INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2001; **50**:2373–2410

Finite cloud method: a true meshless technique based on a fixed reproducing kernel approximation

N. R. Aluru* and Gang Li†

Beckman Institute and Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.

SUMMARY

We introduce fixed, moving and multiple fixed kernel techniques for the construction of interpolation functions over a scattered set of points. We show that for a particular choice of nodal volumes, the fixed, moving and multiple fixed kernel approaches are identical to the fixed, moving and multiple fixed least squares approaches. A finite cloud method, which combines collocation with a fixed kernel technique for the construction of interpolation functions, is presented as a true meshless technique for the numerical solution of partial differential equations. Numerical results are presented for several one- and two-dimensional problems, including examples from elasticity, heat conduction, thermoelasticity, Stokes flow and piezoelectricity. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: meshless method; fixed kernel technique; reproducing kernel; point collocation; finite cloud method

1. INTRODUCTION

There is a growing interest in the development of meshless methods for numerical solution of partial differential equations as meshless techniques do not require the generation of a mesh for complex two- and three-dimensional structures. Instead, meshless techniques require only a scattered set of nodes representing the domain of interest. No connectivity information among the scattered set of nodes is required, unlike finite element, boundary element or classical finite difference techniques. Meshless techniques are also appealing because of their potential in adaptive techniques, where a user can simply add more points in a particular region to obtain more accurate results. Finally, meshless techniques are especially attractive for emerging technologies such as microelectromechanical systems (MEMS), where multiphysics

*Correspondence to: N. R. Aluru, Beckman Institute and Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.
†Doctoral Student

Contract/grant Sponsor: DARPA; contract/grant number: F30602-98-2-0178
Contract/grant Sponsor: NSF.

Received 24 June 1999
Revised 3 July 2000

and multiscale analysis is a common requirement. Both multiphysics and multiscale analysis are radically simplified by dealing with nodes or points instead of a mesh.

Several meshless methods have been proposed in the literature, many of them in the last decade (see Reference [1] for a review). Among the first meshless methods are the generalized finite difference techniques [2] and the classical smooth particle hydrodynamics (SPH) approach based on a kernel approximation [3–6]. The classical SPH technique suffers from poor accuracy and spatial or tensile instability. An approach that significantly improves the stability and accuracy of the classical SPH is the reproducing kernel particle method (RKPM) [7–14]. The key difference in the construction of the RKPM and the classical SPH is the introduction of a correction function in RKPM to ensure the consistency of the method. Other meshless methods developed over the last few years include the diffuse element method (DEM) [15], element free Galerkin method (EFG) [1, 16], finite point method (FPM) [17–19], meshless local Petrov–Galerkin method (MLPG) [20], boundary node method [21], hp-clouds [22], partition of unity [23], local boundary integral equation method [24, 25] and others.

In many of these meshless techniques the interpolation functions are constructed by employing either the reproducing kernel approach (e.g. RKPM) or the moving least-squares approach (e.g. EFG, MLPG, FPM and others). Once the interpolation functions are constructed, the governing partial differential equations can be solved by employing either a Galerkin or a collocation approach. For example, RKPM combines Galerkin with a reproducing kernel approach for the computation of interpolation functions, [26] combines collocation with reproducing kernel approximations, EFG combines Galerkin with a moving least-squares approach and FPM combines collocation with moving and other (these will be elaborated in the next paragraph) least-squares approaches. Galerkin-based approaches are typically not true meshless techniques (an exception to this is the approach presented in Reference [27]) as they require some kind of a background grid for integration purpose. A collocation approach, on the other hand, can be a true meshless technique and does not require any background grid.

One of the goals of this paper is to explore the connection between least squares and kernel approaches for the construction of interpolation functions. In addition, Oñate *et al.* [17, 18] have proposed fixed least squares and multiple fixed least squares as alternative approaches to a moving least squares approach for the construction of interpolation functions. We would also like to ask the question: are there kernel equivalents to fixed least squares and multiple fixed least squares approaches? We explore these ideas in this paper and the primary results are summarized as follows:

1. We introduce three kernel approximations—the fixed reproducing kernel, the moving reproducing kernel and the multiple fixed reproducing kernel approaches. We show that for a particular choice of nodal volumes the fixed, moving, and multiple fixed reproducing kernel approaches are equivalent to fixed, moving, and multiple fixed least squares approaches.
2. We present a new meshless technique, defined as the finite cloud method, which combines collocation with a fixed reproducing kernel technique. Since the construction of interpolation functions by a fixed reproducing kernel technique utilizes overlapping clouds of finite in extent, we refer to this method as a finite cloud method. The finite cloud method is a true meshless technique in which both Dirichlet and Neumann boundary conditions are enforced exactly.

The rest of the paper is organized as follows: The classical reproducing kernel technique is summarized in Section 2, the fixed reproducing kernel technique is introduced in Section 3, and two other kernel approximations—namely the moving reproducing kernel and the multiple fixed reproducing kernel are introduced in Section 4. The discretization technique is described in Section 5, numerical results employing the fixed reproducing kernel technique are shown in Section 6 and finally conclusions are given in Section 7.

2. CLASSICAL REPRODUCING KERNEL TECHNIQUE

In this section we summarize the key ideas in the classical reproducing kernel technique with the hope that other kernel techniques introduced in the following sections can be contrasted with the classical approach. In the classical reproducing kernel technique [10, 11], an approximation to a function is constructed by employing a corrected kernel. In two dimensions, the reproducing kernel technique is stated as

$$u^{\mathrm{a}}(x,y)=\int_{\Omega}\mathscr{C}(x,y,x-s,y-t)\varphi(x-s,y-t)u(s,t)\,\mathrm{d}s\,\mathrm{d}t \tag{1}$$

where $u(x,y)$ is the unknown function, $u^{\mathrm{a}}(x,y)$ is an approximation to the unknown function, $\mathscr{C}(x,y,x-s,y-t)$ is a correction function, and $\varphi(x-s,y-t)$ is the kernel function. The correction function is expressed as

$$\mathscr{C}(x,y,x-s,y-t)=P^{\mathrm{T}}(x-s,y-t)C(x,y) \tag{2}$$

where P is the $m\times 1$ vector of basis functions and C is the $m\times 1$ vector of correction function coefficients. A linear basis in two dimensions is given by

$$P^{\mathrm{T}}(x-s,y-t)=[1,x-s,y-t],\quad m=3 \tag{3}$$

and a quadratic basis is given by

$$P^{\mathrm{T}}(x-s,y-t)=[1,x-s,y-t,(x-s)^2,(x-s)(y-t),(y-t)^2],\quad m=6 \tag{4}$$

The unknown correction function coefficients are computed by satisfying the reproducing conditions (see e.g. Reference [8] for details). The shape function or an interpolation function in the classical reproducing technique is given by

$$N_I(x,y)=\mathscr{C}(x,y,x-x_I,y-y_I)\varphi(x-x_I,y-y_I)\Delta V_I \tag{5}$$

where ΔV_I is the nodal volume associated with node I.
Remarks:

1. The classical reproducing kernel technique in combination with a Galerkin method has been successfully employed to simulate several complex phenomena in solid and fluid mechanics applications [7–14] and MEMS [28].
2. Recently a point collocation technique in combination with the reproducing kernel method has also been presented [26].

3. FIXED REPRODUCING KERNEL TECHNIQUE

In a fixed kernel technique, the approximate solution is given by

$$u^{\mathrm{a}} = \int_{\Omega} \mathscr{C}(x, y, s, t)\varphi(x_K - s, y_K - t)u(s, t)\,\mathrm{d}s\,\mathrm{d}t \tag{6}$$

where $\mathscr{C}(x, y, s, t)$ is the correction function and is given by

$$\mathscr{C}(x, y, s, t) = P^{\mathrm{T}}(s, t)C(x, y) \tag{7}$$

$P^{\mathrm{T}}(s, t) = \{p_1, p_2, \ldots, p_m\}$ is the $m \times 1$ vector of basis functions and $C^{\mathrm{T}}(x, y) = \{c_1, c_2, \ldots, c_m\}$ is the $m \times 1$ vector of correction function coefficients. A linear basis in two dimensions is given by

$$P^{\mathrm{T}}(s, t) = [1, s, t], \quad m = 3 \tag{8}$$

and a quadratic basis is given by

$$P^{\mathrm{T}}(s, t) = [1, s, t, s^2, st, t^2], \quad m = 6 \tag{9}$$

The kernel function $\varphi(x_K - s, y_K - t)$ is centred at the point (x_K, y_K). Note the difference in the construction of an approximation by a fixed kernel (Equation (6)) and by a classical reproducing kernel (Equation (1)). In particular, note the definition of the correction and kernel functions. The unknown correction function coefficients can be determined by satisfying the reproducing or consistency conditions, i.e.

$$\int_{\Omega} P^{\mathrm{T}}(s, t)C(x, y)\varphi(x_K - s, y_K - t)\,p_i(s, t)\,\mathrm{d}s\,\mathrm{d}t = p_i(x, y), \quad i = 1, 2, \ldots, m \tag{10}$$

A discrete approximation of the above consistency conditions is written as

$$\sum_{I=1}^{\mathrm{NP}} P^{\mathrm{T}}(x_I, y_I)C(x, y)\varphi(x_K - x_I, y_K - y_I)\,p_i(x_I, y_I)\Delta V_I = p_i(x, y), \quad i = 1, 2, \ldots, m \tag{11}$$

where NP is the total number of points covering the domain Ω, and ΔV_I is the nodal volume associated with node I. The consistency conditions summarized in Equation (11) can be written in a matrix form as

$$MC(x, y) = P(x, y) \tag{12}$$

where M is the $m \times m$ moment matrix and is a constant i.e. it is not a function of x and y. The entries in the moment matrix are given by

$$M_{ij} = \int_{\Omega} p_i(s, t)\varphi(x_K - s, y_K - t)\,p_j(s, t)\,\mathrm{d}s\,\mathrm{d}t, \quad i, j = 1, \ldots, m \tag{13}$$

In discrete form, the components of the moment matrix are defined as

$$M_{ij} = \sum_{I=1}^{\mathrm{NP}} p_i(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\,p_j(x_I, y_I)\Delta V_I \tag{14}$$

The moment matrix can also be written as

$$M = FWF^{\mathrm{T}} \tag{15}$$

where F is an $m \times \mathrm{NP}$ matrix defined as

$$F = \begin{bmatrix} p_1(x_1, y_1) & p_1(x_2, y_2) & \cdots & p_1(x_{\mathrm{NP}}, y_{\mathrm{NP}}) \\ p_2(x_1, y_1) & p_2(x_2, y_2) & \cdots & p_2(x_{\mathrm{NP}}, y_{\mathrm{NP}}) \\ \vdots & \vdots & \ddots & \vdots \\ p_m(x_1, y_1) & p_m(x_2, y_2) & \cdots & p_m(x_{\mathrm{NP}}, y_{\mathrm{NP}}) \end{bmatrix} \tag{16}$$

and W is an $\mathrm{NP} \times \mathrm{NP}$ diagonal matrix defined as

$$W = \begin{bmatrix} \varphi(x_K - x_1, y_K - y_1)\Delta V_1 & 0 & \cdots & 0 \\ 0 & \varphi(x_K - x_2, y_K - y_2)\Delta V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varphi(x_K - x_{\mathrm{NP}}, y_K - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}} \end{bmatrix} \tag{17}$$

From Equation (12), the unknown correction function coefficients are computed as

$$C(x, y) = M^{-1}P(x, y) \tag{18}$$

Substituting the correction function coefficients into Equation (6)

$$u^{\mathrm{a}}(x, y) = \int_{\Omega} P^{\mathrm{T}}(x, y)M^{-1}P(s, t)\varphi(x_K - s, y_K - t)u(s, t)\,\mathrm{d}s\,\mathrm{d}t \tag{19}$$

The approximation in Equation (19) can be rewritten as

$$u^{\mathrm{a}}(x, y) = \sum_{I=1}^{\mathrm{NP}} N_I(x, y)u_I \tag{20}$$

where u_I is a nodal unknown for node I, and $N_I(x, y)$ is the fixed kernel interpolation function defined as

$$N_I(x, y) = P^{\mathrm{T}}(x, y)M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I \tag{21}$$

Defining

$$B = \begin{bmatrix} p_1(x_1, y_1)\varphi(x_K - x_1, y_K - y_1)\Delta V_1 & \cdots & p_1(x_{\mathrm{NP}}, y_{\mathrm{NP}})\varphi(x_K - x_{\mathrm{NP}}, y_K - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}} \\ p_2(x_1, y_1)\varphi(x_K - x_1, y_K - y_1)\Delta V_1 & \cdots & p_2(x_{\mathrm{NP}}, y_{\mathrm{NP}})\varphi(x_K - x_{\mathrm{NP}}, y_K - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}} \\ \vdots & \ddots & \vdots \\ p_m(x_1, y_1)\varphi(x_K - x_1, y_K - y_1)\Delta V_1 & \cdots & p_m(x_{\mathrm{NP}}, y_{\mathrm{NP}}\varphi(x_K - x_{\mathrm{NP}}, y_K - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}}) \end{bmatrix} \tag{22}$$

the interpolation function can be rewritten as

$$N_I(x, y) = \sum_{l=1}^{m} p_l(x, y)\bar{C}_{lI}^{-1} \tag{23}$$

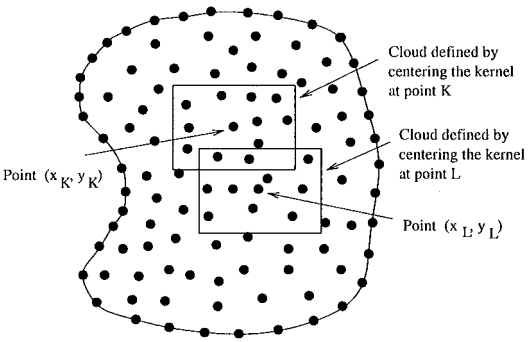

Figure 1. Illustration of the concept of a cloud and overlapping of clouds.

where

$$\bar{C}^{-1} = M^{-1}B \tag{24}$$

and $\bar{C}_{II}^{-1}$ denotes the Ith column of matrix $\bar{C}^{-1}$.
Remarks:

1. When all the nodal volumes are set to unity i.e. $\Delta V_I = 1$, then the definition of the shape function given in Equation (23) is identical to the definition of the shape function given by the fixed least-squares approach [17, 18].
2. The construction of an approximate solution by Equation (6) is referred to as a fixed kernel technique as the kernel in Equation (6) is fixed at the point (x_K, y_K). The point (x_K, y_K) can, however, be any point in the domain Ω. This then leads to the issue of multivalued interpolation functions which is discussed next.

