



Towards an efficient meshless computational technique: the method of finite spheres

Suvranu De and Klaus-Jürgen Bathe

Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

Keywords Finite element method, Meshless method

Abstract Computational efficiency and reliability are clearly the most important requirements for the success of a meshless numerical technique. While the basic ideas of meshless techniques are simple and well understood, an effective meshless method is very difficult to develop. The efficiency depends on the proper choice of the interpolation scheme, numerical integration procedures and techniques of imposing the boundary conditions. These issues in the context of the method of finite spheres are discussed.

1. Introduction

Over the past decade several meshless techniques have been proposed including the smoothed particle hydrodynamics (SPH) method (Lucy, 1977), the diffuse element method (DEM) (Nayroles *et al.*, 1992), the element free Galerkin (EFG) method (Belytschko *et al.*, 1996), the reproducing kernel particle method (RKPM) (Liu *et al.*, 1993), the partition of unity finite element method (PUFEM) (Melenk and Babuška, 1996), the *hp*-clouds method (Duarte and Oden, 1996), the finite point method (Oñate *et al.*, 1996), the local boundary integral equation (LBIE) method (Zhu *et al.*, 1998), the meshless local Petrov-Galerkin (MLPG) method (Alturi and Zhu, 1998) and the method of finite spheres (De and Bathe, 2000).

The primary reason that there is significant interest in meshless computational procedures is that the well established and successful numerical techniques like the finite element/finite volume methods require a mesh. The automatic generation of a good quality mesh poses a significant problem in the analysis of practical engineering systems. Moreover, the simulation and analysis of certain types of problems (like dynamic crack propagation) require an expensive remeshing operation.

Meshless techniques overcome these difficulties associated with meshing by eliminating the mesh altogether. Interpolation is performed in terms of nodal points "sprinkled" on the analysis domain using functions having compact support. A weighted residual technique is used to generate the discrete set of equations corresponding to the governing partial differential equations. While the basic idea behind meshless techniques is rather straightforward, the different meshless techniques found in the literature differ in the implementation details; namely, the choice of the interpolation functions, the particular form of the weighted residual scheme employed, the procedure

of imposing the boundary conditions and the numerical integration rules adopted.

A review of the literature on meshless techniques reveals that the current trend is towards application of the new techniques to diverse problem areas in engineering. But, it is clear that for general applications none of these methods is computationally as efficient as the traditional finite element/finite volume techniques. The issue of efficiency has been given due consideration only recently. Breitkopf *et al.* (2000), for example, have proposed a way of computing the moving least squares interpolants and their derivatives using a consistency based approach (Breitkopf *et al.*, 2000). However, for any meshless technique to find eventually wide application, it must be reasonably efficient compared to the now classical finite element/finite volume techniques and it should, of course, be reliable. With these considerations in mind, we proposed a truly meshless technique – the method of finite spheres (De and Bathe, 2000).

In the method of finite spheres the discretization is performed using functions that are compactly supported on general n-dimensional spheres and the Galerkin weak form of the governing partial differential equations is integrated using specialized numerical integration rules. The primary reason for the selection of the spherical support is that the relative orientation and the region of overlap of two spheres are completely determined by the coordinates of their centers and their radii.

In the traditional finite element/finite volume methods, the interpolation functions are piece-wise continuous polynomials (or mapped polynomials) and satisfy the Kronecker delta property at the nodes. This ensures that Dirichlet boundary conditions are enforced rather easily. Moreover, numerical integration is performed most efficiently using Gauss-Legendre product rules on integration domains that are n-dimensional cubes or tetrahedra. The Gauss-Legendre quadrature rule ensures arbitrary polynomial accuracy and therefore the stiffness terms (for undistorted elements) are exactly integrated with minimal cost (Bathe, 1996).

In the method of finite spheres, however, the interpolation functions do not satisfy the Kronecker delta property at the nodes. Hence, efficient imposition of Dirichlet boundary conditions is an important issue. Moreover, the interpolation functions are rational (nonpolynomial) functions and the integration domains are spheres, spherical shells or general sectors. Hence effective numerical integration rules have to be developed and exact integration can hardly be achieved. It is interesting to note that some of the earlier proposed meshless techniques use compactly supported functions for interpolation, but require a background mesh for numerical integration purposes, and are therefore not truly meshless as is the method of finite spheres.

In this paper we discuss how computational efficiency may be achieved in the method of finite spheres by proper choice of the interpolation functions, techniques of imposing the Dirichlet boundary conditions and efficient numerical integration rules. The organization of this paper is as follows. In

section 2 we review the interpolation scheme used in the displacement-based method of finite spheres and compare it with the other popular interpolation schemes. In section 3 we address the important issue of numerical integration and introduce several different integration rules. In section 4 we briefly review the imposition of Dirichlet boundary conditions. In section 5 we estimate the computational cost in the method of finite spheres and compare the method in this respect with other meshless techniques currently available and also with the classical finite element technique.

2. Choice of interpolation scheme

In the method of finite spheres the interpolation functions are generated using a partition of unity paradigm (Melenk and Babuška, 1996) based on the Shepard partition of unity functions (Shepard, 1968). In this section we review some of the most popular interpolation schemes and show that the one used in the method of finite spheres offers a comparatively less expensive means of generating approximation spaces with a given order of consistency.

Let $\Omega \in R^d$ ($d = 1, 2$ or 3) be an open bounded domain and let S be its boundary (see Figure 1). Let a family of open spheres $\{B(\mathbf{x}_I, r_I); I = 1, 2, \dots, N\}$ form a covering for Ω , i.e. $\Omega \subset \bigcup_{I=1}^N B(\mathbf{x}_I, r_I)$, where \mathbf{x}_I and r_I refer to the center and radius of the sphere I respectively. We associate a “node” with the geometric center \mathbf{x}_I of each sphere. By $S(\mathbf{x}_I, r_I)$ we denote the surface of sphere I . The spheres may be entirely within the domain (interior spheres) or may have nonzero intercepts with the boundary (boundary spheres) (see Figure 1).

In order to obtain localized approximations we define a radial weighting function (or window function) $W_I(\mathbf{x})$ compactly supported on the sphere centered at node I such that:

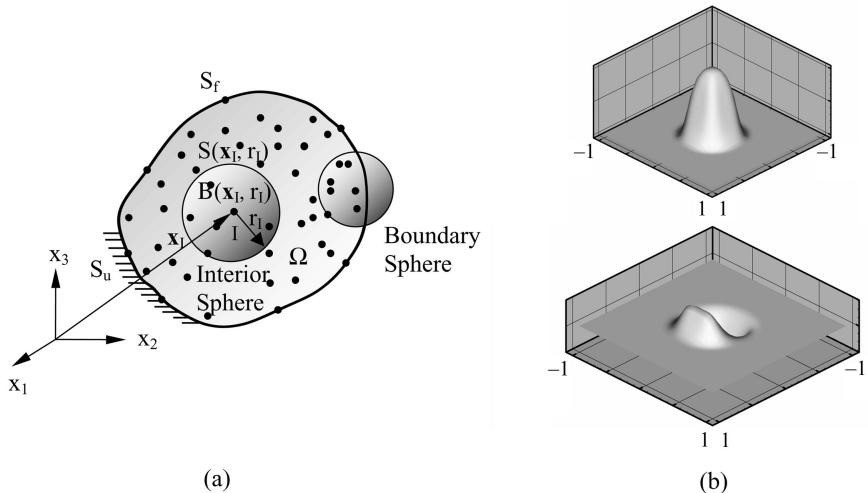


Figure 1.
(a) A schematic of the method of finite spheres and (b) some shape functions in two dimensions

- (1) $W_I(\mathbf{x}) \in C_0^s(B(\mathbf{x}_I, r_I)), s \geq 0;$
- (2) $\text{supp}(W_I) \subset B(\mathbf{x}_I, r_I);$ and
- (3) $W_I(\mathbf{x}) \geq 0 \forall \mathbf{x} \in \Omega.$

We concentrate on radial functions of the form $W_I(\mathbf{x}) = W(s_I)$, where $s_I = \frac{\|\mathbf{x} - \mathbf{x}_I\|_0}{r_I}$.

