(x)^2}{2} \\
&+ \frac{\partial^2 u}{\partial x \partial y}(\mathbf{0})(xy) + \frac{\partial^2 u}{\partial y^2}(\mathbf{0})\frac{(y)^2}{2} + O(\rho^3)
\end{aligned} \tag{3.19}
$$

where $\rho = |\mathbf{x}|$ and the choice of the origin was merely for convenience. From now on we assume without loss of generality the the expansion is done at the origin.

Define a local approximation of u, built at the origin, by

$$
\begin{aligned}
Lu(\mathbf{x}) &= u(\mathbf{0}) + \frac{\partial u}{\partial x}(\mathbf{0})(x) + \frac{\partial u}{\partial y}(\mathbf{0})(y) + \frac{\partial^2 u}{\partial x^2}(\mathbf{0})\frac{(x)^2}{2} \\
&+ \frac{\partial^2 u}{\partial x \partial y}(\mathbf{0})(xy) + \frac{\partial^2 u}{\partial y^2}(\mathbf{0})\frac{(y)^2}{2}
\end{aligned} \tag{3.20}
$$

or

$$
Lu(\mathbf{x}) = \sum_{j=1}^{6} Du_j P_j(\mathbf{x})
$$

where

$$
\mathbf{Du} = \{u(\mathbf{0}), \frac{\partial u}{\partial x}(\mathbf{0}), \frac{\partial u}{\partial y}(\mathbf{0}), \frac{\partial^2 u}{\partial x^2}(\mathbf{0}), \frac{\partial^2 u}{\partial x \partial y}(\mathbf{0}), \frac{\partial^2 u}{\partial y^2}(\mathbf{0})\}
$$

$$
\mathbf{P}(\mathbf{x}) = \{1, x, y, x^2, xy, y^2\}
$$

Define the weighted error of this local approximation by [18]

$$
\mathbf{B}(\mathbf{Du}) = \sum_{I=1}^{N} [u(\mathbf{x}_I) - Lu(\mathbf{x}_I)]^2 W_I(\mathbf{0}) := (u - Lu, u - Lu)_{\mathbf{0}}
$$

where W_I is a weighting function associated to node I.

The coefficients $\mathbf{Du}$ are found solving the following problem [18]

Find $\mathbf{Du}^* \in \Re^6$ such that

$$
\mathbf{B}(\mathbf{Du}^*) \leq \mathbf{B}(\mathbf{Du}) \quad \forall\ \mathbf{Du} \in \Re^6.
$$

The previous implies that

$$\sum_{I=1}^{N}\left[\left(u(\mathbf{x}_I)-\sum_{j=1}^{6}Du_jP_j(\mathbf{x}_I)\right)P_k(\mathbf{x}_I)W_I(\mathbf{0})\right]$$

$$=(u-Lu,P_k)_{\mathbf{0}}=0 \quad k=1,\ldots,6$$

Thus

$$(Lu,P_k)_{\mathbf{0}}=(u,P_k)_{\mathbf{0}} \quad k=1,\ldots,6$$

$$\sum_{j=1}^{6}(P_j,P_k)_{\mathbf{0}}Du_j=(u,P_k)_{\mathbf{0}} \quad k=1,\ldots,6 \tag{3.21}$$

Note that

$$(u,v)_{\mathbf{0}}=\sum_{I=1}^{N}u(\mathbf{x}_I)W_I(\mathbf{0})v(\mathbf{x}_I) \tag{3.22}$$

is exactly the weighted inner product defined in (2.3), if $W_I(\mathbf{0})>0$. Thus (3.21) is equivalent to (2.6) and it can be concluded that if all nodes in the mesh contribute to the weighted inner product then Liszka's method and the MLSM are equivalents. If however not all nodes are used to build the local approximation Lu then the two methods will give different approximating functions. In the MLSM, as discussed in Section 2.2.1, the weighting functions W_I have a compact support equal to an open disc (ball) of radius ρ centered at the origin. The only nodes involved in the computation of the inner product (2.3) are those satisfying $|\mathbf{x}_I|<\rho$. In the Liszka's method the number of nodes used to compute (3.22) is fixed and their choice does not depend on the size of the support of the weighting functions. This criteria may lead in general to a discontinuous approximation of the function u.

Liszka [16–18] has used and pointed out many applications of his interpolation technique in a variety of situations:

- Extractions of derivatives of FEM and FDM solution of boundary value problems.
- Interpolation of the solution from one FE/FD mesh to another.
- Construction of the shape functions used in the FEM (like in the DEM and the EFGM).
- Interpolation of experimental data.

The Liszka's interpolation technique has also been used by Liszka and colleagues [18,32,33] to generalize the finite difference method and used to solve various problems in applied mechanics.

