INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2002; **53**:1937–1958 (DOI: 10.1002/nme.368)

Moving particle finite element method

Su Hao, Harold S. Park and Wing Kam Liu*,†

Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, U.S.A.

SUMMARY

This paper presents the fundamental concepts behind the moving particle finite element method, which combines salient features of finite element and meshfree methods. The proposed method alleviates certain problems that plague meshfree techniques, such as essential boundary condition enforcement and the use of a separate background mesh to integrate the weak form. The method is illustrated via two-dimensional linear elastic problems. Numerical examples are provided to show the capability of the method in benchmark problems. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: meshfree; nodal integration; particle method; large deformation; impact; penetration

1. INTRODUCTION

The finite element method (FEM) is the most popular and widely used method in computational mechanics. However, the FEM is not without shortcomings. One is distortion in the mesh that occurs if the body undergoes a significant deformation; this mesh distortion can either terminate the calculation or result in dramatic deterioration of accuracy. Another is that the FEM often requires a very fine mesh in problems with high gradients or a distinct local character, which can be computationally expensive.

Due to these and other shortcomings of the FEM, so-called meshless, or meshfree methods have recently emerged. These methods rely on constructing interpolation functions at arbitrary discrete points in the domain by enforcing certain reproducing conditions, thereby eliminating the need for elements and a mesh. Examples of meshless methods include the reproducing kernel particle method (RKPM) [1, 2] by Liu *et al.*, element-free Galerkin method (EFG) [3–5] by Belytschko *et al.*, smooth particle hydrodynamics (SPH) [6–8] by Lucy and co-workers, partition of unity methods (PUM) [9–11] by Babuska *et al.*, hp clouds [12] by Oden *et al.* and free mesh methods [13, 14] by Yagawa *et al.* These meshless methods have

*Correspondence to: Wing Kam Liu, Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, U.S.A.
†E-mail: w-liu@northwestern.edu

Contract/grant sponsor: ARO
Contract/grant sponsor: NSF

Received 3 March 2001
Revised 16 May 2001

proven to be quite capable in certain applications where the FEM traditionally fails, such as large deformation problems [15, 16], dynamic shear band propagation [7, 18] and propagation of discontinuities [19–22], such as cracks. Other recent advances include the corrected smooth particle hydrodynamics (CSPH) [23] by Bonet *et al.*, meshless local Petrov–Galerkin (MLPG) by Atluri *et al.* [24, 25], stabilized conforming (SC) nodal integration by Chen *et al.* [26], and coupling of meshfree methods and finite elements by Huerta *et al.* [27]. An excellent summary of the important findings can be found in three special issues of research papers; see References [28–30].

While meshless methods are an improvement over the FEM for the applications mentioned above, this class of methods still faces some issues. For instance, the name 'meshless' is somewhat of a misnomer, as most meshless methods typically employ a background mesh to set up Gaussian quadrature points to integrate the weak form. Others, such as SPH, fail due to a lack of consistency, particularly near or at the boundary of the domain. A problem inherent in some meshless methods is the fact that the shape functions do not satisfy the Kronecker–Delta property, which causes additional problems in enforcing essential boundary conditions. See References [31, 32] for an elegant treatment of this problem. A notable exception to this is the natural element method (NEM) by Sukumar *et al.* [33, 34]. Finally, all meshless methods are to varying degrees more expensive computationally than the FEM.

We now introduce a new method in computational mechanics, the moving particle finite element method (MPFEM). The MPFEM offers the following attributes:

- It should be as efficient and as accurate as the FEM.
- The cost of computing MPFEM shape functions should be competitive with the FEM.
- No background mesh or stress points should be necessary to integrate the weak form. Therefore, all computations of interest, such as stress and strain, are at the particles.
- Essential boundary conditions should be handled exactly as in FEM without recourse to special methods.

This paper is organized as follows. First the governing equations for linear elastostatics are reviewed, then the corresponding weak formulation is derived and discretized. Essential details of the MPFEM, including the calculation of shape functions and their derivatives, essential boundary condition treatment and reproducing conditions are presented via example in Section 3. In Section 4, a 5-node MPFEM element is derived, and the 5-node element is tested in a benchmark problem in Section 5. A summary and concluding discussion comprise the final section.

2. GOVERNING EQUATIONS AND WEAK FORM

In this section, the governing equations for classical linear elastostatics are reviewed, and a weak formulation is derived. As given in Reference [35], the equilibrium equation and boundary conditions in the domain $\Omega \subset \Re^2$ bounded by Γ can be written as follows:

$$\sigma_{ij,j} + f_i = 0 \tag{1a}$$

$$u_i = g_i \quad \text{on } \Gamma_g \tag{1b}$$

$$\sigma_{ij} n_j = h_i \quad \text{on } \Gamma_h \tag{1c}$$

In (1), $\sigma_{ij} = C_{ijkl}\varepsilon_{kl} = C_{ijkl}u_{(k,l)}$ is the Cauchy stress tensor, where (k, l) indicates the symmetric part of the strain, f_i is the body force per unit volume, ε_{kl} the strain tensor, C_{ijkl} are the elastic coefficients, h_i the boundary traction, g_i the prescribed boundary displacement, Γ_h the natural boundary and Γ_g the essential boundary. The comma following a variable indicates a partial derivative with respect to the indicated spatial variable.

Multiplying by a test function δu_i and performing integration by parts, the following weak formulation is obtained:

$$\int_\Omega \delta u_{(i,j)} C_{ijkl} u_{(k,l)}\, \mathrm{d}\Omega = \int_\Omega \delta u_i f_i\, \mathrm{d}\Omega + \int_{\Gamma_{h_i}} \delta u_i h_i\, \mathrm{d}\Gamma_{h_i} \tag{2}$$

For the weak form given above in (2), the left-hand term is approximated as

$$\int_\Omega \delta u_i C_{ijkl} u_{(k,l)}\, \mathrm{d}\Omega \approx \delta \mathbf{u}^{\mathrm{T}} \mathbf{K} \mathbf{u} \tag{3}$$

where $\delta\mathbf{u}$ and $\mathbf{u}$ are the trial function and discrete displacement vectors, respectively. The stiffness matrix $\mathbf{K}$ is approximated by

$$\mathbf{K} = \int_\Omega \mathbf{B}^{\mathrm{T}} \mathbf{C} \mathbf{B}\, \mathrm{d}\Omega \approx \sum_{I=1}^{\mathrm{NP}} \mathbf{B}^{\mathrm{T}}(\mathbf{x}_I) \mathbf{C} \mathbf{B}(\mathbf{x}_I) \Delta V_I \tag{4}$$

where NP is the number of particles and ΔV_I is an integration weight associated with each particle. It is important to note that a particle is equivalent to a node in the FEM formulation.

Similarly, the second term in (2), the body force term, is approximated as

$$\int_\Omega \delta u_i f_i\, \mathrm{d}\Omega \approx \sum_{I=1}^{\mathrm{NP}} \mathbf{N}^{\mathrm{T}}(\mathbf{x}_I) \mathbf{f}(\mathbf{x}_I) \Delta V_I \tag{5}$$

The final term in (2), the traction term, is approximated as

$$\int_{\Gamma_{h_i}} \delta u_i h_i\, \mathrm{d}\Gamma_{h_i} \approx \sum_{I=1}^{\mathrm{NB}} \mathbf{N}^{\mathrm{T}}(\mathbf{x}_I) \mathbf{h}(\mathbf{x}_I) \Delta S_I \tag{6}$$

where NB is the number of particles on the natural boundary of the domain and ΔS_I is an integration weight associated with boundary particles.

In the MPFEM, the weak form is integrated via a nodal integration scheme. The only restriction on such a scheme is that the integration weights sum to the total volume of the domain, i.e.

$$\sum_{I=1}^{\mathrm{NP}} V_I = V \tag{7}$$

where V is the total volume of the domain Ω. For the MPFEM, Delauney triangulation is used to compute the integration weight for each particle. Consider point G in Figure 1. First, certain neighbouring particles are found using the algorithm described in Appendix B, which are the same particles that create the MPFEM5 element discussed in Section 4. Next, a triangulation procedure is initiated in which each neighbouring particle is connected to the middle particle

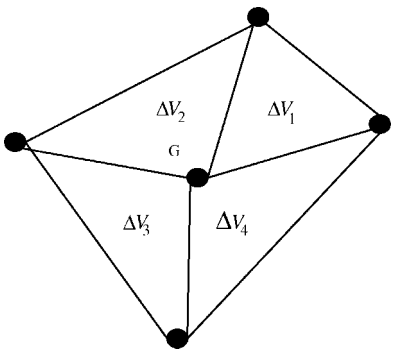

Figure 1. Example of triangulation procedure to determine the integration weight for particle G.

