Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method

J. E. Guilkey[1],*,[†] and J. A. Weiss[2]

[1]*Department of Mechanical Engineering, University of Utah, Salt Lake City, Utah 84112, U.S.A.*
[2]*Department of Bioengineering, University of Utah, Salt Lake City, Utah 84112, U.S.A.*

SUMMARY

An implicit integration strategy was developed and implemented for use with the material point method (MPM). An incremental-iterative solution strategy was developed around Newton's method to solve the equations of motion with Newmark integration to update the kinematic variables. Test problems directly compared the implicit MPM solutions with those obtained using an explicit MPM code and implicit finite element (FE) code. Results demonstrated very good agreement with FE predictions and also illustrated several advantages in comparison to calculations using the explicit MPM code. In particular, the accuracy of the implicit solution was superior to the explicit MPM when compared to validated FE solutions, and by definition the implicit time integration is unconditionally stable. Similarities between the assembly of the implicit MPM equations and those of the FE method were identified and should allow further algorithmic improvement. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: implicit time integration; material point method

1. INTRODUCTION

The material point method (MPM) is a particle method for simulations in computational fluid and solid mechanics [1, 2]. The method uses a regular structured grid as a computational scratchpad for computing spatial gradients of field variables. The grid is convected with the particles during deformations that occur over a time step, eliminating the diffusion problems associated with advection on an Eulerian grid. The grid is restored to its original location at the end of a time step. In addition to avoiding the Eulerian diffusion problem, this approach also circumvents problems with mesh entanglement that can plague fully Lagrangian-based techniques when large deformations are encountered. MPM has also been successful in solving

Copyright © 2003 John Wiley & Sons, Ltd.

problems involving contact, having an advantage over traditional finite element (FE) methods in that the use of the regular grid eliminates the need for doing costly searches for contact surfaces [3].

Coupling of the MPM to computational fluid dynamics simulations is readily achieved because a regular grid is used for gradient calculations [4]. The grid then serves as both an Eulerian reference frame for CFD calculations and an updated Lagrangian reference frame for MPM calculations. Tight coupling can be achieved between the two phases by using a multimaterial CFD formulation, while each phase still enjoys the benefits of computation using its optimal reference frame. When using explicit time integration for both phases, the time step sizes required by the Courant stability condition ($dt < dx/c$, where dx is the grid spacing and c is the speed of sound in the material) are often disparate by several orders of magnitude. The time step size imposed by this restriction is relatively severe compared to the time step limitations of most CFD codes. This can result in prohibitive computational solution times for fluid–solid interaction problems involving traditional engineering materials.

To circumvent the time step restrictions imposed by explicit time integration for low rate dynamic and quasi-static problems, the objectives of this work were to develop and implement an implicit time integration strategy with the MPM and to test the implementation against an explicit MPM code and an implicitly integrated FE code. Our implicit time integration strategy exploits similarities between the function of material points in MPM and integration points in FE calculations to adapt implicit time integration for use with MPM. The implementation uses Newton's method to solve for the incremental grid displacements in the linearized form of the equations of motion and the update of nodal kinematics using the trapezoidal rule. Explicit expressions for the tangent stiffness are derived in terms of the grid displacements used in MPM. Because of the similarities that are identified between MPM and FE methods in this work, further improvements to MPM algorithms and numerical implementations should be able to benefit from the large amount of published research on implicit FE methods. In addition to allowing for much larger time steps, the implicit algorithm has also shown advantage over the explicit algorithm in its ability to obtain a solution for certain types of problems. The accuracy for problems involving large deformation, contact and dynamics is demonstrated through representative numerical simulations. The implicit MPM algorithm described here is intended to give an additional option to analysts for the solution of problems for which MPM is well suited, but hindered by the explicit time step size restriction.

2. REVIEW OF MPM WITH EXPLICIT TIME INTEGRATION

The following section reviews the framework for explicit time integration with MPM as presented by Sulsky and Schreyer [2] to introduce symbols and notation and to provide a starting point for development of the implicit approach. The approaches share the same particle discretization and interpolation procedures. We assume use of trilinear shape functions on the computational grid. After the MPM discretization and assembly on the background computational grid, the equations of motion take the following familiar discretized form:

$$\mathbf{M}_g \cdot \mathbf{a}_g = \mathbf{Fext}_g - \mathbf{Fint}_g \tag{1}$$

Here, $\mathbf{M}_g$ is the mass matrix (typically lumped), $\mathbf{a}_g$ is the acceleration vector, $\mathbf{Fext}_g$ is the external force vector (sum of body forces and surface tractions), and $\mathbf{Fint}_g$ is the internal

force vector resulting from the stress divergence. The subscript g indicates that quantities are evaluated on the computational grid.

The solution procedure begins by interpolating the particle state to the grid to form $\mathbf{M}_g$, which is in turn used to obtain a velocity field $\mathbf{v}_g$ on the grid. Field variables for a particular grid node receive contributions from particles that currently reside in any element that contains the grid node. The mass matrix for the ith grid node is obtained by interpolating the mass from surrounding material points m_p using the shape functions evaluated at each material point, S_{ip}:

$$M_i = \sum_p S_{ip} m_p \tag{2}$$

The grid node velocity $\mathbf{v}_i$ is obtained by interpolating the momentum of surrounding particles to the grid node using the shape functions and then weighting their momentum contribution by the grid node mass M_i:

$$\mathbf{v}_i = \frac{\sum_p S_{ip} m_p \mathbf{v}_p}{M_i} \tag{3}$$

Here, m_p is the particle mass, $\mathbf{v}_p$ is the particle velocity and $\mathbf{G}_{ip}$ is the gradient of the ith node's shape function at $\mathbf{x}_p$. This approach ensures momentum conservation during the interpolations. External forces on the particles $\mathbf{Fext}_p$ are interpolated to the grid similarly:

$$\mathbf{Fext}_i = \sum_p S_{ip} \mathbf{Fext}_p \tag{4}$$

A particle velocity gradient $\nabla \mathbf{v}_p$ is computed for constitutive model evaluation:

$$\nabla \mathbf{v}_p = \sum_i \mathbf{G}_{ip} \mathbf{v}_i \tag{5}$$

The constitutive model is then evaluated to obtain the Cauchy stress $\boldsymbol{\sigma}_p$ at each particle. The internal force is then

$$\mathbf{Fint}_i = \sum_p \mathbf{G}_{ip} \boldsymbol{\sigma}_p v_p \tag{6}$$

where v_p is the particle volume. After assembly, Equation (1) is solved for $\mathbf{a}_g$.

