

A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations

A. Sadeghirad, R. M. Brannon^{*,†} and J. Burghardt

Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, U.S.A.

SUMMARY

A new algorithm is developed to improve the accuracy and efficiency of the material point method for problems involving extremely large tensile deformations and rotations. In the proposed procedure, particle domains are convected with the material motion more accurately than in the generalized interpolation material point method. This feature is crucial to eliminate instability in extension, which is a common shortcoming of most particle methods. Also, a novel alternative set of grid basis functions is proposed for efficiently calculating nodal force and consistent mass integrals on the grid. Specifically, by taking advantage of initially parallelogram-shaped particle domains, and treating the deformation gradient as constant over the particle domain, the convected particle domain is a reshaped parallelogram in the deformed configuration. Accordingly, an alternative grid basis function over the particle domain is constructed by a standard 4-node finite element interpolation on the parallelogram. Effectiveness of the proposed modifications is demonstrated using several large deformation solid mechanics problems. Copyright © 2011 John Wiley & Sons, Ltd.

Received 24 February 2010; Revised 1 October 2010; Accepted 10 November 2010

KEY WORDS: material point method; large deformations; extension instability; nodal integration; verification; particle methods

1. INTRODUCTION

The material point method (MPM) [1, 2] is a particle-grid method suitable for solving large deformations of complicated geometries in which the constitutive model is history-dependent and therefore intolerant of advection errors. A traditional MPM implementation treats each particle as a lumped mass. The generalized interpolation material point (GIMP) method [3] is a generalization of the MPM that accounts for finite spatial extent occupied by each particle. MPM and GIMP have been successfully used in simulation of a range of complicated engineering problems including finite deformation plasticity, thin membranes, cracks and fracture, sea ice dynamics, granular materials, multiscale problems, hypervelocity impact, and dynamic analysis of saturated porous media among others [4–16]. Also, their basic algorithms and formulations have been studied by a number of authors such as [17–29]. In MPM, the material domain is represented by a set of material points or particles and uses a mixed Eulerian–Lagrangian method in which Lagrangian particles carry history-dependent state data and an Eulerian background grid is used for calculation of derivatives and solving the momentum equation. In general, this background mesh is fixed in space and does not experience any distortion. The Lagrangian particles are followed throughout the deformation, and there is no need to map the history data from one point to another. Also, the

*Correspondence to: R. M. Brannon, 2134 MEB, 50 S. Central Campus Dr., Salt Lake City, UT 84112, U.S.A.

†E-mail: Rebecca.Brannon@utah.edu

MPM formulation does not include nonlinear convection terms as appear in Eulerian formulations because the background grid is considered as an updated reference. In summary, the MPM has the advantages of both Eulerian and Lagrangian formulations. Another interesting feature of traditional MPM is that this method enforces no-slip contact between bodies automatically without any further computational costs. However, the MPM suffers from a ‘cell crossing instability’ for problems involving large displacements. This instability occurs whenever particles cross boundaries of any cell in the computational background grid and is due to the lack of smoothness of the grid basis functions used as the interpolation and mapping functions in the MPM. By introducing weighting functions and gradient weighting functions with a higher degree of smoothness, the GIMP is capable of reducing these errors and improving accuracy [3]. These functions, which can be interpreted as averages of the basis function and its gradient over the particle domain, are used for mapping and interpolating data between particles and grid nodes as well as calculating internal and external forces.

GIMP weighting functions are defined by the convolution of a characteristic function and the grid basis function. The gradient weighting function is similarly defined by the convolution of the characteristic function and the gradient of the grid basis function. Characteristic functions are defined for each particle. When the choice of characteristic function is the Dirac delta function, the traditional MPM formulation is recovered exactly. A commonly used characteristic function is the ‘top-hat’ function which is equal to unity within the particle domain, and zero elsewhere. One may also choose fuzzy particle characteristic functions whose support is more extended [3]; however, these have been used only for one-dimensional problems and are not common. Calculation of the weighting and gradient weighting functions requires integration over the current support of the particle characteristic functions. Accurately tracking particle domains is essential for accurate calculation of these functions during finite deformation problems in which particle domains deform and rotate significantly.

In practice, particle domains are often taken to be initially rectangles in 2D. In the literature, two strategies for tracking the particle domains have been proposed. In the first, the particle domains are assumed to be unchanged for all time (denoted as uGIMP here). In the second, particle domain sizes evolve with time such that they continue to be rectangular or cuboid (denoted as cpGIMP here). In cpGIMP, the particle size in each direction is determined from the corresponding diagonal components of the deformation gradient tensor at the particle. Accordingly, the cpGIMP is limited to problems for which deformation is along the grid directions so that off-diagonal deformation gradient components are negligible. Another method for changing particle size has been proposed and is described in [21]. This method is not investigated here because it likewise has the shortcoming of neglecting shear distortion and has some challenging issues reported in [23] even in comparison to cpGIMP.

For two-dimensional deformations, these available methods track particle domains as rectangles that are always aligned with the grid. This assumption is not accurate for analyzing problems with shear deformations or large rotations. Instead initially rectangular particle domains become parallelograms under shear deformations, and they become misaligned with the grid under rotation. In this case, exactly evaluating weighting and gradient weighting integrals entails very high computational cost because of the need to divide the integration domain (i.e. the parallelograms) over cell boundaries. In this paper, a novel alternative to standard grid basis functions is proposed for efficiently calculating weighting function and gradient weighting function integrals. By using alternative grid basis functions, division of particle domains along cell boundaries is avoided. This work is called the convected particle domain interpolation (CPDI) method because it convects initially rectangular particle domains into parallelograms, and it constructs an alternative basis function field over the particle that exactly interpolates the grid basis function at the parallelogram’s corners using the standard 4-node finite element interpolation. The proposed method can be naturally generalized to higher order interpolation, as well as three-dimensional problems; for clarity, however, only two-dimensional case studies are described in this paper. The proposed technique will be shown to eliminate instabilities observed in uGIMP and cpGIMP without significantly increasing computational cost.

The proposed CPDI method uses standard finite element (FE) shape functions of 4-node quadrilateral (Q4) elements associated with the corners of any particle domain for constructing alternative grid basis functions over that particle domain. In this method, alternative grid basis functions are interpolated based on the values of grid basis functions at the corners of particle domains. Numerical investigations have shown that this algorithm drastically improves the accuracy and stability of the GIMP.

The organization of this paper is as follows: a brief review of the MPM and GIMP is presented in Section 2. Section 3 is dedicated to describing the new CPDI method and contrasting it with GIMP. Numerical examples of large deformation solid mechanics are given in Section 4, followed by concluding remarks.

2. REVIEW OF THE MPM AND GIMP METHODS

In the MPM and GIMP method, the problem domain is discretized by a finite number of particles that carry material data and deformation history. In contrast with most other particle methods in which the momentum equations are solved on the particles, the momentum equations are solved on the predefined background grid in the MPM. Since this grid carries no permanent material information, the grid could be reconstructed each time step after deformation for convenience in the computation. Often, the same fixed regular grid is used in all time steps. The solution procedure of MPM consists of three major phases for any time step. In the first phase, material data and external forces are mapped from the particles to the grid. The lumped mass matrix and nodal force array are formed at the grid nodes. During the second phase, nodal accelerations are determined by solving the discretized momentum equations as in the standard finite element method (FEM). Finally at the current time step, the location, velocity, and deformation gradient of the particles are updated based on the kinematic solution on the grid nodes. At the end of time step, the deformed grid is discarded and the same regular grid is used for the next time step to avoid mesh distortion. In other words, large deformations are modeled by moving the particles through the grid. Because field variables are saved at particles, MPM has the advantages of both Eulerian and Lagrangian descriptions.

The algorithm described here is the ‘update stress last (USL)’ version of the GIMP algorithm described by Bardenhagen [17]. Using the cpGIMP, Wallstedt and Guilkey [23] demonstrated that the USL scheme performs dramatically better than an alternative ‘update stress first (USF)’ scheme.

2.1. Equations common to all versions of the MPM

Throughout this paper, the subscript i refers to the grid nodes, p refers to the particles, and α refers to the particle domain corners. At the first of each time step, particle mass, and momentum are mapped to the grid nodes using equations of the form

$$m_i = \sum_p \varphi_{ip} m_p \quad (1)$$

$$\mathbf{v}_i = \frac{\sum_p \varphi_{ip} m_p \mathbf{v}_p}{m_i} \quad (2)$$

where m_i and \mathbf{v}_i are the mass and velocity at grid node i , m_p and \mathbf{v}_p are the mass and velocity at particle p , and φ_{ip} are the components of a mapping matrix that couples grid values to particle values. We refer to φ_{ip} as the effective grid basis function at particle p . As discussed in the following section, the scheme used to assign values to φ_{ip} distinguishes one MPM method from the another.

In the traditional FEM, the nodal force on the grid is defined as the following integral over the entire domain Ω :

$$\mathbf{f}_i^{\text{int}} = - \int_{\Omega} \nabla \varphi_i \cdot \boldsymbol{\sigma} \, dV \quad (3)$$

where φ_i is the basis function associated with grid node i . This integral is then broken into the sum of integrals over elements as

$$\mathbf{f}_i^{\text{int}} = - \sum_e \int_{\Omega_e} \nabla \varphi_i \cdot \boldsymbol{\sigma} \, dV \quad (4)$$

The MPM can be interpreted as instead breaking the integral in Equation (3) into a sum of integrals over nonoverlapping particle domains

$$\mathbf{f}_i^{\text{int}} = - \sum_p \int_{\Omega_p} \nabla \varphi_i \cdot \boldsymbol{\sigma} \, dV \quad (5)$$

Particle domains are often small and the variation of stress over them is negligible in comparison with the variation of basis function gradients, especially when the particle spans an element boundary where there can be a jump discontinuity in the basis function gradient. Therefore, the stress over the particle domain is approximately constant. Treating the stress field as a constant $\boldsymbol{\sigma}_p$ over each particle domain

$$\mathbf{f}_i^{\text{int}} = - \sum_p \int_{\Omega_p} \nabla \varphi_i \, dV \cdot \boldsymbol{\sigma}_p \quad (6)$$

Defining an effective basis function gradient over the particle as

$$\nabla \varphi_{ip} = \frac{1}{V_p} \int_{\Omega_p} \nabla \varphi_i \, dV \quad (7)$$

the nodal internal forces can be calculated

$$\mathbf{f}_i^{\text{int}} = - \sum_p \nabla \varphi_{ip} \cdot \boldsymbol{\sigma}_p V_p \quad (8)$$

Similarly, the φ_{ip} in Equations (1) and (2) can be regarded as the average of the basis function over the particle domain.