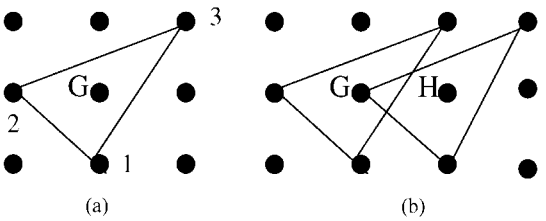

Figure 2. (a) Example of MPFEM triangular element for particle G; (b) demonstration of overlap of two MPFEM triangular elements, for particles G and H.

(G in this case), and the two particles closest to it. The end result is that four triangles will be found, each having a certain area. The integration volume for point G is then computed by

$$\Delta V_{\mathrm{G}} = \tfrac{1}{3}(\Delta V_1 + \Delta V_2 + \Delta V_3 + \Delta V_4) \tag{8}$$

For the case of an individual particle, consider point G in Figure 2(a). The particle stiffness matrix for G is

$$\mathbf{K}(\mathbf{x}_{\mathrm{G}}) = \mathbf{B}^{\mathrm{T}}(\mathbf{x}_{\mathrm{G}})\mathbf{C}\mathbf{B}(\mathbf{x}_{\mathrm{G}})\Delta V_{\mathrm{G}} \tag{9}$$

where the strain displacement matrix $\mathbf{B}$ for point G is

$$\mathbf{B}(\mathbf{x}_{\mathrm{G}}) = \begin{bmatrix} N_1^x(\mathbf{x}_{\mathrm{G}}) & 0 & N_2^x(\mathbf{x}_{\mathrm{G}}) & 0 & N_3^x(\mathbf{x}_{\mathrm{G}}) & 0 \\ 0 & N_1^y(\mathbf{x}_{\mathrm{G}}) & 0 & N_2^y(\mathbf{x}_{\mathrm{G}}) & 0 & N_3^y(\mathbf{x}_{\mathrm{G}}) \\ N_1^y(\mathbf{x}_{\mathrm{G}}) & N_1^x(\mathbf{x}_{\mathrm{G}}) & N_2^y(\mathbf{x}_{\mathrm{G}}) & N_2^x(\mathbf{x}_{\mathrm{G}}) & N_3^y(\mathbf{x}_{\mathrm{G}}) & N_3^x(\mathbf{x}_{\mathrm{G}}) \end{bmatrix} \tag{10}$$

where N_I^x indicates a partial derivative of particle I's shape function with respect to $\mathbf{x}$.

The elastic coefficient tensor $\mathbf{C}$ for isotropic materials under a plane strain state is given by

$$\mathbf{C} = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1-v & v & 0 \\ v & 1-v & 0 \\ 0 & 0 & \dfrac{1-2v}{2} \end{bmatrix} \tag{11}$$

where E is the Young's modulus and v the Poisson's ratio.

Figure 2(b) illustrates some aesthetic differences between the MPFEM and FEM. In a FEM mesh, there is no overlap amongst elements, and the nodes define the boundaries of each element. In the MPFEM, neighbouring particles are used to form elements around a particular point. Due to the notion of forming elements around a point, the MPFEM elements inevitably overlap.

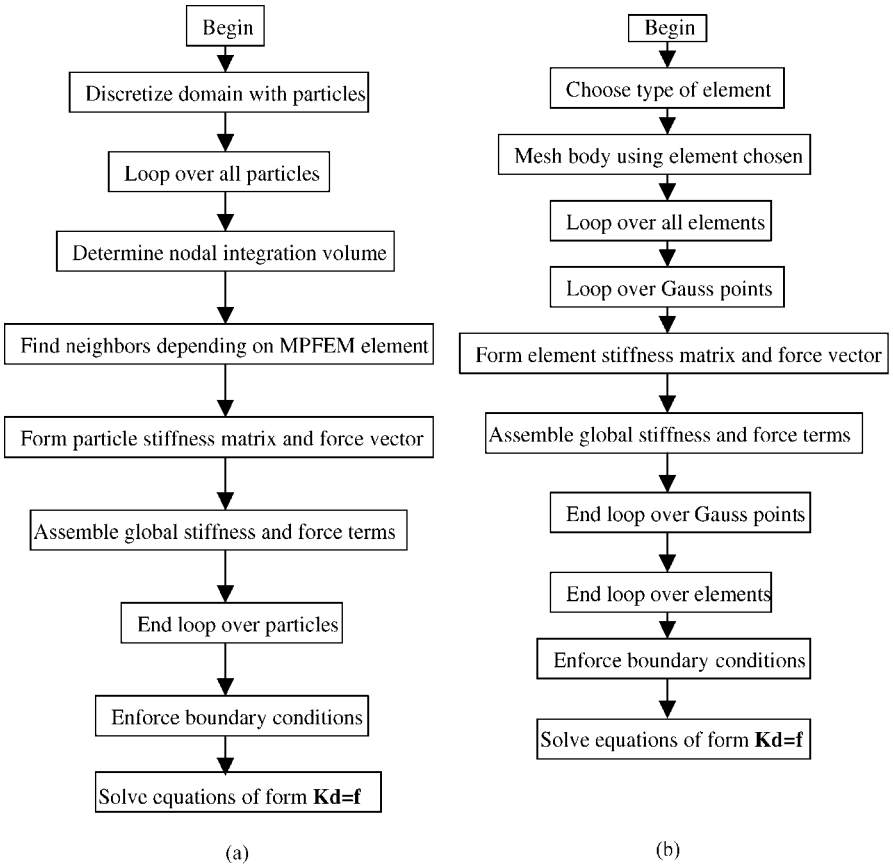

Figure 3. (a) Discretization procedure for MPFEM; (b) discretization procedure for FEM.

Since no mesh is needed and the approximation is constructed at discrete points in the domain, the MPFEM is a particle method. A FEM character is inherent in the MPFEM because the particles chosen to surround a point form an element-like being about the point. Since the MPFEM elements can be shown to be equivalent to their FEM brethren, i.e. the triangular element in Figure 2 reproduces constant as well as linear fields in both x and y like the FEM triangle, the method is thus called a particle finite element method.

It is crucial to note that the shape function derivatives shown above for the **B** matrix are not obtained via direct differentiation of the shape functions. Instead, they are determined by the reproducing conditions given in Section 3, as they are the shape functions themselves. Because of this fact, the MPFEM can be formulated based on an assumed strain method [36].

The programming flowchart for the MPFEM is summarized in Figure 3(a). For comparison, a typical coding flowchart for the FEM is given in Figure 3(b). In the MPFEM, the first step is to discretize the domain of the problem with particles. Then a loop over all particles is initiated. For each particle, nearest neighbours are found which encompass the particle under consideration and which satisfy certain reproducing conditions. The criteria for selecting these neighbours are described in the Appendix B. Then, the nodal integration volume is determined

for the particle. Next, the shape functions and derivatives at the particle are determined and the **B** matrix for each particle is calculated. Finally, the particle stiffness matrix is calculated, integrated using the nodal integration weight, and augmented into the global stiffness matrix.

3. MPFEM FUNDAMENTALS

3.1. Reproducing conditions

In a particle-based method, the reproducing conditions, or order of monomial that the approximation is expected to reproduce, are satisfied between every particle and its neighbours under consideration. As will be demonstrated below, the MPFEM shape functions and derivatives are constructed to reproduce any order of monomial desired.

For zeroth-order consistency, the approximation should exactly reproduce a constant, so 1 is taken for convenience. Mathematically, this says

$$u^h(\mathbf{x}) = 1 = \left(\sum_I N_I(\mathbf{x})\right)(1) \tag{12}$$

because $u_I = 1$. This shows that

$$\sum_I N_I(\mathbf{x}) = 1 \tag{13}$$

This expression is important as it implies partition of unity, which ensures that the numerical scheme is consistent.

First-order consistency states that the approximation should exactly reproduce a linear field, i.e.

$$\sum_I N_I(\mathbf{x})\, x_I = x \tag{14}$$

However, since the reproducing conditions are satisfied between particles, we shall show that for the MPFEM, this implies that

$$\sum_I N_I(\mathbf{x}_G)(x_I - x_G) = 0 \tag{15}$$

where x_G represents the coordinates at an arbitrary point. Expanding this expression, and using (13), it can be shown that

$$\sum_I N_I(\mathbf{x}_G)\, x_I = x_G \tag{16}$$

which proves that the proposed approximation can exactly reproduce a linear field in x. Similar calculations can be carried out to prove that the reproducing conditions in y are also satisfied.