The backward Euler method is used for time integration. The grid velocity $\mathbf{v}_g^L$ is then

$$\mathbf{v}_g^L = \mathbf{v}_g + \mathbf{a}_g \, dt \tag{7}$$

While the following calculation is never carried out *explicitly*, the nodes of the grid also move with the same convective velocity

$$\mathbf{x}_g^L = \mathbf{x}_g + \mathbf{v}_g^L \, dt \tag{8}$$

The particles move with the deforming grid and their kinematics are updated explicitly:

$$\mathbf{v}_p(t + dt) = \mathbf{v}_p(t) + \sum_i S_{ip} \mathbf{a}_i \, dt \tag{9}$$

and

$$\mathbf{x}_p(t + dt) = \mathbf{x}_p(t) + \sum_i S_{ip} \mathbf{v}_i^L \, dt \tag{10}$$

This completes one time step. Since the calculation in Equation (8) is not performed, the deformed grid is effectively reset to its undeformed position at the end of the time step.

This explicit integration strategy is widely used in FEA, and the limitation on the size of the time step from the conditional stability of the backward Euler integration method is well known. An implicit time integration strategy that addresses the limitation is described in the following section.

3. MPM WITH IMPLICIT TIME INTEGRATION

3.1. Incremental-iterative solution of linearized matrix equations

Our approach to implicit time integration with MPM is based on the observation that *calculations on the computational grid are carried out in a manner that is exactly the same as those performed for finite element calculations. The material points essentially function as integration points for the assembly.* The discrete form of the equations of motion linearized about a configuration at time t are identical to those obtained for a traditional displacement-based non-linear finite element formulation, based upon integrals obtained numerically using the Gauss integration points for evaluation (see, e.g. Reference [5]). Assuming an incremental-iterative solution strategy, the discretized version of the linearized equations of motion on the grid at Newton iteration k are

$$\mathbf{KK}_g^{k-1}(t+\mathrm{d}t) \cdot \mathbf{du}_g^k = \mathbf{Fext}_g(t+\mathrm{d}t) - \mathbf{Fint}_g^{k-1}(t+\mathrm{d}t) - \mathbf{M}_g \cdot \mathbf{a}_g^{k-1}(t+\mathrm{d}t)$$

$$:= \mathbf{Q}_g^k \qquad (11)$$

Here $\mathbf{KK}_g$ is the stiffness matrix, $\mathbf{du}_g$ is an estimate of the incremental displacements, $\mathbf{Fext}_g$ is the vector of external forces at the new time $t+\mathrm{d}t$, $\mathbf{Fint}_g$ is the vector of internal forces due to the stress divergence, $\mathbf{M}_g$ is the mass matrix and $\mathbf{a}_g$ is the acceleration vector.

The objective of the Newton iterations is to determine the grid displacement vector $\mathbf{u}$ that minimizes the norm of the residual, $\mathbf{Q}_g$. A new estimate for $\mathbf{u}$ is obtained by solving Equation (11) for the current estimate of incremental displacements $\mathbf{du}_g^k$ and adding it to the previous displacements

$$\mathbf{u}_g^k(t+\mathrm{d}t) = \mathbf{u}_g^{k-1}(t+\mathrm{d}t) + \mathbf{du}_g^k \qquad (12)$$

The other kinematic variables, stiffness matrix and internal forces are then updated using the new displacements.

Note that $\mathbf{u}_g^k(t+\mathrm{d}t)$ is not the *total* displacement, but rather it is the displacement of the grid from t to $t+\mathrm{d}t$. The total material displacements are contained in the positions and total deformation gradient of the particles. This is a fundamental difference from the standard approach used in implicit FEA. This is apparent in the kinematics in Step (5) below.

Newton iterations are continued until convergence is achieved, defined in the present work as satisfaction of both of the following criteria:

$$\frac{\|\mathbf{du}_g^k\|}{\|\mathbf{du}_g^{\max}\|} < \varepsilon_d \quad \text{and} \quad \frac{\|\mathbf{du}_g^k \mathbf{Q}_g^k\|}{\|\mathbf{du}_g^0 \mathbf{Q}_g^0\|} < \varepsilon_e \qquad (13)$$

where $\|\mathbf{du}_{\mathrm{g}}^{\max}\|$ is the maximum value of the norm of the displacement increment and $\|\mathbf{du}_{\mathrm{g}}^{0}\mathbf{Q}_{\mathrm{g}}^{0}\|$ is the initial value of the norm of the product of the displacement increment and the residual. ε_d and ε_e are user-selected values for the convergence criteria.

3.2. Kinematic update

The unconditionally stable trapezoidal rule is used to advance the grid kinematics to $t + \mathrm{d}t$:

$$\mathbf{u}_{\mathrm{g}}(t + \mathrm{d}t) = \frac{\mathrm{d}t}{2}(\mathbf{v}_{\mathrm{g}}(t) + \mathbf{v}_{\mathrm{g}}(t + \mathrm{d}t)) \tag{14}$$

$$\mathbf{v}_{\mathrm{g}}(t + \mathrm{d}t) = \mathbf{v}_{\mathrm{g}}(t) + \frac{\mathrm{d}t}{2}(\mathbf{a}_{\mathrm{g}}(t) + \mathbf{a}_{\mathrm{g}}(t + \mathrm{d}t)) \tag{15}$$

When Equations (14) and (15) are combined with Equation (12), the acceleration for the current iteration k at time $t + \mathrm{d}t$ can be approximated in terms of known quantities at time t and estimates at time $t + \mathrm{d}t$ from the previous iteration $k - 1$:

$$\mathbf{a}_{\mathrm{g}}^{k}(t + \mathrm{d}t) = \frac{4}{\mathrm{d}t^2}\mathbf{u}_{\mathrm{g}}^{k-1}(t + \mathrm{d}t) - \frac{4}{\mathrm{d}t}\mathbf{v}_{\mathrm{g}}(t) - \mathbf{a}_{\mathrm{g}}(t) \tag{16}$$

This value for $\mathbf{a}_{\mathrm{g}}^{k}(t + \mathrm{d}t)$ is used in Equation (11) above.

