ELSEVIER

Computer Physics Communications 87 (1995) 236–252

Computer Physics Communications

Application of a particle-in-cell method to solid mechanics

Deborah Sulsky[a], Shi-Jian Zhou[b], Howard L. Schreyer[b]

[a] *Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131, USA*

[b] *Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM 87131 ,USA*

Received 26 April 1994; revised 16 June 1994

Abstract

An extension to solid mechanics of the FLIP particle-in-cell method is presented. The particle-in-cell method uses two representations of the continuum, one based on a collection of material points and the other based on a computational grid. The material points are followed throughout the deformation of a solid and provide a Lagrangian description that is not subject to mesh tangling. This feature permits constitutive equations with history-dependent variables to be applied at these material points with no requirement for mapping the history parameters from one point to another. A grid, which can be held fixed or adapted as the need arises, is used to determine spatial gradients. Since the grid is used as an updated Lagrangian frame, the nonlinear convection term associated with Eulerian formulations does not appear. With the use of maps between material points and the grid, the advantages of both Eulerian and Lagrangian schemes are utilized. No-slip impact between bodies, inelastic, elastic, or rigid, is handled automatically by the method without resorting to a special contact algorithm.

1. Introduction

Problems in solid mechanics that involve history-dependent constitutive models, like large deformation plasticity, are expressed more naturally in a Lagrangian computational frame. Similarly, a Lagrangian description is useful when following material free surfaces or multiple materials. On the other hand, if large deformations are involved, a purely Lagrangian mesh can become tangled or require unreasonably small time steps to be successful. Remeshing is problematic where history-dependence is involved. For these reasons, particle methods for solid mechanics seem to be increasing in popularity [1,2,9,10].

This paper describes an extension to solid mechanics of FLIP [3–8], a particle-in-cell (PIC) method for fluid flow. FLIP is a fully Lagrangian particle method that uses two discretizations of the material, one based on a computational mesh and the other based on a collection of material points or "particles". This approach combines the advantages of Eulerian and Lagrangian descriptions of the material while avoiding the shortcomings of each. In FLIP, equations of motion are solved in a Lagrangian frame on a computational grid, using standard finite difference or finite element methods. Convection is modeled by moving the material points in the computed velocity field. Each numerical material point carries its material properties without error while it is convected. Since all the properties of the continuum are assigned to the numerical material points, the information carried by these points is enough to characterize the flow and the grid carries no permanent information. Thus, the

SSDI 0010-4655(94)00170-7

grid can be discarded and reconstructed for computational convenience each time step. This allows free use of adaptive, Eulerian, or Lagrangian meshes, as each problem dictates. Because the particles provide a Lagrangian material description, FLIP successfully tracks material discontinuities and can model highly distorted flows.

The properties that make FLIP attractive for fluid flows also make it appealing for some problems in solid mechanics. It is a low dissipation method that is particularly suited to problems involving interactions of objects moving relative to one another since the particles naturally track the position of each object. The particles also provide a means for recording the deformation history of material points for use in history-dependent constitutive equations. Potential applications include a wide range of engineering problems, including collision and rebound, impact, penetration and perforation, as well as fluid-solid interactions. Some of these capabilities have already been demonstrated [8,13–17].

In Sulsky, et al. [15], we give a weak formulation of the FLIP algorithm for solid mechanics and frame the method in terms of finite elements. The method has proven successful in treating problems with large displacements and deformations [15] and has been applied to problems with multiple materials and history-dependent constitutive equations [16,17], as well as to wave propagation [14]. The governing equations are given in the next section and their discretization in Section 3. Numerical implementation is discussed in Section 4, and in Section 5 numerical examples are presented.

2. Governing equations

The governing equations are standard conservation equations for mass and momentum

$$\frac{d\rho}{dt} + \rho\nabla\cdot\mathbf{v} = 0\,, \tag{1}$$

$$\rho\mathbf{a} = \nabla\cdot\sigma + \rho\mathbf{b}\,. \tag{2}$$

In these equations, $\rho(\mathbf{x},t)$ is the mass density, $\mathbf{a}(\mathbf{x},t)$ is the acceleration, $\mathbf{v}(\mathbf{x},t)$ is the velocity, $\sigma(\mathbf{x},\mathbf{t})$ is a symmetric stress tensor, and $\mathbf{b}(\mathbf{x},t)$ is the specific body force. Generally, an energy equation would also be necessary but in this paper thermal effects are neglected.