Deriving reproducing conditions for the shape function derivatives is achieved in a similar fashion. Consider an approximation in the form of

$$\frac{\partial u^h(\mathbf{x})}{\partial x} = \sum_I N_I^x(\mathbf{x}) u_I \tag{17}$$

where the $N_I^x(\mathbf{x})$ are the derivatives of the shape functions. For the derivatives, zeroth-order consistency means that the derivative of a constant field should equal zero, or

$$\frac{\partial u^h(\mathbf{x})}{\partial x} = 0 = \left(\sum_I N_I^x(\mathbf{x})\right)(1) \tag{18}$$

Again, 1 is used as the constant value. This implies for the zeroth-order reproducing condition for derivatives that

$$\sum_I N_I^x(\mathbf{x}) = 0 \tag{19}$$

For first-order consistency, the derivative of a linear field should yield a constant, or

$$1 = \sum_I N_I^x(\mathbf{x})\, x_I \tag{20}$$

Employing the same shifting technique as above, to ensure consistency between discrete particles, we propose that the first-order derivative consistency is equivalent to saying

$$1 = \sum_I N_I^x(\mathbf{x}_G)(x_I - x_G) \tag{21}$$

To prove this, expand the above expression to

$$1 = \sum_I N_I^x(\mathbf{x}_G)\, x_I - \sum_I N_I^x(\mathbf{x}_G)\, x_G \tag{22}$$

However, because x_G are the coordinates of an arbitrary point and thus is independent of the summation on I, the expression becomes

$$1 = \sum_I N_I^x(\mathbf{x}_G)\, x_I - x\sum_I N_I^x(\mathbf{x}_G) \tag{23}$$

Utilizing (19), the expression simplifies to

$$\sum_I N_I^x(\mathbf{x}_G)\, x_I = 1 \tag{24}$$

proving that the MPFEM approximation enforces the first-order reproducing condition for the x-derivative. It can be shown in a similar fashion that the y-derivative also satisfies the same reproducing conditions. The reproducing conditions for the 2D MPFEM triangle are summarized in Table I.

Table I. MPFEM triangle (2D linear) reproducing conditions.

u_I	1	$x - x_G$	$y - y_G$
u^h	1	0	0
$u^h_{,x}$	0	1	0
$u^h_{,y}$	0	0	1

3.2. MPFEM shape functions

Consider a 2D approximation of the form

$$u^h(\mathbf{x}) = \sum_{I=1}^{3} N_I(\mathbf{x}) u_I \tag{25}$$

Here, $N_I(\mathbf{x})$ are the shape functions and u_I are the nodal degrees of freedom. The expression for the x-derivative is then

$$u^h_{,x}(\mathbf{x}) = \sum_{I=1}^{3} N^x_I(\mathbf{x}) u_I \tag{26}$$

where $N^x_I(\mathbf{x})$ are the x-derivatives of the shape functions. Similarly, the expression for the y-derivative is

$$u^h_{,y}(\mathbf{x}) = \sum_{I=1}^{3} N^y_I(\mathbf{x}) u_I \tag{27}$$

where $N^y_I(\mathbf{x})$ are the y-derivatives of the shape functions.

Writing out the expressions for the approximation and its derivatives at point G in Figure 2, one obtains

$$u^h(\mathbf{x}_\mathrm{G}) = \sum_{I=1}^{3} N_I(\mathbf{x}_\mathrm{G}) u_I = N_1(\mathbf{x}_\mathrm{G}) u_1 + N_2(\mathbf{x}_\mathrm{G}) u_2 + N_3(\mathbf{x}_\mathrm{G}) u_3 \tag{28a}$$

$$u^h_{,x}(\mathbf{x}_\mathrm{G}) = \sum_{I=1}^{3} N^x_I(\mathbf{x}_\mathrm{G}) u_I = N^x_1(\mathbf{x}_\mathrm{G}) u_1 + N^x_2(\mathbf{x}_\mathrm{G}) u_2 + N^x_3(\mathbf{x}_\mathrm{G}) u_3 \tag{28b}$$

$$u^h_{,y}(\mathbf{x}_\mathrm{G}) = \sum_{I=1}^{3} N^y_I(\mathbf{x}_\mathrm{G}) u_I = N^y_1(\mathbf{x}_\mathrm{G}) u_1 + N^y_2(\mathbf{x}_\mathrm{G}) u_2 + N^y_3(\mathbf{x}_\mathrm{G}) u_3 \tag{28c}$$

In comparison with the FEM, Equations (28a)–(28c) show that the expressions for the MPFEM triangle are identical to that of the FEM triangle. However, the fundamental difference is that the MPFEM shape functions *are not* used as interpolation functions. The reason for this can be seen in Figure 2(b). Note how the MPFEM elements for particles G and H overlap. Therefore, a problem arises if an approximation is desired at a point intersected by both triangles. The problem lies in determining which element will interpolate the values to that point. Therefore, in order to resolve such inconsistencies, the MPFEM approximation is constructed at individual discrete points where an approximation is desired.

This leads to the explanation of the *moving* in MPFEM. The *moving* idea is that the MPFEM shape functions are computed pointwise in the domain using the reproducing conditions detailed above. Thus, the notion of MPFEM elements is purely for comparative purposes, as the elements are not used to interpolate any field values.

Now the expression for the shape functions and their derivatives are written out in a useful format. For point G in Figure 1, these expressions become

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 - x_\mathrm{G} & x_2 - x_\mathrm{G} & x_3 - x_\mathrm{G} \\ y_1 - y_\mathrm{G} & y_2 - y_\mathrm{G} & y_3 - y_\mathrm{G} \end{bmatrix} \begin{bmatrix} N_1(\mathbf{x}_\mathrm{G}) \\ N_2(\mathbf{x}_\mathrm{G}) \\ N_3(\mathbf{x}_\mathrm{G}) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{29a}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 - x_G & x_2 - x_G & x_3 - x_G \\ y_1 - y_G & y_2 - y_G & y_3 - y_G \end{bmatrix} \begin{bmatrix} N_1^x(\mathbf{x}_G) \\ N_2^x(\mathbf{x}_G) \\ N_3^x(\mathbf{x}_G) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{29b}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 - x_G & x_2 - x_G & x_3 - x_G \\ y_1 - y_G & y_2 - y_G & y_3 - y_G \end{bmatrix} \begin{bmatrix} N_1^y(\mathbf{x}_G) \\ N_2^y(\mathbf{x}_G) \\ N_3^y(\mathbf{x}_G) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{29c}$$

It is important to note that Equations (29a)–(29c) are simply the explicit expressions of the reproducing conditions defined in Section 3.1. Note also that the final terms in (29a)–(29c) are the reproducing vectors that are summarized in Table I. We now define a matrix $\mathbf{W}$, which in this case is

$$\mathbf{W}(\mathbf{x}_G) = \begin{bmatrix} 1 & 1 & 1 \\ x_1 - x_G & x_2 - x_G & x_3 - x_G \\ y_1 - y_G & y_2 - y_G & y_3 - y_G \end{bmatrix} \tag{30}$$

Equations (29a)–(29c) can be combined into one, giving the shape functions and their derivatives at point G:

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 - x_G & x_2 - x_G & x_3 - x_G \\ y_1 - y_G & y_2 - y_G & y_3 - y_G \end{bmatrix} \begin{bmatrix} N_1(\mathbf{x}_G) & N_1^x(\mathbf{x}_G) & N_1^y(\mathbf{x}_G) \\ N_2(\mathbf{x}_G) & N_2^x(\mathbf{x}_G) & N_2^y(\mathbf{x}_G) \\ N_3(\mathbf{x}_G) & N_3^x(\mathbf{x}_G) & N_3^y(\mathbf{x}_G) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{31}$$

We now justify our claim that the MPFEM should be on the same order of efficiency as the FEM:

- There is no numerical integration in the MPFEM.
- The number of nodes is much smaller than the number of Gaussian quadrature points.
- As demonstrated in (31), we have derived explicit expressions for determining the shape functions and their derivatives at each node; thus, only one matrix inversion is necessary to determine all necessary interpolants.