4 Interpolation by Kernel Estimates

The kernel estimate, $u_h(\mathbf{x})$, of the function $u(\mathbf{x})$ defined on a domain Ω is given by [24]

$$u_h(\mathbf{x}) = \int_\Omega \mathcal{W}_h(\mathbf{x}-\mathbf{y})u(\mathbf{y})d\mathbf{y} \tag{4.23}$$

The idea behind (4.23) is that $\mathcal{W}_h$ can be chosen to have the following properties for any real number $h \geq 0$

(i) $\mathcal{W}_h(\mathbf{x}) \in C^\infty(\Re^n)$

(ii)

$$\begin{array}{rcll} \mathcal{W}_h(\mathbf{x}) & = & 0 & \forall\ \mathbf{x}\ s.t.\ \|\mathbf{x}\| \geq h \\ \mathcal{W}_h(\mathbf{x}) & > & 0 & \forall\ \mathbf{x}\ s.t.\ \|\mathbf{x}\| < h \end{array}$$

i.e., $\mathcal{W}_h(\mathbf{x})$ and all its partial derivatives vanish outside a ball $B_h(\mathbf{0})$ of radius h centered at $\mathbf{x} = \mathbf{0}$ which implies that $\mathcal{W}_h(\mathbf{x}) \in C_0^\infty(B_h(\mathbf{0}))$.

(iii)

$$\frac{1}{h^n}\int_{\Re^n} \mathcal{W}_h(\mathbf{x})d\mathbf{x} = 1$$

Such functions do exists, e.g., the function

$$\mathcal{W}_h(\mathbf{x}) = \begin{cases} C_0 \exp\left(\frac{h^2}{\|\mathbf{x}\|^2 - h^2}\right) & for\ \|\mathbf{x}\| < h \\ 0 & for\ \|\mathbf{x}\| \geq h \end{cases}$$

where $\|.\|$ denotes the Euclidian norm, satisfies (i),(ii) and (iii) above if C_0 is properly chosen [28].

Theorem 4.1 *[28] Let $u(\mathbf{x}) \in L_p(\Omega)$ then*

(i) $\|u_h\|_{L_p(\Omega)} \leq \|u\|_{L_p(\Omega)}$

(ii) $\lim_{h\to 0} \|u - u_h\|_{L_p(\Omega)} = 0$

Proof. See [28] $\square$

In practice the kernel function $\mathcal{W}_h$ satisfies the property (iii) above only in the limit when h goes to zero. Some commonly used kernels are

- [31]

$$\mathcal{W}_h(z) = \begin{cases} \frac{C}{h^d}\left(1 - \frac{3}{2}z^2 + \frac{3}{4}z^3\right) & if\, z < 1 \\ \frac{C}{4h^d}(2-z)^3 & if\, 1 \leq z < 2 \\ 0 & if\, z \geq 2 \end{cases}$$

where

$$z = \frac{\|\mathbf{x} - \mathbf{y}\|}{h}$$

d is the dimension of the problem, i.e. 1, 2 or 3, and the constant C is given by

$$C = \begin{cases} \frac{2}{3} & if\, d = 1 \\ \frac{10}{7\pi} & if\, d = 2 \\ \frac{1}{\pi} & if\, d = 3 \end{cases}$$

- [25]

$$\mathcal{W}_h(\mathbf{x} - \mathbf{y}) = \frac{1}{\pi^{\frac{3}{2}} h^3} \exp\left(\frac{-\|\mathbf{x} - \mathbf{y}\|^2}{h^2}\right)$$

It is interesting to note that a local polynomial approximation of a function u can be considered as a special case of interpolation by kernel estimates [24]. If we approximate a function u in the interval $-\pi < x < \pi$ by using its Fourier expansion truncated at the N^{th} term, i.e.,

$$u(x) \approx u_h(x) = \sum_{n=0}^{N} \left(u_n^1 \sin nx + u_n^2 \cos nx\right)$$

the previous expression can be written as [24]

$$u_h(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sin[(2N+1)(x-y)/2]}{\sin[(x-y)/2]} u(y)dy$$

which is a kernel estimate [24].

Other examples of orthogonal function expansion written as kernel estimate can be found in [24]. The major difference between local polynomial interpolation and kernel evaluation is that in the first a fixed set of ordered points, e.g. the nodes of a finite element, is used while the second method interpolates from a moving set of points, i.e. those in the support of $\mathcal{W}(\mathbf{x}-\mathbf{y})$.