Internal forces are combined with any external force, $\mathbf{f}_i^{\text{ext}}$, including body forces and tractions, to calculate grid node accelerations as

$$\mathbf{a}_i = \frac{\mathbf{f}_i^{\text{int}} + \mathbf{f}_i^{\text{ext}}}{m_i} \quad (9)$$

The explicit forward-Euler time discretization is used to update the grid node velocities

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{a}_i \Delta t \quad (10)$$

These updated grid node velocities are used for calculating velocity gradients at the particle positions

$$\nabla \mathbf{v}_p^{n+1} = \sum_i \nabla \varphi_{ip} \mathbf{v}_i^{n+1} \quad (11)$$

Finally, time-dependent particle data such as position, velocity, volume, and deformation gradient are updated according to

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \sum_i \varphi_{ip} \mathbf{a}_i \Delta t \quad (12)$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \sum_i \varphi_{ip} \mathbf{v}_i^{n+1} \Delta t \quad (13)$$

$$\mathbf{F}_p^{n+1} = (\mathbf{I} + \nabla \mathbf{v}_p^{n+1} \Delta t) \mathbf{F}_p^n \quad (14)$$

$$V_p^{n+1} = J V_p^0 \quad (15)$$

in which J is the determinant of the deformation gradient of the particle, $J = |\mathbf{F}_p^{n+1}|$.

Following this update of the kinematic state, the constitutive model is called to then update the particle stresses to the end of the time step.

2.2. Existing material point methods

Various implementations of MPM are distinguished primarily by how they evaluate φ_{ip} and $\nabla \varphi_{ip}$. In the original MPM, sometimes called ‘standard MPM’, the density field was approximated as a collection of point masses using Dirac delta functions. The resulting equations are equivalent to using single-point integration in Equation (7).

In the GIMP method, the effective averages of the basis function and its gradient over the particle domain were generalized by Bardenhagen and Kober [3] as

$$\varphi_{ip} \equiv \overline{S_{ip}} = \frac{1}{V_p} \int_{\Omega_\chi} \chi_p(\mathbf{x} - \mathbf{x}_p) S_i(\mathbf{x}) \, d\mathbf{x} \quad (16)$$

$$\nabla \varphi_{ip} \equiv \overline{\nabla S_{ip}} = \frac{1}{V_p} \int_{\Omega_\chi} \chi_p(\mathbf{x} - \mathbf{x}_p) \nabla S_i(\mathbf{x}) \, d\mathbf{x} \quad (17)$$

where $\chi_p(\mathbf{x})$ is the particle characteristic function and Ω_χ denotes the current support of this function. Typically, a ‘top-hat’ characteristic function is used in the GIMP:

$$\chi_p(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_p \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Noting Equations (16) and (17), it is obvious that when the choice of characteristic function is the Dirac delta function, the standard MPM formulation is recovered exactly.

Two strategies, uGIMP and cpGIMP, have been used to date to track evolving particle domains under large deformations. In the uGIMP, deformation of the particle domain is neglected. Specifically, referring to the physical dimensions labeled in Figure 1.

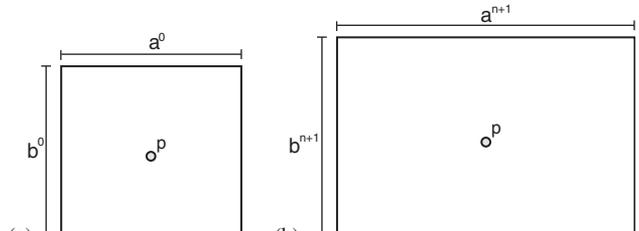
$$\begin{cases} a^{n+1} \\ b^{n+1} \end{cases} = \begin{cases} a^0 \\ b^0 \end{cases} \quad (19)$$


Figure 1. (a) Initial and (b) updated particle domain during the analysis with uGIMP, Equation (19) and cpGIMP, Equation (20).

In the cpGIMP, the particle sizes evolve with deformation such that the particle domain remains rectangular. Specifically, in terms of the diagonal components of the deformation gradient at the particle

$$\begin{Bmatrix} a^{n+1} \\ b^{n+1} \end{Bmatrix} = \begin{bmatrix} F_{xx}^{n+1} & 0 \\ 0 & F_{yy}^{n+1} \end{bmatrix} \begin{Bmatrix} a^0 \\ b^0 \end{Bmatrix} \quad (20)$$

In summary, three methods have been used to date to evaluate the effective basis function and its gradient: a Dirac delta function (or, single-point integration) in the standard MPM, a top-hat average over the stationary domain in the uGIMP method, and a top-hat average over an evolving rectangular domain in the cpGIMP method. Each of these approaches has shortcomings. The standard MPM is subject to a so-called ringing instability caused by a sudden jump in basis function gradient as a particle crosses a cell boundary. The uGIMP suffers from the extension instability when the particles become separated to the extent that their regions of influence no longer overlap. The cpGIMP rectifies the extension instability if the direction of loading is aligned with the rectangular grid. However, the results are not indifferent to a frame change.

3. DESCRIPTION OF THE CPDI METHOD

When using a top-hat function in Equations (16) and (17), the strategy used for tracking particle domains directly affect the results. The particle domains are tracked as rectangles with the uGIMP and cpGIMP algorithms. However, under shear deformations the particle domains should become parallelograms and under rigid body rotations they ought to become rotated rectangles. When the particle domains are tracked as parallelograms, the GIMP needs complicated division of the particle domain along cell boundaries and use of a numerical integration scheme for evaluating φ_{ip} and $\nabla\varphi_{ip}$. In this case, the GIMP algorithm can be applied with considerable computational costs.

In the CPDI method, a novel alternative to standard grid basis functions is proposed for efficiently calculating integrals Equations (16) and (17) when the particle domains are parallelograms. By using these alternative grid basis functions, dividing particle domains along cell boundaries is avoided. Consequently, the CPDI method is capable of more accurately tracking particle domains with a computational cost comparable with original GIMP.

3.1. Strategy for tracking particle domains as parallelograms

By using the fully updated deformation gradient at each particle, all deformations and rotations computed at the particle are used for determining the updated particle domain at any time step. The deformation gradient is typically already available since constitutive models often need it. Assuming that the particle domains are parallelograms and that the deformation gradient is approximately constant over the particle domain, the particle domains deform according to

$$\begin{aligned} \mathbf{r}_1^{n+1} &= \mathbf{F}_p^{n+1} \mathbf{r}_1^0 \\ \mathbf{r}_2^{n+1} &= \mathbf{F}_p^{n+1} \mathbf{r}_2^0 \end{aligned} \quad (21)$$

in which $(\mathbf{r}_1^0, \mathbf{r}_2^0)$ and $(\mathbf{r}_1^{n+1}, \mathbf{r}_2^{n+1})$ are the vectors defining particle domains at the initial and updated configuration, respectively, as shown in Figure 2.

3.2. Proposed alternative evaluation of φ_{ip} and $\nabla\varphi_{ip}$

An alternative to the standard grid basis functions is proposed which eliminates the requirement of dividing of particle domains along cell boundaries in calculation of the integrals in Equations (16)

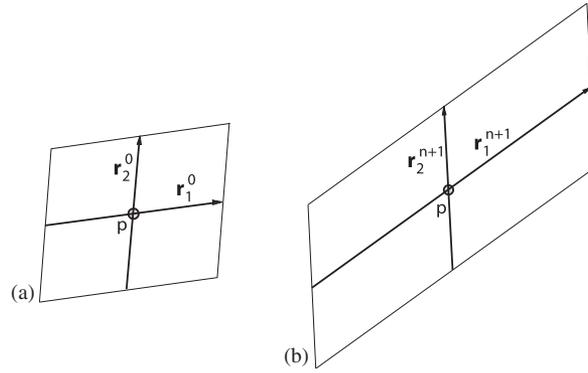


Figure 2. (a) Initial and (b) updated particle domain during the analysis with CPDI method.