The momentum equation is solved in a Lagrangian frame on a finite element mesh. The quantity $\mathbf{x}$ is the current position at time t of the material that started at position $\mathbf{X}$ at time zero; *i.e.*, with $\mathbf{x} = \mathbf{x}(\mathbf{X},t)$, $\mathbf{x}(\mathbf{X},0) = \mathbf{X}$. With the use of the material time derivative

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla, \tag{3}$$

the acceleration is the time derivative of the velocity, $\mathbf{v}(\mathbf{x},t)$,

$$\mathbf{a}(\mathbf{x},t) = \frac{d\mathbf{v}}{dt}(\mathbf{x},t) \tag{4}$$

and the displacement $\mathbf{u}(\mathbf{x},t)$ satisfies

$$\mathbf{v}(\mathbf{x},t) = \frac{d\mathbf{u}}{dt}(\mathbf{x},t). \tag{5}$$

The displacement can also be defined as $\mathbf{u}(\mathbf{x},t) = \mathbf{x} - \mathbf{X}(\mathbf{x},t)$, where $\mathbf{X}(\mathbf{x},t)$ is the inverse function of $\mathbf{x}(\mathbf{X},t)$.

Equations are solved numerically in a two-dimensional region $\Omega \subseteq \mathbf{R}^2$, given initial conditions and appropriate conditions on the boundary $\partial\Omega$. If $\mathbf{n}$ denotes the unit outward normal to the boundary, boundary conditions take the form

$$\mathbf{u}(\mathbf{x},t) = \mathbf{g}(t) \quad \text{on} \quad \partial\Omega_u, \tag{6}$$

and

$$\sigma(\mathbf{x},t)\cdot\mathbf{n}=\tau(t)\quad\text{on }\partial\Omega_\tau. \tag{7}$$

In the above conditions, the boundary has been divided into two disjoint pieces $\partial\Omega=\partial\Omega_u\cup\partial\Omega_\tau$. On the first section of the boundary, displacement is prescribed, and on the second portion a prescribed traction τ is applied.

The continuity equation Eq. (1) and the momentum equation Eq. (2) must be supplemented with a constitutive equation for the stress. For simplicity, constitutive equations are presented in terms of small deformation theory, although more general forms have been implemented [16]. With the small deformation assumption, there is no need to select from the various strain and stress tensors, or gradient operators associated with large deformation theory. Define a strain tensor **e** such that the strain rate is

$$\frac{d\mathbf{e}}{dt}=\tfrac{1}{2}\left\{\nabla\mathbf{v}+(\nabla\mathbf{v})^T\right\}. \tag{8}$$

Then, the constitutive equation relating stress rate to strain rate can be written in terms of the tangent modulus, **T**, a fourth-order tensor defined such that

$$\frac{d\sigma}{dt}=\mathbf{T}:\frac{d\mathbf{e}}{dt}. \tag{9}$$

For linear elastic materials, **T** is just the usual elasticity tensor, $\mathbf{T}^e$. For elastic-plastic materials the tangent modulus is more complicated, and is defined in the Appendix. In either case, Eq. (9) gives a general form of the constitutive equation for many materials.

For the derivation of the discrete equations given in the next section, it is convenient to introduce a specific stress, σ^s, and specific tangent modulus, $\mathbf{T}^s$, so that

$$\rho\sigma^s=\sigma \tag{10}$$

and

$$\rho\mathbf{T}^s=\mathbf{T}. \tag{11}$$

3. Weak form of the governing equations

This section describes the discrete formulation of the problem and prescribes the spatial and temporal integration of the equations. In the Lagrangian view of continuum mechanics, the material is divided into infinitesimal mass elements. Each mass element contains a fixed amount of mass for all time. In FLIP, these infinitesimal mass elements are represented by a finite collection of N_p material points, with fixed mass M_p, $p=1,\ldots,N_p$, that are followed throughout the history of the material deformation. These points are initially located at $\mathbf{X}_p(0)$ at time $t=0$ and at $\mathbf{X}_p(t)$ at time t. Mass density can then be written as a sum of point masses

$$\rho(\mathbf{x},t)=\sum_{p=1}^{N_p}M_p\delta(\mathbf{x}-\mathbf{X}_p(t)), \tag{12}$$

where δ is the Dirac delta function. Since M_p is fixed, mass conservation Eq. (1) is automatically satisfied.

In order to obtain the weak form of the momentum equation, multiply Eq. (2) by the test function **w** and integrate over the current configuration, Ω,

$$\int_\Omega\rho\mathbf{w}\cdot\mathbf{a}\,d\Omega=-\int_\Omega\rho\sigma^s:\nabla\mathbf{w}\,d\Omega+\int_{\partial\Omega_\tau}\mathbf{w}\cdot\tau\,dS+\int_\Omega\rho\mathbf{w}\cdot\mathbf{b}\,d\Omega. \tag{13}$$

Differential volume and surface elements are denoted by $d\Omega$ and dS, respectively. To derive this form of the equation, integration by parts and the divergence theorem are applied to the term involving the stress, and the traction boundary condition is invoked. It is assumed that $\mathbf{w}$ is zero on $d\Omega_u$