3.1. Interpolation functions

From the definition of the interpolation or shape function given in Equation (21), when the kernel is centred at the point (x_K, y_K), one can compute all the interpolation functions at all nodal positions i.e. $N_I(x_Q, y_Q)$, $I, Q = 1, 2, \ldots, \mathrm{NP}$. For those points, that are outside the cloud of point (x_K, y_K), the interpolation function typically vanishes. The definition of a cloud for a random distribution of points is shown in Figure 1. The clouds are finite in extent and each cloud includes sufficient number of points so that the moment matrix does not become singular. To illustrate the multivalued nature of interpolation functions, consider two overlapping clouds defined by kernels centred at points (x_K, y_K) and (x_L, y_L) (see Figure 1). We will simply refer to the cloud defined by centring the kernel at the point (x_K, y_K) as cloud K. Since clouds K and L overlap, some points belong to both clouds K and L. Let (x_Q, y_Q) be a point that belongs to both clouds K and L. If we denote $N_Q^{(K)}$ to be the shape function for point Q computed using cloud K and $N_Q^{(L)}$ to be the shape function for point Q computed using cloud L, then using the fixed kernel technique we note that $N_Q^{(K)} \neq N_Q^{(L)}$ i.e. the interpolation function for point Q is multivalued.

To further illustrate the multivalued nature of interpolation functions, consider a one-dimensional example of 11 equi-spaced points between 0 and 1. The interpolation functions

are constructed for this point distribution using a fixed kernel technique and a cubic spline kernel function. The kernel is defined as

$$\varphi(x - x_I) = \frac{1}{d_x} w\left(\frac{x - x_I}{d_x}\right) \tag{25}$$

$$w(z_I) = \begin{cases} 0, & z_I < -2, \\ \dfrac{1}{6}(z_I + 2)^3, & -2 \leqslant z_I \leqslant -1, \\ \dfrac{2}{3} - z_I^2\left(1 + \dfrac{z_I}{2}\right), & -1 \leqslant z_I \leqslant 0, \\ \dfrac{2}{3} - z_I^2\left(1 - \dfrac{z_I}{2}\right), & 0 \leqslant z_I \leqslant 1, \\ \dfrac{-1}{6}(z_I - 2)^3, & 1 \leqslant z_I \leqslant 2, \\ 0, & z_I > 2 \end{cases} \tag{26}$$

where $z_I = (x - x_I)/d_x$ and d_x denotes the cloud size in the x-direction. A multi-dimensional kernel function can be constructed as a product of one-dimensional kernel functions. In two dimensions, the kernel function is constructed as

$$\varphi(x - x_I, y - y_I) = \frac{1}{d_x} w\left(\frac{x - x_I}{d_x}\right) \frac{1}{d_y} w\left(\frac{y - y_I}{d_y}\right) \tag{27}$$

where d_y is the cloud size in the y-direction. We choose a value of $d_x = 1.17 * \Delta x$, where Δx is the spacing between points, to define 5 point clouds for the one-dimensional example. Shown in Figures 2 and 3 are the interpolation functions obtained when the kernel is centred at points 6 and 5, respectively. When the cloud is centred at point 6, shape functions, $N_I^{(6)}$, $I = 4, \ldots, 8$ are nonzero and the rest are zero. Similarly, when the cloud is centred at point 5, shape functions, $N_I^{(5)}$, $I = 3, \ldots, 7$ are non-zero and the rest are zero. Points 4, 5, 6 and 7 are shared by both clouds 5 and 6 and from Figures 2 and 3 it is clear that $N_I^{(6)} \neq N_I^{(5)}$, $I = 4, \ldots, 7$. From this we conclude that with a fixed kernel technique the interpolation functions are multivalued.

To construct a useful approximation, the interpolation function should be limited to a single value. By using a point collocation approach, the validity of an interpolation function can be limited to a single point. In this paper, we limit the interpolation function to a single value by computing the interpolations at the point where the kernel is fixed i.e. when a kernel is fixed at the point (x_K, y_K), then the only interpolation function values that are computed are $N_I(x_K, y_K)$, $I = 1, 2, \ldots, \mathrm{NP}$. Of course, $N_I(x_K, y_K) = 0$ when the point I does not belong to cloud K. For the one-dimensional example discussed above, when the kernel is fixed at point 6, then the relevant interpolation values that are computed are $N_I(x_6)$, $I = 4, \ldots, 8$ (see Figure 2(b)). Likewise, when the kernel is fixed at point 5, the only useful interpolation values are $N_I(x_5)$, $I = 3, \ldots, 7$ (see Figure 3(b)). By fixing the kernel at every point in the domain, one can compute all the required interpolation functions.

The shape functions constructed by the above manner can be shown to sum to unity i.e.

$$\sum_{I=1}^{\mathrm{NP}} N_I(x_K, y_K) = 1.0, \quad K = 1, 2, \ldots, \mathrm{NP} \tag{28}$$

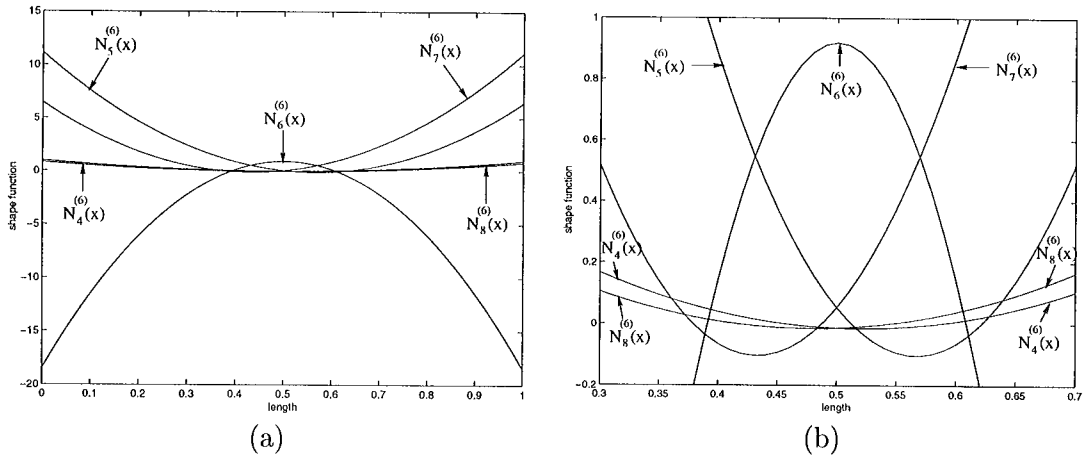

Figure 2. (a) Interpolation functions when the kernel is centred at point 6; (b) A closer look of all the interpolation functions at the point where the kernel is centred.

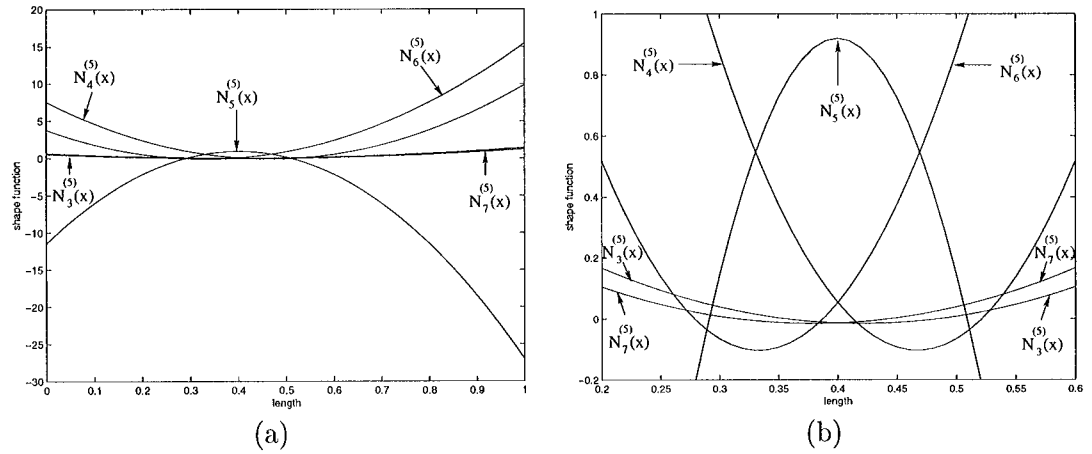

Figure 3. (a) Interpolation functions when the kernel is centred at point 5; (b) A closer look of all the interpolation functions at the point where the kernel is centred.

This property of interpolation functions is also referred to as the concept of partition of unity. To verify this property, consider the first reproducing condition (i.e. $i=1$) given in Equation (10)

$$\int_{\Omega} P^{\mathrm{T}}(s,t)C(x,y)\varphi(x_K-s,y_K-t)\,p_1(s,t)\,\mathrm{d}s\,\mathrm{d}t = p_1(x,y) \tag{29}$$

Substituting $C(x, y) = M^{-1}P(x, y)$ and noting that $p_1(x, y) = 1.0$, a discrete approximation for the above equation can be written as

$$\sum_{I=1}^{\mathrm{NP}} P^{\mathrm{T}}(x, y)M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I = 1.0 \tag{30}$$

Noting that the left-hand side of Equation (30) is the definition of the shape function given in Equation (21), we conclude that

$$\sum_{I=1}^{\mathrm{NP}} N_I(x, y) = 1.0 \tag{31}$$

In particular, when the kernel is fixed at (x_K, y_K), the values of all the shape functions at (x_K, y_K) also sum to unity i.e.

$$\sum_{I=1}^{\mathrm{NP}} N_I(x_K, y_K) = 1.0 \tag{32}$$

The satisfaction of all the consistency conditions given in Equation (10) ensures that the fixed kernel technique can interpolate any function exactly that is included in the definition of $P(x, y)$. Note that the shape functions summing to unity is just the inclusion of the zero-order polynomial in the definition of $P(x, y)$.

3.2. Derivatives of shape functions

From the definition of the shape function given in Equation (21), the derivatives of the shape function can be computed in a straightforward manner as the moment matrix is not a function of x and y. In two dimensions, for a quadratic basis (i.e. $m = 6$), the various derivatives of the shape functions are given as

$$N_{I,x}(x, y) = [0\ \ 1\ \ 0\ \ 2x\ \ y\ \ 0]M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I \tag{33}$$

$$N_{I,y}(x, y) = [0\ \ 0\ \ 1\ \ 0\ \ x\ \ 2y]M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I \tag{34}$$

$$N_{I,xx}(x, y) = [0\ \ 0\ \ 0\ \ 2\ \ 0\ \ 0]M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I \tag{35}$$

$$N_{I,xy}(x, y) = [0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0]M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I \tag{36}$$

$$N_{I,yy}(x, y) = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 2]M^{-1}P(x_I, y_I)\varphi(x_K - x_I, y_K - y_I)\Delta V_I \tag{37}$$

where an inferior comma denotes a partial derivative i.e. $N_{I,x}$ denotes the x-derivative of N_I and $N_{I,xx}$ denotes the second x-derivative of N_I. An advantage with the fixed kernel technique compared with the other reproducing kernel approaches presented in this paper is that the computational effort required to compute the shape functions and its derivatives in much less. This is due to the fact that in the fixed kernel technique the moment matrix is constant within each cloud.

3.3. Comparison of fixed kernel and classical reproducing kernel interpolation functions

A comparison of the interpolation function and its derivatives obtained by the fixed kernel approximation and by the classical reproducing kernel approximation (discused in Section 2)

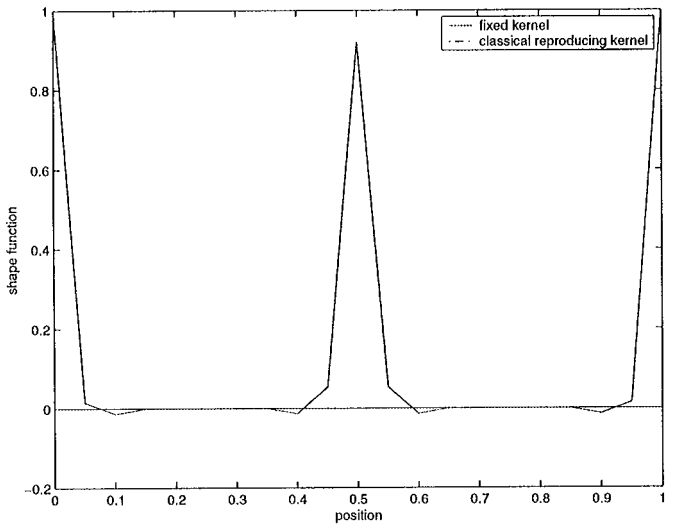

Figure 4. Comparison of the interpolation functions for an interior and boundary nodes obtained with the fixed kernel and the classical reproducing kernel approximations. The interpolation functions are identical.

is shown in Figures 4–6. Interpolation functions are computed for a one-dimensional problem of length 1 using 21 points ($x \in [0,1]$ and $\Delta x = 0.05$). The cloud size for the interior and boundary nodes is $1.17 \times \Delta x$. The nodal volumes are set to unity. Shown in Figures 4–6 are the interpolation functions and the first and second derivatives for an interior node and boundary nodes. We observe that the interpolation function obtained with the fixed kernel approximation is identical to the interpolation function computed by the classical reproducing kernel approximation. However, the first and second derivatives of the interpolation functions in the fixed kernel approach are different from those computed by the classical reproducing kernel technique.