3.3. Imposing essential boundary conditions with MPFEM

As illustrated below, the MPFEM can be constructed to restore the Kronecker–Delta property for boundary particles. By accomplishing this, essential boundary conditions can be enforced simply as in the FEM. One manner in which this can be accomplished is by picking the

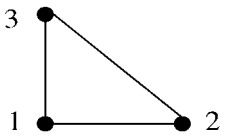

Figure 4. Example of MPFEM boundary element, for particle 1.

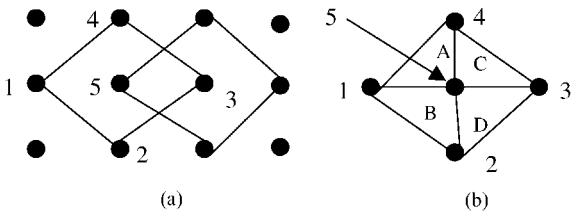

Figure 5. (a) The MPFEM5 element for points 3 and 5; (b) triangulation procedure to compute shape function derivatives, for point 5.

boundary particle itself as one of the neighbours of the particle. For example, consider the case illustrated in Figure 4. In this case, particle 1 is the boundary point for which the shape function is desired. For particle 1, $\mathbf{W}$ becomes

$$\mathbf{W}(x_1) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & x_2 - x_1 & x_3 - x_1 \\ 0 & y_2 - y_1 & y_3 - y_1 \end{bmatrix} \tag{32}$$

Multiplying the inverse of $\mathbf{W}$ by $\mathbf{P}(0)$, where $\mathbf{P}(0)^{\mathrm{T}} = [1, 0, 0]$ yields

$$\begin{bmatrix} N_1(x_1) \\ N_2(x_1) \\ N_3(x_1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{33}$$

so the Kronecker–Delta property is preserved.

4. A 5-NODE MPFEM ELEMENT

4.1. Element description

The MPFEM element which we have chosen to study in detail is a 5-node element that reproduces a constant field, a linear field in x and y, and quadratic fields in x and y. A pictorial description of the element is given in Figure 5(a). A potential problem for all MPFEM elements is when the $\mathbf{W}$ matrix is singular. Therefore, we propose that the MPFEM5 x-derivatives are in the following form:

$$\mathbf{N}_2^x(\mathbf{x}_5) = \tfrac{1}{4}(\mathbf{Q}) + (\eta_1 - \tfrac{1}{4})(\mathbf{R}) + (\eta_2 - \tfrac{1}{4})(\mathbf{S}) \tag{34}$$

where

$$\mathbf{Q}=\begin{bmatrix} N^x_{1A}(\mathbf{x}_5)\\ 0\\ 0\\ N^x_{4A}(\mathbf{x}_5)\\ N^x_{5A}(\mathbf{x}_5)\end{bmatrix}+\begin{bmatrix} N^x_{1B}(\mathbf{x}_5)\\ N^x_{2B}(\mathbf{x}_5)\\ 0\\ 0\\ N^x_{5B}(\mathbf{x}_5)\end{bmatrix}+\begin{bmatrix} 0\\ N^x_{2C}(\mathbf{x}_5)\\ N^x_{3C}(\mathbf{x}_5)\\ 0\\ N^x_{5C}(\mathbf{x}_5)\end{bmatrix}+\begin{bmatrix} 0\\ 0\\ N^x_{3D}(\mathbf{x}_5)\\ N^x_{4D}(\mathbf{x}_5)\\ N^x_{5D}(\mathbf{x}_5)\end{bmatrix} \tag{35a}$$

$$\mathbf{R}=\begin{bmatrix} N^x_{1A}(\mathbf{x}_5)\\ 0\\ 0\\ N^x_{4A}(\mathbf{x}_5)\\ N^x_{5A}(\mathbf{x}_5)\end{bmatrix}+\begin{bmatrix} N^x_{1B}(\mathbf{x}_5)\\ N^x_{2B}(\mathbf{x}_5)\\ 0\\ 0\\ N^x_{5B}(\mathbf{x}_5)\end{bmatrix} \tag{35b}$$

$$\mathbf{S}=\begin{bmatrix} 0\\ N^x_{2C}(\mathbf{x}_5)\\ N^x_{3C}(\mathbf{x}_5)\\ 0\\ N^x_{5C}(\mathbf{x}_5)\end{bmatrix}+\begin{bmatrix} 0\\ 0\\ N^x_{3D}(\mathbf{x}_5)\\ N^x_{4D}(\mathbf{x}_5)\\ N^x_{5D}(\mathbf{x}_5)\end{bmatrix} \tag{35c}$$

and $N^x_{IA}(\mathbf{x}_5)$ are the x-derivatives for particle I associated with triangle A at particle 5.

In the above, the MPFEM5 element is first decomposed into four triangles, as shown in Figure 5(b). This is done because it is known that the **W** matrix for the triangular elements is not singular unless the three points lie in a straight line. Then, for each triangle, a set of equations identical to (29b) and (29c) is solved, yielding the x and y derivatives for each point of each triangle at point 5. This ensures satisfaction of the constant and linear derivative reproducing conditions. Thus, **Q** is simply the average of the derivatives of the four triangles at point 5.

4.2. x-Derivatives

However, the linear and quadratic reproducing conditions are yet to be imposed. For the x-derivatives, the following equations must be solved, which give the values for the η's in (34)

$$\begin{bmatrix}\eta_1\\ \eta_2\end{bmatrix}=\begin{bmatrix}\alpha_1 & \alpha_2\\ \alpha_3 & \alpha_4\end{bmatrix}^{-1}\begin{bmatrix}1\\ 0\end{bmatrix} \tag{36a,b}$$

where

$$\alpha_1 = \sum_I N_{IA}^x(\mathbf{x}_I - \mathbf{x}_5) + \sum_J N_{JB}^x(\mathbf{x}_J - \mathbf{x}_5) \tag{37a}$$

$$\alpha_2 = \sum_I N_{IC}^x(\mathbf{x}_I - \mathbf{x}_5) + \sum_J N_{JD}^x(\mathbf{x}_J - \mathbf{x}_5) \tag{37b}$$

$$\alpha_3 = \sum_I N_{IA}^x(\mathbf{x}_I - \mathbf{x}_5)^2 + \sum_J N_{JB}^x(\mathbf{x}_J - \mathbf{x}_5)^2 \tag{37c}$$

$$\alpha_4 = \sum_I N_{IC}^x(\mathbf{x}_I - \mathbf{x}_5)^2 + \sum_J N_{JD}^x(\mathbf{x}_J - \mathbf{x}_5)^2 \tag{37d}$$

Equation (36a) is derived from enforcing linear reproducing conditions as in (37a) and (37b), while (36b) comes from enforcing quadratic reproducing conditions as in (37c) and (37d).

As can be seen from above, the η's represent the geometric irregularity of the particle discretization. If the mesh is regular, $\eta_1 = \eta_2 = \frac{1}{4}$, and the shape function derivatives for point 5 are the average of the x-derivatives of the triangles that include that point. For this case, the **R** and **S** terms contribute nothing to the derivative approximation. Thus, the first term in (34) can be thought of as the term that satisfies the constant reproducing conditions. If the mesh is irregular, $\eta_1 \neq \frac{1}{4}, \eta_2 \neq \frac{1}{4}$, and the **R** and **S** vectors in (34) are no longer zero. Due to this, these terms can be thought of as the terms that satisfy the linear and quadratic reproducing conditions.

Let us now prove the equivalence of (34) to obtaining the shape function derivatives (for regular spacing) via inverting the **W** matrix and multiplying by the appropriate reproducing vector. Consider the MPFEM5 element for particle 5, shown in Figure 5(a). Define the uniform grid spacing to be Δ. The x-derivatives are calculated by solving a system of equations

$$\mathbf{N}^x(\mathbf{x}_5) = \mathbf{W}^{-1}\mathbf{P}^x(0) \tag{38a}$$

where

$$\mathbf{W}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -\Delta & 0 & \Delta & 0 & 0 \\ 0 & -\Delta & 0 & \Delta & 0 \\ \Delta^2 & 0 & \Delta^2 & 0 & 0 \\ 0 & \Delta^2 & 0 & \Delta^2 & 0 \end{bmatrix}^{-1} \tag{38b}$$

$$\mathbf{P}^x(0)^{\mathrm{T}} = [0, 1, 0, 0, 0] \tag{38c}$$

For this regularly spaced configuration,

$$\mathbf{N}^x(\mathbf{x}_5)^{\mathrm{T}} = \left[-\frac{1}{2\Delta}, 0, \frac{1}{2\Delta}, 0, 0\right] \tag{39}$$

Applying the theory described above, first, the MPFEM5 element is decomposed into four triangles, as in Figure 4(b). For these four triangles, it can be shown that

$$N_A^x(\mathbf{x}_5)^{\mathrm{T}} = \left[-\frac{1}{\Delta}, 0, 0, 0, \frac{1}{\Delta}\right] \tag{40a}$$

$$N_B^x(\mathbf{x}_5)^{\mathrm{T}} = \left[-\frac{1}{\Delta}, 0, 0, 0, \frac{1}{\Delta}\right] \tag{40b}$$

$$N_C^x(\mathbf{x}_5)^{\mathrm{T}} = \left[0, 0, \frac{1}{\Delta}, 0, -\frac{1}{\Delta}\right] \tag{40c}$$

$$N_D^x(\mathbf{x}_5)^{\mathrm{T}} = \left[0, 0, \frac{1}{\Delta}, 0, -\frac{1}{\Delta}\right] \tag{40d}$$

Taking these results and using (34), it can easily be proven that the result of (34) is equivalent to (39) for the regularly spaced case.