4.1 Smoothed Particle Hydrodynamics Methods

In the Smoothed Particle Hydrodynamics Method (SPH) the integral

$$\mathbf{u}_h(\mathbf{x}) = \int_\Omega \mathcal{W}_h(\mathbf{x}-\mathbf{y})\mathbf{u}(\mathbf{y})d\mathbf{y} \tag{4.24}$$

is approximated by imagining that the matter is divided into N small volume elements with masses $m_1, m_2, \ldots, m_N$. The contribution to the integral from the volume element I with mass m_I and center of mass $\mathbf{x}_I$ is

$$\mathcal{W}_h(\mathbf{x}-\mathbf{x_I})\mathbf{u}(\mathbf{x}_I)m_I/\rho(\mathbf{x}_I)$$

where $\rho(\mathbf{x}_I)$ is the density at node $\mathbf{x}_I$ and (4.24) is estimated by (if $\mathcal{W}_h$ has compact support)

$$\mathbf{u}_h(\mathbf{x}) = \sum_{I\in Q(\mathbf{x})} \mathcal{W}_h(\mathbf{x}-\mathbf{x_I})\mathbf{u}(\mathbf{x}_I)\Delta V_I \tag{4.25}$$

where $Q(\mathbf{x})$ is the set of nodes in the domain of influence of $\mathbf{x}$, i.e., the set of nodes $\mathbf{x}_I$ s.t. $\mathbf{x} \in supp(\mathcal{W}(\mathbf{x}-\mathbf{x}_I))$ and $\Delta V_I = \frac{m_I}{\rho(\mathbf{x}_I)}$. Note that although in the SPH method a mesh is not used, i.e. there is no fixed connection between the nodes, the domain have to be partitioned to compute the ΔV_I quantities. This partition may not be easy to do in the case of, for example, three-dimensional domains with complex geometries.

The approximation for spatial derivatives of $\mathbf{u}$ is obtained by substituting $\nabla \cdot \mathbf{u}(\mathbf{x})$ for $\mathbf{u}(\mathbf{x})$ in (4.24)

$$\nabla \cdot \mathbf{u}_h(\mathbf{x}) = \int_\Omega \mathcal{W}_h(\mathbf{x}-\mathbf{y}) \nabla \cdot \mathbf{u}(\mathbf{y}) d\mathbf{y} \tag{4.26}$$

After some manipulation [31] it can be showed that

$$\nabla \cdot \mathbf{u}_h(\mathbf{x}) = -\int_\Omega \mathbf{u}(\mathbf{y}) \cdot \nabla \mathcal{W}_h(\mathbf{x}-\mathbf{y}) d\mathbf{y} \tag{4.27}$$

Thus the kernel approximation allows spatial gradients to be determined from the values of the function and the derivatives of the kernel, rather than the derivatives of the function itself. As before the continuous integral is converted to sums over discrete interpolation points by [31]

$$\nabla \cdot \mathbf{u}_h(\mathbf{x}) = -\sum_{I \in Q(\mathbf{x})} \Delta V_I \mathbf{u}(\mathbf{x}_I) \cdot \nabla \mathcal{W}_h(\mathbf{x}-\mathbf{x}_I) \tag{4.28}$$

Expressions (4.25) and (4.28) are used in the SPH method to solve a boundary value problem by directly applying the PDE at the nodes $\mathbf{x}_I$ [31]. This collocation process gives a discrete system of equations that can be solved for the coefficients $\mathbf{u}(\mathbf{x}_I)$. The major disadvantage of the SPH method is its poor accuracy [3]. It has been reported that a large number of nodes are usually required in the SPH method to achieve reasonable accuracy in practical applications [11]. It is shown in the next section that the interpolant (4.25) used in the SPH method is equivalent to the Shephard interpolant defined in Section 2.2.

4.2 Relation Between the Shephard Interpolant and Kernel Estimates in SPH Methods [22]

The following result is due to Lu et al. [22].

In Section 2.2 the Shephard interpolant was defined by

$$(Su)(\mathbf{y}) = \sum_I^N v_I(\mathbf{y}) u(\mathbf{x}_I) \tag{4.29}$$

where

$$v_I(\mathbf{y}) = \frac{W_I(\mathbf{y})}{\sum_J^N W_J(\mathbf{y})} \tag{4.30}$$

and $W_I(\mathbf{y})$ is the weighting function. In this section the kernel function used in the SPH method will be denoted by $\mathcal{W}_h^{sph}(\mathbf{y})$.

If the kernel used in the SHP method is defined by

$$\mathcal{W}_h^{SPH}(\mathbf{x}-\mathbf{x}_I) = \frac{W_I(\mathbf{x})}{\Delta V_I \sum_J^N W_J(\mathbf{y})}$$

Then (4.25) gives that

$$u_h(\mathbf{x}) = \sum_I^N v_I(\mathbf{x}) u(\mathbf{x}_I)$$

Thus the two methods use equivalent interpolation functions [22].