Since the interpolation functions with a fixed kernel technique are multivalued, it is possible to construct interpolation functions that are different from the classical reproducing kernel interpolation functions. However, the approach discussed in Section 3.1 gives rise to identical interpolation functions. The derivatives of the interpolation functions are different in both approaches as the moment matrix and the correction function coefficients are constant within each cloud in the fixed kernel approximation whereas the moment matrix and the correction function coefficients are a function of position within each cloud in the classical reproducing kernel approximation. If the moment matrix is assumed constant within each cloud in the classical reproducing kernel technique, then the derivatives of the interpolation functions in the classical reproducing kernel technique can be shown to match the derivatives of the fixed kernel approximation. It is important to note that the moment matrix within each cloud in the fixed kernel technique is constant by definition (refer Equations (6) and (13)), whereas in the classical reproducing kernel technique the moment matrix is a function of position by definition. Hence, the derivatives of the moment matrix cannnot be ignored in the classical reproducing kernel technique. When the derivatives of the moment matrix are ignored in the classical reproducing kernel technique, it can be shown [29, 30, 11] that the approach

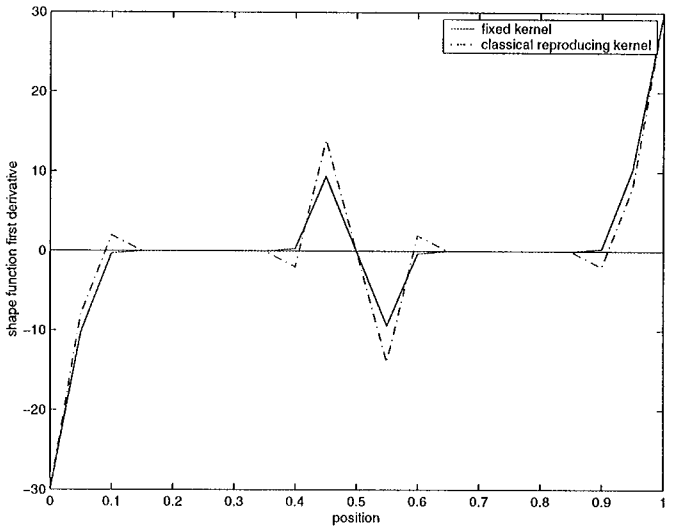

Figure 5. Comparison of the x-derivative of the interpolation functions for an interior and boundary nodes obtained with the fixed kernel and the classical reproducing kernel approximations.

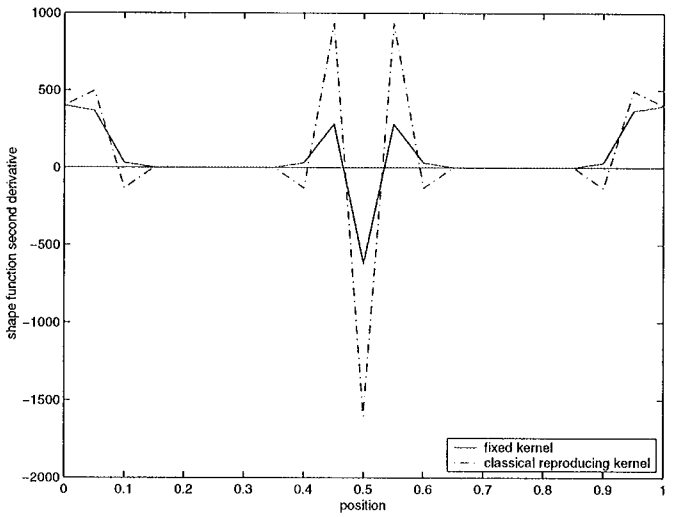

Figure 6. Comparison of the second x-derivative of the interpolation functions for an interior and boundary nodes obtained with the fixed kernel and the classical reproducing kernel approximations.

is equivalent to the diffuse element method [15]. The diffuse element method suffers from poor accuracy [31] as the derivatives in the diffuse element method are consistent but not integrable. In the results section, we compare the convergence characteristics of the fixed kernel approximation and the classical reproducing kernel approximation using a collocation method. We have observed that the convergence characteristics are very similar and the error

with the fixed kernel technique is typically smaller compared to the error with the classical reproducing kernel technique.

4. OTHER KERNEL APPROXIMATIONS

In this section we introduce two other kernel approaches, the moving reproducing kernel and the multiple fixed reproducing kernel technique. In particular, we will show that the moving reproducing kernel and the multiple fixed reproducing kernel approaches are equivalent to the moving least squares and the multiple fixed least squares approaches [18], respectively, for a specific choice of the nodal volumes.

4.1. Moving reproducing kernel technique

In a moving kernel technique, the approximate solution is given by

$$u^{\mathrm{a}}(x,y)=\int_{\Omega}\mathscr{C}(x,y,s,t)\varphi(x-s,y-t)u(s,t)\,\mathrm{d}s\,\mathrm{d}t \tag{38}$$

where $\mathscr{C}$ is the correction function and is given by

$$\mathscr{C}(x,y,s,t)=P^{\mathrm{T}}(s,t)C(x,y) \tag{39}$$

$P^{\mathrm{T}}(s,t)=\{p_1,p_2,\ldots,p_m\}$ is the vector of basis functions and $C^{\mathrm{T}}(x,y)=\{c_1,c_2,\ldots,c_m\}$ is the vector of unknown correction function coefficients. The kernel function $\varphi(x-s,y-t)$ is centred at the point (x,y). The unknown correction function coefficients can be determined by satisfying the consistency conditions, i.e.

$$\int_{\Omega}P^{\mathrm{T}}(s,t)C(x,y)\varphi(x-s,y-t)\,p_i(s,t)\,\mathrm{d}s\,\mathrm{d}t=p_i(x,y),\quad i=1,2,\ldots,m \tag{40}$$

A discrete approximation of the above consistency conditions is written as

$$\sum_{I=1}^{\mathrm{NP}}P^{\mathrm{T}}(x_I,y_I)C(x,y)\varphi(x-x_I,y-y_I)\,p_i(x_I,y_I)\Delta V_I=p_i(x,y),\quad i=1,2,\ldots,m \tag{41}$$

The consistency conditions summarized in Equation (41) can be written in a matrix form as

$$M(x,y)C(x,y)=P(x,y) \tag{42}$$

where $M(x,y)$ is the $m\times m$ moment matrix. The entries in the moment matrix are given by

$$M_{ij}(x,y)=\int_{\Omega}p_i(s,t)\varphi(x-s,y-t)\,p_j(s,t)\,\mathrm{d}s\,\mathrm{d}t,\quad i,j=1,\ldots,m \tag{43}$$

Unlike the fixed kernel technique, the moment matrix in this approach is not constant and is a function of x and y. In discrete form, the components of the moment matrix are defined as

$$M_{ij}(x,y)=\sum_{I=1}^{\mathrm{NP}}p_i(x_I,y_I)\varphi(x-x_I,y-y_I)\,p_j(x_I,y_I)\Delta V_I \tag{44}$$

The moment matrix can also be written as

$$M = FWF^{\mathrm{T}} \tag{45}$$

where F is an $m \times \mathrm{NP}$ matrix as defined in Equation (16) and W is an $\mathrm{NP} \times \mathrm{NP}$ diagonal matrix

$$W = \begin{bmatrix} \varphi(x - x_1, y - y_1)\Delta V_1 & 0 & \cdots & 0 \\ 0 & \varphi(x - x_2, y - y_2)\Delta V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varphi(x - x_{\mathrm{NP}}, y - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}} \end{bmatrix} \tag{46}$$

From Equation (42), the unknown correction function coefficients are computed as

$$C(x, y) = M^{-1}(x, y)P(x, y) \tag{47}$$

Substituting the correction function coefficients into Equation (38), the approximate solution is written as

$$u^{\mathrm{a}}(x, y) = \int_{\Omega} P^{\mathrm{T}}(x, y)M^{-1}(x, y)P(s, t)\varphi(x - s, y - t)u(s, t)\,\mathrm{d}s\,\mathrm{d}t \tag{48}$$

The approximation in Equation (48) can be rewritten as

$$u^{\mathrm{a}}(x, y) = \sum_{I=1}^{\mathrm{NP}} N_I(x, y)u_I \tag{49}$$

where u_I is a nodal unknown for node I, and $N_I(x, y)$ is the moving kernel interpolation function defined as

$$N_I(x, y) = P^{\mathrm{T}}(x, y)M^{-1}(x, y)P(x_I, y_I)\varphi(x - x_I, y - y_I)\Delta V_I \tag{50}$$

Defining

$$B = \begin{bmatrix} p_1(x_1, y_1)\varphi(x - x_1, y - y_1)\Delta V_1 & \cdots & p_1(x_{\mathrm{NP}}, y_{\mathrm{NP}})\varphi(x - x_{\mathrm{NP}}, y - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}} \\ p_2(x_1, y_1)\varphi(x - x_1, y - y_1)\Delta V_1 & \cdots & p_2(x_{\mathrm{NP}}, y_{\mathrm{NP}})\varphi(x - x_{\mathrm{NP}}, y - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}} \\ \vdots & \ddots & \vdots \\ p_m(x_1, y_1)\varphi(x - x_1, y - y_1)\Delta V_1 & \cdots & p_m(x_{\mathrm{NP}}, y_{\mathrm{NP}})\varphi(x - x_{\mathrm{NP}}, y - y_{\mathrm{NP}})\Delta V_{\mathrm{NP}}) \end{bmatrix} \tag{51}$$

the interpolation function can be rewritten as

$$N_I(x, y) = \sum_{l=1}^{m} p_l(x, y)\bar{C}_{lI}^{-1} \tag{52}$$

where

$$\bar{C}^{-1} = M^{-1}B \tag{53}$$

Note that the $\bar{C}$ matrix for the moving kernel technique is different from the $\bar{C}$ matrix introduced in the fixed kernel technique as the definition of the M and B matrices is different in each case.

Remarks:

1. When all the nodal volumes are set to unity i.e. $\Delta V_I = 1$, then the definition of the shape function given in Equation (50) is identical to the definition of the shape function given by the moving least squares approach [17, 18].
2. In a moving reproducing kernel approach the interpolation functions are not multivalued. They can be computed uniquely and can be shown to sum to unity. The moving kernel technique can also interpolate any function exactly that is included in the definition of $P(x, y)$.
3. As mentioned earlier, the moment matrix in the moving reproducing kernel approach is a function of x and y. The computation of the various derivatives of the shape functions requires the calculation of the derivatives of the correction function coefficients which would require the calculation of various derivatives of the moment matrix (refer Equation (42)). This is a more expensive procedure compared to the calculation of derivatives in the fixed reproducing kernel technique.

4.2. *Multiple fixed reproducing kernel technique*

In a multiple fixed kernel technique, the approximate solution is given by

$$u^{\mathrm{a}}(x, y) = \int_{\Omega} \mathscr{C}(x, y, s, t)\varphi_{s,t}(s - x, t - y)u(s, t)\,\mathrm{d}s\,\mathrm{d}t \tag{54}$$

where $\mathscr{C}$ denotes the correction function and is as defined in Equation (39). The kernel function $\varphi_{s,t}(s-x, t-y)$ is centred at the point (s, t) and the subscript (s, t) denotes that. Note the difference between the moving kernel technique where the kernel function is centred at the point (x, y) and the multiple fixed kernel technique where the kernel function is centred at (s, t). This method is referred to as a multiple fixed kernel technique as it makes use of different fixed kernels $(\varphi_{s,t})$ to construct the approximation $u^{\mathrm{a}}(x, y)$ in Equation (54). The unknown correction function coefficients in Equation (54) are determined by satisfying the consistency conditions, i.e.

$$\int_{\Omega} P^{\mathrm{T}}(s, t)C(x, y)\varphi_{s,t}(s - x, t - y)\,p_i(s, t)\,\mathrm{d}s\,\mathrm{d}t = p_i(x, y), \quad i = 1, 2, \ldots, m \tag{55}$$

A discrete approximation of the above consistency conditions is written as

$$\sum_{I=1}^{\mathrm{NP}} P^{\mathrm{T}}(x_I, y_I)C(x, y)\varphi_I(x_I - x, y_I - y)\,p_i(x_I, y_I)\Delta V_I = p_i(x, y), \quad i = 1, 2, \ldots, m \tag{56}$$

Note that we have simply denoted φ_{x_I, y_I} by φ_I, it being understood that they are the same. The consistency conditions summarized in Equation (56) can be written in a matrix form as

$$M(x, y)C(x, y) = P(x, y) \tag{57}$$

where $M(x, y)$ is the $m \times m$ moment matrix. Unlike the fixed kernel technique, the moment matrix is not constant and is a function of x and y. The entries in the moment matrix are given by

$$M_{ij}(x, y) = \int_{\Omega} p_i(s, t)\varphi_{s,t}(s - x, t - y)\,p_j(s, t)\,\mathrm{d}s\,\mathrm{d}t, \quad i, j = 1, \ldots, m \tag{58}$$

In discrete form, the components of the moment matrix are defined as

$$M_{ij}(x,y)=\sum_{I=1}^{\mathrm{NP}} p_i(x_I,y_I)\varphi_I(x_I-x,y_I-y)\,p_j(x_I,y_I)\Delta V_I \tag{59}$$