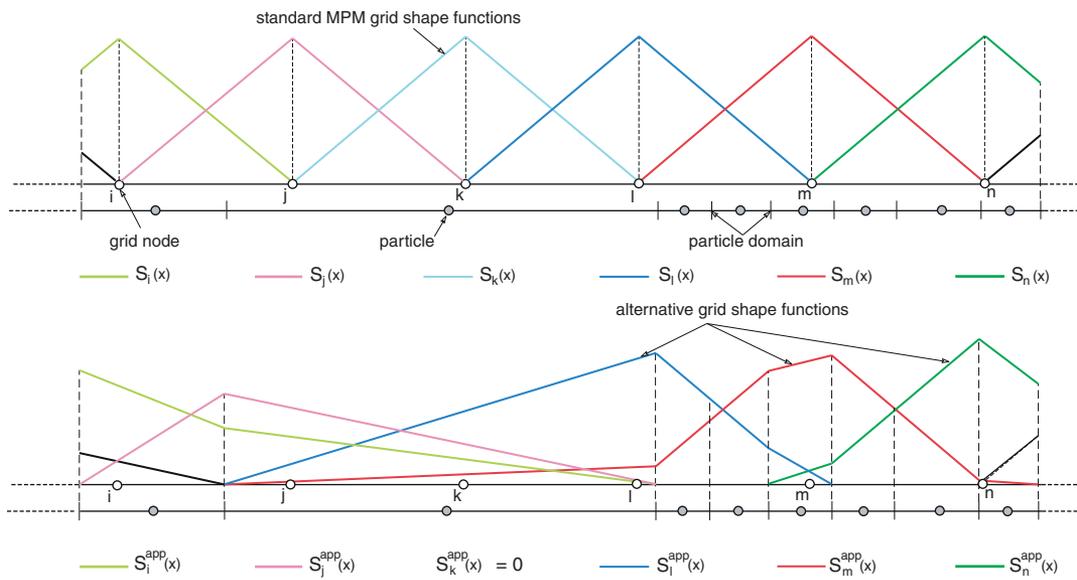


Figure 3. Alternative grid basis functions in a simple one-dimensional case.

and (17). Here, the alternative grid basis functions are constructed to be an interpolation of standard grid basis functions at the four corners of each particle domain

$$S_i^{app}(\mathbf{x}) = \sum_{\alpha=1}^4 Q_{\alpha}^p(\mathbf{x}) S_i(\mathbf{x}_{\alpha}^p) \quad \text{on } \Omega_p \tag{22}$$

in which Q_{α}^p is the standard FE 4-node (Q4) shape function related to the α th corner of the domain corresponding to particle p and \mathbf{x}_{α}^p is the position of this corner. The original and alternative basis functions differ from each other in the interior of the particle domain. However, since the alternative basis function is an interpolation, it identically equals the exact basis function at the particle corners. This property makes the CPDI evaluation of nodal forces exact in 1D. The proof is a straightforward application of the fundamental theorem of calculus: the integral of a function gradient depends only on the function value at the boundaries. Thus, since the original and alternative basis functions are, by design, identical at the particle boundaries, both functions give identical results for the nodal force integrals.

These alternative grid basis functions are depicted in Figure 3 for a simple one-dimensional case. When the particles are far apart in massive tensile deformations, the particles in the CPDI method are connected together based on these new grid basis functions.

By using Equation (22), the top-hat versions of Equations (16) and (17) can be written as

$$\varphi_{ip} \equiv \overline{S_{ip}} \cong \frac{1}{V_p} \int_{\Omega_p} S_i^{\text{app}}(\mathbf{x}) \, d\mathbf{x} = \frac{1}{V_p} \sum_{\alpha=1}^4 \left(\int_{\Omega_p} Q_{\alpha}^p(\mathbf{x}) \, d\mathbf{x} \right) S_i(\mathbf{x}_{\alpha}^p) \quad (23)$$

$$\nabla \varphi_{ip} \equiv \overline{\nabla S_{ip}} \cong \frac{1}{V_p} \int_{\Omega_p} \nabla S_i^{\text{app}}(\mathbf{x}) \, d\mathbf{x} = \frac{1}{V_p} \sum_{\alpha=1}^4 \left(\int_{\Omega_p} \nabla Q_{\alpha}^p(\mathbf{x}) \, d\mathbf{x} \right) S_i(\mathbf{x}_{\alpha}^p) \quad (24)$$

These functions are analytically calculated as

$$\varphi_{ip} \cong \frac{1}{4} \{S_i(\mathbf{x}_1^p) + S_i(\mathbf{x}_2^p) + S_i(\mathbf{x}_3^p) + S_i(\mathbf{x}_4^p)\} \quad (25)$$

$$\nabla \varphi_{ip} \cong \frac{1}{2V_p} \left\{ (S_i(\mathbf{x}_1^p) - S_i(\mathbf{x}_3^p)) \begin{bmatrix} r_{1y}^n - r_{2y}^n \\ r_{2x}^n - r_{1x}^n \end{bmatrix} + (S_i(\mathbf{x}_2^p) - S_i(\mathbf{x}_4^p)) \begin{bmatrix} r_{1y}^n + r_{2y}^n \\ -r_{1x}^n - r_{2x}^n \end{bmatrix} \right\} \quad (26)$$

in which (r_{1x}^n, r_{1y}^n) and (r_{2x}^n, r_{2y}^n) are, respectively, the components of vectors \mathbf{r}_1^n and \mathbf{r}_2^n which define the particle domain at the current time step (Figure 2).

4. NUMERICAL EXAMPLES

Four nonlinear large deformation dynamic examples are considered here for investigating the accuracy and stability of the CPDI method. The first two examples were constructed based on the method of manufactured solutions (MMS) [23, 30]. In the MMS, the solution of the model equations is assumed *a priori*, then the external force required to achieve this solution is analytically determined. This allows verification of nonlinear codes or algorithms by running them with the computed external force and demonstrating that the assumed solution is recovered. Two examples constructed based on the MMS in this paper are similar to those that were constructed and described by Wallstedt and Guilkey [23] for studying the effects of different time integration schemes in the GIMP algorithm. The following error norm was adopted to investigate spatial convergence of these two examples:

$$\text{Err} = \frac{\sqrt{\sum_{i=1}^{N_t} \sum_{p=1}^{N_p} \|\mathbf{u}^{\text{app}}(\mathbf{x}_p, t_i) - \mathbf{u}^{\text{exact}}(\mathbf{x}_p, t_i)\|^2}}{N_t \times N_p} \quad (27)$$

in which N_t and N_p are total number of particles and time steps, respectively, $\mathbf{u}^{\text{app}}(\mathbf{x}_p, t_i)$ and $\mathbf{u}^{\text{exact}}(\mathbf{x}_p, t_i)$ are the calculated and analytical displacement vectors, respectively.

In the spatial convergence studies performed for these two examples, simulations are done with time steps as

$$\Delta t = 0.4 \frac{h}{C} \quad (28)$$

where h is the cell spacing and C is a constant defined by

$$C = \sqrt{\frac{E}{\rho_0}} \quad (29)$$

in which E and ρ_0 are the modulus of elasticity and initial density, respectively. C is the thin bar wave speed, not the uniaxial strain wave speed.

The last two examples present the performance of the CPDI method for the problems in which the standard MPM and GIMP method encounter instability. In all numerical examples, the Neo-Hookean material model was assumed

$$\boldsymbol{\sigma} = \frac{\lambda \ln J}{J} \mathbf{I} + \frac{\mu}{J} (\mathbf{F}\mathbf{F}^T - \mathbf{I}) \quad (30)$$

where \mathbf{I} is the identity tensor, \mathbf{F} is the deformation gradient, J is the determinant of \mathbf{F} , μ is the shear modulus, and λ is the Lamé constant. While many complicated material models have been successfully used in many implementation of the MPM, this simple and frame-indifferent material model is sufficient for investigating the proposed modifications.

4.1. Axis-aligned displacement in a unit square

The following displacement field is assumed in a unit square ($L \times L$, $L = 1$ m).

$$u_x = A \sin\left(\frac{2\pi X}{L}\right) \sin\left(\frac{C\pi t}{L}\right) \quad (31)$$

$$u_y = A \sin\left(\frac{2\pi Y}{L}\right) \sin\left(\frac{C\pi t}{L} + \pi\right) \quad (32)$$

in which X and Y are the components of the global coordinate system in the reference configuration, A is the maximum amplitude of displacement (in the numerical simulations, this parameter is assumed to be 0.05 m), t is the time, and C is the constant defined in Equation (29).

Using the MMS, the following body forces are found to be required to achieve the assumed displacement field (detailed calculations are presented in Appendix A):

$$b_x = \frac{\pi^2 u_x}{L^2} \left(\frac{4\mu}{\rho_0} - C^2 - 4 \frac{\lambda[\ln F_{xx} F_{yy} - 1] - \mu}{\rho_0 F_{xx}^2} \right) \quad (33)$$

$$b_y = \frac{\pi^2 u_y}{L^2} \left(\frac{4\mu}{\rho_0} - C^2 - 4 \frac{\lambda[\ln F_{xx} F_{yy} - 1] - \mu}{\rho_0 F_{yy}^2} \right) \quad (34)$$

in which F_{xx} and F_{yy} are the non-zero components of the deformation gradient tensor. It should be noted that, for the axis-aligned problem, only the diagonal terms of \mathbf{F} are non-zero

$$F_{xx} = 1 + \frac{2A\pi}{L} \cos\left(\frac{2\pi X}{L}\right) \sin\left(\frac{C\pi t}{L}\right) \quad (35)$$

$$F_{yy} = 1 + \frac{2A\pi}{L} \cos\left(\frac{2\pi Y}{L}\right) \sin\left(\frac{C\pi t}{L} + \pi\right) \quad (36)$$

The problem domain, a unit square, is discretized by using four different grid resolutions 8×8 , 16×16 , 32×32 , and 64×64 , with two particles per cell in each direction. The modulus of the elasticity and Poisson's ratio are $E = 1 \times 10^7$ Pa and $\nu = 0.3$, respectively. Also, initial density is assumed to be $\rho_0 = 1000$ kg/m³ and real-time simulation is $T = 0.02$ s, one full period of oscillation. Convergence curves for this example using the standard MPM, uGIMP, cpGIMP, and CPDI method are depicted in Figure 4. This example is not representative of general material deformations because displacements are axis aligned. In this case, both cpGIMP and the CPDI methods track the particle domains accurately as rectangles. Since convergence curves of the cpGIMP and CPDI methods approximately coincide, it can be concluded that the alternative grid basis functions proposed in the CPDI method can accurately evaluate φ_{ip} and $\nabla\varphi_{ip}$. Also, the uGIMP that does not track the particle domains leads to less accurate results in comparison with the cpGIMP and CPDI methods. Three snapshots of the CPDI results using 16×16 cells at three representative times are shown in Figure 5.