5 Wavelet-Galerkin Solution for PDEs

In this section the use of wavelets to solve ordinary and partial differential equations is discussed. Section 5.1 reviews the construction of the Daubechies scaling and wavelet functions and lists some of its properties. Section 5.2 illustrates how this functions can be used to solve partial differential equations.

5.1 Construction of Daubechies' Wavelets and Scaling Functions

Definitions [34]:

Consider the sequence of embedded subspaces V_i of $L^2(\Re)$

$$\emptyset \subset \ldots \subset V_{-1} \subset V_0 \subset \ldots \subset L^2(\Re) \tag{5.31}$$

with the following properties:

- $$\bigcup_{j \in \mathbf{Z}} V_j \text{ is dense in } L^2(\Re), \text{ where } \mathbf{Z} \text{ denote the set of integers.} \tag{5.32}$$

- $$\bigcap_{j\in\mathbf{Z}} V_j = \emptyset \tag{5.33}$$

- The embedded subspaces are related by a *scaling law*

$$g(x) \in V_j \Leftrightarrow g(2x) \in V_{j+1} \tag{5.34}$$

- Each subspace is spanned by integer translates of a single function g(x) such that

$$g(x) \in V_0 \Leftrightarrow g(x+1) \in V_0 \tag{5.35}$$

We seek a scaling function $\phi(x) \in V_0$ such that its integer translates form a basis for the space V_0, i.e.

$$V_0 = span\overline{\{\phi(x-k), k \in \mathbf{Z}\}} \tag{5.36}$$

Consider the binary dilation (i.e. dilation by a power of two) and a dyadic translation of the scaling function ϕ given by

$$\phi_{m,k}(x) = 2^{m/2}\phi(2^m x - k) \quad m, k \in \mathbf{Z}$$

then

$$V_m = span\overline{\{\phi(2^m x-k), k \in \mathbf{Z}\}} \tag{5.37}$$

The dilation parameter m is called the *scale* .

Since $V_0 \subset V_1$, $\phi(x)$ can be represented using the basis functions of V_1

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \phi(2x-k) \tag{5.38}$$

This equation is denoted as the *dilation equation* and the constant coefficients are called *filter coefficients*. It is often the case that only a finite number of these coefficients are non-zero. The above equation plays a fundamental role in the construction of the wavelet functions.

Consider now the orthogonal complement of the space V_{m-1} in V_m

$$W_{m-1} = \{\phi_{m,k} \in V_m : (\phi_{m,k}, \phi_{m-1,l}) = 0 \, \forall \, \phi_{m-1,l} \in V_{m-1}\}$$

Then we can write

$$V_m = V_{m-1} \oplus W_{m-1}$$

where $\oplus$ represents a direct sum. It follows that the spaces W_j are orthogonal and that

$$\oplus_{j \in \mathbf{Z}} W_j = L^2(\Re)$$

The wavelet function $\psi(x)$ is defined as the function such that

$$W_0 = span\overline{\{\psi(x-k), k \in \mathbf{Z}\}} \tag{5.39}$$

Then its binary dilation and translation is given by

$$\psi_{m,k}(x) = 2^{m/2}\psi(2^m x - k)$$

and it has the property that

$$W_n = span\overline{\{\psi_{n,k}, k \in \mathbf{Z}\}} \tag{5.40}$$

Note that since the space W_0 is contained in the space V_1 the wavelet function ψ can be written as

$$\psi(x) = \sum_{k=-\infty}^{\infty} b_k \phi(2x - k)$$

This property is used to define $\psi(x)$ using the filter coefficients defined in (5.38) and the scaling functions ϕ.

$$\psi(x) = \sum_{k=-\infty}^{\infty} (-1)^k a_{N-1-k} \phi(2x - k) \tag{5.41}$$

where N is an even integer.

Thus, to construct the wavelet ψ first it is necessary to compute the filter coefficients and construct the scaling function $\phi(x)$.

5.1.1 Computation of the Filter Coefficients and Construction of the Daubechies' Wavelets

The filter coefficients, a_k, defined in equation (5.38) are derived by imposing certain conditions on the scaling function ϕ which lead to corresponding conditions on the filter coefficients [34]. The conditions listed below were used by Daubechies to construct, for any given integer $N \geq 1$, a pair of functions ϕ and ψ that are the classical Haar scaling function and Haar wavelet, respectively, for $N = 1$, and that generalize these functions for $N > 1$ [9]. The Haar scaling function and the Haar wavelet are showed in Figures 5.11 and 5.12, respectively.