2.1 Moving least squares shape functions

The moving least squares (MLS) method of generating smooth approximation spaces was used by Nayroles *et al.* (1992) in the context of the diffuse element method and was later adopted by Belytschko *et al.* (1996) in their element free Galerkin method. The MLS method has also been used by Atluri and Zhu (1998) in their implementation of the meshless local Petrov-Galerkin method). The MLS and closely related techniques described in this section require an expensive matrix inversion/equation solution process at every evaluation point and therefore result in shape functions that are computationally very expensive.

In the moving least squares method, an element v_h of the global approximation space V_h (the subscript h is a measure of the size of the spheres) can be written as:

$$v_h(\mathbf{x}) = \sum_{I=1}^N h_I(\mathbf{x}) \alpha_I, \quad (1)$$

where $h_I(\mathbf{x})$ is the shape function associated with the nodal degree of freedom α_I . If $v_h(\mathbf{x}) \in \text{span}\{p_m(\mathbf{x})\}_{m=1}^n \forall \mathbf{x} \in \Omega$, where $p_m(\mathbf{x})$ is a polynomial or other function, then the MLS shape function corresponding to node I is expressed as:

$$h_I(\mathbf{x}) = W_I(\mathbf{x}) \mathbf{P}(\mathbf{x})^T \mathbf{A}^{-1}(\mathbf{x}) \mathbf{P}(\mathbf{x}_I), \quad (2)$$

where:

$$\mathbf{P}(\mathbf{x})^T = [p_1(\mathbf{x}) \ p_2(\mathbf{x}) \ \dots \ p_n(\mathbf{x})], \quad (3)$$

and:

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^N W_I(\mathbf{x}) \mathbf{P}(\mathbf{x}_I) \mathbf{P}(\mathbf{x}_I)^T. \quad (4)$$

The derivative of $h_I(\mathbf{x})$ with respect to a spatial variable x_i ($i = 1, 2$ or 3) is:

$$\begin{aligned} \frac{\partial h_I(\mathbf{x})}{\partial x_i} &= W_I(\mathbf{x}) \frac{\partial \mathbf{P}(\mathbf{x})^T}{\partial x_i} \mathbf{A}^{-1}(\mathbf{x}) \mathbf{P}(\mathbf{x}_I) \\ &\quad + \mathbf{P}(\mathbf{x})^T \mathbf{A}^{-1}(\mathbf{x}) \left(\frac{\partial W_I(\mathbf{x})}{\partial x_i} - W_I(\mathbf{x}) \frac{\partial \mathbf{A}(\mathbf{x})}{\partial x_i} \mathbf{A}^{-1}(\mathbf{x}) \right) \mathbf{P}(\mathbf{x}_I). \end{aligned} \quad (5)$$

From equation (2) we observe that $\text{supp}(h_I) = \text{supp}(W_I)$. We make the following remarks (for details, see Belytschko *et al.*, 1996).

Remark 2.1. Reproducing property: $p_m(\mathbf{x}) = \sum_{I=1}^N h_I(\mathbf{x}) p_m(\mathbf{x}_I)$ for $m = 1, \dots, n$.

This directly follows from equations (2) and (4). Hence to obtain linear consistency in R^2 for example, it is sufficient to choose $\mathbf{P}(\mathbf{x})^T = [1, x, y]$.

An interesting case arises when $\mathbf{P}(\mathbf{x}) = \{1\}$ and zeroth order consistency is assured by the shape functions in equation (2) which assume the particularly simple form:

$$h_I(\mathbf{x}) = \frac{W_I(\mathbf{x})}{\sum_{J=1}^N W_J(\mathbf{x})}. \quad (6)$$

These functions are also known as the Shepard partition of unity functions (also described in section 2.2).

Furthermore, if $1 \in \mathbf{P}(\mathbf{x})$ then the shape functions form a partition of unity subordinate to the cover $\{B(\mathbf{x}_I, r_I)\}_{I=1}^N$, i.e.:

$$\sum_{I=1}^N h_I(\mathbf{x}) = 1. \quad (7)$$

Remark 2.2. Continuity: let $W_I \in C_0^s(B(\mathbf{x}_I, r_I))$, $I = 1, 2, \dots, N$ and let $p_i(\mathbf{x}) \in C^l(\Omega)$, $i = 1, \dots, m$ for $s, l \geq 0$; then the shape functions $h_I(\mathbf{x})$ in equation (2) satisfy $h_I(\mathbf{x}) \in C_0^{\min(s,l)}(B(\mathbf{x}_I, r_I) \cap \Omega)$.

This property may be used to choose the functional form of W .

Remark 2.3. Invertibility of $\mathbf{A}(\mathbf{x})$: a necessary condition for the matrix $\mathbf{A}(\mathbf{x}) \in R^{n \times n}$ to be invertible at every $\mathbf{x} \in \Omega$ is:

$$\mathbf{x} \in \bigcap_{J \in \mathcal{Q}} B(\mathbf{x}_J, r_J), \quad (8)$$

where \mathcal{Q} is an index set with $\text{card}\{\mathcal{Q}\} \geq n$.

This directly follows from equation (4), which represents $\mathbf{A}(\mathbf{x})$ as the sum of rank one dyadic products. In practice the matrix $\mathbf{A}(\mathbf{x})$ should be invertible at the integration points and therefore at every integration station at least n spheres (to be consistent in the treatment of the various meshless schemes, we assume that the support of the weighting function, W_I , is a sphere throughout our discussion) should have nonzero support. To obtain second order consistency, for example, six spheres in R^2 and ten spheres in R^3 should have nonzero support at every integration point. This is quite a formidable requirement and of course it is not sufficient to ensure invertibility.

If $\mathbf{P}(\mathbf{x}) = \{1\}$ then condition (8) is also a sufficient condition for invertibility. For $d > 1$ if linear consistency is desired then an additional condition is required, namely, no three nodes $J \in \mathcal{Q}$ should be colinear. If the

nodal arrangement is close to this pathological condition, then the matrix $\mathbf{A}(\mathbf{x})$ becomes ill-conditioned resulting in large approximation errors.

The fact that the $n \times n$ matrix $\mathbf{A}(\mathbf{x})$ has to be generated and then inverted at every integration point makes the MLS approach of generating shape functions very expensive.

Remark 2.4. The weighted least squares (WLS) interpolation scheme used in the finite point method (Oñate *et al.*, 1996) is related to the MLS technique. A multivalued global interpolation is obtained by requiring that:

$$\forall \mathbf{x} \in B(\mathbf{x}_I, r_I) \quad v_h(\mathbf{x}) = \sum_{J=1}^N h_J^I(\mathbf{x}) \alpha_J, \quad (9)$$

where $h_J^I(\mathbf{x})$ and α_J are the shape function and degree of freedom, respectively, at node J such that $\mathbf{x}_J \in B(\mathbf{x}_I, r_I)$ and:

$$h_J^I(\mathbf{x}) = W(\mathbf{x}_J - \mathbf{x}_I) \mathbf{P}(\mathbf{x})^T \mathbf{A}_I^{-1} \mathbf{P}(\mathbf{x}_I), \quad (10)$$

where:

$$\mathbf{P}(\mathbf{x})^T = [p_1(\mathbf{x}) \ p_2(\mathbf{x}) \dots p_n(\mathbf{x})],$$

as before and:

$$\mathbf{A}_I = \sum_{I=1}^N W(\mathbf{x}_J - \mathbf{x}_I) \mathbf{P}(\mathbf{x}_I) \mathbf{P}(\mathbf{x}_I)^T. \quad (11)$$

Note that a discontinuous global approximation is obtained. The generalized finite difference scheme of Liszka and Orkisz (1980) may be considered as a special case of the WLS technique where the weights are so chosen that their supports always include the same number of nodal points.