Because the grid is reset each time step, it is not possible to use one-time step old values of $\mathbf{v}_{\mathrm{g}}(t + \mathrm{d}t)$ and $\mathbf{a}_{\mathrm{g}}(t + \mathrm{d}t)$ for $\mathbf{v}_{\mathrm{g}}(t)$ and $\mathbf{a}_{\mathrm{g}}(t)$ respectively. Because particles migrate into previously empty cells, some of the nodes of these cells will not have had values for $\mathbf{v}_{\mathrm{g}}(t + \mathrm{d}t)$ and $\mathbf{a}_{\mathrm{g}}(t + \mathrm{d}t)$. Therefore, the values for $\mathbf{v}_{\mathrm{g}}(t)$ and $\mathbf{a}_{\mathrm{g}}(t)$ come from the particle values for $\mathbf{v}_p$ and $\mathbf{a}_p$ interpolated to the grid as in Equation (3).

3.3. Computational algorithm for implicit MPM

The approach for implicit integration in the context of MPM proceeds in the above framework, taking into account the appropriate interpolation of quantities from the particles to the grid at each step to be consistent with the MPM framework. The equilibrium iterations occur on the computational grid. At the beginning of each implicit time step, the following particle quantities are known at time t: mass m_p, volume v_p, position $\mathbf{x}_p(t)$, velocity $\mathbf{v}_p(t)$, deformation gradient $\mathbf{F}_p(t)$ and Cauchy stress $\boldsymbol{\sigma}_p(t)$. Also known are the external forces at time $t + \mathrm{d}t$, $\mathbf{Fext}_p(t + \mathrm{d}t)$ and the acceleration on the grid at time t, $\mathbf{a}_{\mathrm{g}}(t)$. With the new applied increment in external forces, $\mathbf{Q}_{\mathrm{g}}^{k}$ in Equation (11) is no longer minimized and Newton iterations begin.

1. *Interpolate to the grid*: Particle data are interpolated to the grid to obtain $\mathbf{M}_{\mathrm{g}}$, $\mathbf{v}_{\mathrm{g}}(t)$ and $\mathbf{Fext}_{\mathrm{g}}(t + \mathrm{d}t)$. The interpolations use the same equations as the explicit formulation, namely Equations (2)–(4).
2. *Initialization*: For the first iteration ($k = 1$), assume

$$\mathbf{u}_{\mathrm{g}}^{k-1}(t + \mathrm{d}t) = 0$$

Since $\mathbf{u}_{\mathrm{g}}$ refers to the displacement of the grid between t and $t + \mathrm{d}t$, setting $\mathbf{u}_{\mathrm{g}} = 0$ corresponds to a reset of the grid to its original undeformed configuration. The material

points remain in their deformed locations. The 'footprint' of the particles on the grid has thus changed and new estimates for $\mathbf{Fint}_g^{k-1}(t+dt)$ and $\mathbf{KK}_g^{k-1}(t+dt)$ are needed:

$$\mathbf{Fint}_g^{k-1}(t+dt) = \sum_e \int_{\Omega_e} \mathbf{B}_L^T \boldsymbol{\sigma}_p(t)\, dv$$
$$\mathbf{Kmat}_g^{k-1}(t+dt) = \sum_e \int_{\Omega_e} \mathbf{B}_L^T \mathbf{D}_p(t) \mathbf{B}_L\, dv$$
$$\mathbf{Kgeo}_g^{k-1}(t+dt) = \sum_e \int_{\Omega_e} \mathbf{B}_{NL}^T \boldsymbol{\sigma}_p(t) \mathbf{B}_{NL}\, dv$$
$$\mathbf{KK}_g^{k-1}(t+dt) = \mathbf{Kmat}_g^{k-1}(t+dt) + \mathbf{Kgeo}_g^{k-1}(t+dt) + \frac{4}{dt^2}\mathbf{M}_g$$

Here $\mathbf{B}_L$ and $\mathbf{B}_{NL}$ are the standard linear and non-linear strain–displacement matrices encountered in a displacement-based non-linear finite element formulation [5], composed of gradients of the trilinear shape functions evaluated at the current particle positions $\mathbf{x}_p$. $\mathbf{D}_p(t)$ is the spatial version of the fourth-order second elasticity tensor c [6] in Voigt notation. $\sum_e$ represents the element assembly, processing the grid point contributions into the global arrays.

3. *Solve for* $\mathbf{du}_g^k$: For iteration k, invert the following to get the current estimate for the displacement increment:

$$\mathbf{KK}_g^{k-1}(t+dt) \cdot \mathbf{du}_g^k = \mathbf{Fext}_g(t+dt) - \mathbf{Fint}_g^{k-1}(t+dt)$$

$$-\mathbf{M}_g\left(\frac{4}{dt^2}\mathbf{u}_g^{k-1}(t+dt) - \frac{4}{dt}\mathbf{v}_g(t) - \mathbf{a}_g(t)\right)$$

$$:= \mathbf{Q}_g^k$$

4. *Update kinematics on the grid*:

$$\mathbf{u}_g^k(t+dt) = \mathbf{u}_g^{k-1}(t+dt) + \mathbf{du}_g^k$$
$$\mathbf{v}_g^k(t+dt) = \frac{2}{dt}\mathbf{u}_g^k(t+dt) - \mathbf{v}_g(t)$$

5. *Update stress divergence and tangent stiffness on the grid*: The *total* deformation gradient $\mathbf{F}(t+dt)$ is decomposed into the deformation gradient up to time t, $\mathbf{F}(t)$, and the *incremental* deformation gradient from t to $t+dt$, $\mathbf{F}(dt)$, so that $\mathbf{F}(t+dt) = \mathbf{F}(dt)\mathbf{F}(t)$.

$$\nabla \mathbf{u}_p^k(t+dt) = \mathbf{G}_p \mathbf{u}_g^k(t+dt)$$
$$\mathbf{F}_p^k(t+dt) = \mathbf{F}_p^k(dt)\mathbf{F}_p(t) = (\nabla \mathbf{u}_p^k(t+dt) + \mathbf{1})\mathbf{F}_p(t)$$
$$\boldsymbol{\sigma}_p^k(t+dt) \text{ is determined from } \mathbf{F}_p^k \text{ and any relevant history variables}$$
$$\mathbf{D}_p^k(t+dt) \text{ follows from } \boldsymbol{\sigma}_p^k$$

Here $\mathbf{G}_p$ represents the gradient of the shape functions evaluated at $\mathbf{x}_p$. The internal force vector, material stiffness and geometric stiffness on the grid are obtained as defined by the expressions in Step 2.