The moment matrix can also be written as

$$M=FWF^{\mathrm{T}} \tag{60}$$

where F is as defined in Equation (16) and

$$W=\begin{bmatrix} \varphi_1(x_1-x,y_1-y)\Delta V_1 & 0 & \cdots & 0 \\ 0 & \varphi_2(x_2-x,y_2-y)\Delta V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varphi_{\mathrm{NP}}(x_{\mathrm{NP}}-x,y_{\mathrm{NP}}-y)\Delta V_{\mathrm{NP}} \end{bmatrix} \tag{61}$$

From Equation (57), the unknown correction function coefficients are computed as

$$C(x,y)=M^{-1}(x,y)P(x,y) \tag{62}$$

Substituting the correction function coefficients into Equation (54)

$$u^{\mathrm{a}}(x,y)=\int_{\Omega} P^{\mathrm{T}}(x,y)M^{-1}(x,y)P(s,t)\varphi_{s,t}(s-x,t-y)u(s,t)\,\mathrm{d}s\,\mathrm{d}t \tag{63}$$

The approximation in Equation (63) can be rewritten as

$$u^{\mathrm{a}}(x,y)=\sum_{I=1}^{\mathrm{NP}} N_I(x,y)u_I \tag{64}$$

where u_I is a nodal unknown for node I, and $N_I(x,y)$ is the multiple fixed kernel interpolation function defined as

$$N_I(x,y)=P^{\mathrm{T}}(x,y)M^{-1}(x,y)P(x_I,y_I)\varphi_I(x_I-x,y_I-y)\Delta V_I \tag{65}$$

Defining

$$B=\begin{bmatrix} p_1(x_1,y_1)\varphi_1(x_1-x,y_1-y)\Delta V_1 & \cdots & p_1(x_{\mathrm{NP}},y_{\mathrm{NP}})\varphi_{\mathrm{NP}}(x_{\mathrm{NP}}-x,y_{\mathrm{NP}}-y)\Delta V_{\mathrm{NP}} \\ p_2(x_1,y_1)\varphi_1(x_1-x,y_1-y)\Delta V_1 & \cdots & p_2(x_{\mathrm{NP}},y_{\mathrm{NP}})\varphi_{\mathrm{NP}}(x_{\mathrm{NP}}-x,y_{\mathrm{NP}}-y)\Delta V_{\mathrm{NP}} \\ \vdots & \ddots & \vdots \\ p_m(x_1,y_1)\varphi_1(x_1-x,y_1-y)\Delta V_1 & \cdots & p_m(x_{\mathrm{NP}},y_{\mathrm{NP}})\varphi_{\mathrm{NP}}(x_{\mathrm{NP}}-x,y_{\mathrm{NP}}-y)\Delta V_{\mathrm{NP}}) \end{bmatrix} \tag{66}$$

the interpolation function can be rewritten as

$$N_I(x,y)=\sum_{l=1}^{m} p_l(x,y)\bar{C}_{lI}^{-1} \tag{67}$$

where

$$\bar{C}^{-1} = M^{-1}B \tag{68}$$

Note that the definitions of M and B for the multiple fixed kernel technique are different from the definitions given earlier for the fixed kernel and moving kernel techniques.
Remarks:

1. When all the nodal volumes are set to unity i.e. $\Delta V_I = 1$, then the definition of the shape function given in Equation (65) is identical to the definition of shape function given by the multiple fixed least squares approach introduced in References [17, 18].
2. Like in a moving kernel technique, the shape functions in a multiple fixed kernel technique are not multivalued and can be computed uniquely.
3. Since the moment matrix in a multiple fixed kernel technique is a function of x and y, the calculation of the shape function and its derivatives is more expensive compared to the calculations in the fixed kernel technique.
4. A multiple fixed kernel technique is more flexible than the moving kernel technique in the sense that different kernel functions of varying shape can be defined at each point. In the particular case of when the same kernel function with an invariant shape is defined at every point, the multiple fixed kernel technique is identical to the moving kernel technique.

5. DISCRETIZATION

The discretization of the governing partial differential equations is accomplished by a point collocation approach. Collocation methods for boundary value problems using piecewise polynomial functions are well known in the literature (see e.g. Reference [32]). An alternative to a collocation approach would be to employ a Galerkin technique, but this would require complex procedures such as a background grid [1] for volume integration or stabilization techniques [27] for nodal integration. Either a point collocation method or a Galerkin method can be combined with any of the reproducing kernel techniques described above for construction of the interpolation function. In particular, in this paper, we construct interpolation functions by a fixed kernel technique and apply a point collocation method for discretization. To illustrate the point collocation approach, consider the following model problem:

$$\mathscr{L}u = f \quad \text{in } \Omega \tag{69}$$

$$u = g \quad \text{on } \Gamma_g \tag{70}$$

$$\frac{\partial u}{\partial n} = h \quad \text{on } \Gamma_h \tag{71}$$

where $\mathscr{L}$ is the differential operator, u is the unknown function, f is the forcing term, Γ_g is the portion of the boundary where Dirichlet boundary conditions are specified and Γ_h is the portion of the boundary where Neumann boundary conditions are specified. The goal is to find an approximate solution $u^{\mathrm{a}}(x, y)$ that approaches the exact solution $u(x, y)$ as the number of points NP increases. $u^{\mathrm{a}}(x, y)$ then satisfies the governing equations introduced in

Equations (69)–(71)

$$\mathscr{L}u^{\mathrm{a}} \simeq f \quad \text{in } \Omega \tag{72}$$

$$u^{\mathrm{a}} \simeq g \quad \text{on } \Gamma_g \tag{73}$$

$$\frac{\partial u^{\mathrm{a}}}{\partial n} \simeq h \quad \text{on } \Gamma_h \tag{74}$$

In a point collocation approach, Equations (72)–(74) are satisfied at every point or a node. Let N_{d} to be the number of points carrying a Dirichlet boundary condition, N_{n} to be the number of points carrying a Neumann boundary condition, N_{r} to be the remaining nodes (typically interior nodes) with no boundary conditions and $\mathrm{NP} = N_{\mathrm{r}} + N_{\mathrm{d}} + N_{\mathrm{n}}$. For a node that is in the interior and is not constrained, the collocation approach satisfies the equation

$$\mathscr{L}u^{\mathrm{a}}(x_I, y_I) = f(x_I, y_I), \quad I = 1, 2, \ldots, N_{\mathrm{r}} \tag{75}$$

For points that are constrained by a Dirichlet boundary condition, the point collocation technique satisfies

$$u^{\mathrm{a}}(x_I, y_I) = g(x_I, y_I), \quad I = 1, 2, \ldots, N_{\mathrm{d}} \tag{76}$$

and for points with Neumann boundary conditions, we satisfy

$$\frac{\partial u^{\mathrm{a}}}{\partial n}(x_I, y_I) = h(x_I, y_I), \quad I = 1, 2, \ldots, N_{\mathrm{n}} \tag{77}$$

Employing a discrete approximation for u^{a}, the point collocation approach gives rise to a matrix problem

$$Ku = b \tag{78}$$

where $K \in \mathscr{R}^{\mathrm{NP}\times\mathrm{NP}}$ is the coefficient matrix, $u \in \mathscr{R}^{\mathrm{NP}\times 1}$ is the unknown vector and $b \in \mathscr{R}^{\mathrm{NP}\times 1}$ is the known right-hand side vector. The entries in K can be calculated from Equations (75)–(77). If a node I is in the interior and is not constrained by a Dirichlet or a Neumann boundary condition, the entries in the Ith row of K are given by

$$K_{IJ} = \mathscr{L}N_J(x_I, y_I), \quad J = 1, 2, \ldots, \mathrm{NP} \tag{79}$$

where K_{IJ} is the Ith row and Jth column entry of K, and $\mathscr{L}N_J(x_I, y_I)$ is the differential operator applied on the Jth node shape function and evaluated at the point (x_I, y_I). Similarly, if a node M is constrained by a Dirichlet boundary condition and a node N is constrained by a Neumann boundary condition, the entries in K are given by

$$K_{MJ} = N_J(x_M, y_M), \quad J = 1, 2, \ldots, \mathrm{NP} \tag{80}$$

$$K_{NJ} = \frac{\partial N_J}{\partial n}(x_N, y_N), \quad J = 1, 2, \ldots, \mathrm{NP} \tag{81}$$

Once u_I, $I = 1, 2, \ldots, \mathrm{NP}$, are computed by a solution of Equation (78), the approximation u^{a} at any point (x_I, y_I) can be computed by

$$u^{\mathrm{a}}(x_I, y_I) = \sum_{J=1}^{\mathrm{NP}} N_J(x_I, y_I) u_J \tag{82}$$

6. RESULTS

In this section we show results for several one- and two-dimensional problems using the fixed kernel technique (or hereafter simply referred to as the finite cloud method). Unless otherwise mentioned we use the cubic spline kernel function as given in Equation (26) and a multi-dimensional kernel function is constructed as given in Equation (27). For a uniform distribution of points with a quadratic basis, the cloud sizes are chosen to be $r_x = 1.17 * \Delta x$ and $r_y = 1.17 * \Delta y$. The convergence of the method is measured by using a global error measure [33]

$$\varepsilon = \frac{1}{|u^{(e)}|_{\max}} \sqrt{\frac{1}{\mathrm{NP}} \sum_{I=1}^{\mathrm{NP}} \left[u_I^{(\mathrm{e})} - u_I^{(\mathrm{c})}\right]^2} \tag{83}$$

where ε is the error in the solution and the superscripts (e) and (c) denote, respectively, the exact and the computed solutions.

6.1. 1-D Examples

The first one-dimensional example is a Poisson equation with a forcing term that is a function of x. The governing equation and boundary conditions are

$$\frac{\partial^2 u}{\partial x^2} = \frac{105}{2} x^2 - \frac{15}{2}, \quad -1 < x < 1 \tag{84}$$

$$u(x = -1) = 1 \tag{85}$$

$$\frac{\partial u}{\partial x}(x = 1) = 10 \tag{86}$$

The exact solution for this problem is given by

$$u = \frac{35}{8} x^4 - \frac{15}{4} x^2 + \frac{3}{8} \tag{87}$$

Since the exact solution is outside the quadratic reproducing basis, this problem is analysed by employing a uniform distribution of $21, 41, 81, 161, 321$ and 641 points to study the convergence behaviour of the method. A comparison of the exact and computed solutions for this example is shown in Figure 7. The convergence of the method is shown in Figure 8. The convergence plots shown in Figure 8 are obtained with exact and unit nodal volumes. In the case of exact nodal volumes, the sum of all the nodal volumes ($\sum_{I=1}^{\mathrm{NP}} \Delta V_I$) equals to the length of the domain and in the unit nodal volume case ΔV_I is simply set to 1 for each node. The convergence rates obtained with exact and unit nodal volumes are almost identical. The convergence rate in u is found to be 1.99 and the convergence rate in $u_{,x}$ is found to be 2.0. For this example, the error in the derivative is found to be smaller than the error in the solution.

The second 1-D example we consider has a high gradient in a local area and it would be interesting to see how the finite cloud method performs. The governing equation and boundary conditions for this example are

$$\frac{\partial^2 u}{\partial x^2} = -6x - \left[\frac{2}{\alpha^2} - 4\left(\frac{x-\beta}{\alpha^2}\right)^2\right] \exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2\right], \quad 0 \leqslant x \leqslant 1 \tag{88}$$

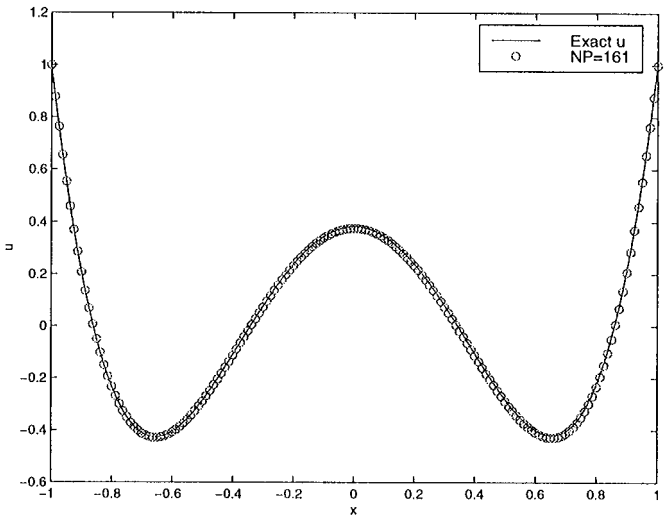

Figure 7. Comparison of exact and computed solutions for the 1-D Poisson equation.

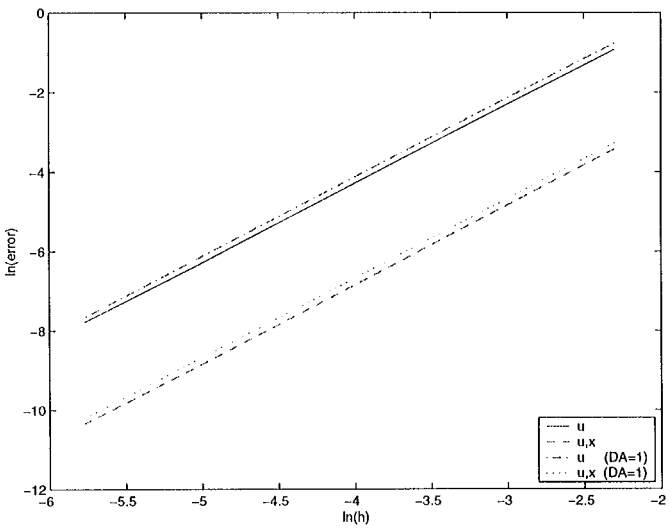

Figure 8. Convergence of the finite cloud method for the 1-D Poisson equation. DA = 1 is the case for unit nodal volumes.

$$u(x=0)=\exp\left(-\frac{\beta^2}{\alpha^2}\right) \tag{89}$$

$$\frac{\partial u}{\partial x}(x=1)=-3-2\left(\frac{1-\beta}{\alpha^2}\right)\exp\left[-\left(\frac{1-\beta}{\alpha}\right)^2\right] \tag{90}$$

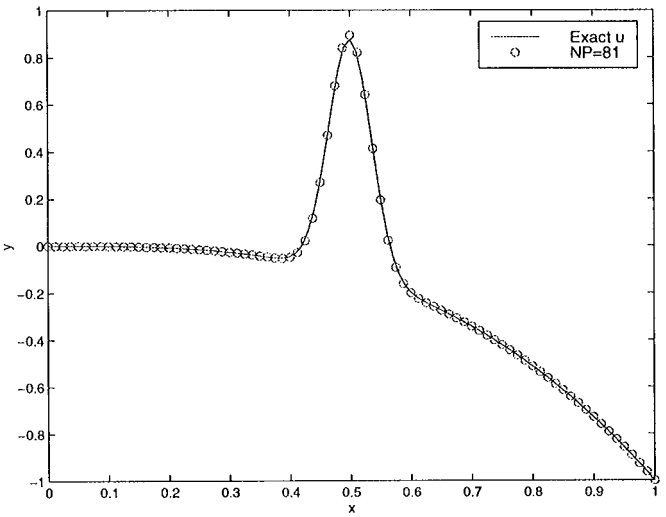

Figure 9. Comparison of exact and computed solutions for the 1-D Poisson equation with a high local gradient.