For completeness, it should be noted that the MPFEM5 shape function satisfies the Kronecker–Delta property. This can be shown by inverting $\mathbf{W}$ as in (38b), and multiplying by the appropriate reproducing vector. The resulting shape functions display the Kronecker–Delta property at point $\mathbf{x}_5$.

4.3. *y*-Derivatives

For the y-derivatives, a procedure nearly identical to computing the x-derivatives is followed. First, the y-derivatives can be written as

$$\mathbf{N}^y(\mathbf{x}_5) = \tfrac{1}{4}(\mathbf{Q}) + (\eta_1 - \tfrac{1}{4})(\mathbf{R}) + (\eta_2 - \tfrac{1}{4})(\mathbf{S}) \tag{41}$$

where

$$\mathbf{Q} = \begin{bmatrix} N_{1A}^y(\mathbf{x}_5) \\ 0 \\ 0 \\ N_{4A}^y(\mathbf{x}_5) \\ N_{5A}^y(\mathbf{x}_5) \end{bmatrix} + \begin{bmatrix} N_{1B}^y(\mathbf{x}_5) \\ N_{2B}^y(\mathbf{x}_5) \\ 0 \\ 0 \\ N_{5B}^y(\mathbf{x}_5) \end{bmatrix} + \begin{bmatrix} 0 \\ N_{2C}^y(\mathbf{x}_5) \\ N_{3C}^y(\mathbf{x}_5) \\ 0 \\ N_{5C}^y(\mathbf{x}_5) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ N_{3D}^y(\mathbf{x}_5) \\ N_{4D}^y(\mathbf{x}_5) \\ N_{5D}^y(\mathbf{x}_5) \end{bmatrix} \tag{42a}$$

$$\mathbf{R} = \begin{bmatrix} N_{1A}^y(\mathbf{x}_5) \\ 0 \\ 0 \\ N_{4A}^y(\mathbf{x}_5) \\ N_{5A}^y(\mathbf{x}_5) \end{bmatrix} + \begin{bmatrix} 0 \\ N_{2C}^y(\mathbf{x}_5) \\ N_{3C}^y(\mathbf{x}_5) \\ 0 \\ N_{5C}^y(\mathbf{x}_5) \end{bmatrix} \tag{42b}$$

$$\mathbf{S} = \begin{bmatrix} N^y_{1B}(\mathbf{x}_5) \\ N^y_{2B}(\mathbf{x}_5) \\ 0 \\ 0 \\ N^y_{5B}(\mathbf{x}_5) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ N^y_{3D}(\mathbf{x}_5) \\ N^y_{4D}(\mathbf{x}_5) \\ N^y_{5D}(\mathbf{x}_5) \end{bmatrix} \tag{42c}$$

Note the similarity of the equations above to (34) and (35), except that there is a different grouping of triangles; triangles A and C are grouped together and triangles B and D are grouped together. Again, higher-order reproducing conditions are yet to be satisfied. In this case, linear and quadratic reproducing conditions in y are enforced. The equations corresponding to (36) and (37) are

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{43}$$

where

$$\alpha_1 = \sum_I N^y_{IA}(\mathbf{y}_I - \mathbf{y}_5) + \sum_J N^y_{JC}(\mathbf{y}_J - \mathbf{y}_5) \tag{44a}$$

$$\alpha_2 = \sum_I N^y_{IB}(\mathbf{y}_I - \mathbf{y}_5) + \sum_J N^y_{JD}(\mathbf{y}_J - \mathbf{y}_5) \tag{44b}$$

$$\alpha_3 = \sum_I N^y_{IA}(\mathbf{y}_I - \mathbf{y}_5)^2 + \sum_J N^y_{JC}(\mathbf{y}_J - \mathbf{y}_5)^2 \tag{44c}$$

$$\alpha_4 = \sum_I N^y_{IB}(\mathbf{y}_I - \mathbf{y}_5)^2 + \sum_J N^y_{JD}(\mathbf{y}_J - \mathbf{y}_5)^2 \tag{44d}$$

4.4. Boundary treatment

One fact to note when using the MPFEM5 element is that the boundary particles will undergo a different discretization than the interior particles. This is because it is impossible to form a MPFEM5 element for boundary particles. Therefore, a 4-particle quadrilateral element is constructed for boundary particles. The boundary quadrilateral is constructed to reproduce a constant, linear fields in x and y, and a quadratic field in either x or y. The quadratic field is chosen to ensure that the $\mathbf{W}$ matrix is invertible. Like the boundary element shown in Section 3.3, this 4-particle quadrilateral can be shown to preserve the Kronecker–Delta property, ensuring simple enforcement of essential boundary conditions. The 4-particle quadrilateral is illustrated in Figure 6.

A special case on regular boundaries involves corner particles. For these cases, the $\mathbf{W}$ matrix is singular if one attempts to reproduce a quadratic field. The fields reproduced for this special case are constant, linear in x and y, and bilinear xy.

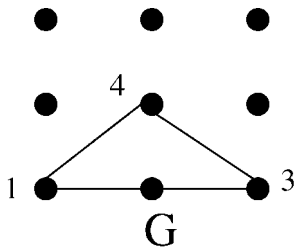

Figure 6. The 4-particle MPFEM boundary element, for point G.

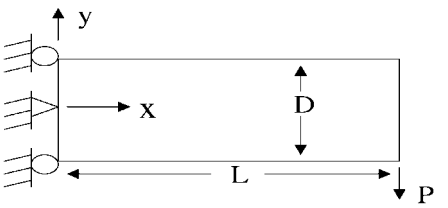

Figure 7. Diagram of shear-loaded beam.

5. NUMERICAL EXAMPLES

5.1. Shear-loaded beam

In this section, the MPFEM5 element is used in solving the problem of a beam that is shear loaded on its free end. The detailed problem statement is

Displacement:

$$u_1(0,0)=u_2(0,0)=0, \quad u_1(0,\pm c)=0 \tag{45a}$$

Traction:

$$t_1(x,\pm c)=t_2(x,\pm c)=0, \quad x\in(0,L) \tag{45b}$$

$$t_1(L,y)=0, \quad y\in(-c,c) \tag{45c}$$

$$t_2(L,y)=\frac{P(c^2-y^2)}{2I}, \quad y\in(-c,c) \tag{45d}$$

$$t_1(0,y)=\frac{PLy}{I}, \quad y\in(-c,0)\cup(0,c) \tag{45e}$$

$$t_2(0,y)=\frac{-P(c^2-y^2)}{2I}, \quad y\in(-c,0)\cup(0,c) \tag{45f}$$

P is a constant, $c=D/2$, and the moment of inertia I is given by

$$I=\frac{2c^3}{3} \tag{45g}$$

The exact solution to the problem is given as follows:

$$\sigma_{11}=\frac{-P(L-x)y}{I}, \quad \sigma_{22}=0, \quad \sigma_{12}=\frac{P(c^2-y^2)}{2I} \tag{45h}$$

$$u_1=\frac{-Py}{6\tilde{E}I}\left((6L-3x)x+(2+\tilde{\nu})\left(y^2-\frac{D^2}{4}\right)\right) \tag{45i}$$

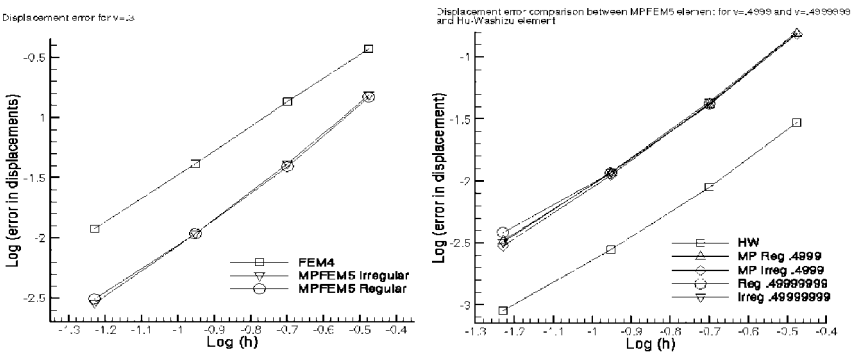

Figure 8. Displacement error norms for MPFEM5.

$$u_2 = \frac{P}{6\tilde{E}I}\left(3\tilde{\nu}y^2(L-x) + (4+5\tilde{\nu})\frac{D^2x}{4} + (3L-x)x^2\right) \tag{45j}$$

For plane strain,

$$\tilde{\nu} = \frac{\nu}{1-\nu} \quad \text{and} \quad \tilde{E} = \frac{E}{1-\nu^2} \tag{45k}$$

Here, $P = -1,\ D = 2,\ L = 10,\ E = 2000$ and $\nu = 0.3$ and 0.4999. The governing equations and weak formulation are identical to that presented in Section 2 (see Figure 7).