Remark 2.5. The so-called “kernel interpolation techniques” like the smoothed particle hydrodynamics (SPH) (Lucy, 1977) and the reproducing kernel particle methods (RKPM) (Liu *et al.*, 1993) generate nodal shape functions that are very similar to those obtained using the MLS technique. The SPH shape functions provide zeroth order consistency and hence if the functions $W_I(\mathbf{x})$ are used as the discrete kernels then the SPH shape functions are identical to the Shepard partition of unity functions (see equation (6)). The RKPM shape functions provide higher order consistency by applying a correction to the SPH kernel. Of course, if the same function $W_I(\mathbf{x})$ is used as the kernel function and the consistency requirement of order p is imposed, the resulting RKPM shape functions are identical to the MLS shape functions in equation (2) satisfying p^{th} order consistency (Belytschko *et al.*, 1996).

2.2 Method of finite spheres shape functions: the partition of unity paradigm
In this section we describe the procedure we have adopted in generating low cost shape functions in the method of finite spheres (De and Bathe, 2000) using

Efficient meshless computational technique

the partition of unity paradigm (Melenk and Babuška, 1996). We consider here a pure displacement-based discretization scheme. The displacement/pressure mixed discretization scheme is more expensive since both displacements and pressures are discretized but is required in incompressible (or nearly incompressible) analyses (De and Bathe, nd).

The first step is to define, at each node I , a basis function, $\varphi_I^0(\mathbf{x})$, satisfying:

- (1) $\sum_{I=1}^N \varphi_I^0(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \Omega;$
- (2) $\text{supp}(\varphi_I^0(\mathbf{x})) \subset B(\mathbf{x}_I, r_I);$ and
- (3) $\varphi_I^0(\mathbf{x}) \in C_0^s(\mathbb{R}^n), s \geq 0.$

This system of functions $\{\varphi_I^0(\mathbf{x})\}_{I=1}^N$ is said to form a “partition of unity” subordinate to the open cover $\{B(\mathbf{x}_I, r_I)\}$.

The functions $\{\varphi_I^0(\mathbf{x})\}$ satisfy zeroth order consistency, i.e. they ensure that rigid body modes are exactly represented. To attain higher order consistency, at each node I , a local approximation space $V_I^h = \text{span}_{m \in \mathcal{I}} \{p_m(\mathbf{x})\}$ is defined, where $p_m(\mathbf{x})$ is a polynomial or other function and \mathcal{I} is an index set. The superscript h is a measure of the size of the spheres.

The global approximation space V_h is generated by multiplying the partition of unity function at each node I with the functions from the local basis:

$$V_h = \sum_{I=1}^N \varphi_I^0 V_I^h.$$

Hence, any function $v_h \in V_h$ can now be written as:

$$v_h(\mathbf{x}) = \sum_{I=1}^N \sum_{m \in \mathcal{I}} h_{Im}(\mathbf{x}) \alpha_{Im}, \quad (12)$$

where:

$$h_{Im}(\mathbf{x}) = \varphi_I^0(\mathbf{x}) p_m(\mathbf{x}), \quad (13)$$

and h_{Im} is a basis/shape function associated with the m^{th} degree of freedom α_{Im} of node I .

It should be observed that no inversion of matrices is involved in the generation of the shape functions and the computation of the shape functions and their derivatives:

$$\frac{\partial h_{Im}}{\partial x_i} = \varphi_I^0(\mathbf{x}) \frac{\partial p_m(\mathbf{x})}{\partial x_i} + p_m(\mathbf{x}) \frac{\partial \varphi_I^0(\mathbf{x})}{\partial x_i}, \quad i = 1, 2 \text{ or } 3, \quad (14)$$

is rather straightforward.

Remark 2.6. Reproducing property: if a function $p_i(\mathbf{x})$ is included in the local basis of all the nodes, then it is possible to exactly reproduce $p_i(\mathbf{x})$ on the entire domain.

The proof directly follows from equations (12) and (13) and noting that $\sum_{I=1}^N \varphi_I^0(\mathbf{x}) = \mathbf{1} \quad \forall \mathbf{x} \in \Omega$.

In our work we use the Shepard partitions of unity functions (Shepard, 1968) defined by:

$$\varphi_I^0(\mathbf{x}) = \frac{W_I}{\sum_{J=1}^N W_J}. \quad (15)$$

Efficient meshless
computational
technique

Important consideration should be given to the choice of the functions $W_I(\mathbf{x})$ so that a low cost partition of unity is obtained. We have chosen a cubic spline weight function of the following form:

$$W(s_I) = \begin{cases} \frac{2}{3} - 4s_I^2 + 4s_I^3 & 0 \leq s_I < \frac{1}{2} \\ \frac{4}{3} - 4s_I + 4s_I^2 - \frac{4}{3}s_I^3 & \frac{1}{2} < s_I \leq 1 \\ 0 & s_I < 1 \end{cases} \quad (16)$$

The choice of the functional form of W_I is guided by the following remark.

Remark 2.7. Continuity: let $W_I \in C_0^s(B(\mathbf{x}_I, r_I)), I = 1, 2, \dots, N$ and let $p_m(\mathbf{x}) \in C^l(\Omega)$ for $s, l \geq 0$; then the shape functions $h_{Im}(\mathbf{x})$ satisfy $h_{Im}(\mathbf{x}) \in C_0^{\min(s, l)}(B(\mathbf{x}_I, r_I) \cap \Omega)$.

The proof is immediate from equation (13). It therefore follows that:

- (1) *Displacement continuity.* The displacement field is continuous so long as the functions W_I and $p_m(\mathbf{x})$ are continuous.
- (2) *Stress continuity.* The stress fields, obtained by differentiating the displacement field (12), are continuous on Ω if each of the functions W_I has zero slope at the center, \mathbf{x}_I , and on the surface, $S(\mathbf{x}_I, r_I)$ of the sphere on which it is defined, and provided the functions $p_m(\mathbf{x})$ and their derivatives are sufficiently smooth.

For sufficiently smooth functions $p_m(\mathbf{x})$, the stress fields are continuous provided the derivatives of W_I with respect to the spatial coordinates $x_i (i \in \{1, 2, 3\})$:

$$\frac{\partial W(s_I)}{\partial x_i} = \frac{x_i - x_{Ii}}{r_I^2} \left[\frac{1}{s_I} \frac{dW(s_I)}{ds_I} \right], \quad (17)$$

are continuous in $B(\mathbf{x}_I, r_I)$ and on $S(\mathbf{x}_I, r_I)$.

This derivative exists as $s_I \rightarrow 0$ if W_I has zero slope at the center of the sphere. Moreover, the derivative in equation (17) is continuous on $S(\mathbf{x}_I, r_I)$, i.e. as $s_I \rightarrow 1$ if W_I has zero slope on the surface $S(\mathbf{x}_I, r_I)$.

Equation (17) introduces two conditions on the first derivative of the function W_I if a continuous stress field is to be obtained. A third condition arises from the constraint that the function W_I vanishes on $S(\mathbf{x}_I, r_I)$, i.e. $W(s_I = 1) = 0$. To satisfy these three conditions, the function W_I needs to be at least a cubic in s_I . This justifies the choice of the cubic spline in equation (16).

Remark 2.8. It was pointed out by Duarte and Oden that the use of Shepard functions to generate the partitions of unity is probably the least expensive for a given level of accuracy (Duarte and Oden, 1996). However, the more general hp -clouds functions of Duarte and Oden use moving least squares functions $\varphi_I^p(\mathbf{x})$ satisfying p^{th} order consistency to provide the partition of unity, i.e.:

$$V_h = \sum_{I=1}^N \varphi_I^p V_I^h, \quad (18)$$

where:

$$\varphi_I^p(\mathbf{x}) = W_I(\mathbf{x}) \mathbf{P}(\mathbf{x})^T \mathbf{A}^{-1}(\mathbf{x}) \mathbf{P}(\mathbf{x}_I), \quad (19)$$

with $\mathbf{P}(\mathbf{x})$ containing complete polynomials of order p and $\mathbf{A}(\mathbf{x})$ defined in equation (4). It should be recognized that this procedure of generating the partition of unity functions is extremely expensive and we do not consider these functions in our discussion in section 5.