The exact solution for this problem is given by

$$u = -x^3 + \exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2\right] \tag{91}$$

For the results shown in this paper, we use $\beta = 0.5$ and $\alpha = 0.05$. A comparison between exact and computed solutions is shown in Figure 9. Note that the solution has a high local gradient near $x = 0.5$. The convergence of the finite cloud method for this example is shown in Figure 10. From Figure 10, the convergence rate of the solution is 2.0 and the convergence rate of the x-derivative of the solution is 2.01.

6.2. 2-D Laplace equation

Consider a Laplace equation with cubic exact solution and with Dirichlet boundary conditions along the four edges. The governing equation is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad 0 < x < 1, \quad 0 < y < 1 \tag{92}$$

The Dirichlet boundary conditions along the four edges are given by

$$u(x = 0) = -y^3 \tag{93}$$

$$u(x = 1) = -1 - y^3 + 3y^2 + 3y \tag{94}$$

$$u(y = 0) = -x^3 \tag{95}$$

$$u(y = 1) = -1 - x^3 + 3x^2 + 3x \tag{96}$$

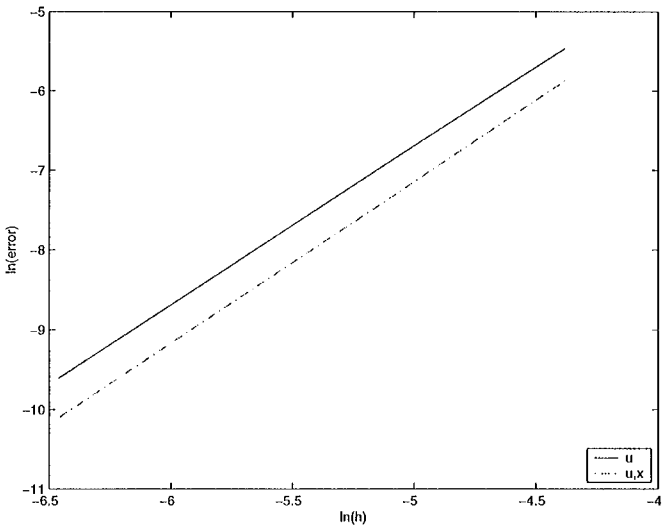

Figure 10. Convergence of the finite cloud method for the 1-D Poisson equation with a high local gradient.

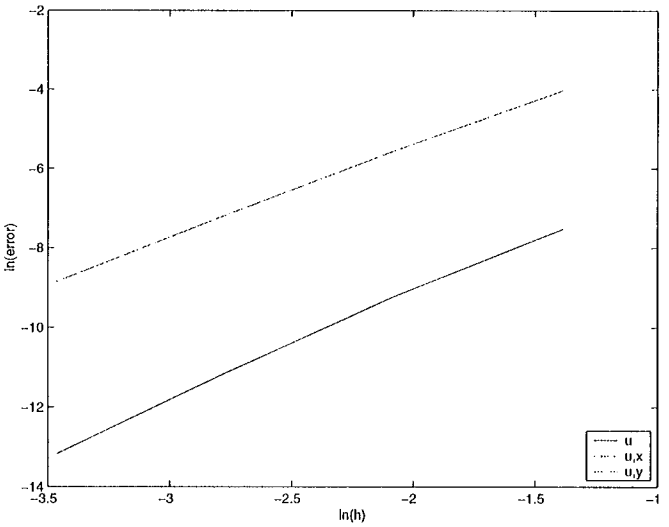

Figure 11. Convergence of the finite cloud method for the Dirichlet Laplace problem.

The exact solution for this problem is given by

$$u(x, y) = -x^3 - y^3 + 3xy^2 + 3x^2y \tag{97}$$

A plot of the exact solution can be found in Reference [26]. The computed solution aproaches the exact solution as we increase the point distribution and the convergence of the method for this example is shown in Figure 11. From Figure 11, the convergence rate of the solution is found to be 2.6 and the convergence rate of the x and y derivatives is found to be 2.3.

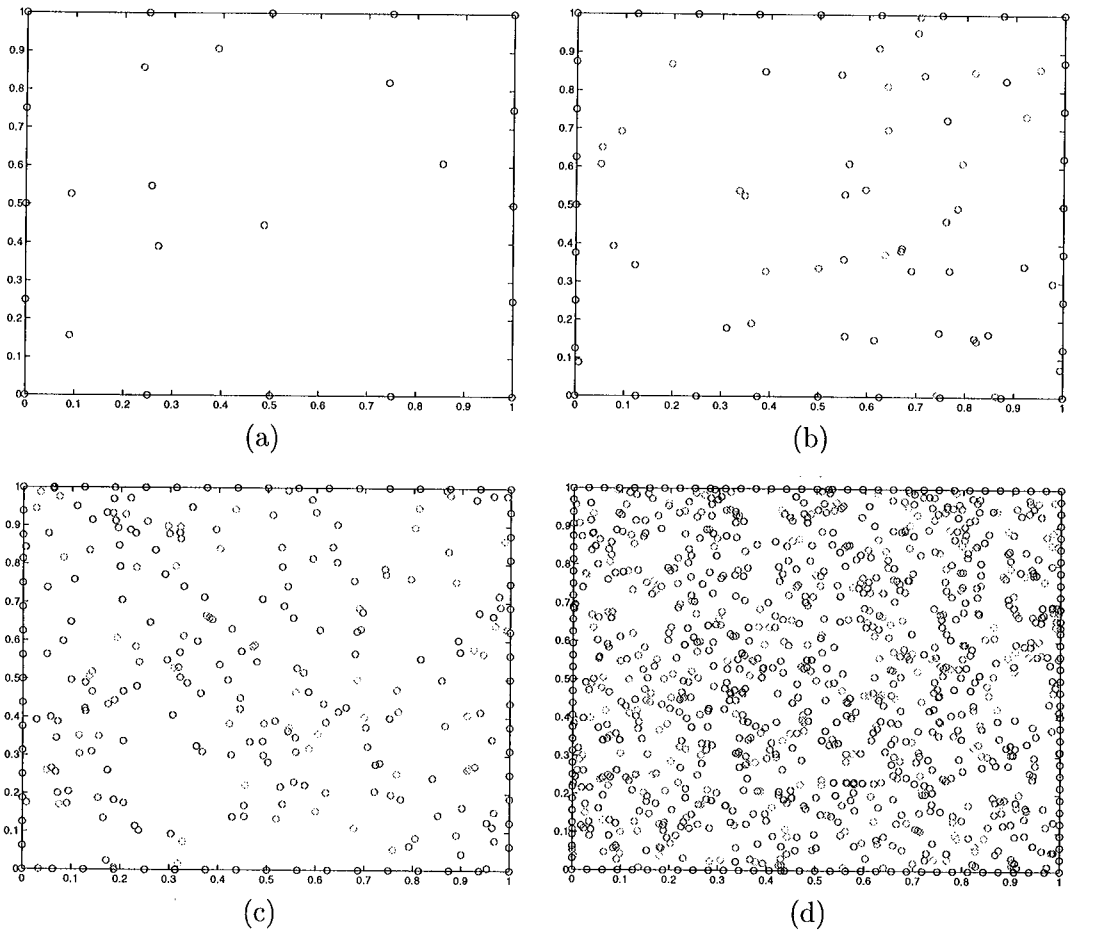

Figure 12. Random point distributions considered for the solution of the Dirichlet Laplace example: (a) $5^2 = 25$ points; (b) $9^2 = 81$ points; (c) $17^2 = 289$ points; and (d) $33^2 = 1089$ points.

6.2.1. Convergence for random point distributions. Let us now consider the convergence of the finite cloud method for random point distributions. The Dirichlet Laplace problem is solved by employing a random distribution of $25, 81, 289$ and 1089 points as shown in Figure 12. For this example, we use $d_x = 3.0h_x$ and $d_y = 3.0h_y$ where

$$h = h_x = h_y = \frac{\text{Length}}{\sqrt{(\text{NP}) - 1}} \tag{98}$$

The convergence of the method for the random point distribution case is shown in Figure 13. The convergence rate of the solution is found to be 1.5 and the convergence rate in the derivatives is 1.9. These results indicate that the convergence rate of the method is affected when a random distribution of points is considered.

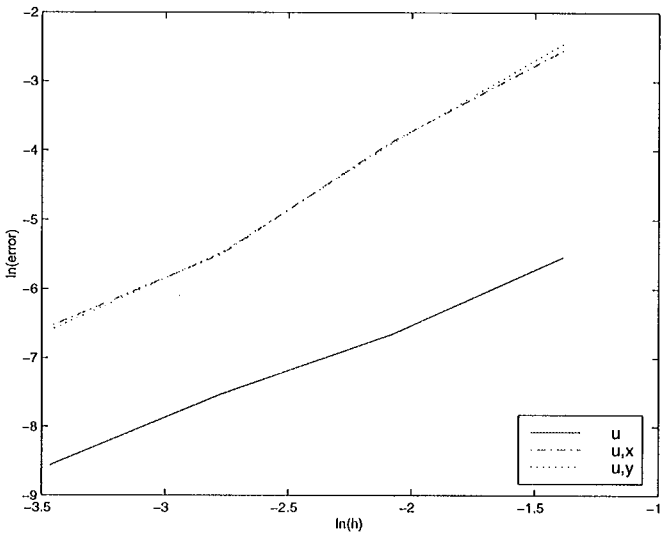

Figure 13. Convergence of the finite cloud method for random point distributions.

6.2.2. Laplace equation with Dirichlet and Neumann boundary conditions. Now consider the solution of the Laplace equation given in Equation (92), but with Neumann boundary conditions along the two edges. The boundary conditions are summarized as

$$u(x=0) = -y^3 \tag{99}$$

$$u(x=1) = -1 - y^3 + 3y^2 + 3y \tag{100}$$

$$\frac{\partial u}{\partial y}(y=0) = 3x^2 \tag{101}$$

$$\frac{\partial u}{\partial y}(y=1) = -3 + 6x + 3x^2 \tag{102}$$

The exact solution for this case is also given by Equation (97). This example is solved by employing a uniform distribution of $5 \times 5, 9 \times 9, 17 \times 17$ and 33×33 points along the x- and y-directions. The convergence of the method for exact and unit nodal volumes is shown in Figure 14. For exact nodal volumes, the convergence rate in the solution is 2.04 and the convergence rates in the x and y derivatives are 2.19 and 1.92, respectively. For the case of unit nodal volumes, the convergence rate in the solution is 2.05 and the convergence rates in the x and y derivatives are 2.26 and 1.96, respectively. Since the convergence rates for both cases are approximately equal, the easiest choice for nodal volumes is to set them to unity. Also, note that when a Neumann boundary condition is imposed, the convergence rate of the method does not change significantly.

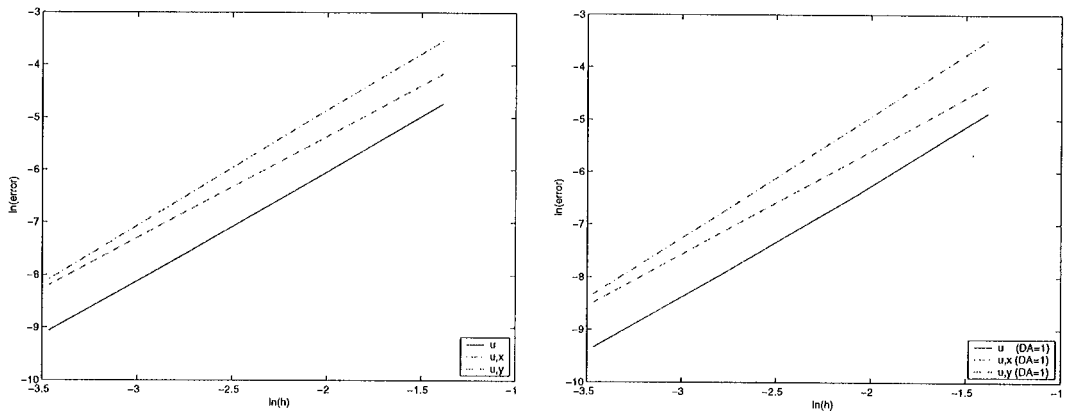

Figure 14. (a) Convergence of the method with exact nodal volumes; (b) Convergence of the method with unit nodal volumes.