6. *Convergence criteria*: Convergence is checked using Equations (13). If convergence has not been achieved, return to Step 3 above and continue iterations.

7. *Save converged state and update kinematics*:

Save $\mathbf{F}_p(t+dt)$, $\mathbf{Fint}_g(t+dt)$, $\mathbf{KK}_g(t+dt)$.

Compute $\mathbf{a}_g(t+dt)$ using Equation (16).

Interpolate $\mathbf{u}_g(t + \mathrm{d}t)$ and $\mathbf{a}_g(t + \mathrm{d}t)$ to the particles to find the particle displacement and acceleration, $\mathbf{u}_p(t + \mathrm{d}t)$ and $\mathbf{a}_p(t + \mathrm{d}t)$:

$$\mathbf{u}_p(t + \mathrm{d}t) = \sum_i S_{ip} \mathbf{u}_i(t + \mathrm{d}t)$$
$$\mathbf{a}_p(t + \mathrm{d}t) = \sum_i S_{ip} \mathbf{a}_i(t + \mathrm{d}t)$$

Update the particle position and velocity:

$$\mathbf{x}_p(t + \mathrm{d}t) = \mathbf{x}_p(t) + \mathbf{u}_p(t + \mathrm{d}t)$$
$$\mathbf{v}_p(t + \mathrm{d}t) = \mathbf{v}_p(t) + \tfrac{1}{2}(\mathbf{a}_p(t) + \mathbf{a}_p(t + \mathrm{d}t))\,\mathrm{d}t$$

8. *Continue to next time step*:
   Note that this incremental-iterative approach uses an *exact linearization* about time t. With appropriate numerical evaluation of the integrals in Step 2, an exact expression for the spatial version of the second elasticity tensor, and a time step size that ensures the solution remains within an appropriate radius of convergence, we should expect quadratic convergence from Newton's method. However, the nature of MPM is such that the material points will not in general correspond to the optimal sampling locations within elements of the computational grid. This may affect convergence via the accuracy of the integral evaluations. Monitoring particle distributions within the grid elements and adjusting either the grid geometry and/or particle distributions accordingly is likely to improve convergence behaviour and also increase the accuracy of converged solutions.

Step 3 involves the solution of a sparse symmetric linear system of equations. Both iterative (conjugate gradient) and direct solvers were implemented in the present work. The direct solver was much more robust than the iterative solver but also much slower. An iterative solution of the linear system was attempted at each time step, and if it failed, the direct solver was used. Although we have not performed a thorough investigation of the general ineffectiveness of the iterative solver, the likely cause is numerical ill-conditioning in the stiffness matrix.

The algorithm described here is for a fully dynamic solution. A quasi-static solution is readily obtained by simply eliminating the mass matrix contribution to the total stiffness matrix (Step 2) and removing the inertial (third) term in Step 3. Also, in what is described above, no assumptions have been made about the material model used to compute the Cauchy stress.

As mentioned in the Introduction, the algorithm described above is very similar to an incremental-iterative solution strategy used with implicit time integration when using the FE method. The material points serve many of the same functions as the integration points in an FE solution. If the grid was not reset after each time step, the approach would be exactly the same as the FE method. However, it is largely this resetting of the mesh that distinguishes MPM from FEM by avoiding mesh distortion issues. Because the mesh is reset in MPM, each time step begins with interpolation of the particle data to the computational grid. This step defines the time t data on the mesh, whereas in FEM the time t data comes from the previous time step's solution. Construction of the initial mesh is trivial with MPM and mesh entanglement is completely avoided.

The migration of material points can create some problems for the robustness of the solution. Whereas a finite element will always have the same number of integration points throughout a solution and these points will typically be located at the optimal sampling points

for Gauss integration, the number of material points in a computational cell in MPM varies. The problems that this fluctuation creates are mitigated by using an increased particle density, leading to a more robust and accurate solution. Improvements in MPM accuracy could likely be made by controlling the times at which the computational grid is reset, thereby ensuring that the material points remain optimally located until grid entanglement is imminent.

4. CONSTITUTIVE FRAMEWORK

Hyperelasticity was chosen as the primary constitutive framework for our MPM code because it is objective and path independent for arbitrarily large deformations and it is readily extended to other material behaviours. Further, constitutive formulations based on energy functionals ensure that a so-called 'consistent tangent' [7] will always be available to ensure optimal convergence behaviour during the Newton iterations, and the resulting elasticity tensor will always lead to a symmetric tangent stiffness matrix.

The strain energy was decoupled into volumetric and deviatoric components via the use of the deviatoric deformation gradient $\tilde{\mathbf{F}}$ and the deviatoric version of the right Cauchy–Green deformation tensor $\tilde{\mathbf{F}}$ [8, 9]:

$$W = U(J) + \tilde{W}(\tilde{\mathbf{C}}) \tag{17}$$

This assumption ensures that decoupling of the pressure variable from the stress is trivial so that alternate multi-field variational principles can be applied for future analyses of nearly and fully incompressible material behaviour. The Cauchy stress is then

$$\boldsymbol{\sigma} = \frac{2}{J}\mathbf{F}\frac{\partial W}{\partial \mathbf{C}}\mathbf{F}^{\mathrm{T}} = p\mathbf{1} + \frac{2}{J}\operatorname{dev}\left[\tilde{\mathbf{F}}\frac{\partial \tilde{W}}{\partial \tilde{\mathbf{C}}}\tilde{\mathbf{F}}^{\mathrm{T}}\right] \tag{18}$$

where the identification $p = \partial U/\partial J$ has been made and the operator 'dev' extracts the deviatoric part of a second-order tensor in the spatial configuration

$$\operatorname{dev}[\,\cdot\,] = [\,\cdot\,] - \tfrac{1}{3}\left([\,\cdot\,]:\mathbf{1}\right)\mathbf{1} \tag{19}$$