4.2. Radial expansion of a ring

In this example, the displacement field is prescribed with radial symmetry for a ring as

$$u(r, t) = A \sin\left(\frac{C\pi t}{2\bar{r}}\right) (c_3 r^3 + c_2 r^2 + c_1 r) \quad (37)$$

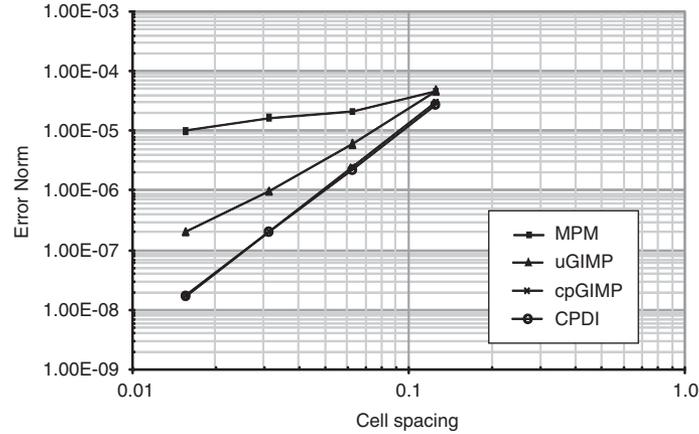


Figure 4. Convergence curves of the standard MPM, uGIMP, cpGIMP, and CPDI method for the axis-aligned displacement in a unit square problem. (Convergence curves of the cpGIMP and CPDI methods overlap, which is to be expected since this verification test involves no shear distortions from rectangles to parallelograms.)

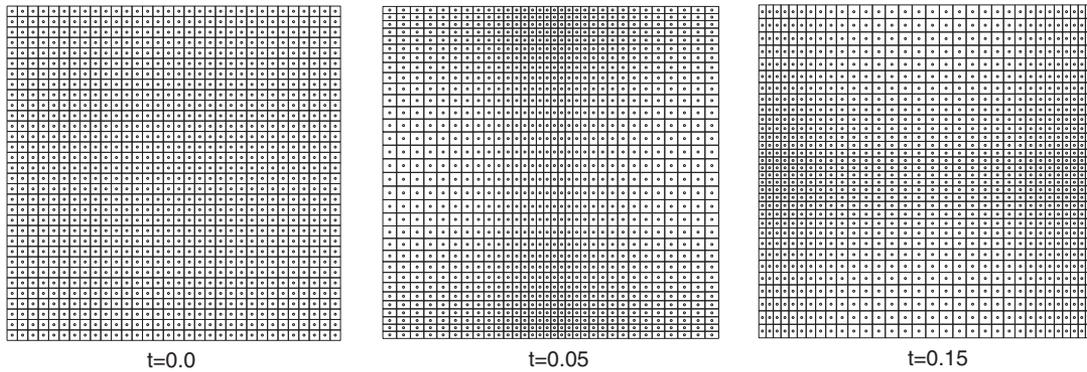


Figure 5. Three snapshots of the CPDI method simulation of the axis-aligned displacement in a unit square problem using 16×16 cells at three representative times.

where r (and θ) are cylindrical coordinates in the reference configuration, t is the time, C is the constant defined in Equation (29), A is the maximum magnitude of displacement, constants c_1 , c_2 , and c_3 are determined for the case in which normal stress vanishes on the inner (r_i) and outer (r_o) surfaces of the ring and $u(r_o, t) = A \sin(C\pi t/2\bar{r})$, and \bar{r} is the mean radius of the ring, $\bar{r} = (r_i + r_o)/2$.

$$c_1 = \frac{-6r_i}{r_o(r_o - 3r_i)}, \quad c_2 = \frac{3(r_o + r_i)}{r_o^2(r_o - 3r_i)}, \quad c_3 = \frac{-2}{r_o^2(r_o - 3r_i)} \quad (38)$$

The required calculations to obtain the necessary body forces are described in Appendix A. The following values are chosen for numerical simulations: modulus of elasticity $E = 1 \times 10^7$ Pa, Poisson's ratio $\nu = 0$, initial density $\rho_0 = 1000$ kg/m³, inner radius $r_i = 0.4$ m, outer radius $r_o = 0.6$ m, maximum magnitude of displacement $A = 0.1$ m, and real-time simulation $T = 0.02$ s which is one full period of oscillation. Exploiting symmetry, only one-quarter of the ring is modeled. The problem domain is discretized by using four different resolutions including 60, 248, 1001, and 4019 particles with two particles per cell in each direction as shown in Figure 6. Figure 7 shows convergence curves for this example by using the standard MPM, uGIMP, cpGIMP, and CPDI methods. As shown in Figure 7, the results obtained from the CPDI method are more accurate because the CPDI method tracks the particle domains more accurately than other methods do.

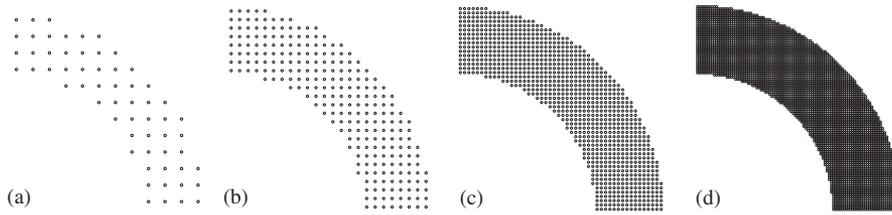


Figure 6. Initial configuration of four different resolutions used in modeling one-quarter of the ring.

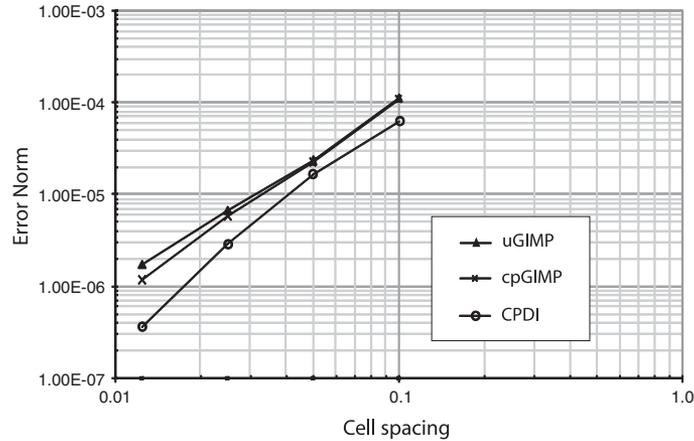


Figure 7. Convergence curves of the standard MPM, uGIMP, cpGIMP, and CPDI method for the radial expansion of a ring problem. The simulations using the standard MPM are unstable at all four resolutions.

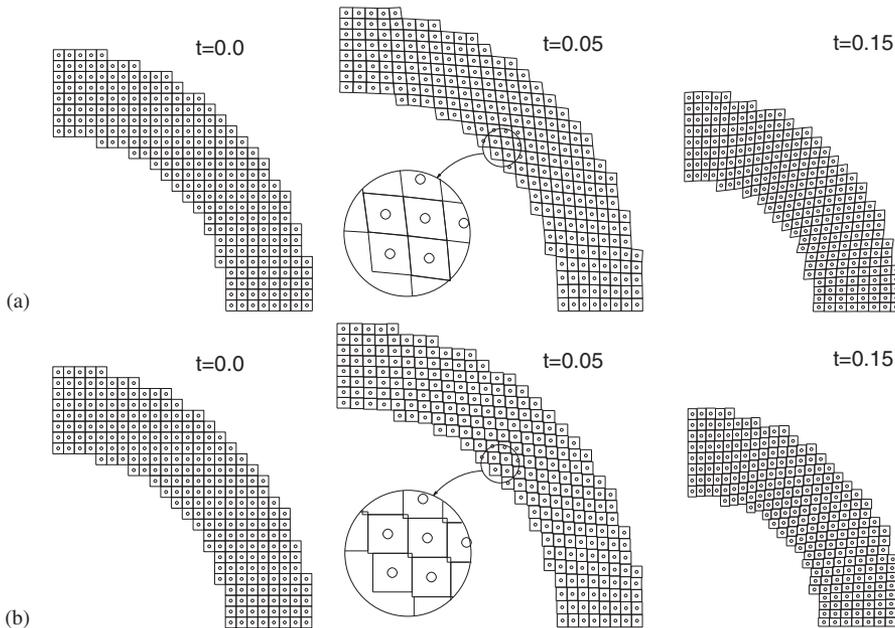


Figure 8. Three snapshots of the model with 248 particles in simulation of the radial expansion of a ring problem using: (a) CPDI method and (b) cpGIMP.

Comparison between tracking particle domains in the cpGIMP and CPDI method is shown in Figure 8. Three snapshots of the simulation with 248 particles using cpGIMP and CPDI method at three representative times are depicted in Figure 8.

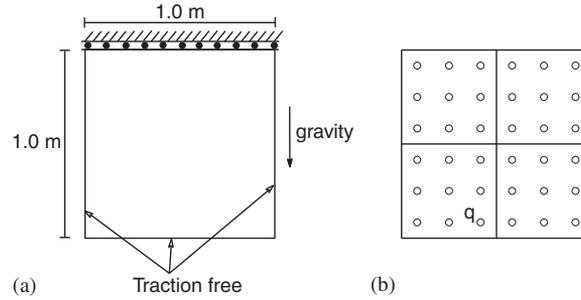


Figure 9. The vertical bar problem: (a) Geometry and (b) discretized model.