(i) The area under the scaling function is normalized to unity, i.e.

$$\int_{-\infty}^{\infty} \phi(x)dx = 1 \tag{5.42}$$

which leads to the following condition on the filter coefficients

$$\sum_{k=-\infty}^{\infty} a_k = 2 \tag{5.43}$$

(ii) The scaling function is orthogonal to its translates, i.e.

$$\int_{-\infty}^{\infty} \phi(x)\phi(x+k)dx = \delta_{0,k},\ k \in \mathbf{Z} \tag{5.44}$$

This yields the condition

$$\sum_{k=-\infty}^{\infty} a_k a_{k+2l} = 2\delta_{0,l},\ l \in \mathbf{Z} \tag{5.45}$$

(iii) In an N coefficient system, i.e. when N of the filter coefficients are non zero, (5.43) and (5.45) yield a total of $N/2+1$ equations. To obtain another $N/2-1$ equations Daubechies required the scaling functions to be able to exactly represent polynomials of order up to, but not greater than, $p = N/2$. This leads to the following conditions on the filter coefficients

$$\sum_{k=-\infty}^{\infty} (-1)^k a_k k^l = 0,\ l = 0, \ldots, N/2-1 \tag{5.46}$$

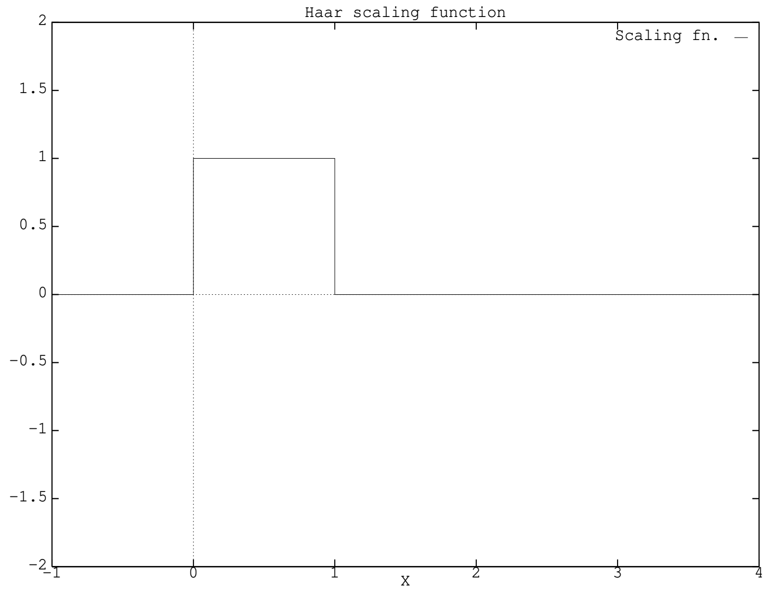

Figure 5.11: Haar scaling function

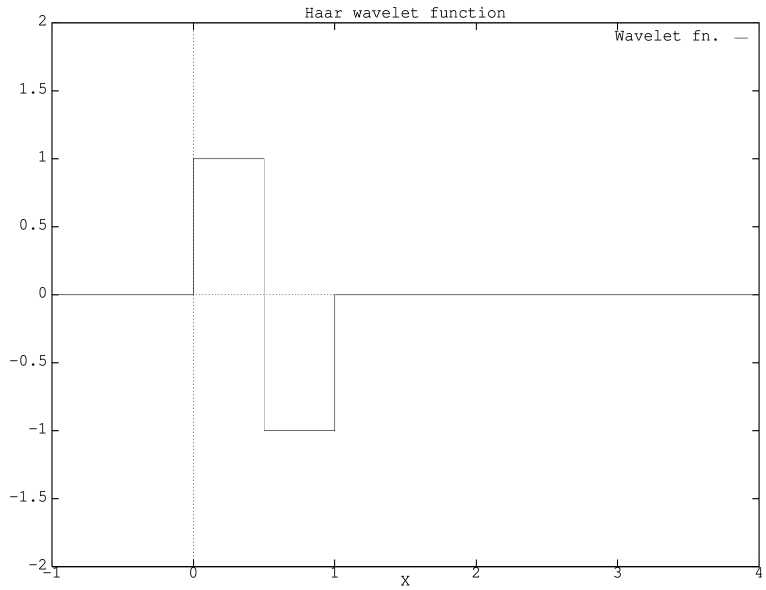

Figure 5.12: Haar wavelet function

Equations (5.43), (5.45) and (5.46) above, uniquely define the Daubechies filter coefficients $a_k, k = 0, \ldots, N-1$ for an N coefficient system [34]. Table I lists the coefficients for the Daubechies D4 system, i.e., for $N = 4$.

k	D4
0	0.68301
1	1.18301
2	0.31698
3	-0.18301

The Daubechies scaling and wavelet functions are considerably more complex than the more familiar elementary functions. They do not have a closed-form. Instead they are generated recursively from the dilation equation (5.38). First the values of $\phi(x)$ at the integer values of x are computed. Then the values of $\phi(x)$ at the half integers can be found from (5.38)

$$\phi(\frac{x}{2}) = \sum_{k=-\infty}^{\infty} a_k \phi(x-k) \tag{5.47}$$

The process is repeated as many times as necessary to find the values of $\phi(x)$ at all dyadic points $\{i/2^n; i, n \in \mathbf{Z}\}$.