A compressible neo-Hookean constitutive model was employed in the examples that follow:

$$\tilde{W}(\tilde{\mathbf{C}}) = \frac{\mu}{2}(\tilde{I}_1 - 3), \quad U(J) = \frac{\kappa}{2}\,[\ln(J)]^2 \tag{20}$$

Here, μ is the shear modulus, κ is the bulk modulus, and $\tilde{I}_1 = \operatorname{tr}(\tilde{\mathbf{C}})$ is the first invariant of the deviatoric version of the right deformation tensor. The spatial version of the second elasticity tensor is, after multiple applications of the chain rule

$$\mathsf{c} = (JU')'J\mathbf{1}\otimes\mathbf{1} - 2JU'\mathbf{I} + 2\mu\left[\mathbf{I} - \frac{1}{3}\,\mathbf{1}\otimes\mathbf{1}\right] - \frac{2J}{3}\,[\mathbf{s}\otimes\mathbf{1} + \mathbf{1}\otimes\mathbf{s}] \tag{21}$$

where $\mathbf{s}$ is the deviatoric part of the Cauchy stress [10].

5. NUMERICAL EXAMPLES

5.1. Compression of an elastic billet

The first test problem consisted of a billet of hyperelastic material bonded between two plates. The plates move toward each other and the material is compressed, bulging out and eventually contacting the plates from the sides (Figure 1(a) and 1(b)). The objective was to achieve 50% compression. This is a common test problem in non-linear solid mechanics and demonstrates both large deformations and contact. Although these analyses were carried out in two dimensions using plane strain assumptions for ease of illustration, our IMPM implementation is completely three dimensional.

The total size of the billet was 40.0×40.0 cm. In all cases, the platen was represented as rigid and a quarter-symmetry model was employed. Unless otherwise indicated below, the material properties were assigned as $\mu = 6.0e6$ dyne/cm^2, $\kappa = 6.0e7$ dyne/cm^2 (Poisson's ratio $v = 0.45$), and $\rho = 10.0$ g/cm^3.

5.1.1. Static compression of an elastic billet. For this test, solutions obtained using IMPM were compared to solutions from NIKE3D, a standard FE code using implicit time integration [11]. Because this problem was quasi-static, the mass matrix contribution to the stiffness matrix was neglected and the inertial (third) term in Equation (11) was also eliminated.
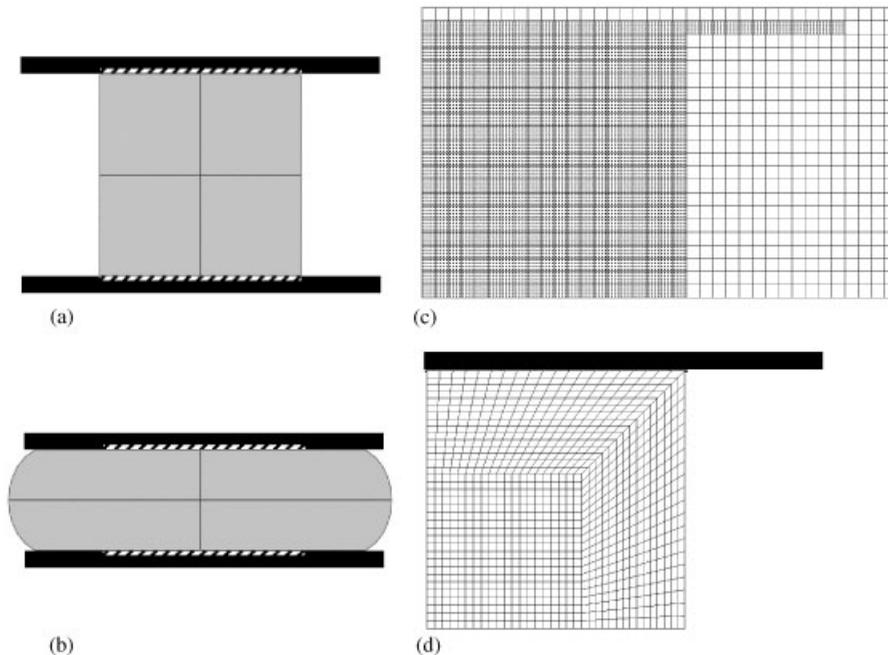
Figure 1. Compression of a billet: (a) initial state; (b) compressed state. Initial configurations used in; (c) the material point method solution; and (d) the finite element solution.

The IMPM computational grid had $0.5\,\mathrm{cm}^2$ cell size, and the movement of the platen was discretized in quasi-time so that it would move through half of a grid cell during each time step. To accommodate the large strains experienced by the billet, a particle distribution of four particles along both the X and Y directions ($4X4$) was used (Figure 1(c)). This particle density ensured that, even at large deformations, each cell would have a reasonable number of particles. Element size for the FE mesh was approximately 0.3–0.5 cm (Figure 1(d)) and an automatic time stepping strategy was employed. Trilinear hexahedral elements were used, with the boundary conditions adjusted to ensure plane-strain deformations.

Figure 2 shows the deformed billet at 50% compression using IMPM and FEM. In both, colours indicate von Mises stress. In Figure 2(a), the particle locations are shown to depict the deformed shape of the billet. Both the shape of the billet and the stress contours within the billet indicate excellent agreement between the two methods. The horizontal displacements of the lower right particle (IMPM) and node (FEM) were 7.38 and 7.23 cm, a 2.1% difference. Note that there is a gap between the particles of the billet and those of the platen in the deformed MPM particle geometry. This is a result of the way contact between bodies is represented with MPM. Contact occurs when information from the two bodies is interpolated to common nodes. While each field remains independent, this interaction may begin to take place when the particles are as much as two grid cells apart.