5.2. *Numerical results*

In order to evaluate the error in the solution, we use the following displacement error norm:

$$\|\mathbf{u} - \mathbf{u}^h\| = \frac{\sum_{I=1}^{\mathrm{NP}} (\mathbf{u}_I - \mathbf{u}_I^h) \cdot (\mathbf{u}_I - \mathbf{u}_I^h)}{\sum_{J=1}^{\mathrm{NP}} (\mathbf{u}_J \cdot \mathbf{u}_J)} \tag{46}$$

In the above, $\mathbf{u}$ is the exact solution and $\mathbf{u}^h$ is the approximate solution.

Before discussing the numerical results, we first address the definition of displacement error used in this problem. Typically, the displacement error is defined by interpolating the nodal data to Gaussian quadrature points, then summing the error from those points multiplied by an integration weight. However, because the MPFEM does not employ Gauss points to integrate the weak form, it seems appropriate to calculate the error based solely on the field values at the particles.

Figure 8 shows the results of the convergence study, which was performed using uniform and non-uniform grids of $4 \times 2,\ 8 \times 4,\ 16 \times 8$ and 32×16 nodes. For the incompressible case, the MPFEM was compared against a Hu–Washizu (HW) assumed strain element, whose formulation can be found in Reference [37]. The HW displayed better accuracy than the MPFEM5 element, but converges at the same rate. The fact that the MPFEM5 is not as accurate as the HW element is not surprising, as the HW element is specially engineered to completely alleviate locking in incompressible media. However, the fact that the MPFEM5 element converges at the same rate indicates its effectiveness in relieving locking for incompressible media. Note

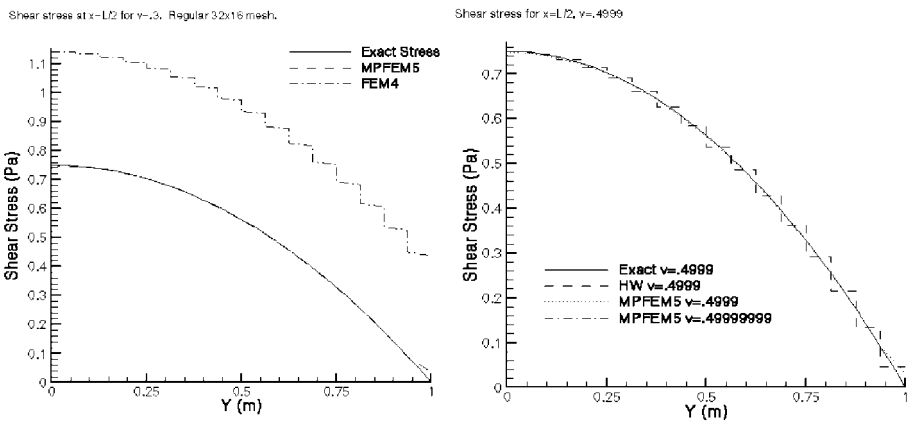

Figure 9. Shear stress comparison between MPFEM5 and various FEM.

that both regular and irregular spacing, compressible and incompressible cases, the error and convergence rate is nearly identical for the MPFEM5 element. It is also worth noting that for the case of $v = 0.49999999$, the MPFEM5 element showed no degradation in either accuracy or rate of convergence.

The simplest explanation as to why the MPFEM5 element can alleviate locking is as follows. The element can be viewed as a standard FEM4 (4-node quadrilateral) element with a bubble mode. By standard FEM4, we mean the isoparametric, bilinear 4-node quadrilateral element. This additional bubble mode gives it an extra degree of freedom as compared to the FEM4, and enables it to perform well in constrained problems. It is well known that the FEM4 element locks in the incompressible limit, so we expect that the additional degree of freedom in the MPFEM5 element will allow it to alleviate this problem.

Figure 9 details the calculation of the shear stress along the midpoint of the beam ($x = L/2$). In the compressible case, the MPFEM5 clearly gives better results than the FEM4 for the same mesh, as the FEM4 is well off, while the MPFEM5 solution is dead on. For the incompressible case, the MPFEM5 again gives a nearly exact solution. The HW element also gives a nice solution, but is restricted by the fact that it is a constant stress element, which manifests itself in the step-like distribution of the stress for that element.

It should be noted for the sake of completeness that the triangular MPFEM element detailed in Section 3 was also tested for the beam bending problems. For the compressible case, it was found that for very coarse meshes, the MPFEM3 (MPFEM triangular element defined in Section 3) element performed worse than the FEM4. However, as the mesh was refined, the MPFEM3 gave a more accurate solution than the FEM4. In the incompressible case, both the MPFEM3 and the FEM4 displayed locking. However, the MPFEM3 element still gave a better result; for the FEM4 to obtain the same results, it was necessary to use a mesh that was twice as fine.

Lastly, a further note on the MPFEM stress calculations. While the MPFEM stresses in the interior of the beam demonstrated excellent accuracy, those at the boundary were not as accurate. We believe the reasons for this are because of the under integration of the traction terms in the weak form and because the MPFEM boundary elements do exhibit the

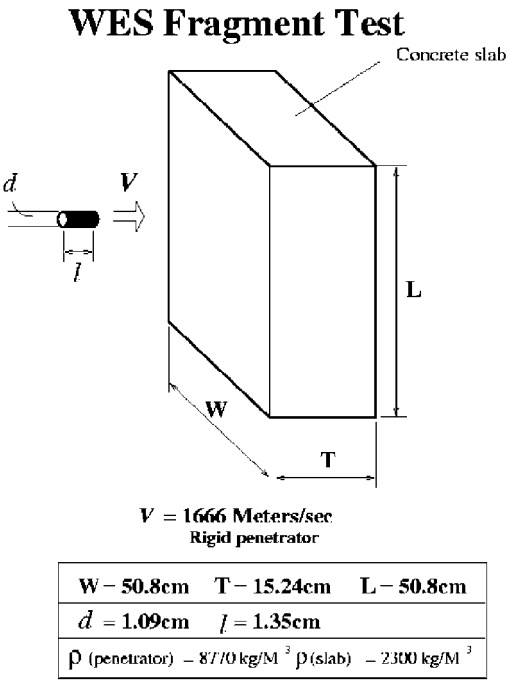

Figure 10. Problem statement for penetration simulation.

same polynomial completeness of the interior elements. We hope to verify this hypothesis shortly.

5.3. 3D penetration simulation

The dynamic, explicit, large deformation version of MPFEM has also been developed for three dimensions. To test it in a real-world engineering application, it has been applied to a high-speed penetration simulation. The computation has been compared with experimental data obtained by the Army Water Station. The problem statement is given in Figure 10. Two cases have been studied: a single projectile case and a multiple projectile case. In both cases the projectile has an initial velocity more than Mach 5. The target is made of rebar reinforced concrete. A multi-scale damage constitutive law [38] is used in the simulation. Plates 1 and 2 show damage contours for the single and multiple penetrator cases, respectively. Figure 11 shows the result of the computed penetrator depth compared with the experimental data. Note in Plates 1 and 2 that the particles that have been displaced by the penetrator have been squashed, representing the degree of damage in the material element lumped at these particles. The mass of the material is thus preserved, and no artificial erosion algorithm is necessary.