3. Choice of numerical integration schemes

In the finite element techniques, the functions to be integrated are usually polynomials (or mapped polynomials), the elements do not overlap and they can be mapped to n-dimensional cubes. Hence Gauss-Legendre product rules are used for numerical integration with relatively low computational cost and high accuracy.

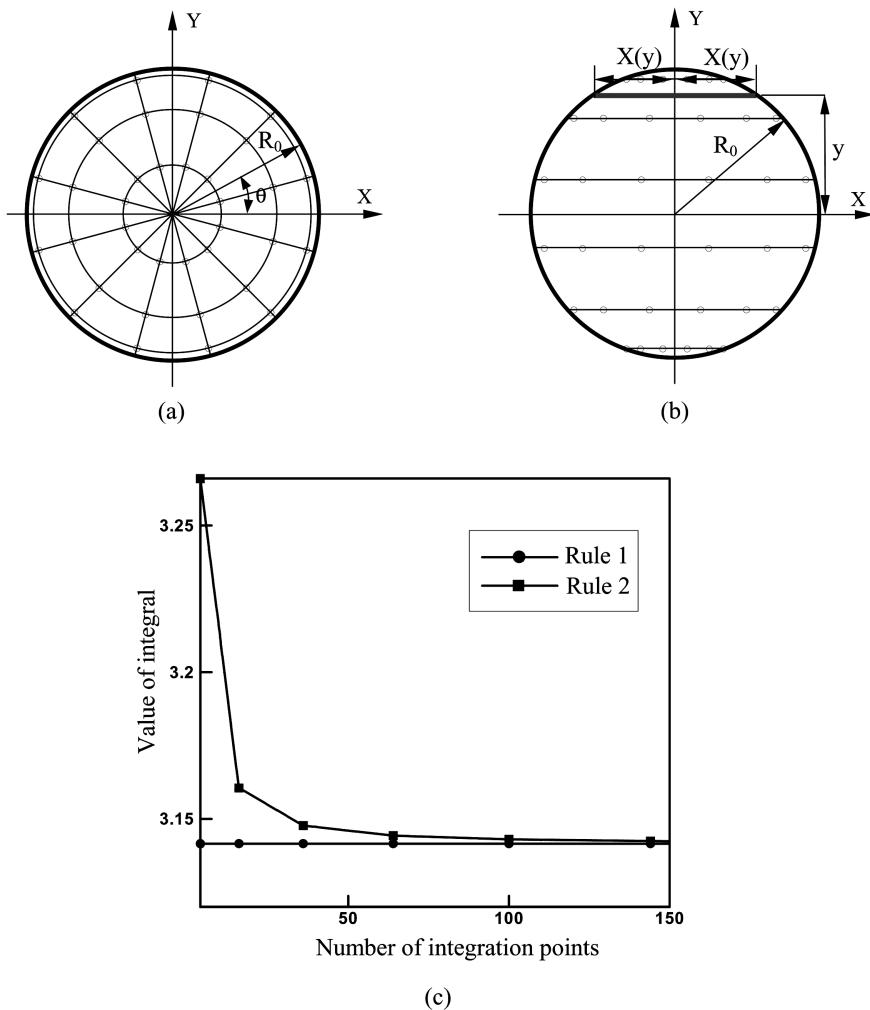
In the method of finite spheres, however, the shape functions are rational (non-polynomial) functions and the integration domains are spheres, spherical shells or general sectors. Moreover, the overlaps of spheres give rise to general lens shaped regions. Hence, a separate class of integration rules is required. Exact integration of the stiffness terms is not possible since the integration rules can, at best, guarantee polynomial accuracy. Hence the focus is on obtaining rules that use a minimal number of integration points for sufficient accuracy.

In this section we discuss the numerical integration rules for two-dimensional integration domains in the method of finite spheres. First we introduce efficient numerical integration rules on interior disks and boundary disks. We realize that integration on the lens shaped region of overlap of two disks is an important issue and propose and compare two different integration rules.

3.1 Integration on an interior disk

It is possible to develop integration rules having arbitrary polynomial accuracy on the unit disk (see Figure 2(a)). In this section we state such a product rule with polynomial accuracy k . The rule has the following form:

$$\iint_{\Omega} f(x, y) dx dy \doteq \sum_i \sum_j D_{ij} f(x_i, y_j). \quad (20)$$



Notes: In (a) integration points corresponding to the rule in theorem 3.1 (Rule 1) are shown. To integrate a polynomial of degree 11 exactly 36 integration points are required. In (b) integration stations corresponding to a Gauss-Legendre product rule (Rule 2) are shown. In (c) Rules 1 and 2 are used to evaluate the area of the disk

Figure 2.
Integration on an “interior” disk of radius 1.0

The region Ω under consideration is a disk with radius R_o (see Figure 2). The dot over the equality signifies that the relationship is a strict equality if the function $f(x,y)$ is a polynomial of order at most k in x and y , otherwise it is an approximation. The following theorem is due to Peirce (Peirce, 1957).

Theorem 3.1. If it is required that the rule in equation (20) has accuracy $k = 4m + 3$, $m = 0, 1, 2, \dots$, in $x = r \cos \theta$ and $y = r \sin \theta$, and if it is required to have a minimum number of evaluation points which are taken at the intersection of concentric arcs (radius r_j) with rays emanating from the origin

(angle θ_i), then it is both necessary and sufficient for the existence of a unique set of weights $D_{ij} \in R$ that:

$$\theta_{i+1} - \theta_i = \frac{2\pi}{k+1} \quad i = 1, 2, \dots, k, \quad (21)$$

and that the r_i^2 be the $m + 1$ zeros of $P_{m+1}(r^2)$, the Legendre polynomial in r^2 of degree $m + 1$, orthogonalized on $[0, R_o^2]$. The (unique) weights D_{ij} are of the form $A_i B_j$, where:

$$A_i = \frac{2\pi}{k+1} \quad i = 1, 2, \dots, k+1, \quad (22)$$

and:

$$B_j = \frac{1}{2P'_{m+1}(r_j^2)} \int_0^{R_o^2} \frac{P_{m+1}(r^2)}{r^2 - r_j^2} dr^2 \quad j = 1, 2, \dots, m+1. \quad (23)$$

The integration points are on equally spaced radii and the integration weights are independent of angular position (Gauss-Chebyshev rule in the θ -direction).

To demonstrate the efficiency of this rule we compare it with the Gauss-Legendre product rule on the disk:

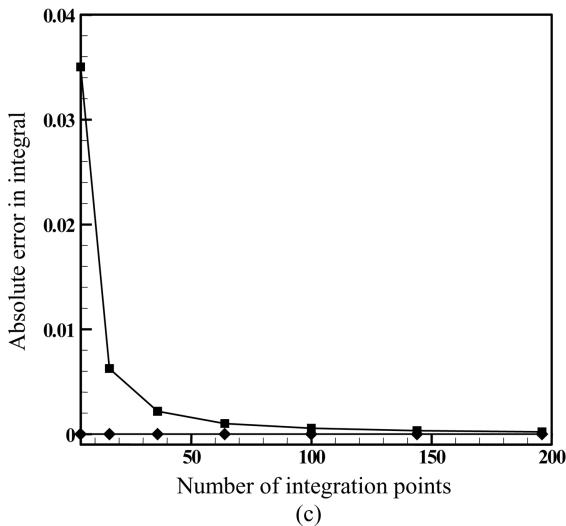
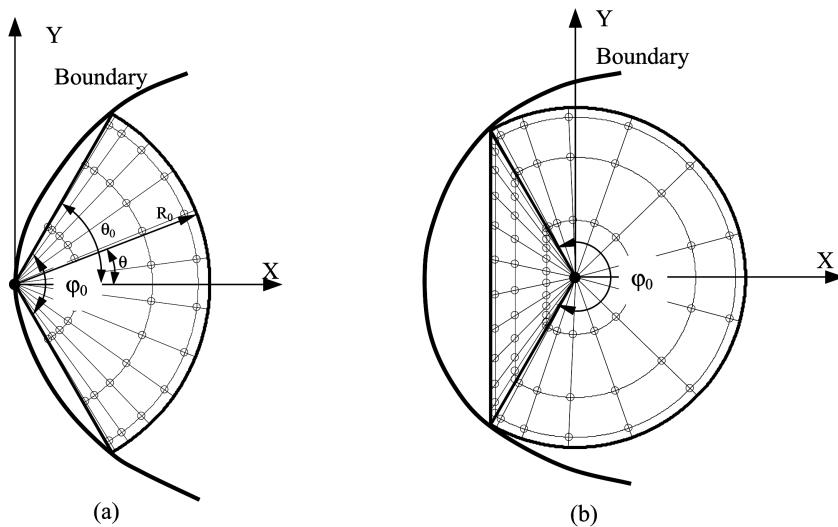
$$\iint_{\Omega} f(x, y) dx dy = \int_{y=-R_o}^{R_o} \int_{x=-X(y)}^{X(y)} f(x, y) dx dy \simeq \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} D_{ij} f(x_i, y_j), \quad (24)$$

where N_x and N_y are the number of integration points chosen along the x and y -directions, respectively, and $D_{ij} = W_i^x W_j^y$ is the product of the usual Gaussian weights W_i^x and W_j^y in the x and y -directions. It is not possible to guarantee exact integration of polynomials of any degree using this rule. For demonstration we consider the simple problem of computing the area of a unit circle (where $f(x, y) = 1$) in Figure 2(c).

3.2 Integration on a boundary sector

We categorize the boundary sectors into two major groups depending on the angle $\varphi_0 = 2\theta_0$ that the radii joining the center of the disk to the two intercepts of the disk on S make interior to the domain.

Type I sector: $\varphi_0 \leq \pi$ (see Figure 3(a)). The rule that allows us to perform numerical cubature on this sector to any desired order of accuracy is computationally expensive since it requires the evaluation of the zeros of a different orthogonal polynomial for every φ_0 (De and Bathe, 2000). We, therefore, propose the following “engineering solution”:



Notes: (a) Type I boundary sector with $\phi_0 \leq \pi$. (b) Type II boundary sector with $\phi_0 > \pi$. (c) Absolute error as a function of number of integration points when $f(x,y) = 1$ (◆) and $f(x,y) = x$ (■) are integrated on a Type I sector of unit radius and $\phi_0 = 2\pi/3$ using the rule in section 3.2

Figure 3.
Integration points on a boundary sector

$$\iint_{\Omega} f(x,y) dx dy \simeq \sum_{i=1}^{4(m+1)} \sum_{j=1}^{m+1} D_{ij} f(r_i \cos \theta_j, r_i \sin \theta_j), \quad (25)$$

where $D_{ij} = A_i B_j$ with A_i being the usual Gaussian weights on an interval

$[-\theta_0, \theta_0]$ and B_j as in equation (23). The integration points r_j are the positive square roots of the zeros of the Legendre polynomial $P_{m+1}(r^2)$ in r^2 of degree $m+1$ ($k = 4m+3$), orthogonalized on $[0, R_o^2]$ as in theorem 3.1. However, the radial coordinates of the integration points θ_j are chosen as the zeros of the Legendre polynomial $P_{4(m+1)}(\theta)$ in θ of degree $4(m+1)$, orthogonalized on $[-\theta_0, \theta_0]$.

This rule guarantees arbitrary polynomial accuracy if the integrand in equation (25) is a purely radial function. Hence, as Figure 3(c) shows, the area of the sector ($f(x, y) = 1$) can be exactly integrated but if the integrand is $f(x, y) = x$, for example, exact integration is not possible. Notice, however, that this rule is better than a pure Gauss-Legendre product rule on the sector as in equation (24).

Type II sector: $\varphi_0 > \pi$ (see Figure 3(b)). This type of boundary sector is more expensive to handle. We decompose a type II sector into a sector for which the rules of the type I sector can be used and a triangle as shown in Figure 3(b). For the triangle we use a product rule based on Gauss-Legendre quadrature.

3.3 Integration on the lens

To be able to integrate efficiently on the lens shaped region of intersection of two disks, we need specialized integration rules. In this section we propose and compare two schemes for numerically evaluating:

$$\iint_{\Omega_{IJ}} f(x, y) dx dy,$$

where $\Omega_{IJ} = B(\mathbf{x}_I, r_I) \cap B(\mathbf{x}_J, r_J) \neq 0$.

Scheme 1. In this scheme (see Figure 4(a)) we use a Gauss-Legendre product rule of the form:

$$\iint_{\Omega_{IJ}} f(x, y) dx dy = \int_{y=-y_o}^{y_o} \int_{x=-X_1(y)}^{X_2(y)} f(x, y) dx dy \simeq \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} D_{ij} f(x_i, y_j), \quad (26)$$

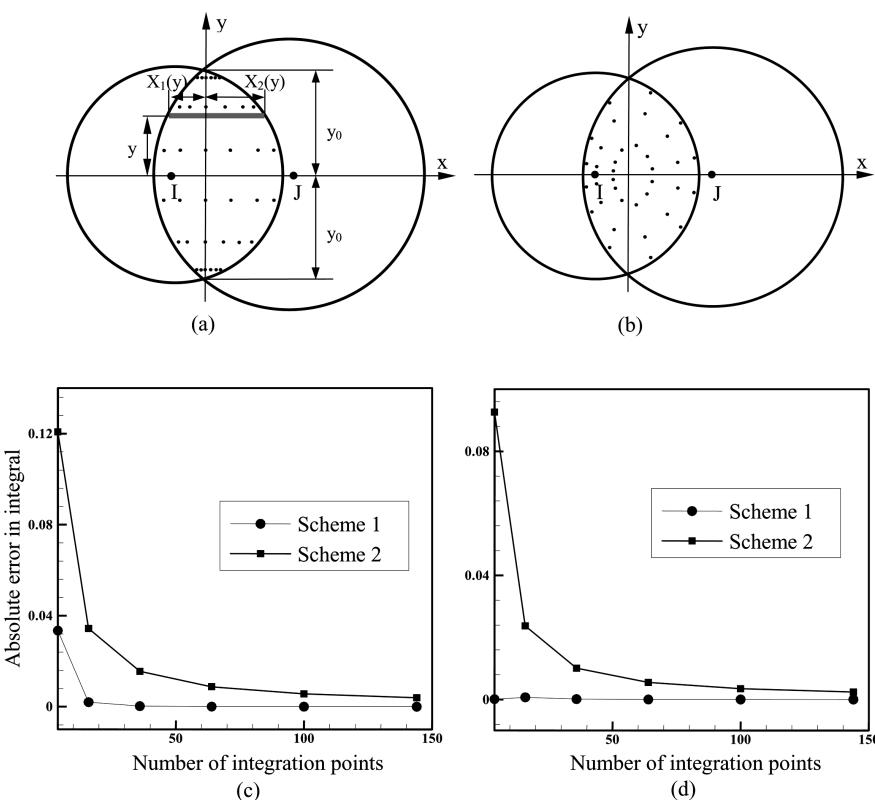
where N_x and N_y are the number of integration points chosen along the x and y -directions, respectively and $D_{ij} = W_i^x W_j^y$ is the product of the usual Gaussian weights W_i^x and W_j^y in the x and y -directions.

Scheme 2. In this scheme (see Figure 4(b)), we map the domain Ω_{IJ} onto a unit disk and compute the resulting integral using the scheme in theorem 3.1:

$$\iint_{\Omega_{IJ}} f(x, y) dx dy = \int_{\rho=0}^1 \int_{\theta=0}^{2\pi} F(\rho, \theta) J \rho d\rho d\theta, \quad (27)$$

where J is the Jacobian of the transformation.