In general, we found that a higher density of particles per cell lead to a more accurate and robust solution. Lower particle densities (e.g. $2X2$), yielded good results when deformations were small, but the use of higher particle densities was required to achieve accurate results at larger deformations. For this particular example, convergence was not obtained beyond 30% compression when a $2X2$ particle distribution was used. Converged solutions were achieved
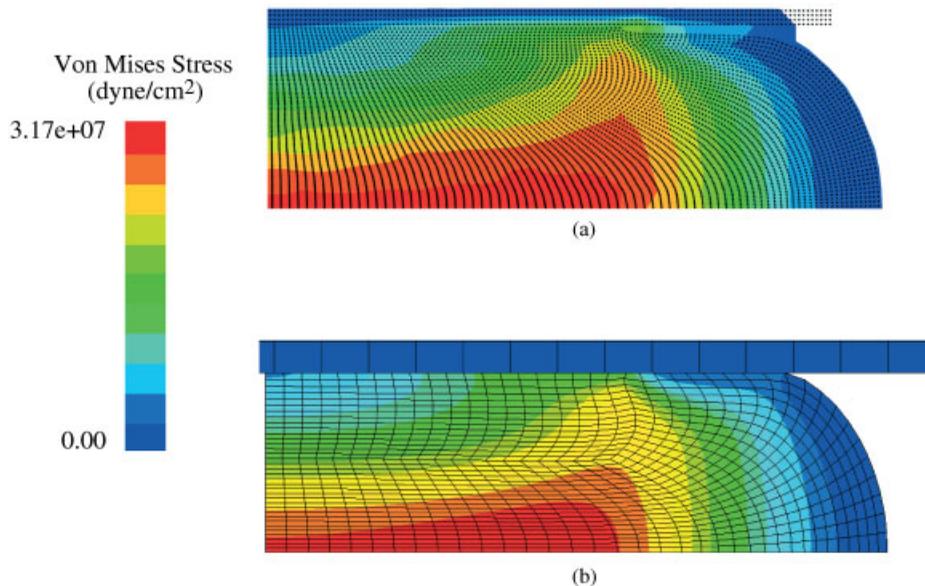
Figure 2. Elastic billet compressed quasi-statically to 50% using: (a) the material point method; and (b) the finite element method. Colours indicate von Mises stress on the grid.

Int. J. Numer. Meth. Engng 2003; **57**:1323–1338

with particle distributions of $3X2$ and $3X3$, but the quality of the solution improved noticeably when the distribution was changed from $3X3$ to $4X4$. There is additional computational expense and storage associated with increased particle density, but the number of equations in the linear system remains unchanged since only the grid nodes contribute degrees of freedom to the system of equations. This observation indicates the need for particle refinement techniques.

5.1.2. Dynamic compression of an elastic billet. The same problem was analysed with full dynamic effects using IMPM, EMPM and the FEM. The downward velocity of the platen was specified as 50 cm/s. Again, a $4X4$ particle distribution was used.

Results for the three simulations are shown in Figure 3. As in the static case, results for the IMPM solution and the FEM solution compared quite favourably. Maximum effective stress differed by 4.3% between these two cases, and the horizontal displacements of the lower right particle (IMPM) and node (FEM) were 7.44 and 7.26 cm, a 2.5% difference. A comparison of the IMPM and EMPM results indicated reasonable agreement. Here, the maximum effective stress differs by 14% and the horizontal displacement of the lower right particle for the EMPM case is 7.94 cm, 6.7% greater than the IMPM result. It is worth noting that the comparisons between results for the IMPM and EMPM are much better at smaller amounts of platen compression. With increasing compression and the associated larger deformations experienced by the particles, the explicit results deviated farther from the FEM solution.

The results of these comparison illustrate the distinct advantage of IMPM over EMPM for low rate dynamic problems. Generally, in problems where the kinetic energy is small in comparison to the strain energy, we have found that IMPM yields more accurate results than EMPM. The increasing deviation of the EMPM solution from the FEM and IMPM solutions likely has two main causes. First, with any explicit method, since there is no iterative procedure
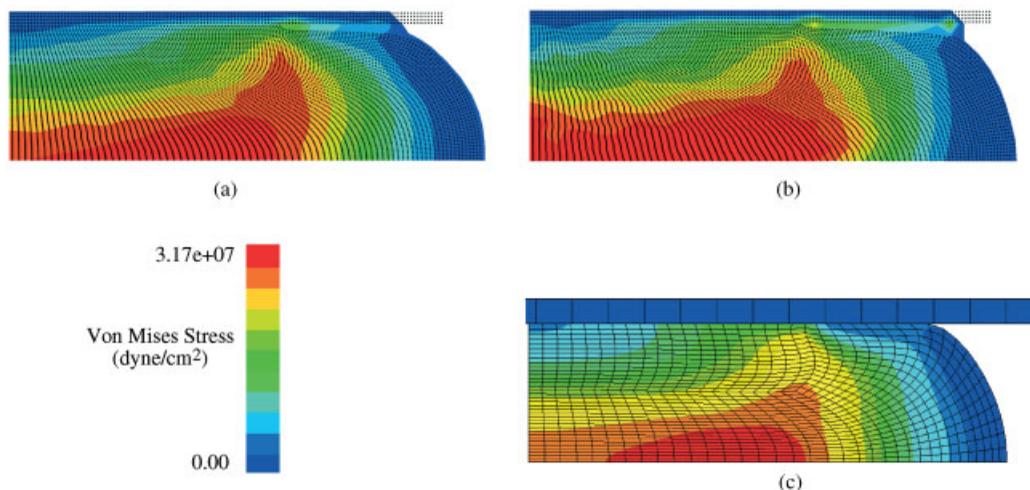
Figure 3. Elastic billet compressed dynamically to 50% using: (a) the implicit material point method; (b) the explicit material point method; and (c) the finite element method. Colours indicate von Mises stress on the grid.

to minimize the error in each time step, error builds up over subsequent time steps. As expected, it was observed that lowering the time step size for the explicit calculations yielded improved results. The second, more destructive effect is particular to MPM. Specifically, as particles move through the computational grid, adjacent cells may contain differing numbers of particles. It has been shown [12] that this can lead to an imbalance in internal force at the nodes that is proportional to the difference in the number of particles in the neighbouring cells. When the internal forces are small relative to the inertia, this artifact is not significant. However, for quasi-static solutions this artifact can be so large that the explicit solution can be destroyed. For example, attempts to use a lower particle density resulted in failure to reach a solution with the EMPM algorithm. The implicit algorithm was able to reach a converged solution for particle distributions of $3X2$ and $3X3$ for this case. Because of the iterative nature of the implicit algorithm, IMPM is much less susceptible to this problem. Given the difficulty in obtaining a solution with the EMPM code for this case and the excellent agreement between the IMPM and FEM results, the implicit solution strategy is clearly the better choice for this problem.