The values of $\phi(x)$ at the integer values of x are computed solving the following problem [34]:

Find $\mathbf{\Phi} \in \Re^N$ such that

$$\mathbf{M\Phi} = \mathbf{\Phi} \tag{5.48}$$

where

$$\mathbf{M} = \begin{bmatrix} a_0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ a_2 & a_1 & a_0 & \ldots & 0 & 0 & 0 \\ a_4 & a_3 & a_2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & a_{N-3} & a_{N-4} & a_{N-5} \\ 0 & 0 & 0 & \ldots & a_{N-1} & a_{N-2} & a_{N-3} \\ 0 & 0 & 0 & \ldots & 0 & 0 & a_{N-1} \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \vdots \\ \phi(N-3) \\ \phi(N-2) \\ \phi(N-1) \end{bmatrix}$$

i.e., $\mathbf{\Phi}$ is the eigenvector associated to the eigenvalue $\lambda = 1$ of the matrix of filter coefficients $\mathbf{M}$. The eigenvector $\mathbf{\Phi}$ is normalized by imposing that

$$\sum_{i=-\infty}^{\infty} \phi(i) = 1, \; i \in \mathbf{Z}$$

Example: The Daubechies 4 coefficient scaling and wavelet system (D4).

The matrix $\mathbf{M}$ defined in (5.48) can be constructed using the filter coefficients given in Table I. The solution of the corresponding eigenvalue problem gives:

$$\phi(0) = 0, \; \phi(1) = \frac{1+\sqrt{3}}{2}, \; \phi(2) = \frac{1-\sqrt{3}}{2}, \; \phi(3) = 0$$

$\phi(x)$ was plotted in Figure 5.13 using the above values and the dilation equation (5.38). The wavelet function $\psi(x)$ was plotted in Figure 5.14 using the computed values of the scaling function and (5.41).

The Daubechies scaling functions and wavelet functions satisfy the properties (5.31) - (5.44) and, in addition, the following properties [9]:

- $\phi_{n,k}$ *and* $\psi_{n,k}$ have compact support for all integers n,k.
- $\phi_{n,k}$ *and* $\psi_{n,k} \in C^{\lambda(N)}$ = space of Holder continuous functions with exponent $\lambda(N)$, where $\lambda(4) \approx 0.5500$, $\lambda(6) \approx 1.087833$, $\lambda(8) \approx 1.6179$, and $\lambda(N) \approx 0.3485N/2$ for large N.

Theorem 5.1 *[9] Let $m \geq 0$ and choose $N > m$ such that the associated scaling function ϕ and wavelet function ψ belong to the Hilbert space $H^m(\Re)$. Let $g \in H^m(\Omega)$, then for any $\epsilon > 0$ there exists an integer n such that*

$$\|g - h\|_m < \epsilon$$

where h is the restriction to Ω of a function in V_n.

Proof. See [9] $\Box$.