We note that none of the schemes can guarantee polynomial accuracy for the integrand $f(x, y)$. This is because the integration limits in rule (26) are not



Notes: Some integration points generated using schemes 1, see (a), and 2, see (b), of section 3.3 are shown on the intersection of two disks of radii 0.8 and 1.0, respectively, with center-to-center distance of 0.9. Absolute errors as a function of number of integration points when $f(x,y) = 1$, see (c) and $f(x,y) = \sqrt{1-x^2-y^2}$, see (d), are integrated on the intersection region using schemes 1 and 2 are also shown

Figure 4.
Integration on the lens

simple. On the other hand, the Jacobian, in equation (27), is a complex function to integrate. In Figure 4(c) the absolute error in integrating the area of the intersection region is shown as a function of the number of integration points. Note that neither scheme can integrate $f(x,y) = 1$ exactly on the domain. In Figure 4(d) the absolute value of the error in integrating a more complex function $f(x,y) = \sqrt{1-x^2-y^2}$ is shown as a function of the number of integration points. Such numerical experiments show that scheme 1 requires fewer integration points for the same accuracy than scheme 2.

4. Imposition of essential boundary conditions

In a weighted residual technique, the governing differential equation is not satisfied point-wise in the interior of the domain. Point-wise satisfaction of the essential boundary conditions on a general Dirichlet boundary is similarly not

possible. Considering the finite element technique, the shape functions satisfy the Kronecker delta property at the nodes (i.e. the shape function at any node is unity at that node and is zero at all other nodes). Furthermore, only those nodes that lie on the Dirichlet boundary participate in the imposition of the Dirichlet boundary conditions.

Therefore, if homogeneous Dirichlet boundary conditions are prescribed, then these conditions can be exactly satisfied along element edges on the Dirichlet boundary. When a nonhomogeneous boundary condition is prescribed the finite element approximation along the element edges on the Dirichlet boundary $u^h(\mathbf{x})|_{S_u} = \sum h_I(\mathbf{x})|_{S_u} u_I$ (where $h_I(\mathbf{x})|_{S_u}$ is the trace of the finite element shape function $h_I(\mathbf{x})$ on the Dirichlet boundary S_u and u_I is the prescribed boundary condition at node I) converges to the applied boundary condition in a weak sense. For example, if the finite elements exhibit polynomial consistency of order p then $\|u - u^h\|_0 \leq Ch^{p+1}$ (h denotes the element size and C is a constant depending on the problem considered but is independent of h).

In the method of finite spheres the shape functions do not satisfy the Kronecker delta property at the nodes. This is also true for the MLS (and related) shape functions. Moreover, nodes not lying on the Dirichlet boundary but with nonzero intercepts of their spheres with the Dirichlet boundary are also involved in enforcing the boundary conditions. Indeed, to retain the flexibility of “sprinkling” the nodes relatively arbitrarily on the domain, we should be able to satisfy the Dirichlet conditions (in some sense) without even a single node directly on the Dirichlet boundary (so long as the spheres cover the domain). Therefore, rather than trying to satisfy the Dirichlet boundary conditions point-wise at the nodes it is more important to be able to enforce them in a weak sense along the boundary. We have therefore not considered in our work the so-called collocation techniques (Zhu and Atluri, 1998; Wagner and Liu, nd).

Some of the other procedures of imposing Dirichlet boundary conditions that have been employed in the context of meshless methods are techniques involving Lagrange multipliers, penalty formulations, use of finite elements along Dirichlet boundaries and modified variational principles (Belytschko *et al.*, 1996). The use of Lagrange multipliers results in indefinite systems of equations and increases the number of unknowns considerably. Penalty formulations can result in ill-conditioned matrices. The use of finite elements along the Dirichlet boundaries destroys the meshless character of the approximation.

A technique of much potential is the use of modified variational principles. In the method of finite spheres we have used this technique for the imposition of boundary conditions (for details, refer to De and Bathe, 2000). This technique enforces the Dirichlet boundary conditions in a weak sense without increasing the number of unknowns. Furthermore, we have also shown that for a special arrangement of nodes on the Dirichlet boundary, it is possible to exactly satisfy the Dirichlet conditions at the nodes (De and Bathe, 2000).

5. Computational costs

In this section we estimate the computational cost in the displacement-based method of finite spheres and compare this cost with the expense of a similar meshless scheme using the moving least squares approximants as well as the classical finite element techniques. We adopt the \mathcal{O} -notation to imply the asymptotic upper bound to within a constant, i.e. for a given function $g(n)$, we denote by $\mathcal{O}(g(n))$ the set of functions:

$$\mathcal{O}(g(n)) = \{f(n) : \exists \text{ constants } c, n_0 > 0 \text{ such that } 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}.$$

For example, if $f(n) = a_o n + a_1$ with $a_o > 0$, then we may write $f(n) \in \mathcal{O}(n)$. A constant is represented as $\mathcal{O}(1)$.

We assume that the major computational cost may be broken down into:

- (1) cost of computation of the global stiffness matrix; and
- (2) cost of solution of the resulting set of algebraic equations.

This approach neglects the computation of the loads, the application of the boundary conditions, memory traffic and other overheads associated with the execution of a general numerical software and therefore gives only a rough estimate of the efficiency. Moreover, we do not consider the preprocessing or postprocessing time. However, a main advantage of the meshless techniques over the traditional finite element techniques is that preprocessing time is reduced as no mesh is required. Postprocessing in meshless techniques is also relatively straightforward since, for example, no additional stress smoothening is required.

We consider a general elliptic problem in d dimensions ($d = 1, 2$ or 3) and assume a discretization scheme using N nodal points and a consistency of order p . The superscripts MLS, FEM and MFS will be used in the following discussion to differentiate the same variable for different methods. We assume banded symmetric matrices with constant column height and a constant half bandwidth m_K . Each node is assumed to have an average connectivity of M , i.e. the support of each node is assumed to have nonzero overlaps with an average of $(M - 1)$ other nodal supports.

5.1 Cost of computation of the global stiffness matrix

The finite element method. In the finite element method, we assume that the stiffness matrix has “ dN^{FEM} ” rows and “ dM^{FEM} ” non zero columns per row. Hence, the computational time for the global stiffness matrix may be assessed as

$$T_K^{\text{FEM}} = \mathcal{O}(d^2 N^{\text{FEM}} M^{\text{FEM}} T_{K_{ij}}^{\text{FEM}}), \quad (28)$$

where $T_{K_{ij}}^{\text{FEM}}$ is the computational time for a single term of the stiffness matrix, which is assumed to be a volume integral of the sum of inner products of the derivatives of the shape functions. This integral is evaluated using numerical integration over each finite element. Let N_g^{FEM} denote the number of Gaussian

integration points per finite element ($N_g^{FEM} = (p + 1)^2$ for a tensor product element in R^2 using a complete polynomial of order p). We model $T_{K_{ij}}^{FEM}$ as:

$$T_{K_{ij}}^{FEM} = \mathcal{O}(N_g^{FEM} T_h^{FEM}), \quad (29)$$

where T_h^{FEM} is the average computational time for a finite element shape function (or its derivative). We neglect the fact that multiple derivatives need be taken and different elements connected to a nodal point contribute. Therefore:

$$T_K^{FEM} = \mathcal{O}(d^2 N_g^{FEM} N^{FEM} M^{FEM} T_h^{FEM}). \quad (30)$$

Meshless method using moving least squares interpolants. We consider a meshless method which uses a moving least squares type of interpolant. The stiffness matrix has “ dN^{MLS} ” rows and “ dM^{MLS} ” non zero columns per row. Hence, the computational time for the global stiffness matrix may be modeled as:

$$T_K^{MLS} = \mathcal{O}(d^2 N_g^{MLS} M^{MLS} T_{K_{ij}}^{MLS}), \quad (31)$$

where $T_{K_{ij}}^{MLS}$ is the computational time for a single term of the stiffness matrix. This integral is evaluated using numerical integration. Let N_g^{MLS} denote the number of integration points per sphere. We model $T_{K_{ij}}^{MLS}$ as:

$$T_{K_{ij}}^{MLS} = \mathcal{O}(N_g^{MLS} T_h^{MLS}), \quad (32)$$

where T_h^{MLS} is the average computational time for a MLS shape function (or its derivative). The MLS shape functions and their derivatives are very complex (see equations (2) and (5)) and involve the inversion of a $n \times n$ matrix at each integration point, where $n = (p + 1)(p + 2)/2$, if a consistency of order p is required in R^2 . We may model T_h^{MLS} as:

$$T_h^{MLS} = \mathcal{O}(n^2 M^{MLS} T_W), \quad (33)$$

where T_W is the computational cost of evaluating the weighting function in equation (16) or its derivative at a single evaluation point. This is justified since matrix inversion and matrix-matrix multiplies are $\mathcal{O}(n^3)$ operations and $M^{MLS} = \mathcal{O}(n)$ at least to ensure invertibility of the $\mathbf{A}(\mathbf{x})$ matrix in equation (4). Therefore:

$$T_K^{MLS} = \mathcal{O}(d^2 n^2 N_g^{MLS} N^{MLS} (M^{MLS})^2 T_W). \quad (34)$$

Method of finite spheres. In the method of finite spheres with n functions in the local basis enforcing p^{th} order consistency, the global stiffness matrix has “ dnN^{MFS} ” rows and “ dnM^{MFS} ” non zero columns per row. Hence, the computational time for the global stiffness matrix may be modeled as:

$$T_K^{MFS} = \mathcal{O}(d^2 n^2 N^{MFS} M^{MFS} T_{K_{ij}}^{MFS}), \quad (35)$$

where $T_{K_{ij}}^{MFS}$ is the computational time for a single term of the stiffness matrix. Let N_g^{MFS} denote the number of integration points per sphere. We model $T_{K_{ij}}^{MFS}$ as:

$$T_{K_{ij}}^{MFS} = \mathcal{O}(N_g^{MFS} T_h^{MFS}), \quad (36)$$

where T_h^{MFS} is the average computational time for a MFS shape function (or its derivative). The MFS shape functions and their derivatives are much simpler to compute than the MLS shape functions (see equations (13) and (14)) and do not involve matrix inversions. The most expensive operation is the computation of the Shepard partition of unity functions (and their derivatives) which requires the evaluation of M^{MFS} weighting functions at each integration point. Therefore we may model T_h^{MFS} as:

$$T_h^{MFS} = \mathcal{O}(M^{MFS} T_W), \quad (37)$$

where T_W is the computational cost of evaluating the weighting function in equation (16) or its derivative at a single evaluation point. Therefore:

$$T_K^{MFS} = \mathcal{O}(d^2 n^2 N_g^{MFS} N^{MFS} (M^{MFS})^2 T_W). \quad (38)$$

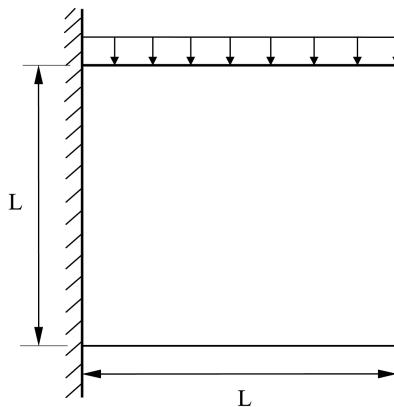
Comparisons. It is interesting to observe that estimates (34) and (38) have the same form. It may appear that the cost advantage in computing the simpler stiffness terms is lost in the number of terms that have to be computed in the MFS. This is true if the same number of nodes, connectivity and, of course number of integration points, are used in both techniques. But due to the condition mentioned in remark 2.3 regarding the invertibility of the $\mathbf{A}(\mathbf{x})$ matrix, $M^{MLS} = \mathcal{O}(n)$, while essentially $M^{MFS} = \mathcal{O}(1)$. Indeed, it has been reported that M^{MLS} for the MLS functions used in the element free Galerkin method can be as high as 50 in R^2 (Krysl and Belytschko, 1995). In the method of finite spheres, however, M^{MFS} can be four. Furthermore, the invertibility of the $\mathbf{A}(\mathbf{x})$ matrix also necessitates that $N^{MLS} \gg N^{MFS}$ for comparable accuracy (for example, in the method of finite spheres, the problem in Figure 5(a) may be solved with quadratic consistency using only four nodes at the four corners). Therefore:

$$\frac{T_K^{MLS}}{T_K^{MFS}} \gg 1.0. \quad (39)$$

Next, we compare the computation time estimates (30) and (38):

$$\frac{T_K^{MFS}}{T_K^{FEM}} = O\left(\frac{n^2 N_g^{MFS} N^{MFS} (M^{MFS})^2 T_W}{N_g^{FEM} N^{FEM} M^{FEM} T_h^{FEM}}\right).$$

It is true that $T_K^{MFS}/T_K^{FEM} > 1.0$ but the ratio is not very large since $N^{MFS} \ll N^{FEM}$ for the same accuracy in solution. This is because in the



(a)

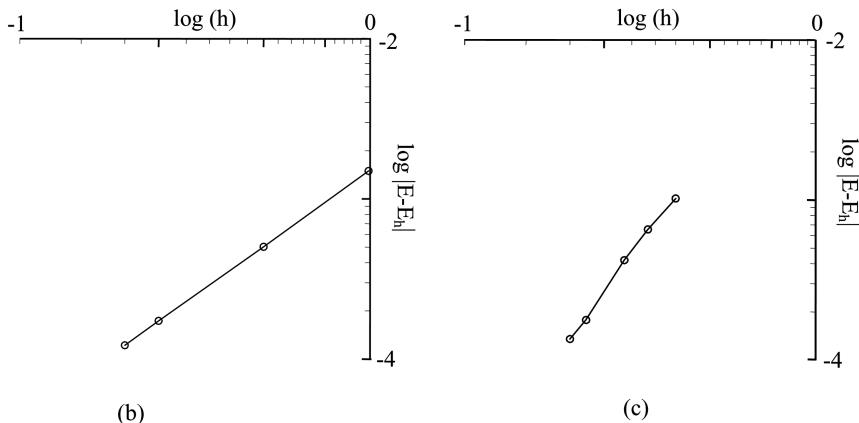


Figure 5.
Solving a problem using
nine-noded finite
elements and the
method of finite spheres

Notes: (a) A cantilever plate ($L = 2.0$) in plane strain with a uniformly distributed load of magnitude $w = 1.0$ per unit length on the top surface. Poisson's ratio, $\nu = 0.3$. The convergence in strain energy (E_h), computed using nine-noded finite elements (see (b)) and the method of finite spheres with second order consistency (see (c)), is shown when a uniform h-type refinement is performed. E is an accurate estimate of the strain energy (reference solution)

method of finite spheres, the observed convergence rate is much better than in the finite element method (De and Bathe, 2000).

Let us consider an example of the square cantilevered plate in plane strain shown in Figure 5(a) with uniformly distributed loading on the top surface. The convergence in strain energy corresponding to a uniform h-type refinement is presented in Figures 5(b) and (c) when the problem was solved using nine-noded finite elements and the method of finite spheres respectively. It is observed that a 8×8 regular nodal arrangement (with quadratic consistency) provides a solution (in strain energy) which is

comparable in accuracy with the solution provided by a 25×25 mesh of nine-noded finite elements. Hence, for comparable accuracy and uniform meshes, we require 5,202 degrees of freedom in the finite element technique and 768 (64 nodes \times 12 degrees of freedom per node) degrees of freedom in the method of finite spheres. We estimate:

$$\begin{aligned} N^{FEM} &= 2,601; N^{MFS} = 64; \\ N_g^{FEM} &= 9; N_g^{MFS} = 144; \\ M^{FEM} &\sim 25; M^{MFS} \sim 4; \\ n &= 6, \end{aligned} \tag{40}$$

Efficient meshless
computational
technique

189

and obtain:

$$\frac{T_K^{MFS}}{T_K^{FEM}} \sim 9, \tag{41}$$

assuming that T_h^{FEM} and T_W are of the same order of magnitude. This estimate is quite close to the one obtained by comparing the actual computational times (the observed ratio is around eight).