4.3. A vertical bar with large tensile deformation

Extension instability, which is a common shortcoming of many particle methods, is investigated. Vibration of a vertical bar under its own weight with very large deformations is considered in this example. The upper end of the bar has roller boundary conditions, and the other boundaries are traction free as shown in Figure 9. The modulus of elasticity, Poisson’s ratio, and initial density are $E = 1 \times 10^6 \text{ Pa}$, $\nu = 0.3$, and $\rho_0 = 1050 \text{ kg/m}^3$, respectively. Real-time simulation is $T = 0.24 \text{ s}$ and time steps are chosen as $\Delta t = 0.00006 \text{ s}$. In order to obtain very large deformations, gravity of magnitude $g = 1000 \text{ m/s}^2$ is applied suddenly as a step function at $t = 0 \text{ s}$. The initial problem domain is discretized by using four grid cells with 3×3 particles per cell as depicted in Figure 9. The simulations using the standard MPM, uGIMP, and cpGIMP exhibit the extension instability. However, more accurately tracking the particle domains in the CPDI method results in a stable simulation. For comparative study, five snapshots of the simulation using cpGIMP and CPDI method are depicted in Figure 10. Because the cpGIMP fails to account for the shear part of the deformation, it predicts unrealistic gaps and overlapping of the particle domains, ultimately leading to the extension instability shown in Figure 10. These gaps and overlapping, although present, are much smaller when using the CPDI method. Thus this method does not suffer from extension instability in this example. Vertical displacements of particle q which is shown in Figure 9 during the simulation with the CPDI method are depicted in Figure 11. In order to show the robustness of the CPDI method in simulation of extraordinarily large tensile deformations, this example is re-solved using the CPDI method when the gravity is more than tripled to the value of $g = 3500 \text{ m/s}^2$. Three snapshots of this simulation are depicted in Figure 12(a). The results show that the CPDI method is stable. The proposed alternative nodal basis functions over the particle domain (Equation (22)) lead to particles that will remain ‘in contact’ even when they are far from each other. Figure 12(b) also shows the vertical displacements of particle q which is shown in Figure 9.

4.4. Large deformation vibration of a cantilever beam

Tracking the particle domains as rectangles as is done in the cpGIMP does not account for large rotations. This shortcoming and the modification presented in this work by tracking the particle domains as parallelograms are investigated in this example. Large deformation vibration of a highly compliant cantilever beam under its weight is induced by suddenly applying gravity at $t = 0 \text{ s}$. As shown in Figure 13, the left end of this beam is fixed and other boundaries are traction free. The modulus of elasticity, Poisson’s ratio, and initial density are $E = 1 \times 10^6 \text{ Pa}$, $\nu = 0.3$, and $\rho_0 = 1050 \text{ kg/m}^3$, respectively. Also, the gravity is considered as $g = 10 \text{ m/s}^2$. This example is analyzed using time steps of $\Delta t = 0.001 \text{ s}$ and real-time simulation is $T = 3 \text{ s}$. The beam at its initial configuration is discretized by using 16 grid cells with 3×3 particles per cell as depicted in Figure 13. The standard MPM and cpGIMP simulations failed by spurious material separation. Three snapshots of simulations using uGIMP, cpGIMP, and CPDI methods are depicted in Figure 14. The cpGIMP uses only diagonal terms of the deformation gradient at the particle for updating the particle domain presented in Equation (20). Pure rotation of 90° results in zero diagonal terms in

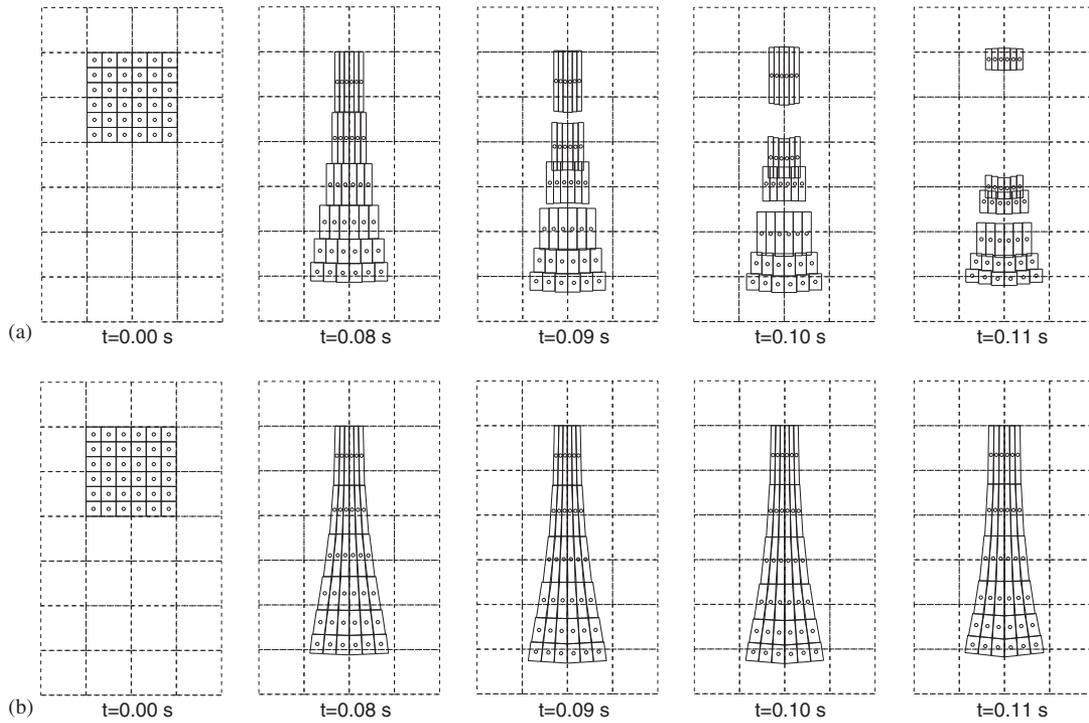


Figure 10. Five snapshots of simulations of the vertical bar problem using: (a) cpGIMP and (b) CPDI method.

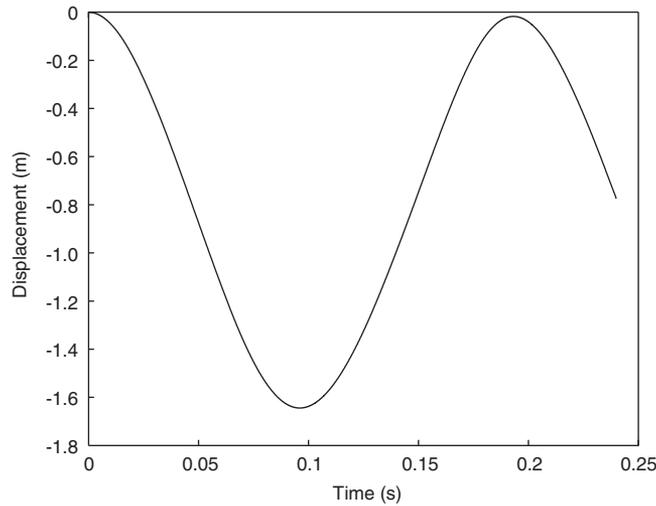


Figure 11. Vertical displacements of particle q during the simulation of the vertical bar problem with CPDI method.

the deformation gradient and the particle domain vanishes in this case. As shown in Figure 14, the particle domain sizes decrease by increasing the rotation in the simulation done using the cpGIMP and finally the analysis fails. On the other hand, the full deformation gradient is used for tracking particle domains in the CPDI method as presented in Equation (21). Appropriately tracking particle domains in the simulation using the CPDI method is shown in Figure 14. Figure 15 shows the vertical displacements of particle q which is shown in Figure 13 during the simulation with uGIMP and CPDI method.

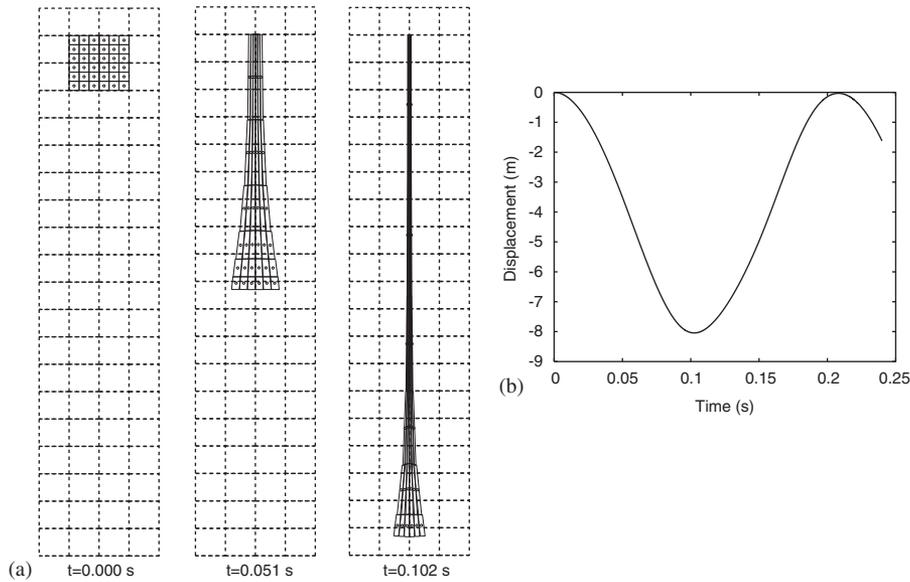


Figure 12. (a) Three snapshots of simulations of very large tensile deformations of the vertical bar problem using CPDI method and (b) vertical displacements of particle q during the simulation with CPDI method.

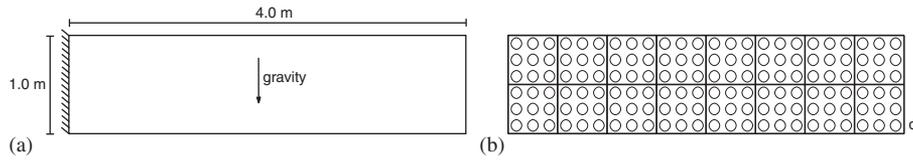


Figure 13. The cantilever beam problem: (a) Geometry and (b) discretized model.