The 3D equivalent to the 2D MPFEM5 element has seven nodes, and the formulation is identical to that given in Section 4, except that the reproducing conditions must be applied in the z-direction as well. Since the formulation is similar to the 2D case, the detailed algebra is not shown here.

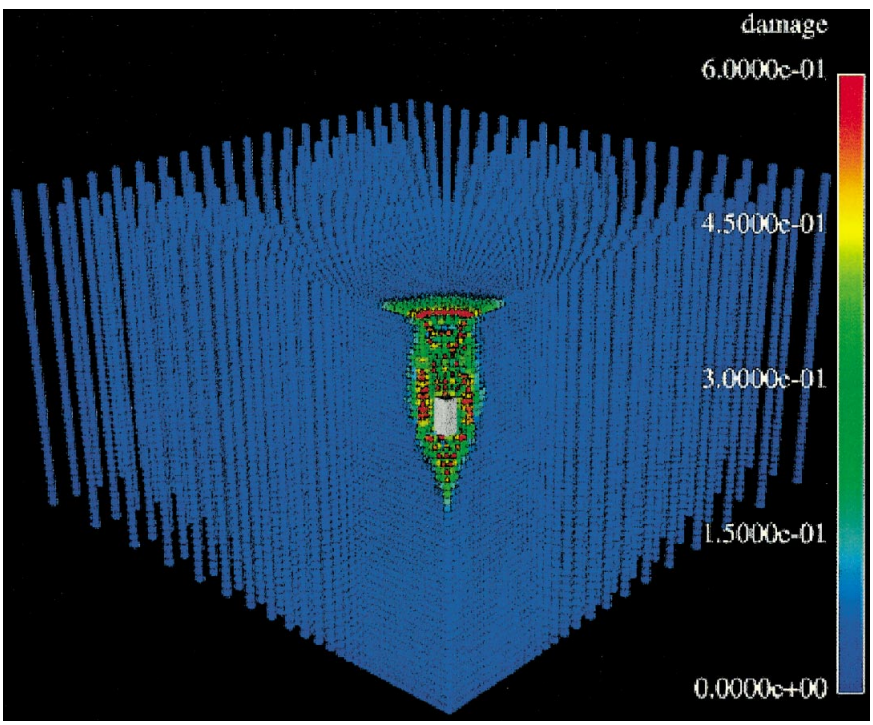

Plate 1. Damage contours for single penetrator case.

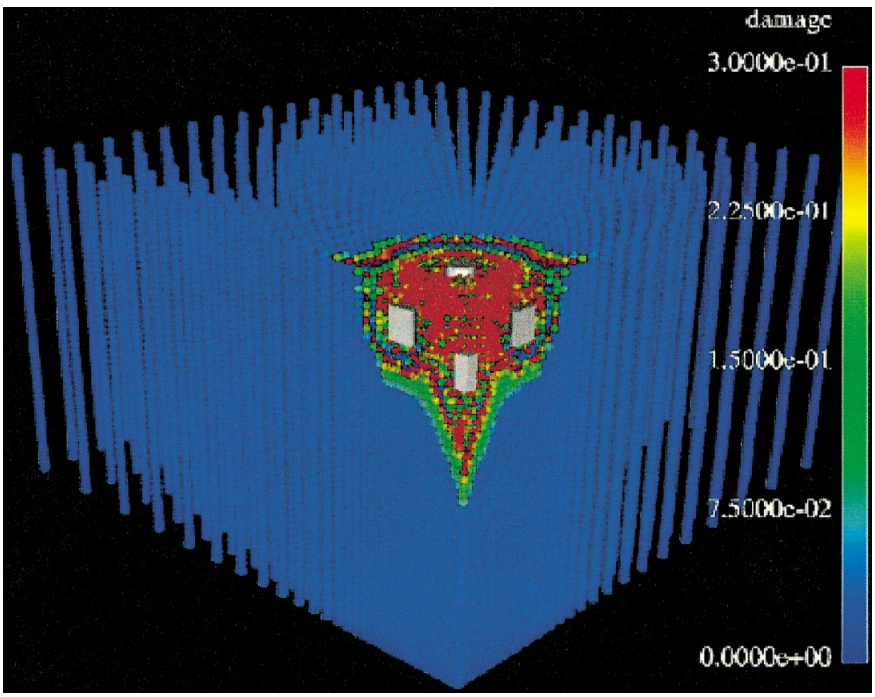

Plate 2. Damage contours for multiple penetrator simulation.

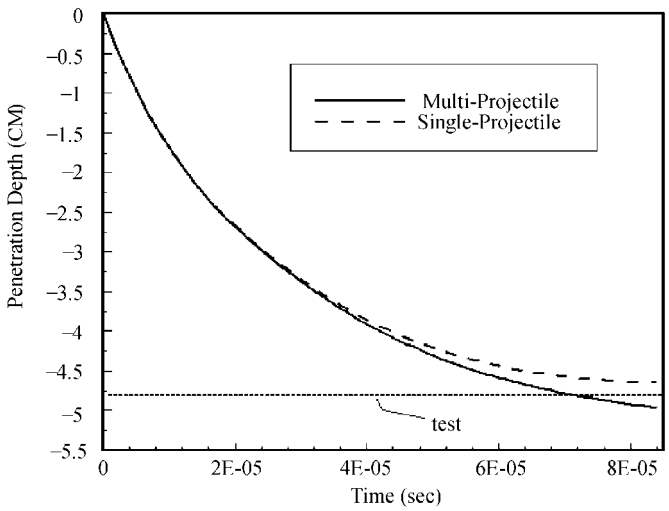

Figure 11. Comparison of MPFEM simulation to experimental result.

6. SUMMARY AND CONCLUSIONS

A new particle method that combines the salient features of both finite elements and meshfree methods is presented. The computational expense of the method is estimated to be considerably less than traditional meshfree methods, and on the same order as the FEM. The shape functions at each point are constructed by using neighbouring particles that form element-like configurations that satisfy certain desired reproducing conditions. However, unlike FEM shape functions, the MPFEM shape functions are not used to interpolate values within the domain of the problem, and are instead constructed at discrete points in the domain.

The method was tested in a benchmark beam bending problem, and was shown to give superior performance than the standard FEM4 element for both the compressible and incompressible cases. For the incompressible case, the MPFEM was shown to alleviate locking, and maintained full rates of convergence in displacement as the mesh was refined. Finally, the method was tested in a 3D large deformation problem, and was shown to accurately simulate a penetration problem.

APPENDIX A: MPFEM—A REPRODUCING KERNEL DERIVATION

Let us generalize the reproducing conditions by deriving them from RKPM [39]. In RKPM, a function $\mathbf{u}^h(\mathbf{x})$ is approximated via the convolution expression

$$\mathbf{u}^h(\mathbf{x}) = \int_\Omega \mathbf{C}(\mathbf{x}-\mathbf{x}')\mathbf{\Phi}(\mathbf{x}-\mathbf{x}')\mathbf{u}(\mathbf{x}')\,\mathrm{d}\mathbf{x}' \tag{A1}$$

where $\mathbf{C}(\mathbf{x}-\mathbf{x}')$ is the correction function from RKPM, and is determined by the product of a polynomial vector $\mathbf{P}(\mathbf{x}-\mathbf{x}')$ and a coefficient vector $\mathbf{b}(\mathbf{x})$. $\mathbf{\Phi}(\mathbf{x}-\mathbf{x}')$ is the window function.