6.3. 2-D Poisson equation

The example we consider here is a two-dimensional extension of the 1-D Poisson example with a high local gradient. The governing equation along with the boundary conditions are

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=-6x-6y-\left[\frac{4}{\alpha^2}-4\left(\frac{x-\beta}{\alpha^2}\right)^2-4\left(\frac{y-\beta}{\alpha^2}\right)^2\right]\times\exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2-\left(\frac{y-\beta}{\alpha}\right)^2\right] \tag{103}$$

$$u(x=0)=-y^3+\exp\left[-\left(\frac{\beta}{\alpha}\right)^2-\left(\frac{y-\beta}{\alpha}\right)^2\right] \tag{104}$$

$$u(x=1)=-1-y^3+\exp\left[-\left(\frac{1-\beta}{\alpha}\right)^2-\left(\frac{y-\beta}{\alpha}\right)^2\right] \tag{105}$$

$$u_{,y}(y=0)=\frac{2\beta}{\alpha^2}\exp\left[-\left(\frac{\beta}{\alpha}\right)^2-\left(\frac{x-\beta}{\alpha}\right)^2\right] \tag{106}$$

$$u_{,y}(y=1)=-3-2\frac{1-\beta}{\alpha^2}\exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2-\left(\frac{1-\beta}{\alpha}\right)^2\right] \tag{107}$$

The exact solution for this problem is given by

$$u=-x^3-y^3+\exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2-\left(\frac{y-\beta}{\alpha}\right)^2\right] \tag{108}$$

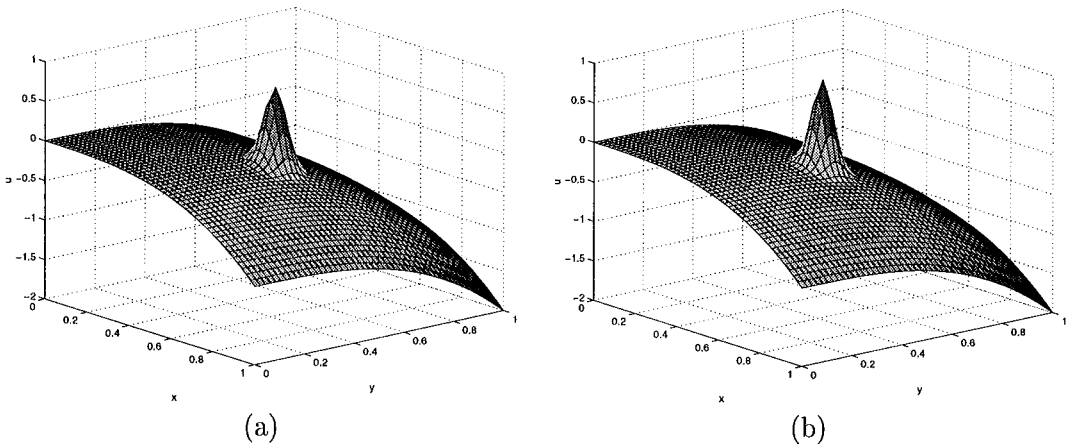

Figure 15. Comparison of: (a) exact; and (b) computed results for the 2-D Poisson equation with a high local gradient.

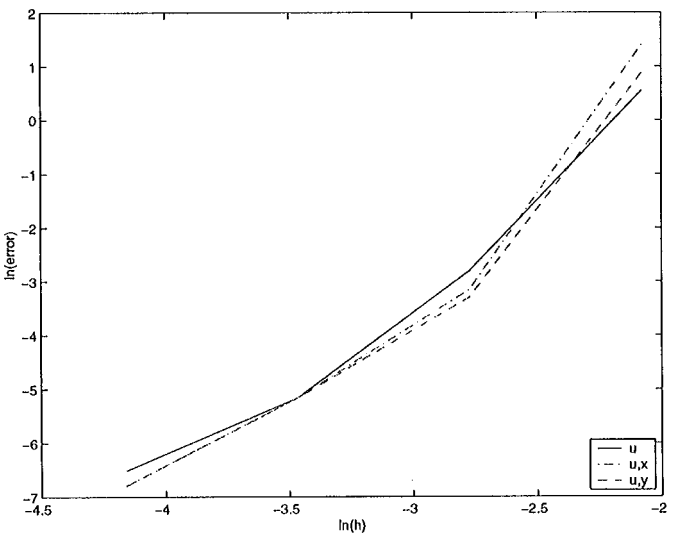

Figure 16. Convergence of the finite cloud method for the 2-D Poisson equation with a high local gradient.

A comparison between exact and computed solutions is shown in Figure 15. The convergence of the finite cloud method using a uniform distribution of $9 \times 9, 17 \times 17, 33 \times 33$ and 65×65 points is shown in Figure 16. From Figure 16, the convergence rate of the solution is 1.95 and the convergence rate of the x and y derivatives of the solution is found to be 2.6.

6.4. Heat conduction

The steady-state heat conduction problem considered here is a rectangular plate ($0.5 \times 1\text{in}^2$) with a heat source. The governing equation is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = -2s^2 \operatorname{sech}^2[s(y-0.5)] \tanh[s(y-0.5)] \tag{109}$$

The boundary conditions are given by

$$T(y=0) = -\tanh(3s) \tag{110}$$

$$T(y=1) = \tanh(3s) \tag{111}$$

$$\frac{\partial T}{\partial x}(x=0) = 0 \tag{112}$$

$$\frac{\partial T}{\partial x}(x=0.5) = 0 \tag{113}$$

The exact solution for this problem is given by

$$T = \tanh[s(y-0.5)] \tag{114}$$

In the above equations, s is a free parameter. For the results shown in this paper, we use a value of $s=30$. Shown in Figure 17 is the variation of the temperature and the gradient of the temperature in the y-direction. The computed solution matches with the exact solution. This example has a high gradient at $y=0.5$ and the finite cloud method captures the gradient nicely. The convergence of the finite cloud method for this problem is shown in Figure 18. The convergence rate of the temperature is found to be 1.98 and the convergence rate of the derivative of the temperature is found to be 2.05. Also shown in Figure 18 is the convergence of the point collocation method using classical reproducing kernel approximation [26]. We observe that the convergence rates for both techniques are very similar, but the error in the solution with the finite cloud method is smaller compared to the error in the solution with the classical reproducing kernel technique. Also, the error in the derivative is smaller compared to the error in the solution with the finite cloud method, while this is not true with the point collocation method based on classical reproducing kernel approximations.

6.5. Elasticity

The governing equations for elasticity (in a plane stress situation) are

$$\frac{2}{1-v}\frac{\partial^2 u}{\partial x^2} + \frac{1+v}{1-v}\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{115}$$

$$\frac{2}{1-v}\frac{\partial^2 v}{\partial y^2} + \frac{1+v}{1-v}\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2} = 0 \tag{116}$$

where u and v are the x and y components of the displacement and v is the Poisson's ratio.

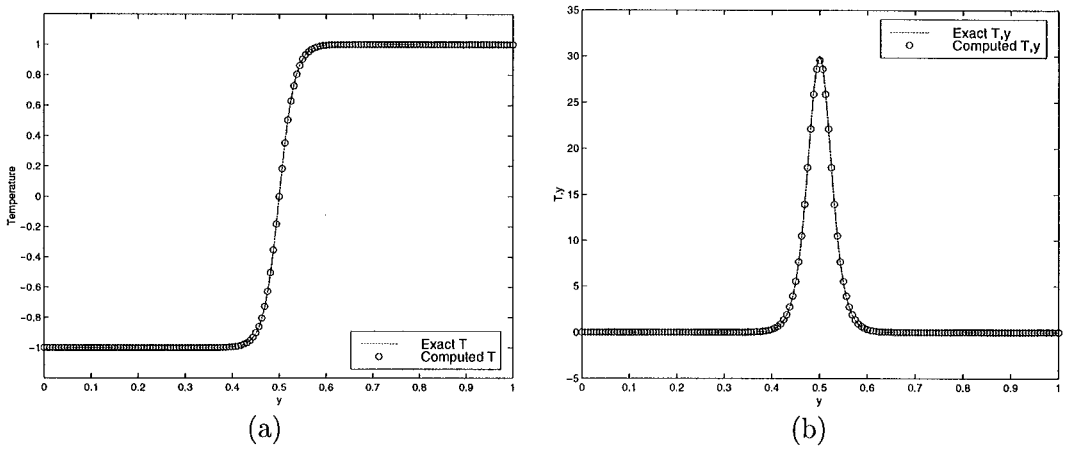

Figure 17. Comparison of computed and exact results for the heat conduction problem: (a) temperature variation along the y-direction; (b) gradient of the temperature along the y-direction.

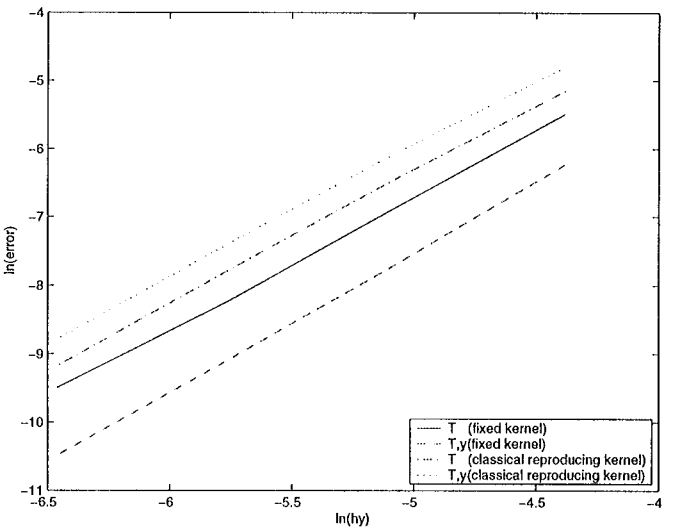

Figure 18. Comparison of the convergence of the finite cloud method and the point collocation method based on classical reproducing kernel approximations [26] for the heat conduction problem.

The finite cloud method satisfies the patch tests shown in Figure 19. With a modulus of unity and a Poisson's ratio of 0.25, the exact solution for the example shown in Figure 19(a) is

$$u = x \tag{117}$$

$$v = -\frac{y}{4} \tag{118}$$

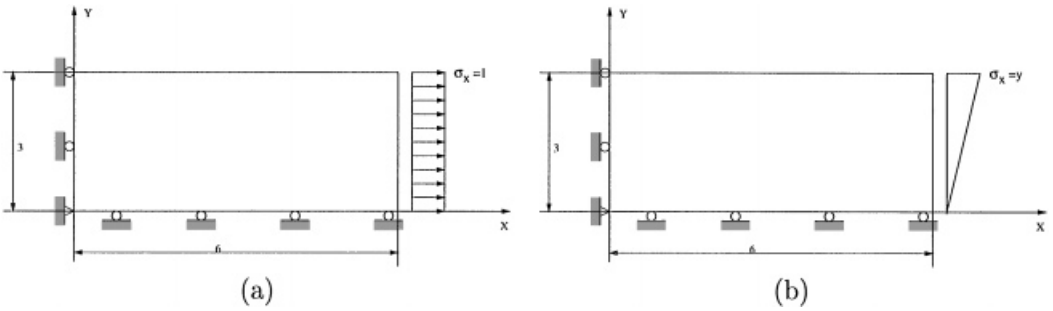

Figure 19. Patch tests for elasticity.

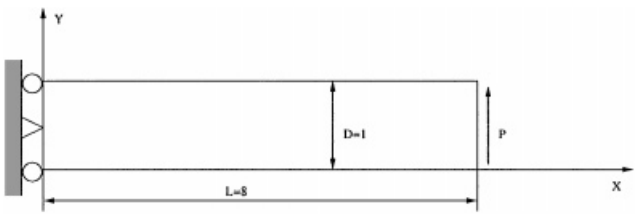

Figure 20. A cantilever beam subjected to an end load.

and the exact solution for the example shown in Figure 19(b) is

$$u = xy \tag{119}$$

$$v = -\frac{x^2}{2} - 8y^2 \tag{120}$$

The finite cloud method gives nodally exact results for all point distributions for both the patch tests as the exact solution is in the reproducing basis. A slightly more difficult example is the cantilever beam subjected to an end load as shown in Figure 20. The exact solution for this problem can be found in Reference [34]. The convergence of the finite cloud method for this example is shown in Figure 21. The convergence rates for u and v are found to be 2.13 and 2.07, respectively. The convergence rates for the x and y derivatives of u are found to be 2.10 and 2.02, respectively, and the convergence rates for the x and y derivatives of v are found to be 2.01 and 2.25, respectively. Shown in Figure 22 is the comparison of the convergence of the finite cloud method and the point collocation method based on classical reproducing kernel approximations [26]. We again observe that the convergence rates for both techniques are very similar, but the error in the solution and its derivative with the finite cloud method is smaller compared to the error in the solution and its derivative with the classical reproducing kernel technique.

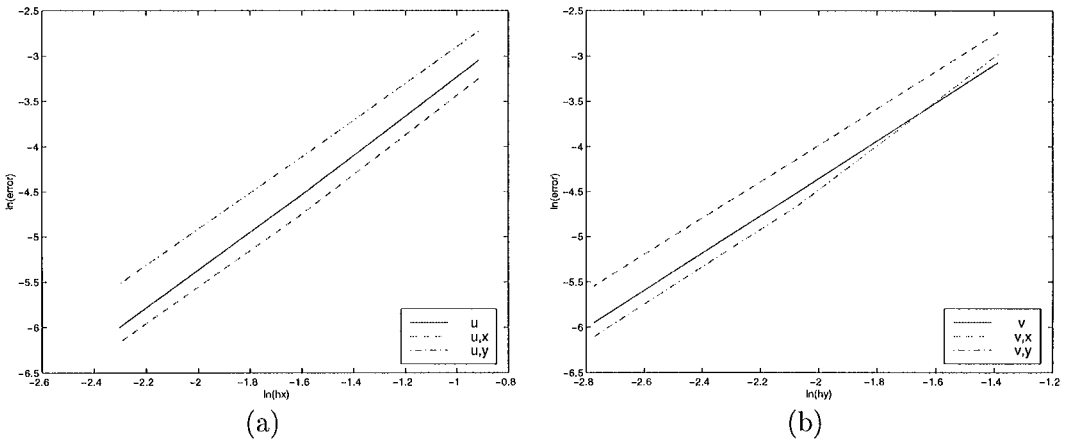

Figure 21. Convergence of the finite cloud method for the beam example with an end load: (a) convergence in u and its gradients; (b) convergence in v and its gradients.