4.5. Investigation of particle domain shape in the initial configuration

Particle domain shapes have an important role in the performance of the uGIMP, cpGIMP, and CPDI methods. The uGIMP and cpGIMP formulations are limited to initial particle distributions corresponding to rectangular particle domains whose edges are parallel with global coordinate axes. However, particle domains can be parallelograms even in the initial configuration in the CPDI method. This capability of the CPDI method is investigated in this section. For this purpose, the examples in Sections 4.3 and 4.4 are re-solved when their geometry is rotated 45° . In this case, the problem domain cannot be accurately discretized by the rectangles whose edges are parallel with global coordinate axes. Such discretization of the bar and cantilever beam are depicted in Figures 16(b) and 17(b), respectively. In these rotated problem domains, only the CPDI method can use the same discretizations as those previously used in examples of Sections 4.3 and 4.4. These discretizations are shown in Figures 16(a), (c) and 17(a), (c). The bar with large tensile deformation and large deformation vibration of the cantilever beam problems are re-analyzed using uGIMP, cpGIMP, and CPDI methods. The symmetry boundary condition in the rotated bar problem, is enforced by adding the same bar on the other side of the boundary which is subjected to gravity with the same magnitude but opposite direction. For the rotated cantilever beam example, the fixed boundary conditions are enforced by fixing the related grid nodes as shown in Figure 17. For the bar problem, the simulations using the uGIMP, cpGIMP, and CPDI methods with mesh(b) encounter the extension instability and cannot be completed. Only the CPDI method with mesh(c) results in a stable simulation. Numerical simulations which are not presented here show that both mesh types (b) and (c) lead to stable solutions using the uGIMP, cpGIMP, and CPDI methods if sufficient refinement is used. One snapshot of the simulation using the CPDI method with mesh(c) and displacements of particle q during the simulation are depicted in Figure 18(a) and (b),

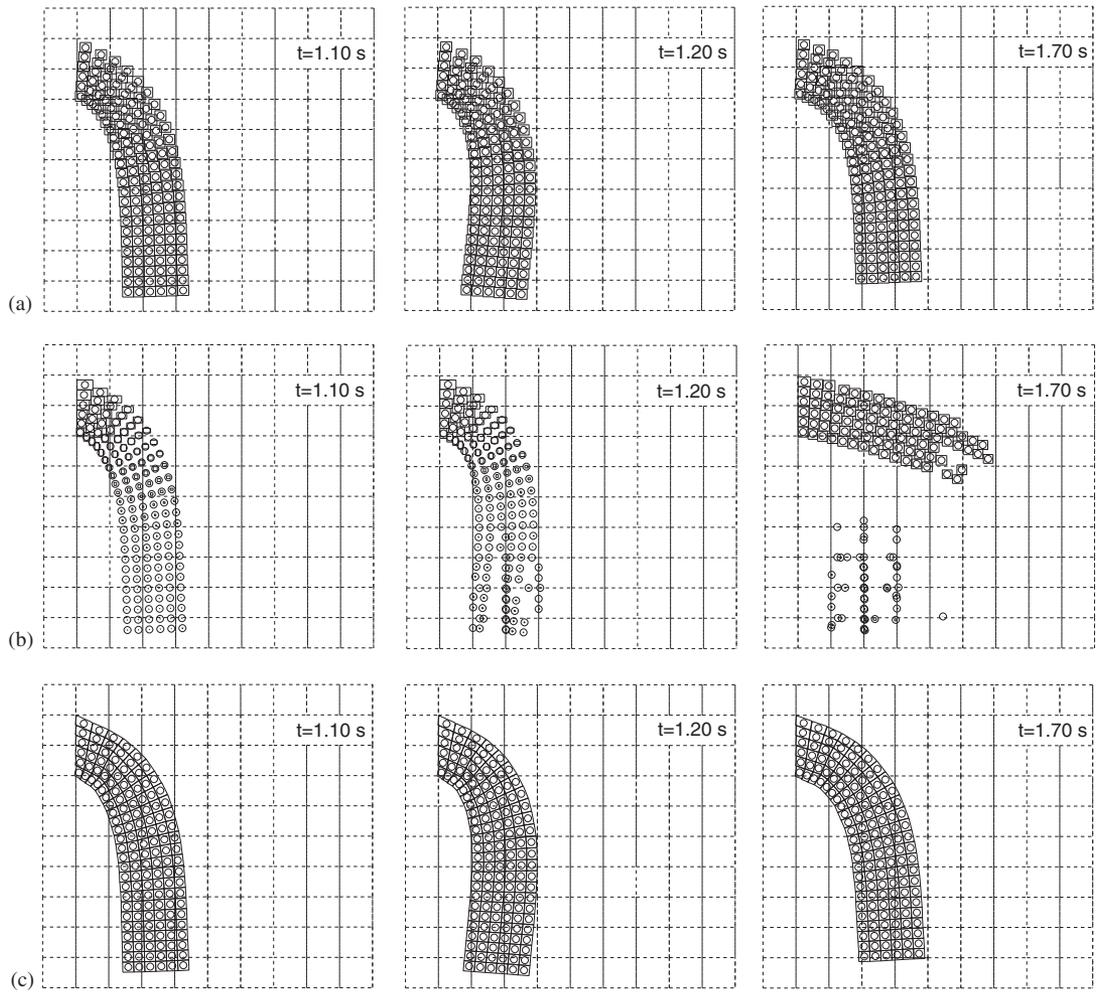


Figure 14. Three snapshots of simulations of the cantilever beam problem using:
 (a) uGIMP; (b) cpGIMP; and (c) CPDI method.

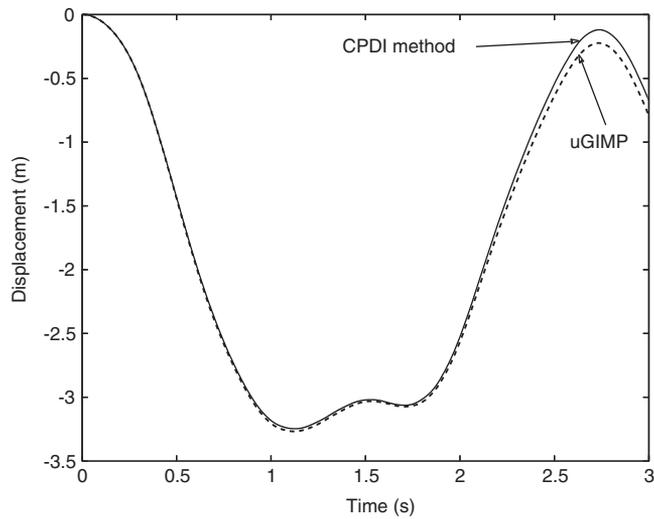


Figure 15. Vertical displacements of particle q during the simulation of the cantilever beam problem with uGIMP and CPDI method.

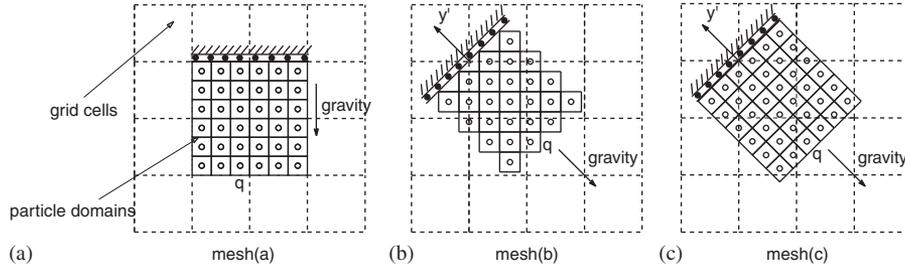


Figure 16. (a) Discretization of the bar problem used in example Section 4.3; (b) and (c) discretizations of the rotated bar problem.

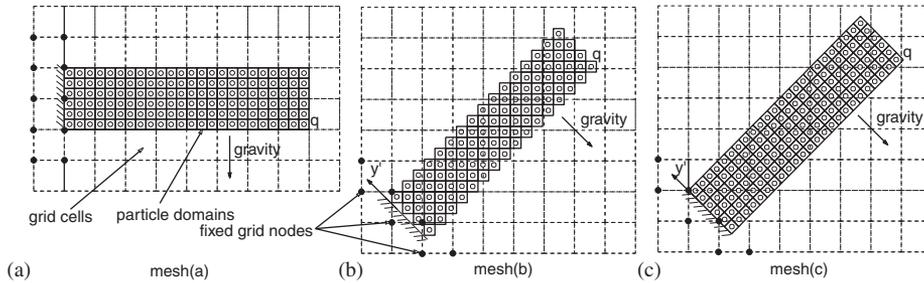


Figure 17. (a) Discretization of the cantilever beam problem used in example Section 4.4; (b) and (c) discretizations of the rotated beam problem.

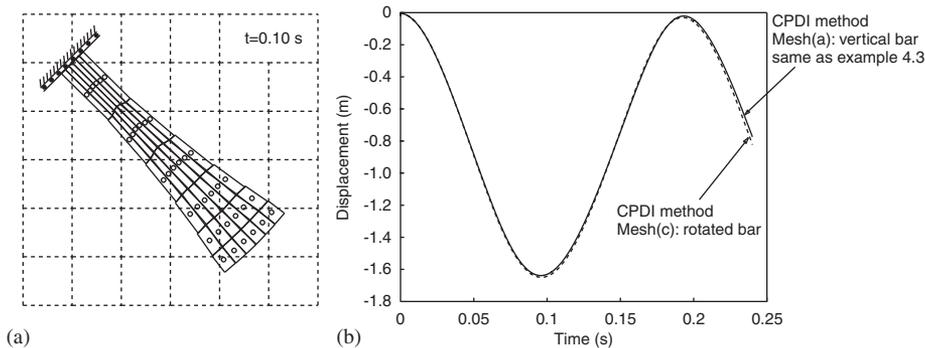


Figure 18. (a) One snapshot of simulation of the rotated bar problem using CPDI method with mesh(c); (b) displacements of particle q during the simulation of the bar problem with CPDI method.

respectively. As shown in Figure 18(b), displacements of particle q are in a very good agreement with those obtained in the example of Section 4.3.