The time-step size advantage of IMPM is also demonstrated by this example. Time-step size for EMPM was selected automatically each time step based on the Courant stability criterion by monitoring material wave speed and particle velocities, and was approximately 7×10^{-5} s. As in the static case, the time-step size for IMPM was chosen to allow movement of the platen through one-half of a computational cell during each time step. This choice is based on the limitations of deformation of the computational grid during a single iteration of the Newton method. If the platen was allowed to completely crush or invert a cell in a single time step, Newton iterations diverged and a solution could not be obtained. For this example, the time step was on the order of 5×10^{-3} s, or more than 70 times larger than the explicit time-step size.

EMPM time steps required 0.39 s of CPU time on one processor of an SGI Origin 200 (225 MHz), while IMPM time steps required 121 s on the same platform, a factor of 309 larger. Considering only time to solution, to be considered 'efficient' it would be necessary to take implicit time steps at least 309 times larger than the explicit time step. Thus, for a stiffer billet material or a slower platen velocity, IMPM would provide both a more efficient *and* a more accurate solution.

5.2. Stress waves in granular media

Stress wave propagation in granular materials is a problem of significant research interest, and EMPM has been applied successfully to this class of problems [3, 13, 14]. This is in part due to the importance of contact between individual grains, which MPM is able to handle with relative ease. Bardenhagen [14] simulated four co-linear disks subjected to loading via a Hopkinson bar with EMPM and compared predictions to results from photoelasticity experiments carried out by Roessig [15, 16]. Good qualitative agreement between the experimental and computational results was demonstrated. More recently, Cummins and Brackbill [13] noted that noise frequently appears in the EMPM solution. They modelled a similar version of this problem using an implicit MPM algorithm that is different from that described here in that it uses a Newton–Krylov solution technique. While their results were not compared to experimental measurements, the noise in the solution was significantly diminished with the use of their implicit strategy.

IMPM was used to study this same problem and comparisons were made to EMPM and experimental measurements. Specifically, four 1 cm disks with material properties approximating Plexiglas were arranged co-linearly edge to edge. Material properties were: $\mu = 72$ GPa, $\kappa = 102$ GPa and $\rho = 1900$ kg/m^3. This resulted in a longitudinal wave speed of 1 cm/µs. The velocity of the striker that loaded the disks was 5.6 m/s. The problem was modelled in two dimensions, assuming plane stress. Fifty computational cells were used across the diameter of each disk (Bardenhagen used 80 cells for the explicit calculations). A particle distribution of $2X2$ was used for the IMPM calculations. IMPM time-step size for was chosen to correspond to the time for a longitudinal wave to cross one-half of a computational cell.

In order to numerically simulate the technique of photoelasticity, fringe patterns were generated by first calculating the in-plane principal stresses, σ_1 and σ_2. The fringe intensity F was then computed as [13]:

$$F = \cos\left(\frac{2\pi f_{\mathrm{no}}(\sigma_1 - \sigma_2)}{\max(\sigma_1 - \sigma_2)}\right) \qquad (22)$$

For this work, a value of 7 was chosen for f_{no}, the fringe number. This value was selected to give the best visual match to the experimental results.

A sample of the results from the current work, the EMPM results of Bardenhagen and the experimental results of Roessig are shown in Figure 4 at three different times. Frame times are non-dimensionalized by the wave transit time across a single disk [3]. Owing to experimental uncertainties in both the time of arrival of the striker and the material properties of the disks, Bardenhagen chose matching frames by eye, subject to the constraint of a constant time contraction factor for all frames. Thus, the experimental results in Figure 4 are from non-dimensional times of 2.6, 6.6 and 9.2, while the explicit simulation results are at times of 2.2, 5.5 and 7.7. Subjecting the current results to the same constraints as those used by Bardenhagen, the implicit simulation results are at time of 1.9, 4.8 and 6.7. The difference in scaling factors between the explicit and implicit results may be due to the difference in the resolution used for the calculations. The contact between adjacent disks is modelled more accurately with increased resolution.

Good agreement is again noted between the IMPM, EMPM and experimental results. The implicit solution shows less noise than the explicit results. While a relatively small time-step size was used for the results shown here, tests on similar problems have indicated that a time-step size at least 4 times larger gives results that are practically indiscernible from those obtained under the time-step constraint used here. Because of the desire to resolve the stress wave propagation, a larger implicit time-step size was not tested for this case.

6. CONCLUSIONS

An implicit integration strategy was developed and implemented for the material point method. Solutions from the IMPM algorithm were compared to those obtained with the implicit finite-element code NIKE3D. Very favourable quantitative agreement was demonstrated. Comparisons with the traditional EMPM strategy were also made and advantages of IMPM were identified for certain classes of problems.