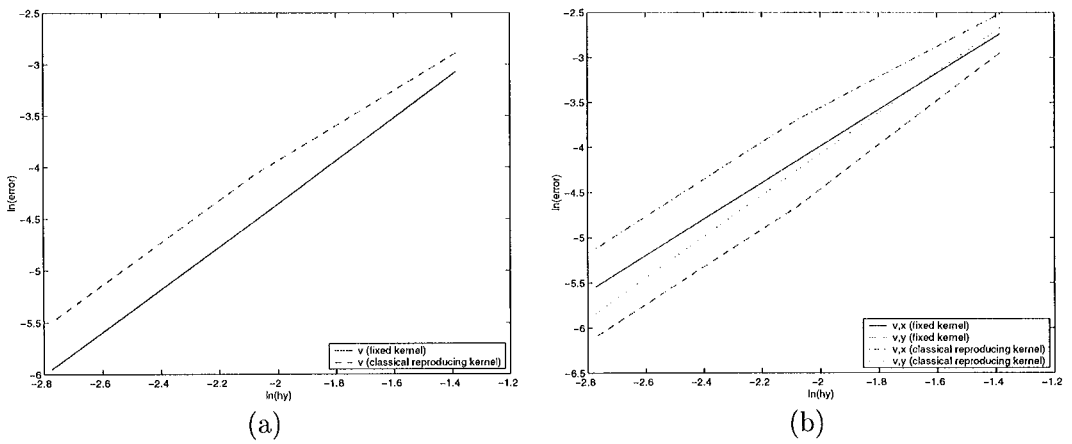

Figure 22. Comparison of the convergence of the finite cloud method and the point collocation method based on classical reproducing kernel [26] for the beam example; (a) convergence in v; (b) convergence in the derivatives of v.

6.6. *Thermoelasticity*

In thermoelasticity, the heat conduction equation is coupled to the elasticity equations through the inclusion of the temperature in the stress definition. The governing equations for thermoelasticity (for a plane stress situation) are

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f \tag{121}$$

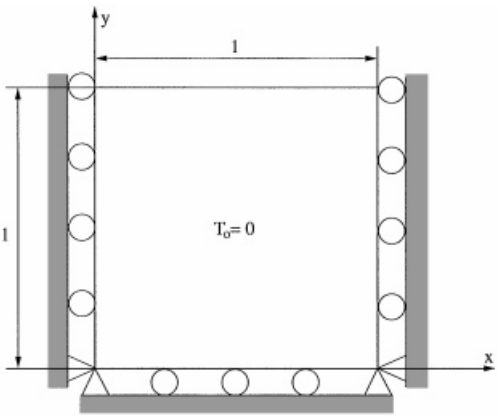

Figure 23. Thermoelasticity example.

$$\frac{2}{1-\nu}\frac{\partial^2 u}{\partial x^2} + \frac{1+\nu}{1-\nu}\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} - 2\alpha\frac{1+\nu}{1-\nu}\frac{\partial(T-T_0)}{\partial x} = 0 \tag{122}$$

$$\frac{2}{1-\nu}\frac{\partial^2 v}{\partial y^2} + \frac{1+\nu}{1-\nu}\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2} - 2\alpha\frac{1+\nu}{1-\nu}\frac{\partial(T-T_0)}{\partial y} = 0 \tag{123}$$

In the above equations, T is the temperature, T_0 is the reference temperature, f is the heat source, u and v are the x and y components of the displacement, respectively, and α is the coefficient of thermal expansion.

We solve the above equations using the finite cloud method for the example problem shown in Figure 23. In the solution of the thermoelasticity equations for the example in Figure 23, we use $T_0 = 0,\ f = 20,\ E = 3.0 \times 10^7,\ \nu = 0.25,\ \alpha = 0.001$ and the following boundary conditions:

$$\begin{aligned} T_{,x}(x=0) &= 0, \quad T_{,x}(x=1) = 0 \\ T(y=0) &= 0, \quad T(y=1) = 10 \\ u(x=0) &= 0, \quad u(x=1) = 0 \\ v(y=0) &= 0, \quad \sigma_{xy}(x=0) = 0 \\ \sigma_{xy}(x=1) &= 0, \quad \sigma_{xy}(y=0) = 0 \\ \sigma_{xy}(y=1) &= 0, \quad \sigma_{yy}(y=1) = 0 \end{aligned} \tag{124}$$

The exact solution for the above example problem is

$$T = 10y^2 \tag{125}$$

$$u = 0 \tag{126}$$

$$v = \tfrac{10}{3}(1+\nu)\alpha y^3 \tag{127}$$

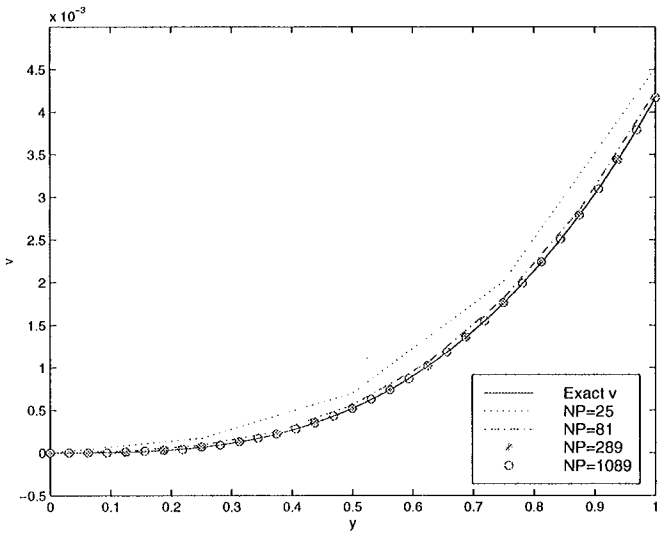

Figure 24. Comparison of the exact and computed displacements for the thermoelasticity example.

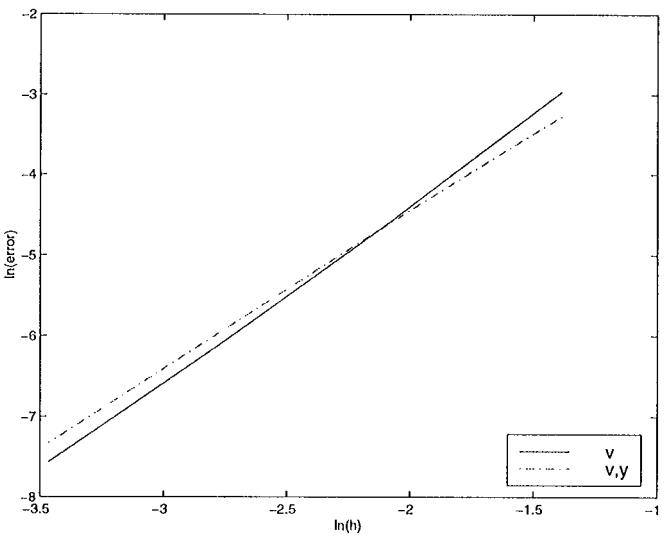

Figure 25. Convergence of the finite cloud method for the thermoelasticity example.

A comparison of the exact and the computed displacement in the y-direction is shown in Figure 24, and the convergence of the finite cloud method for the thermoelasticity example is shown in Figure 25. The convergence rate in the displacement is found to be 2.2 and convergence rate in the derivative of the displacement is found to be 1.95.

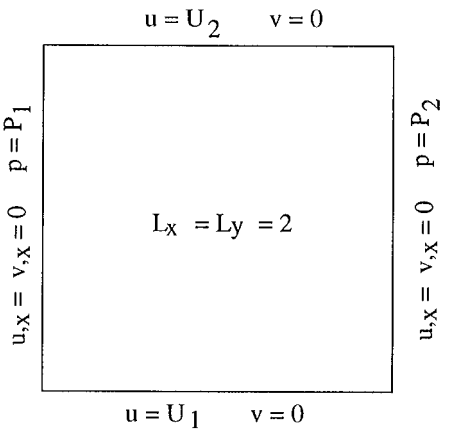

Figure 26. Fluid flow between parallel plates. The boundary conditions applied along the four edges are shown in the figure. The fluid viscosity is assumed to be $\mu = 13.5$.

6.7. *Stokes flow*

In this section we show results on the application of the finite cloud method for Stokes flow problems. The governing equations for Stokes flow are

$$\mu \nabla^2 \mathbf{u} - \nabla p = 0 \tag{128}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{129}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure and μ is the viscosity of the fluid.

The first example we consider is a simple situation of Couette flow between parallel plates where the pressure is prescribed at the inlet and outlet boundaries and the walls are moving with a horizontal velocity. This example has an analytical solution given by

$$u = U_1 + \frac{y}{h}(U_2 - U_1) - \frac{h^2}{2\mu}\frac{\mathrm{d}p}{\mathrm{d}x}\frac{y}{h}\left(1 - \frac{y}{h}\right) \tag{130}$$

$$v = 0 \tag{131}$$

where u is the x-component of the velocity, v is the y-component of the velocity, U_1 is the horizontal velocity of the bottom plate, U_2 is the horizontal velocity of the top plate and h is the separation between the top and bottom plates.

The specific example we consider here along with the boundary conditions is shown in Figure 26. Assuming the velocity of the bottom plate to be $U_1 = -0.5$, the velocity of the top plate to be $U_2 = 1.0$, the inlet pressure to be $P_1 = 54.0$ and the outlet pressure to be $P_2 = 0$, the horizontal velocity profile obtained with the finite cloud method is shown in Figure 27. The pressure drops linearly from the inlet to the outlet with no variation in the y-direction and the y-component of the velocity vanishes. Since the exact solution for the x-component of the velocity varies quadratically, the finite cloud method with a quadratic basis is able to produce nodally exact results for all point distributions.

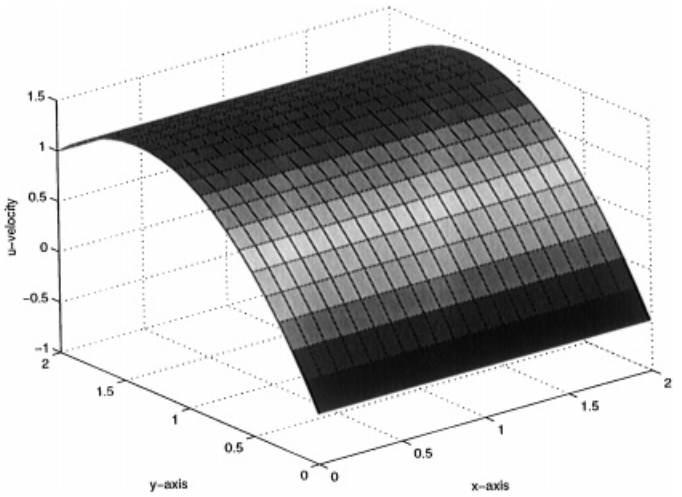

Figure 27. X-component of the velocity for the Couette flow example.

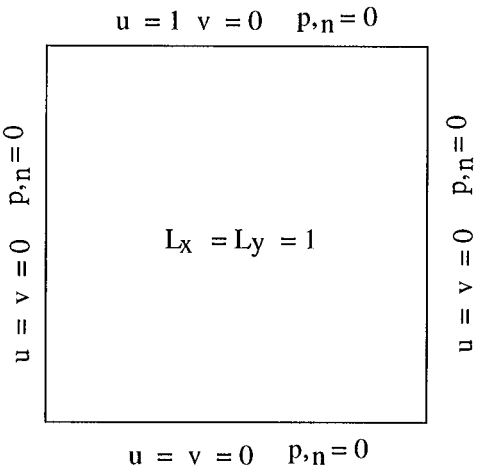

Figure 28. Geometry for the cavity driven flow example.

The second example we consider is the cavity-driven flow, the geometry of which is shown in Figure 28. We use a uniform distribution of 21×21 points. The x and y components of the velocity for this example are shown in Figure 29. The velocity vector and the pressure contour plot for the cavity-driven problem are shown in Figure 30.

6.8. Piezoelectricity

The physical behaviour of a piezoelectric material is described by coupled electrical and mechanical equations. The governing mechanical equilibrium equation is

$$\sigma_{ij,j} = 0 \tag{132}$$

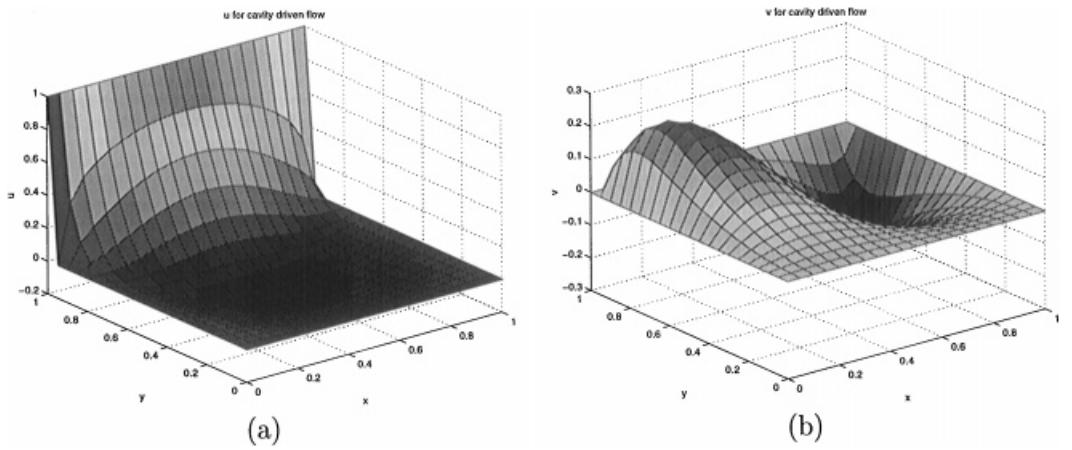

Figure 29. (a) X-component of the velocity for the cavity driven flow; (b) Y-component of the velocity for the cavity driven flow.