In the cantilever beam problem, the simulations using the uGIMP with mesh(b) and CPDI method with both mesh(b) and mesh(c) are stable. However, simulation using cpGIMP with mesh(b) is unstable. If sufficient refinement is used, both mesh types (b) and (c) lead to the same results using uGIMP and CPDI methods but corresponding results are not presented here. One snapshot of the simulation using the CPDI method with mesh(c) and displacements of particle q during stable simulations are depicted in Figure 19(a) and (b), respectively. The displacements of particle q during the simulation of the rotated cantilever beam are measured with respect to the y' axis as shown in Figure 17. As depicted in Figure 19(b), the results obtained from the uGIMP and CPDI methods with mesh(b) are not accurate. However, displacements of particle q obtained from the simulation using CPDI with mesh(c) are in a very good agreement with those obtained in the example of Section 4.4. For investigating poor quality of mesh(b), this example is solved

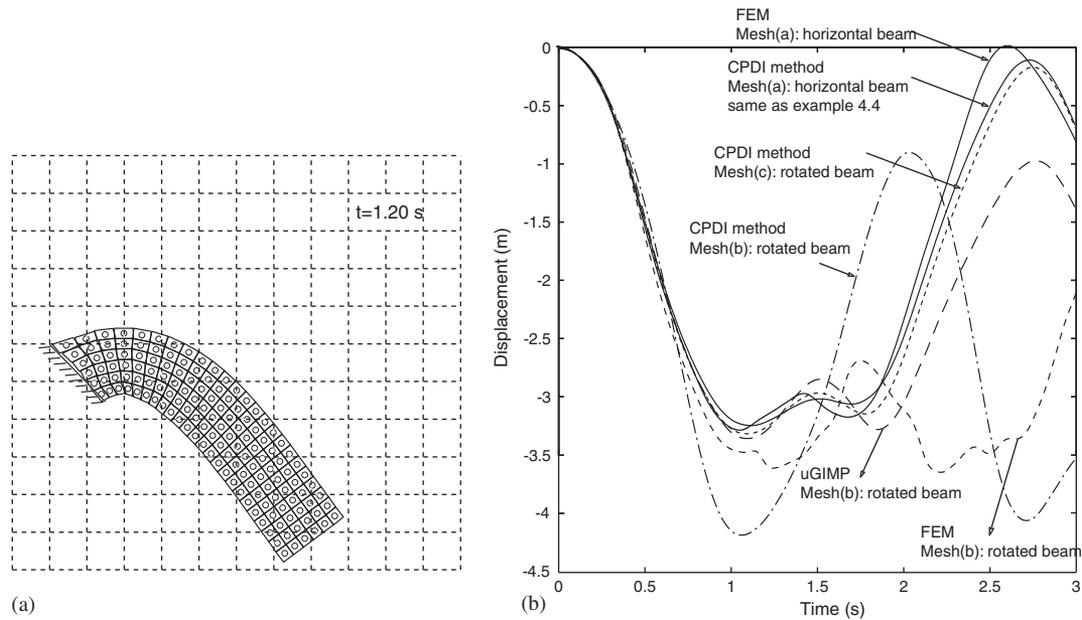


Figure 19. (a) One snapshot of simulations of the rotated cantilever beam problem using CPDI method with mesh(c); (b) displacements of particle q during the simulation of the cantilever beam problem with uGIMP, CPDI, and FEM methods.

using the FEM by considering the particle domains as the Q4 finite elements. Similar to the CPDI, the FEM with mesh(b) also failed to capture accurate results as shown in Figure 19(b). The results obtained from the FEM with mesh(a) do not exactly coincide with those obtained from the CPDI with mesh(a) because the CPDI uses the proposed alternative grid functions, which are coarser than the FEM basis functions. It can be concluded that the CPDI method capability of dealing with parallelograms as the particle domains in the initial configuration improves the results where the problem domain boundaries are not aligned with the global coordinate axes. Although we have limited the scope in this paper to parallelogram particle domains, the concept should be straightforward to generalize to more general convex polygons that conform to complicated boundaries, provided that an efficient interpolator (i.e. generalization of Q4 shape functions) is available. A boundary conforming particle distribution can be generated using centroidal Voronoi tessellation (CVT) [31].

5. CONCLUSION

The CPDI method has been developed and demonstrated to be generally more accurate and efficient than the uGIMP and cpGIMP methods. The accuracy improvement is achieved by allowing particle domains to track with the material motion in such a way that an initially rectangular particle domain deforms into a parallelogram according to the local tangent mapping. The efficiency improvement, which allows evaluation of the MPM integrals without having to partition the integration domain into element intersections, is achieved by replacing the standard grid basis functions with alternative grid basis functions that are a finite element interpolation of the standard grid basis to the parallelogram corners. Similar to the standard grid basis, the alternative grid basis is a partition of unity.

The new CPDI method has been demonstrated to be efficient, accurate, and stable in several large deformation solid mechanics problems that were not solved satisfactorily with any previously existing MPM techniques. The cpGIMP and uGIMP methods develop inaccurate results and even instability because they inaccurately track particle domains as rectangles. By using the full deformation gradient at the particles, these problems are eliminated in the new CPDI method by

convecting the particle domains with the material. In principle, the conventional GIMP approach could also convect its integration domain with the material but doing so would lead to a very high computational cost associated with dividing the particle domain along cell boundaries. Using the alternative grid basis functions avoids the need to divide particle domains along cell boundaries. In the interior of a particle, the CPDI alternative grid basis functions can differ significantly from the standard grid basis functions, yet they still provide very efficient and accurate approximations to the MPM integrals. These integrals are essentially *averages* of the standard grid basis function and its gradient (φ_{ip} and $\nabla\varphi_{ip}$) over the particle. The CPDI method evaluates these averages accurately and efficiently even if the particle domain is stretched over many grid cells. In 1D, for example, the fundamental theorem of calculus ensures that the nodal force integral (i.e. the integral of the basis function gradient over the particle) depends only on the basis function values at the particle's boundaries; since the alternative grid basis is an interpolation of the standard grid basis, it has identical boundary values and hence identical force integral values.

The first two verification examples presented in this paper were constructed based on the MMS. The first example involved simultaneously applied horizontal and vertical extension and compression waves. In this special case where the particle domains remain rectangular, consistent with the cpGIMP assumptions, it was demonstrated that the accuracy and rate of convergence of the CPDI method was comparable to the cpGIMP. The second MMS verification problem involved deformation of rectangles to parallelograms to demonstrate superior accuracy of the CPDI method in comparison with three most common MPM techniques currently used. The third example, which was a compliant bar elongating under its own weight in response to the sudden application of a large gravitational field, demonstrated that the CPDI method does not suffer from the extension instability that was evident in the other MPM methods considered in this study. Finally, the fourth example explains the disadvantage of the cpGIMP for analysis of problems including very large rotations and how the CPDI method resolves this shortcoming. Moreover, numerical investigations of the rotated problem domains show that the capability of the CPDI method to allow parallelograms as the *initial* particle domains improves the results in comparison with simulations for which the problem domain boundaries are not aligned with the global coordinate axes. More broadly, this simulation suggests that particle distributions that conform to a boundary are desirable.

Although the presentation of the new CPDI method was limited to 2D deformations, it should be self-evident how to generalize the techniques to 3D. Furthermore, although the CPDI method has been here limited to the case of parallelograms deforming to parallelograms, the principles are applicable to general polygons (or polyhedra in 3D).

APPENDIX A: APPLICATION OF THE METHOD OF MANUFACTURED SOLUTIONS FOR CALCULATING BODY FORCES OF EXAMPLES 4.1 AND 4.2

The displacement solutions of the equations of motion are assumed *a priori* in the MMS. Given the assumed prescribed displacements, the constitutive model can be evaluated to determine the corresponding stress field. Therefore, the divergence of stress and acceleration can be evaluated, and the required body forces to achieve these solutions are analytically determined from the equations of motion. Two forms of the equations of motion are often presented in the FEM texts: total Lagrangian and updated Lagrangian forms. Both of them are used successfully with the FEM formulations, and solutions from both forms are equivalent [32]. The solutions are typically manufactured in the total Lagrangian form in the MMS. However, MPM and GIMP are always implemented in the updated Lagrangian form. Noting equivalence of the total and updated Lagrangian forms, a numerical solution in the updated Lagrangian form can be compared with a manufactured solution in the total Lagrangian form using mapping equations between two forms.

The equations of motion in the total and updated Lagrangian forms are considered, respectively

$$\nabla_0 \cdot \mathbf{P} + \rho_0 \mathbf{b} = \rho_0 \ddot{\mathbf{u}} \quad (\text{A1})$$

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} = \rho \ddot{\mathbf{u}} \quad (\text{A2})$$

where $\nabla_0 \cdot$ and $\nabla \cdot$ are the divergence operators with respect to the initial and current configurations, respectively, \mathbf{P} is the first Piola–Kirchhoff stress, $\boldsymbol{\sigma}$ is the Cauchy stress, \mathbf{b} is the body force per unit mass, ρ_0 and ρ are initial and current densities, respectively, and \mathbf{u} and $\ddot{\mathbf{u}}$ are displacement and acceleration, respectively.

The Neo-Hookean material model which is presented in the updated Lagrangian form in Equation (30) was used in this paper. Equation (30) can be rewritten in the total Lagrangian form

$$\mathbf{P} = \lambda \ln J \mathbf{F}^{-1} + \mu \mathbf{F}^{-1} (\mathbf{F} \mathbf{F}^T - \mathbf{I}) \quad (\text{A3})$$

where \mathbf{F} is the deformation gradient, J is the determinant of \mathbf{F} , μ and λ are the shear modulus and Lamé constant, respectively.

$$\mathbf{F} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \quad (\text{A4})$$

in which \mathbf{X} is the position in the reference configuration.

A.1. Calculation of body forces of axis-aligned displacement in a unit square problem

The following displacement vector is assumed in this example.

$$\mathbf{u} = \begin{Bmatrix} A \sin\left(\frac{2\pi X}{L}\right) \sin\left(\frac{C\pi t}{L}\right) \\ A \sin\left(\frac{2\pi Y}{L}\right) \sin\left(\frac{C\pi t}{L} + \pi\right) \\ 0 \end{Bmatrix} \quad (\text{A5})$$

in which X and Y are the components of the global coordinate system in the reference configuration, A is the maximum amplitude of displacement, t is the time, and C is the constant defined in Equation (29).