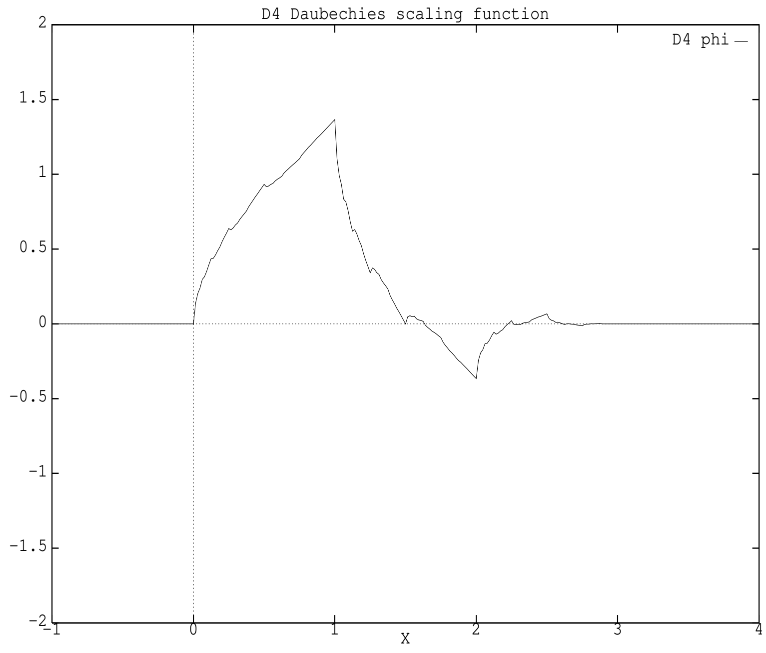

Figure 5.13: Daubechies D4 scaling function

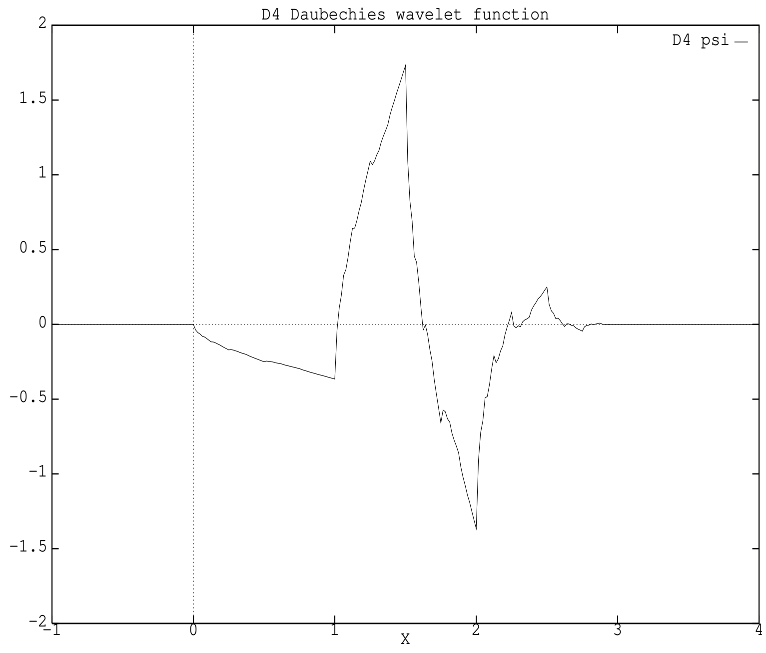

Figure 5.14: Daubechies D4 wavelet function

5.2 The Wavelet-Galerkin Method

Let $N = 6$ and let ϕ be the corresponding Daubechies' scaling function as defined in Section 5.1. Let n be any integer and let V_n be defined as in Section 5.1. Define $V_n(0,1)$ to be the restriction to $(0,1)$ of all functions in V_n. By Theorem 5.1 every function in $H^m(0,1)$ can be approximated arbitrarily closely by an element in $V_n(0,1)$ for sufficient large n, hence

$$H^m(0,1) = \overline{\bigcup_{n \in \mathbf{Z}} V_n(0,1)}$$

Therefore the family of subspaces $V_n(0,1)$ of $H^m(0,1)$ is well suited to the Galerkin solution of, e.g., second order elliptic problems.

Example: [29]

Consider a PDE of the form

$$F(u, u_x, u_{xx}, \ldots) = 0$$

An approximation to the solution is defined by

$$U = \sum_{k=M}^{L} U_k \phi(x-k) \tag{5.49}$$

In effect, the solution is projected onto the subspace spanned by

$$V(M,L) = span\{\phi(x-k) : k = M, \ldots, L, k \in \mathbf{Z}\}$$

To determine the coefficients U_k, (5.49) is substituted into the differential equation and the resulting expression is again projected onto the space $V(M,L)$. The projection requires that

$$\int_{[M,L]} \phi(x-k) F(U, U_x, U_{xx}, \ldots) dx = 0 \; for \; k = M, \ldots, L$$

To evaluate this expression it is necessary to know coefficients of the form

$$\int_{[M,L]} \phi(x)\phi_x(x-k_1)\ldots\phi_{xx}(x-k_2)\ldots dx$$

Since the scaling function used to define compact wavelets has a limited number of derivatives, the numerical evaluation of these expressions is often unstable and inaccurate [29]. Resnikoff [9] developed methods for the explicitly evaluation of such functionals when the coefficients of the differential operator are piecewise polynomial functions whose associated intervals have dyadic rational endpoints [9].

Acknowledgment: *The CNPq Graduate Fellowship Program's support of Armando Duarte is gratefully acknowledged.*

References

[1] K. Amarantuga, J. R. Williams, S. Qian, and J. Weis. Wavelet-galerkin solutions for one-dimensional partial differential equations. *International Journal for Numerical Methods in Engineering*, 37:2703–2716, 1994.