5.2 Cost of solution including solving the set of algebraic equations

Given the dimensions of the stiffness matrices considered in the previous section, the solution times for the three techniques may be assessed as:

(1) The finite element method:

$$T_s^{FEM} = \mathcal{O}(dN^{FEM}(m_K^{FEM})^2). \tag{42}$$

(2) Meshless method using moving least squares interpolants:

$$T_s^{MLS} = \mathcal{O}(dN^{MLS}(m_K^{MLS})^2). \tag{43}$$

(3) Method of finite spheres:

$$T_s^{MFS} = \mathcal{O}(dnN^{MFS}(m_K^{MFS})^2). \tag{44}$$

We are interested in the order of magnitude estimates and not any particular solution technique.

From our discussion above we realize that for a large problem, the half bandwidth $m_K^{MFS} \ll m_K^{MLS}$ and $N^{MFS} \ll N^{MLS}$. Therefore it is reasonable to estimate that $T_s^{MFS} \ll T_s^{MLS}$.

Comparing estimates (40) and (42) we observe that $N^{MFS} \ll N^{FEM}$ for comparable accuracy and $m_K^{MFS} \ll m_K^{FEM}$. For example, from the data in equation (40) we see that:

$$N^{MFS} = 64; N^{FEM} = 2,601;$$

$$m_K^{FEM} = 209; m_K^{MFS} = 119$$

and therefore:

$$\frac{T_s^{MFS}}{T_s^{FEM}} \sim 4.8 \times 10^{-2}. \quad (45)$$

Estimates (41) and (45) are important in comparing the total computational costs of the method of finite spheres (T_{total}^{MFS}) and the finite element technique (T_{total}^{FEM}). We assume:

$$\begin{aligned} T_{total}^{FEM} &= T_K^{FEM} + T_s^{FEM} \\ T_{total}^{MFS} &= T_K^{MFS} + T_s^{MFS}. \end{aligned} \quad (46)$$

We observe, by solving the problem in Figure 5(a) using a sequence of nine-noded finite elements using the commercial finite element software package ADINA, that for fine discretizations:

$$\frac{T_K^{FEM}}{T_s^{FEM}} \sim 1.0. \quad (47)$$

From equation (46) we have:

$$\frac{T_{total}^{MFS}}{T_{total}^{FEM}} = \frac{(T_K^{MFS}/T_K^{FEM})}{1 + T_s^{FEM}/T_K^{FEM}} + \frac{(T_s^{MFS}/T_s^{FEM})}{1 + T_K^{FEM}/T_s^{FEM}}. \quad (48)$$

From equation (45) it may be observed that the ratio T_s^{MFS}/T_s^{FEM} is very small compared to the denominator (which is of the order of two). From equation (41) we observe that the ratio T_K^{MFS}/T_K^{FEM} is of the order of ten. Therefore, it is reasonable to estimate that the method of finite spheres is about five (or say ten) times slower than the finite element technique for elastostatic problems in two dimensions.

6. Concluding remarks

Over the past several years an increasing interest in meshless computational techniques is observed. While current research efforts are primarily aimed at applying these techniques to the analysis of various complex engineering problems, not enough emphasis is placed on computational efficiency. However, for a meshless technique to be widely applicable in industry, computational efficiency is clearly a most important factor to be considered. We have developed the method of finite spheres with this issue in mind. In this paper we have discussed various aspects of the efficiency of the method.

We note that even though multiple shape functions are used at a node the method of finite spheres is more efficient than the techniques based on the

moving least squares interpolants since no matrix inversion is required at every integration point and there are no stringent overlap criteria. In the current form of implementation, when measured roughly theoretically and as seen in an example solution, the method of finite spheres is about five times slower than the traditional finite element techniques for representative problems in two-dimensions since it offers comparable accuracy in solution with considerably fewer nodes on the domain. This is quite encouraging since the preprocessing time is considerably less than in the finite element techniques. However, future developments to obtain more efficient shape functions and integration rules along the lines discussed in this paper would be of much value.

References

- Atluri, S.N. and Zhu, T. (1998), "A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics", *Computational Mechanics*, Vol. 22, pp. 117-27.
- Bathe, K.J. (1996), *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ.
- Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. and Krysl, P. (1996), "Meshless methods: an overview and recent developments", *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, pp. 3-47.
- Breitkopf, P., Rassineux, A., Touzot, G. and Villon, P. (2000), "Explicit form and efficient computation of MLS shape functions and their derivatives", *International Journal for Numerical Methods in Engineering*, Vol. 48, pp. 451-66.
- De, S. and Bathe, K.J. (2000), "The method of finite spheres", *Computational Mechanics*, Vol. 25, pp. 329-45.
- De, S. and Bathe, K.J. (nd), "Displacement/pressure mixed interpolation in the method of finite spheres", *International Journal for Numerical Methods in Engineering*, (submitted for publication).
- Duarte, C.A. and Oden, J.T. (1996), " $h\text{-}p$ clouds – an $h\text{-}p$ meshless method", *Num. Meth. Partial Differential Equations*, Vol. 12, pp. 673-705.
- Krysl, P. and Belytschko, T. (1995), "Analysis of thin plates by the element-free Galerkin method", *Computational Mechanics*, Vol. 17, pp. 26-35.
- Liszka, T. and Orkisz, J. (1980), "The finite difference method at arbitrary irregular grids and its applications in applied mechanics", *Computers and Structures*, Vol. 11, pp. 83-95.
- Liu, W.K., Adey, J. and Jun, S. (1993), "Reproducing kernel particle methods for elastic and plastic problems", in Benson, D.J. and Asaro, R.A. (Eds), *Advanced Computational Methods for Material Modeling*, AMD 180 and PVP 33, ASME, pp. 175-90.
- Lucy, L.B. (1977), "A numerical approach to the testing of the fission hypothesis", *The Astronomical Journal*, Vol. 82 No. 12, pp. 1013-24.
- Melenk, J.M. and Babuška, I. (1996), "The partition of unity finite element method: basic theory and applications", *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, pp. 289-314.
- Nayroles, B., Touzot, G. and Villon, P. (1992), "Generalizing the FEM: diffuse approximation and diffuse elements", *Computational Mechanics*, Vol. 10, pp. 307-18.
- Oñate, E., Idelsohn, S., Zienkiewicz, O.C. and Taylor, R.L. (1996), "A finite point method in computational mechanics: applications to convective transport and fluid flow", *International Journal for Numerical Methods in Engineering*, Vol. 39, pp. 3839-66.

- Peirce, W.H. (1957), "Numerical integration over the planar annulus", *Journal of the Society for Industrial and Applied Mathematics*, Vol. 5 No. 2, pp. 66-73.
- Shepard, D. (1968), "A two-dimensional interpolation function for irregularly spaced data", *Proceedings of the 23rd Nat. Conf. ACM*, pp. 517-24.
- Wagner, G.J. and Liu, W.K. (nd), "Application of essential boundary conditions in mesh-free methods: a corrected collocation method", *International Journal for Numerical Methods in Engineering* (submitted for publication).
- Zhu, T. and Atluri, S.N. (1998), "A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element-free Galerkin method", *Computational Mechanics*, Vol. 21, pp. 211-22.
- Zhu, T., Zhang, J.D. and Atluri, S.N. (1998), "A local boundary integral equation (LBIE) method in computational mechanics and a meshless discretization approach", *Computational Mechanics*, Vol. 21, pp. 223-35.