The deformation gradient is found by taking derivatives with respect to position according to Equation (A4)

$$\mathbf{F} = \begin{bmatrix} 1 + \frac{2A\pi}{L} \cos\left(\frac{2\pi X}{L}\right) \sin\left(\frac{C\pi t}{L}\right) & 0 & 0 \\ 0 & 1 + \frac{2A\pi}{L} \cos\left(\frac{2\pi Y}{L}\right) \sin\left(\frac{C\pi t}{L} + \pi\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A6})$$

For any diagonal \mathbf{F} , Equation (A3) may be expressed as

$$\mathbf{P} = \begin{bmatrix} \frac{\lambda}{F_{xx}} \ln(F_{xx} F_{yy}) + \frac{\mu}{F_{xx}} (F_{xx}^2 - 1) & 0 & 0 \\ 0 & \frac{\lambda}{F_{yy}} \ln(F_{xx} F_{yy}) + \frac{\mu}{F_{yy}} (F_{yy}^2 - 1) & 0 \\ 0 & 0 & \lambda \ln(F_{xx} F_{yy}) \end{bmatrix} \quad (\text{A7})$$

After substituting stress \mathbf{P} from Equation (A7) and the acceleration which is determined by twice differentiating the displacement vector \mathbf{u} in Equation (A5), Equation (A1) can be solved for the body force \mathbf{b}

$$\mathbf{b} = \begin{Bmatrix} \frac{\pi^2 u_x}{L^2} \left(\frac{4\mu}{\rho_0} - C^2 - 4 \frac{\lambda[\ln(F_{xx} F_{yy}) - 1] - \mu}{\rho_0 F_{xx}^2} \right) \\ \frac{\pi^2 u_y}{L^2} \left(\frac{4\mu}{\rho_0} - C^2 - 4 \frac{\lambda[\ln(F_{xx} F_{yy}) - 1] - \mu}{\rho_0 F_{yy}^2} \right) \\ 0 \end{Bmatrix} \quad (\text{A8})$$

A.2. Calculation of body forces of radial expansion of a ring problem

The displacement field with radial symmetry in Equation (37) is rewritten as a displacement vector in the Cartesian coordinate system

$$\mathbf{u} = \begin{Bmatrix} A \sin\left(\frac{C\pi t}{2\bar{r}}\right) [c_3(X^2 + Y^2) + c_2\sqrt{X^2 + Y^2} + c_1]X \\ A \sin\left(\frac{C\pi t}{2\bar{r}}\right) [c_3(X^2 + Y^2) + c_2\sqrt{X^2 + Y^2} + c_1]Y \\ 0 \end{Bmatrix} \quad (\text{A9})$$

in which X and Y are the components of the global coordinate system in the reference configuration, A is the maximum amplitude of displacement, t is the time, C is the constant defined in Equation (29), constants c_1 , c_2 , and c_3 were defined in Equation (38), and \bar{r} is the mean radius of the ring.

The procedure discussed for the axis-aligned displacement in a unit square problem in Appendix A.1 is used to determine the body force vector for the radial expansion of a ring problem with zero Poisson's ratio. However, the resultant equations are quite unwieldy in this case. The body force for this problem is determined by using the following symbolic calculations in MATLAB:

```
syms A X Y t rho mu C c1 c2 c3 rb
u=[A*sin((C*pi*t)/2/rb)*(c3*(X^2+Y^2)+c2*sqrt(X^2+Y^2)+c1)*X
  A*sin((C*pi*t)/2/rb)*(c3*(X^2+Y^2)+c2*sqrt(X^2+Y^2)+c1)*Y
  0];
F=[1+diff(u(1),X) diff(u(1),Y) 0;diff(u(2),X) 1+diff(u(2),Y) 0;0 0 1];
I=[1 0 0;0 1 0;0 0 1];
J=det(F);
Fp=[F(1,1) F(2,1) 0;F(1,2) F(2,2) 0;0 0 1];
P=simple(mu*inv(F)*(F*Fp-I));
dP=[diff(P(1,1),X)+diff(P(1,2),Y);diff(P(2,1),X)+diff(P(2,2),Y);0];
BodyForce=(diff(diff(u,t),t)-1/rho*dP);
```

ACKNOWLEDGEMENTS

Funding support of this work by Schlumberger Technology Corporation is gratefully acknowledged. The authors thank Dr J. E. Guilkey for his helpful comments and contribution in this work. Also, we are grateful to Prof. Deborah Sulsky of the University of New Mexico for helpful clarification about the distinction between the tensile instability reported for the smoothed particle hydrodynamics (SPH) method in contrast to what we have referred to as the extension instability in this work. Finally, the final manuscript was much improved by our incorporation of suggestions from participants in the MPM session of the World Congress on Computational Mechanics, 2010, and the MPM workshop in Albuquerque, NM, 2010.

REFERENCES

1. Sulsky D, Chen A, Schreyer H. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:179–196.
2. Sulsky D, Zhou S, Schreyer H. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**:236–252.
3. Bardenhagen S, Kober E. The generalized interpolation material point method. *CMES—Computer Modeling in Engineering and Sciences* 2004; **5**:477–495.
4. York A, Sulsky D, Schreyer H. The material point method for simulation of thin membranes. *International Journal for Numerical Methods in Engineering* 1999; **44**:1429–1456.
5. York A, Sulsky D, Schreyer H. Fluid–membrane interaction based on the material point method. *International Journal for Numerical Methods in Engineering* 2000; **48**:901–924.
6. Love E, Sulsky D. An energy-consistent material-point method for dynamic finite deformation plasticity. *International Journal for Numerical Methods in Engineering* 2005; **65**:1608–1638.
7. Bardenhagen S, Guilkey J, Roessig K, Brackbill J, Witzel W, Foster J. An improved contact algorithm for the material point method and application to stress propagation in granular material. *Computer Modeling in Engineering and Sciences* 2001; **2**:509–522.
8. Nairn J. Material point method calculations with explicit cracks. *Computer Modeling in Engineering and Sciences* 2003; **4**:649–663.
9. Guo Y, Nairn J. Three-dimensional dynamic fracture analysis using the material point method. *Computer Modeling in Engineering and Sciences* 2006; **16**:141–155.
10. Sulsky D, Schreyer H, Peterson K, Kwok R, Coon M. Using the material point method to model sea ice dynamics. *Journal of Geophysical Research* 2007; **112**(c2):C02S90.
11. Ma S, Zhang X, Qiu X. Comparison study of MPM and SPH in modeling hypervelocity impact problems. *International Journal of Impact Engineering* 2009; **36**:272–282.
12. Zhang H, Wang K, Chen Z. Material point method for dynamic analysis of saturated porous media under external contact/impact of solid bodies. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:1456–1472.
13. Schreyer H, Sulsky D, Zhou S. Modeling delamination as a strong discontinuity with the material point method. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:2483–2507.
14. Ma J, Lu H, Wang B, Hornung R, Wissink A, Komanduri R. Multiscale simulation using generalized interpolation material point (GIMP) method and molecular dynamics (MD). *Computer Modeling in Engineering and Sciences* 2006; **14**:101–117.
15. Ma J, Liu Y, Lu H, Komanduri R. Multiscale simulation of nanoindentation using the generalized interpolation material point (GIMP) method, dislocation dynamics (DD) and molecular dynamics (MD). *Computer Modeling in Engineering and Sciences* 2006; **16**:41–55.
16. Daphalapurkar N, Lu H, Coker D, Komanduri R. Simulation of dynamics crack growth using the generalized interpolation material point (GIMP) method. *International Journal of Fracture* 2007; **143**:79–102.
17. Bardenhagen S. Energy conservation error in the material point method for solid mechanics. *Journal of Computational Physics* 2002; **180**:383–403.
18. Guilkey J, Weiss J. Implicit time integration for the material point method: quantitative and algorithmic comparisons with the finite element method. *International Journal for Numerical Methods in Engineering* 2003; **57**:1323–1338.
19. Sulsky D, Kaul A. Implicit dynamics in the material-point method. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1137–1170.
20. Love E, Sulsky D. An unconditionally stable, energy momentum consistent implementation of the material-point method. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3903–3925.
21. Ma J, Lu H, Komanduri R. Structured mesh refinement in generalized interpolation material point method (GIMP) for simulation of dynamic problems. *Computer Modeling in Engineering and Sciences* 2006; **12**:213–227.
22. Wallstedt P, Guilkey J. Improved velocity projection for the material point method. *Computer Modeling in Engineering and Sciences* 2007; **19**:223–232.
23. Wallstedt P, Guilkey J. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics* 2008; **227**:9628–9642.
24. Steffen M, Kirby R, Berzins M. Analysis and reduction of quadrature errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering* 2008; **76**:922–948.
25. Steffen M, Wallstedt P, Guilkey J, Kirby R, Berzins M. Examination and analysis of implementation choices within the material point method (MPM). *Computer Modeling in Engineering and Sciences* 2008; **31**:107–127.
26. Steffen M, Kirby R, Berzins M. Decoupling and balancing of space and time errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering* 2010; DOI: 10.1002/nme.2787.
27. Shen L, Chen Z. A silent boundary scheme with the material point method for dynamic analyses. *Computer Modeling in Engineering and Sciences* 2005; **7**:305–320.
28. Buzzi O, Pedroso D, Giacomini A. Caveats on the implementation of the generalized material point method. *Computer Modeling in Engineering and Sciences* 2008; **1**:1–21.

29. Mackenzie-Helnwein P, Arduino P, Shin W, Moore J, Miller G. Modeling strategies for multiphase drag interactions using the material point method. *International Journal for Numerical Methods in Engineering* 2010; **83**: 295–322.
30. Knupp P, Salari K. *Verification of Computer Codes in Computational Science and Engineering*. Chapman & Hall/CRC: London, 2003.
31. Burkardt J, Gunzberger M, Peterson J, Brannon R. User manual and supporting information for library of codes for centroidal Voronoi point placement and associated zeroth, first, and second moment determination. *Sandia National Laboratories Report SAND2002-0099*, 2002.
32. Belytschko T, Liu W, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: New York, 2000.