[2] T. Belytschko, Y. Y. Lu, and L. Gu. Crack propagation by element free galerkin methods. In *Advanced Computational Methods for Material Modeling*, pages 191–205, 1993. AMD-Vol. 180/PVP-Vol. 268, ASME 1993.

[3] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[4] C. K. Chui. *An Introduction to Wavelets*. Academic Press, Boston, 1992.

[5] C. K. Chui, editor. *Wavelets: A Tutorial in Theory and Applications*. Academic Press, Boston, 1992.

[6] L. Demkowicz, A. Karafiat, and T. Liszka. On some convergence results for FDM with irregular mesh. *Computer Methods in Applied Mechanics and Engineering*, 42:343–355, 1984.

[7] Richard Franke. Scattered data interpolation: Test of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.

[8] R. A. Gingold and J. J. Monaghan. Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, 46:429–453, 1982.

[9] R. Glowinski, M. Lawton, M. Ravachol, and E. Tenenbaum. Wavelet solutions of linear and nonlinear elliptic, parabolic and hyperbolic problems in one space dimension. In 9^{th} *Int. Conf. on Numerical Methods in Applied Sciences and Engineering, SIAM*, Philadelphia, USA, 1990.

[10] R. L. Hardy. Theory and applications of the multiquadric–biharmonic method. *Computers and Mathematics with Applications*, 19(8/9):163–208, 1990.

[11] G. R. Johnson, E. H. Peterson, and R.A. Stryrk. Incorporation of an sph option into the epic code for a wide range of high velocity impact computations. In *Colloquium on Advances in Smoothed Particle Hydrodynamics*, 1993.

[12] E. J. Kansa. Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics–i surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, 19(8/9):127–145, 1990.

[13] E. J. Kansa. Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics–ii solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, 19(8/9):147–161, 1990.

[14] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[15] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting, an Introduction.* Academic Press, San Diego, 1986.

[16] Tadeusz Liszka. An interpolation method for an irregular net of nodes. *International Journal for Numerical Methods in Engineering*, 20:1599–1612, 1984.

[17] Tadeusz Liszka and J. Orkisz. Finite difference method for arbitrary irregular meshes in nonlinear problems of applied mechanics. In *IV SMiRt*, San Francisco, 1977.

[18] Tadeusz Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11:83–95, 1980.

[19] Wing Kam Liu and Yijung Chen. Wavelet and multiple scale reproducing kernel methods. Accepted for publication in the International Journal for Numerical Methods in Fluids.

[20] Wing Kam Liu, Sukky Jun, Shaofan Li, Jonathan Adee, and Ted Belytschko. Reproducing kernel particle methods for structural dynamics.

Accepted for publication in the International Journal for Numerical Methods in Engineering.

[21] Y. Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element free galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 113:397–414, 1993.

[22] Y. Y. Lu, Ted Belytschko, and M. Tabbara. Element-free galerkin method for wave propagation and dynamic fracture. Accepted for publication in Computer Methods in Applied Mechanics and Engineering.

[23] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

[24] J. J. Monaghan. Why particle methods work. *SIAM J. Sci. Stat. Comput.*, 3(4):422–433, 1982.

[25] J. J. Monaghan. An introduction to sph. *Computer Physics Communications*, 48:89–96, 1988.

[26] R. A. Nay and S. Utku. An alternative for the finite element method. In *Variational Methods in Engineering*, volume 1, University of Southampton, September 1972.

[27] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.

[28] J. T. Oden and J. N. Reddy. *An Introduction to the Mathematical Theory of Finite Elements*. John Wiley and Sons, New York, 1976.

[29] Sam Qian and J. Weiss. Wavelet and the numerical solution of partial differential equations. *Journal of Computational Physics*, 106:155–175, 1993.

[30] D. Shepard. A two-dimensional function for irregularly spaced data. In *ACM National Conference*, pages 517–524, 1968.

[31] J. W. Swegle, S. W. Attaway, M. W. Heinstein, F. J. Mello, and D. L. Hicks. An analysis of the smoothed particle hydrodynamics. Technical Report SAND93-2513 UC-705, Sandia, 1994.

[32] W. Tworzydlo. Analysis of large deformations of membrane shells by the generalized finite difference method. *Computers and Structures*, 27(1):39–59, 1987.

[33] W. Tworzydlo. The fdm in arbitrary curvilinear co-ordinates - formulation, numerical approach and applications. *International Journal for Numerical Methods in Engineering*, 28:261–277, 1989.

[34] J. R. Williams and Amarantuga. Introduction to wavelets in engineering. *International Journal for Numerical Methods in Engineering*, 37:2365–2388, 1994.

[35] Genki Yagawa. Some remarks on free mesh method: A kind of meshless finite element method. In *International Conference on Computational Engineering Science*, Hawaii, USA, August 1995.