Note that times to solution between the IMPM code and NIKE3D were not presented. Our IMPM code was written to demonstrate a new algorithm, whereas NIKE3D is a production FE
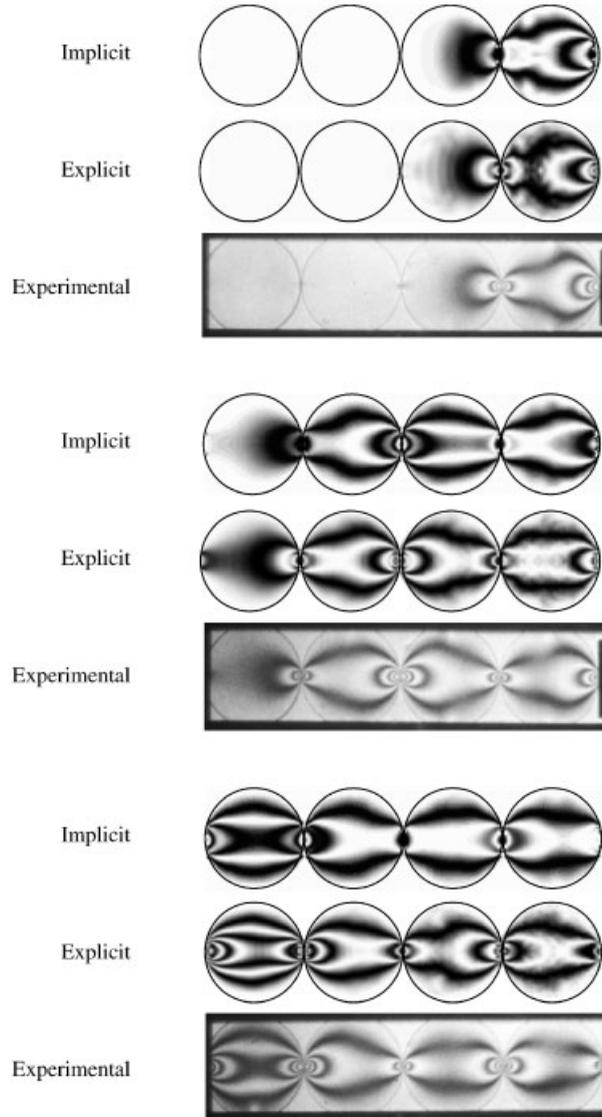
Figure 4. Stress wave propagation through four co-linear disks. Matching frames are presented together with the implicit results on top, explicit results in the middle and experimental results on the bottom. Non-dimensional frame times are 1.9, 4.8 and 6.7 for the implicit results, 2.2, 5.5 and 7.7 for the explicit results and 2.6, 6.6 and 9.2 for the experimental results.

code that has undergone over 20 years of continuous development and optimization. Suffice it to say that, at this time, solution times with NIKE3D are much shorter than those with IMPM for the same simulation. The implicit FEM will likely continue to be faster even after optimization as the IMPM (and EMPM) requires a number of additional interpolation steps between the particles and the computational grid that are not necessary with implicit FEM.

However, the advantage of avoiding mesh entanglement with IMPM clearly makes it the superior choice for certain problems that demonstrate extreme deformation and contact.

The principal advantage of IMPM over EMPM is the accommodation of much larger time steps. Time steps hundreds of times larger than that dictated by the CFL condition have been used successfully. The only time-step restriction for IMPM is related to the magnitude of cell deformation in the computational grid during a single iteration of the Newton method. An additional significant benefit of IMPM is that it frequently performs much better in both its ability to obtain a solution and in the accuracy of that solution than its explicit counterpart, particularly for quasi-static loading scenarios.

The implicit solution strategy described here differs substantially from another implicit MPM technique recently reported by Cummins and Brackbill [13]. Their work utilized a Newton–Krylov solver. This 'matrix-free' technique avoids the inversion of a large linear system of equations at each iteration, which is the most costly part of the algorithm described here. However, the reported time step restrictions associated with their method were rather severe, much more so than in the present work. Specifically, they report that an inability to achieve a solution for time steps greater than 10 times the explicit time step size (when using a CFL number of 0.25 for the explicit case). Additionally, as in this work, they limit the time step size based on strain, but while the method reported here is limited by a strain of 50% per time step, that of Cummins and Brackbill is limited to 1% strain per time step. The ability of the presently reported method to use larger time steps is likely due to the fact that our approach employs a consistent tangent, providing the best convergence behaviour possible with Newton methods in the context of implicit MPM analysis.

Because of the similarities between MPM and FEM, the implicit solution strategy described here should be easily modifiable to accommodate quasi-Newton solution methods. The BFGS method introduced by Matthies and Strang [16] is an obvious choice as it has proven to be robust and computationally efficient for a wide range of non-linear problems in solid mechanics. Implementation of BFGS will reduce the number of times a large linear system will need to be solved, and should further improve the efficiency of this technique.

REFERENCES

1. Sulsky D, Chen Z, Schreyer HL. A particle method for history dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:179–196.
2. Sulsky D, Zhou S, Schreyer HL. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**:236–252.
3. Bardenhagen SG, Brackbill JU, Sulsky D. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering* 2000; **187**:529–541.
4. Kashiwa BA, Lewis MW. Fluid-structure interaction modeling. *LA-13255-PR*, 1997; 283–295.
5. Bathe K-J. *Finite Element Procedures*. Prentice-Hall: New Jersey, 1996.
6. Marsden JE, Hughes TJR. *The Mathematical Foundations of Elascitity*. Dover Publications, Inc.: New York, 1993.
7. Simo JC, Taylor RL. Consistent tangent operators for rate-independent elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* 1985; **48**:101–118.

8. Flory PJ. Thermodynamic relations for high elastic materials. *Transactions of the Faraday Society* 1961; **57**: 829–838.
9. Simo JC, Taylor RL, Pister KS. Variational and projection methods for the volume constraint in finite deformation elastoplasticity. *Computer Methods in Applied Mechanics and Engineering* 1985; **51**:177–208.
10. Simo JC, Hughes TJR. *Computational Inelasticity*. Springer-Verlag, New York, 1998.
11. Maker BN, Ferencz RM, Hallquist JO. NIKE3D: A nonlinear, implicit, three-dimensional finite-element code for solid and structural mechanics. *Lawrence Livermore National Laboratory Technical Report*, *UCRL-MA-105268*, 1990.
12. Zhou S. *The Numerical Prediction of Material Failure Based on the Material Point Method*. PhD Dissertation, Department of Mechanical Engineering, University of New Mexico, 1998.
13. Cummins SJ, Brackbill JU. An implicit particle-in-cell method for granular materials. *Journal of Computational Physics* 2002; **180**:506–548.
14. Bardenhagen SG, Guilkey JG, Roessig KM, Brackbill JU, Witzel WM, Foster JC. An improved contact algorithm for the material point method and applications to stress propagation in granular material. *Computer Modeling in Engineering and Sciences* 2001; **2**:509–522.
15. Roessig KM. Mesoscale mechanics of plastic bonded explosives. *Proceedings of the American Physical Society Topical Group on Shock Compression of Condensed Matter*, 2001.
16. Roessig KM, Foster JC, Bardenhagen SG. Dynamic stress chain formation in a two-dimensional particle bed. *Experimental Mechanics* 2002; **42**(3):329–337.
17. Matthies H, Strang G. The solution of nonlinear finite-element equations. *International Journal for Numerical Methods in Engineering* 1979; **14**:1613–1626.