To solve for $\mathbf{b}(\mathbf{x})$, the convolution expression is utilized to satisfy any order of completeness desired. The general equation for this becomes

$$\mathbf{M}(\mathbf{x})\mathbf{b}(\mathbf{x}) = \mathbf{P}(0) \tag{A2}$$

where

$$\mathbf{P}^{\mathrm{T}}(0) = [1, 0, 0, \ldots, 0] \tag{A3}$$

and

$$\mathbf{M}(\mathbf{x}) = \int_{\Omega} \mathbf{P}(\mathbf{x} - \mathbf{x}')\mathbf{\Phi}(\mathbf{x} - \mathbf{x}')\mathbf{P}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\,\mathrm{d}\Omega \tag{A4}$$

The previous expression is called the moment equation, and solving it for $\mathbf{b}(\mathbf{x})$ gives the necessary coefficient vector. Here, instead of solving for the moment matrix $\mathbf{M}$ as above, it is instead decomposed as

$$\mathbf{M}(\mathbf{x}) = \int_{\Omega} \mathbf{W}(\mathbf{x} - \mathbf{x}')\mathbf{\Phi}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\,\mathrm{d}\Omega \tag{A5}$$

where in 2D using three neighbours, and thus satisfying linear completeness,

$$\mathbf{W}(\mathbf{x} - \mathbf{x}') = \begin{bmatrix} 1 & 1 & 1 \\ x - x_1 & x - x_2 & x - x_3 \\ y - y_1 & y - y_2 & y - y_3 \end{bmatrix} \tag{A6}$$

Discretizing the moment matrix, the new moment equation becomes

$$\mathbf{W}(\mathbf{x} - \mathbf{x}')\mathbf{H}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{b}(\mathbf{x})\Delta V = \mathbf{P}(0) \tag{A7}$$

where

$$\mathbf{H}(\mathbf{x} - \mathbf{x}') = \begin{bmatrix} \mathbf{\Phi}(\mathbf{x}_1 - \mathbf{x}') & 0 & 0 \\ 0 & \mathbf{\Phi}(\mathbf{x}_2 - \mathbf{x}') & 0 \\ 0 & 0 & \mathbf{\Phi}(\mathbf{x}_3 - \mathbf{x}') \end{bmatrix} \tag{A8}$$

Solving for $\mathbf{b}(\mathbf{x})$ yields

$$\mathbf{b}(\mathbf{x}) = \mathbf{W}^{-\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{-1}(\mathbf{x} - \mathbf{x}')\mathbf{P}(0)\Delta V^{-1} \tag{A9}$$

The RKPM shape functions can be expressed as

$$N_I(\mathbf{x}) = \mathbf{H}(\mathbf{x} - \mathbf{x}')\mathbf{P}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{b}(\mathbf{x})\Delta V_I \tag{A10}$$

Rewriting this expression in terms of the new variables $\mathbf{W}(\mathbf{x} - \mathbf{x}')$ and $\mathbf{H}(\mathbf{x} - \mathbf{x}')$, the new expression for the shape functions becomes

$$N_I(\mathbf{x}) = \mathbf{H}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{b}(\mathbf{x})\Delta V_I \tag{A11}$$

Substituting in (54), one obtains

$$N_I(\mathbf{x}) = \mathbf{H}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{-\mathrm{T}}(\mathbf{x} - \mathbf{x}')\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}')\mathbf{W}^{-1}(\mathbf{x} - \mathbf{x}')\mathbf{P}(0)\Delta V\Delta V^{-1} \tag{A12}$$

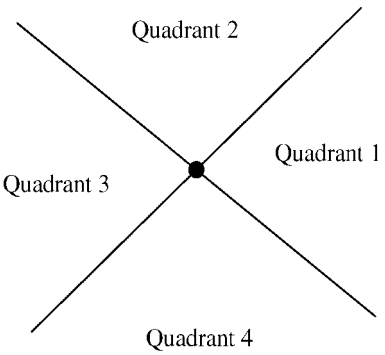

Figure A1. The concepts of quadrants for choosing the neighbour particles for the MPFEM5 element.

Cancelling terms, one obtains the final expression for the MPFEM shape functions

$$N_I(\mathbf{x}) = \mathbf{W}^{-1}(\mathbf{x} - \mathbf{y})\mathbf{P}(0) \tag{A13}$$

APPENDIX B: CHOOSING NEIGHBOURS FOR THE MPFEM5 ELEMENT

To determine which particles will form the basis for the MPFEM5 element, a simple searching algorithm is utilized. First, the domain surrounding each particle is split up into quadrants, as illustrated in Figure A1, with each quadrant encompassing a 90° range of space. Next, all particles within a reasonable range are found within each quadrant. The points chosen to be nodes in the MPFEM5 element are dictated by the principle of closest distance. For quadrants 1 and 3, the point chosen is the one closest in the x-direction. For quadrants 2 and 4, the point chosen is the one closest in the y-direction. After one particle has been found from each quadrant, the MPFEM5 element is known, and the method continues with the next particle.

ACKNOWLEDGEMENTS

The support of the Army Research Office (ARO) and the National Science Foundation (NSF) is gratefully acknowledged.

REFERENCES

1. Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering* 1995; **20**:1081–1106.
2. 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**:1655–1680.
3. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
4. Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:49–74.
5. Belytschko T, Krongauz K, Fleming M, Organ D, Liu WK. Smoothing and accelerated computations in the element-free Galerkin method. *Journal of Computational and Applied Mathematics* 1996; **74**:111–126.
6. Gingold RA, Monaghan JJ. Smooth particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375–389.
7. Randles P. Libersky L. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:375–408.

8. Swegle JW, Hicks DL, Attaway SW. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics* 1995; **116**:123–134.
9. Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* 1996; **39**:289–314.
10. Babuška I, Melenk JM. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
11. Strouboulis T, Babuška I, Copps K. The design and analysis of the generalized finite element method. *International Journal for Numerical Methods in Engineering* 2000; **181**:43–69.
12. Oden JT, Duarte CAM, Zienkiewicz OC. A new cloud-based hp finite element method. *Computer Methods in Applied Mechanics and Engineering* 1998; **153**:117–126.
13. Yagawa G, Furukawa T. Recent developments of free mesh method. *International Journal for Numerical Methods in Engineering* 2000; **47**:1419–1443.
14. Furukawa T, Yang C, Yagawa G, Chang-Chun W. Quadrilateral approaches for accurate free mesh method. *International Journal for Numerical Methods in Engineering* 2000; **47**:1445–1462.
15. Chen JS, Pan C, Wu CT, Liu WK. Reproducing kernel particle methods for large deformation analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:195–227.
16. Chen JS, Pan C, Roque CMOL, Wang HP. A lagrangian reproducing kernel particle method for metal forming analysis. *Computational Mechanics* 1998; **22**:289–307.
17. Li S, Liu WK. Numerical simulations of strain localization in inelastic solids using mesh-free methods. *International Journal for Numerical Methods in Engineering* 2000; **48**:1285–1309.
18. Li S, Liu WK. Reproducing kernel hierarchical partition of unity, part II—applications. *International Journal for Numerical Methods in Engineering* 1999; **45**:289–317.
19. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**:131–150.
20. Belytschko T, Moës N, Usui S, Parimi C. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering* 2001; **50**:993–1013.
21. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical methods in Engineering* 1999; **45**:601–620.
22. Dolbow J. An extended finite element method with discontinuous enrichment for applied mechanics. *PhD Thesis*, Theoretical and applied mechanics, Northwestern University, 1999.
23. Bonet J, Kulasegaram S. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering* 2000; **47**:1189–1214.
24. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (mlpg) approach to nonlinear problems in computer modeling and simulation. *Computer Modeling and Simulation in Engineering* 1998; **3**:187–196.
25. Atluri SN, Zhu T. The meshless local Petrov–Galerkin (mlpg) approach for solving problems in elasto-statics. *Computational Mechanics* 2000; **25**:169–179.
26. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:435–466.
27. Huerta A, Fernandez-Mendez S. Enrichment and coupling of the finite element and meshfree methods. *International Journal for Numerical Methods in Engineering* 2000; **48**:1615–1636.
28. Liu WK, Belytschko T, Oden JT (eds). *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:1–2.
29. Liu WK, Idelsohn SR, Oñate JT (eds). *International Journal for Numerical Methods in Engineering* 2000; **49**:721–723.
30. Chen JS, Liu WK (eds). *Computational Mechanics* 2000; **25**:99–316.
31. Wagner GJ, Liu WK. Application of essential boundary conditions in mesh-free methods: a corrected collocation method. *International Journal for Numerical Methods in Engineering* 2000; **47**:1367–1379.
32. Wagner GJ, Liu WK. Hierarchical enrichment for bridging scales and mesh-free boundary conditions. *International Journal for Numerical Methods in Engineering* 2001; **50**:507–524.
33. Sukumar N, Moran B, Belytschko T. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering* 1998; **43**:839–887.
34. Sukumar N, Moran B, Semenov A, Belikov VV. Natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:1–27.
35. Hughes TJR. *The Finite Element Method*. Prentice-Hall: Engelwood Cliffs, NJ, 1987.
36. Simo JC, Hughes TJR. On the variational foundations of assumed strain methods. *Journal of Applied Mechanics* 1986; **53**:51–54.
37. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: New York, 2000.
38. Hao S, Liu WK, Klein P, Rosakis AJ. Multi-scale damage model. 2001, manuscript to be submitted.
39. Liu WK, Chen Y. Wavelet and multiple scale reproducing kernel methods. *International Journal for Numerical Methods in Fluids* 1995; **21**:901–931.