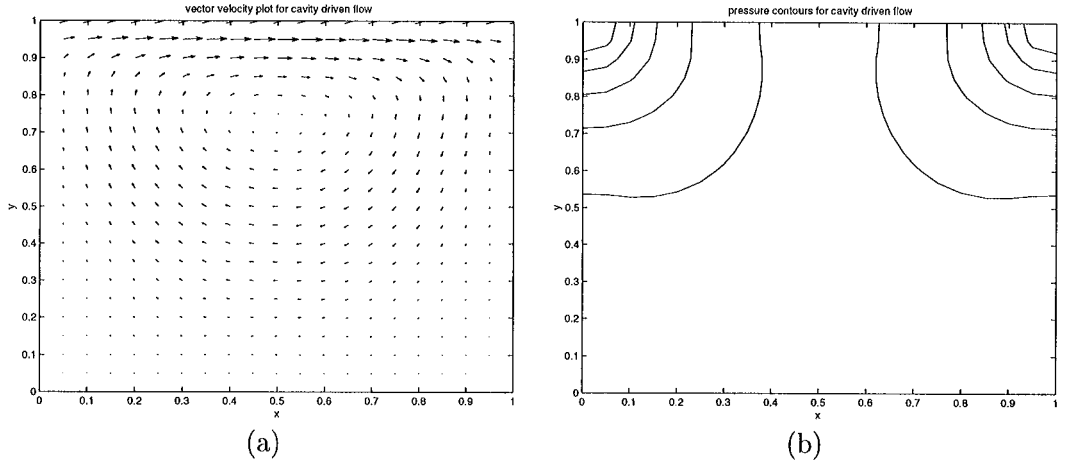

Figure 30. (a) Velocity vector; and (b) Pressure contour plot for the cavity-driven flow.

and the governing electrostatic equation is

$$D_{i,i} = 0 \tag{133}$$

The constitutive equations for a piezoelectric material are expressed in terms of the strains and the electric field [35]:

$$\sigma_p = c^{\mathrm{E}}_{pq}\varepsilon_q - e_{kp}E_k \tag{134}$$

$$D_i = e_{iq}\varepsilon_q + \zeta^{\varepsilon}_{ik}E_k \tag{135}$$

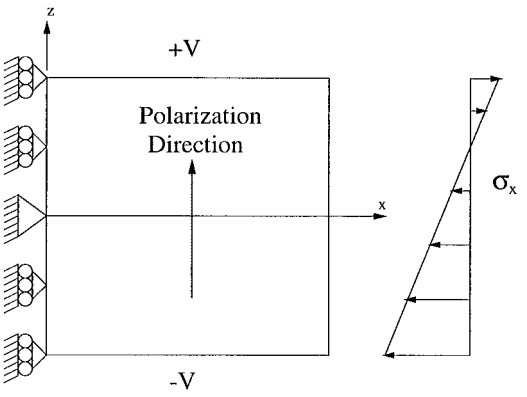

Figure 31. Piezo-strip subjected to a linearly varying stress and an applied voltage.

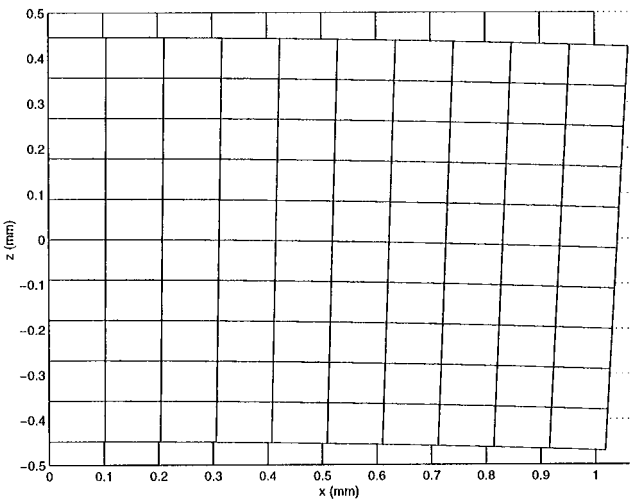

Figure 32. Mechanical deformation of the piezostrip with a 11×11 uniform distribution of points.

In the above equations, σ_p, ε_q, D_i and E_k are, respectively, the stress tensor, the strain tensor, the electric displacement vector and the electric field. c^E, e, and ζ^ε are the elastic stiffness, piezoelectric and dielectric constants, respectively. Superscript E and ε represent coefficients measured at constant electric field and strain, respectively. The strains are related to the displacements by

$$\varepsilon_{ij} = \tfrac{1}{2}(u_{i,j} + u_{j,i}) \tag{136}$$

and the electric field is related to the electric potential by

$$E_i = -\phi_{,i} \tag{137}$$

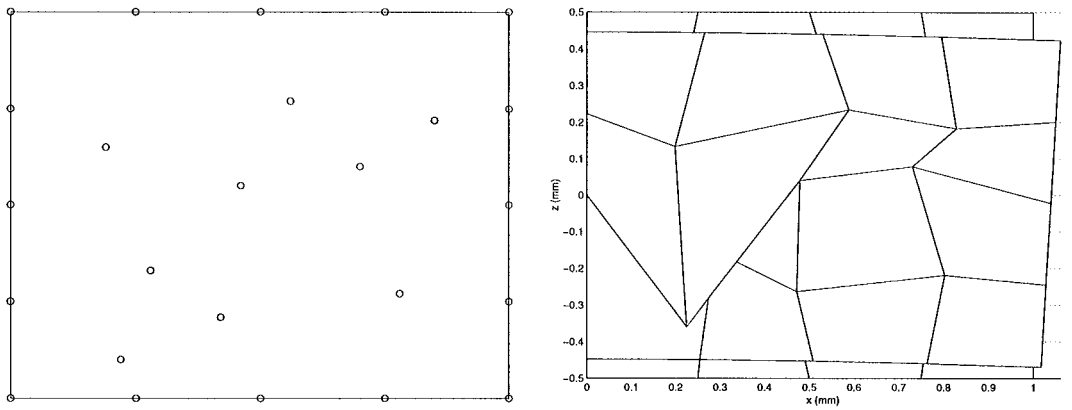

Figure 33. (a) Random distribution of points for the example presented in Figure 31; (b) Mechanical deformation for the random distribution of points.

The finite cloud method has been employed to solve a 1 mm $\times$ 1 mm PZT-5 strip subjected to an applied voltage and a linearly varying stress. The geometry of the problem is shown in Figure 31. The material properties for PZT-5 are summarized in Reference [36]. The example in Figure 31 has an analytical solution which is given in Reference [37]. The analytical solution for the displacement and the potential varies quadratically. With a quadratic basis in the finite cloud method, we were able to get nodally exact results for this problem. The mechanical deformation of the piezo-strip obtained with a 11×11 uniform distribution of points is shown in Figure 32. The grid in the background of Figure 32 shows the original shape of the strip before the potential and the axial loading is applied. The same example was also solved using a 5×5 distribution of random points. A plot of the random point distribution is shown in Figure 33(a) and the computed mechanical deformation is shown in Figure 33(b). For both random and uniform point distributions, nodally exact results were obtained for both mechanical and electrical quantities.

7. CONCLUSION

In this paper we have introduced fixed, moving and multiple fixed kernel techniques to construct interpolation functions. These approaches require only a scattered set of nodes and no connectivity information among the nodes is assumed or required. When the nodal volumes required in reproducing kernel techniques are set to unity we have shown that fixed, moving and multiple fixed kernel techniques are identical to the fixed, moving and multiple fixed least squares approaches, respectively.

We have also presented a true meshless technique, called the finite cloud method, which combines collocation with the fixed kernel technique. The other two approaches, the moving kernel and the multiple fixed kernel approaches, can also be combined with collocation to obtain true meshless techniques. With the finite cloud method we have observed that the convergence rates with exact and unit nodal volumes are almost identical. This suggests that

for complicated geometries and with random distribution of points, where the assignment of nodal volumes is complicated, the simplest choice would be to set the nodal volumes to unity. Compared to our earlier method, the point collocation method based on reproducing kernel approximations [26], we found that the finite cloud method, which uses fixed kernel approximation, is more efficient and accurate. The results with the finite cloud method also indicate that for some problems the error in the derivative is smaller than the error in the solution.

Numerical results were presented for several one- and two-dimensional problems including examples from heat conduction, elasticity, thermoelasticity, Stokes flow and piezoelectricity. The results indicate that finite cloud method is a promising technique for meshless analysis of partial differential equations encountered in several engineering applications.

ACKNOWLEDGEMENTS

This work was supported by an NSF CAREER award to N. R. Aluru and by a grant from DARPA under agreement number F30602-98-2-0178. This support is gratefully acknowledged.

REFERENCES

1. Belytschko T, Lu YY, Gu L. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
2. Liszka T, Orkisz J. Finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures* 1980; **11**:83–95.
3. Lucy LB. A numerical approach to the testing of the fission hypothesis. *The Astronomy Journal* 1977; **8**(12):1013–1024.
4. Monaghan JJ. Why particle methods work. *SIAM Journal on Scientific and Statistical Computing* 1982; **3**(4): 422–433.
5. Monaghan JJ. An introduction to SPH. *Computer Physics Communications* 1988; **48**:89–96.
6. Libersky LD, Petschek AG. Smooth particle hydrodynamics with strength of materials. In: *Advances in the Free-Lagrange Method*, Trease HE, Fritts MJ, Crowley WP. (eds), Lecture Notes in Physics, Springer: Berlin, 1990; 248–257.
7. Liu WK, Chen Y. Wavelet and multiple scale reproducing kernel methods. *International Journal for Numerical Methods in Engineering* 1995; **21**:901–931.
8. Liu WK, Chen Y, Chang CT, Belytschko T. Advances in multiple scale kernel particle methods. *Computational Mechanics* 1996; **18**:73–111.
9. Liu WK, Chen Y, Jun S, Chen JS, Belytschko T, Pan C, Uras RA, Chang CT. Overview and applications of the reproducing kernel particle methods. *Archives of Computational Methods in Engineering*: *State of the Art Reviews* 1996; **3**:3–80.
10. Liu WK, Jun S, Li S, Adee J, Belytschko T. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering* 1995; **38**:1655–1679.
11. Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
12. Chen J-S, Pan C, Wu C, Liu WK. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:195–227.
13. Chen J-S, Pan C, Wu C-T. Large deformation analysis of rubber based on a reproducing kernel particle method. *Computational Mechanics* 1997; **19**:211–227.
14. Chen J-S, Pan C, Roque CMOL, Wang H-P. A Lagrangian reproducing kernel particle method for metal forming analysis. *Computational Mechanics* 1998; **22**:289–307.
15. Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
16. Lu YY, Belytschko T, Gu L. A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1994; **113**:397–414.
17. Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL. A finite point method in computational mechanics. Applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering* 1996; **39**:3839–3866.

18. Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL, Sacco C. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:315–346.
19. Oñate E, Idelsohn S. A mesh-free finite point method for advective–diffusive transport and fluid flow problems. *Computational Mechanics* 1998; **21**:283–292.
20. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
21. Mukherjee YX, Mukherjee S. The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering* 1997; **40**:797–815.
22. Duarte CA, Oden JT. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:237–262.
23. Melenk JM, Babuska I. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:289–314.
24. Zhu T, Zhang J-D, Atluri SN. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. *Computational Mechanics* 1998; **21**:223–235.
25. Zhu T, Zhang J, Atluri SN. A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Computational Mechanics* 1998; **22**:174–186.
26. Aluru NR. A point collocation method based on reproducing kernel approximations. *International Journal for Numerical Methods in Engineering* 2000; **47**:1083–1121.
27. Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:49–74.
28. Aluru NR. A reproducing kernel particle method for meshless analysis of microelectromechanical systems. *Computational Mechanics* 1999; **23**:324–338.
29. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.
30. Liu WK, Li S, Belytschko T. Moving least square reproducing kernel method Part I: methodology and convergence. *Computer Methods in Applied Mechanics and Engineering* 1997; **143**:113–154.
31. Krongauz Y, Belytschko T. A Petrov–Galerkin diffuse element method (PG DEM) and its comprison to EFG. *Computational Mechanics* 1997; **19**:327–333.
32. Russell RD, Shampine LF. A collocation method for boundary value problems. *Numerical Mathematics* 1972; **19**:1–28.
33. Mukherjee YX, Mukherjee S. On boundary conditions in the element-free Galerkin method. *Computational Mechanics* 1997; **19**:264–270.
34. Timoshenko SP, Goodier JN. *Theory of Elasticity* (3rd edn). McGraw-Hill: New York, 1970.
35. IEEE Standard on Piezoelectricity. ANSI/IEEE Std 176, 1987; 9–15.
36. Ohs R, Aluru N. Modeling of piezoelectric MEMS using the finite cloud method. *Technical Proceedings of the 2nd International Conference on Modeling and Simulation of Microsystems*, *MSM 99*, San Juan, Puerto Rico, April 19–21, 1999; 293–296.
37. Parton VZ. *et al. Applied Mechanics: Soviet Reviews*, *Electromagnetoelasticity*, vol. 2. 1989; 1–58.
38. Krongauz Y, Belytschko T. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering* 1996; **131**:133–145.