The substitution of Eq. (12) into Eq. (13) converts integrals to sums of quantities evaluated at material points

$$\sum_{p=1}^{N_p} M_p \mathbf{w}(\mathbf{X}_p(t),t) \cdot \mathbf{a}(\mathbf{X}_p(t),t) = -\sum_{p=1}^{N_p} M_p \sigma^s(\mathbf{X}_p(t),t) : \nabla \mathbf{w}(\mathbf{x},t)|_{\mathbf{x}=\mathbf{X}_p(t)}$$

$$+ \int_{\partial\Omega_\tau} \mathbf{w} \cdot \tau dS + \sum_{p=1}^{N_p} M_p \mathbf{w}(\mathbf{X}_p(t),t) \cdot \mathbf{b}(\mathbf{X}_p(t),t). \tag{14}$$

Now, to continue the discretization, suppose a computational grid is constructed of isoparametric, 4-node, quadrilateral elements. These elements are then used to define standard nodal basis functions, $N_i(\mathbf{x})$, associated with spatial nodes $\mathbf{x}_i(t)$, $i = 1,\dots,N_n$, with N_n being the total number of nodes. The nodal basis functions are assembled from conventional finite element shape functions. The coordinates of any point can be written compactly using the nodal basis functions as,

$$\mathbf{x} = \sum_{i=1}^{N_n} \mathbf{x}_i(t) N_i(\mathbf{x}). \tag{15}$$

If the displacements in the element are defined by the nodal displacements, $\mathbf{u}_i(t)$, then the displacement has a similar representation

$$\mathbf{u}(\mathbf{x},t) = \sum_{i=1}^{N_n} \mathbf{u}_i(t) N_i(\mathbf{x}). \tag{16}$$

Since the same basis functions apply to both spatial coordinates and displacements, the material time derivative of the nodal basis functions must vanish. Therefore, the velocity and acceleration are represented by

$$\mathbf{v}(\mathbf{x},t) = \sum_{i=1}^{N_n} \mathbf{v}_i(t) N_i(\mathbf{x}) \tag{17}$$

and

$$\mathbf{a}(\mathbf{x},t) = \sum_{i=1}^{N_n} \mathbf{a}_i(t) N_i(\mathbf{x}). \tag{18}$$

The test function $\mathbf{w}$ also has this form,

$$\mathbf{w}(\mathbf{x},t) = \sum_{i=1}^{N_n} \mathbf{w}_i(t) N_i(\mathbf{x}). \tag{19}$$

The representations, Eqs. (15)–(19), ensure that the associated quantities are continuous.

The numerical solution will be obtained at a discrete set of times, t^k, $k = 1,\dots K$. Substituting Eq. (18) and Eq. (19) into Eq. (14) and evaluating the result at time t^k yields the equation

$$\sum_{i=1}^{N_n} \mathbf{w}_i^k \cdot \sum_{j=1}^{N_n} m_{ij}^k \mathbf{a}_j^k = -\sum_{i=1}^{N_n} \mathbf{w}_i^k \cdot \sum_{p=1}^{N_p} M_p \sigma_p^{s,k} \cdot \nabla N_i(\mathbf{x})|_{\mathbf{x}=\mathbf{X}_p^k} + \sum_{i=1}^{N_n} \mathbf{w}_i^k \cdot \hat{\tau}_i^k + \sum_{i=1}^{N_n} \mathbf{w}_i^k \cdot \mathbf{b}_i^k. \tag{20}$$

In Eq. (20), a superscript k denotes evaluation at time t^k; so, for example $\mathbf{w}_i^k$ means $\mathbf{w}_i(t^k)$. Also in Eq. (20), a mass matrix has been introduced,

$$m_{ij}^k = \sum_{p=1}^{N_p} M_p N_i(\mathbf{X}_p^k) N_j(\mathbf{X}_p^k). \tag{21}$$

The discrete applied traction is

$$\hat{\tau}_i^k = \int_{\partial\Omega_\tau} N_i(\mathbf{x}) \tau(\mathbf{x}, t^k) \, dS, \tag{22}$$

while the specific stress at a material point position is

$$\sigma_p^{s,k} = \sigma^s(\mathbf{X}_p^k, t^k). \tag{23}$$

The specific body force is discretized as

$$\mathbf{b}_i^k = \sum_{p=1}^{N_p} M_p \mathbf{b}(\mathbf{X}_p, t^k) N_i(\mathbf{X}_p^k). \tag{24}$$

The components of $\mathbf{w}_i^k$ are arbitrary except where the components of displacement are prescribed. With the understanding that the proper constraints on displacement are invoked, the weak form of the equation of motion reduces to

$$\sum_{j=1}^{N_n} m_{ij}^k \mathbf{a}_j^k = \mathbf{f}_i^{\mathrm{int},k} + \mathbf{f}_i^{\mathrm{ext},k}, \tag{25}$$

where the internal force vector is

$$\mathbf{f}_i^{\mathrm{int},k} = -\sum_{p=1}^{N_p} M_p \mathbf{G}_{ip}^k \cdot \sigma_p^{s,k}, \tag{26}$$

and the external force vector is

$$\mathbf{f}_i^{\mathrm{ext},k} = \mathbf{b}_i^k + \hat{\tau}_i^k. \tag{27}$$

To simplify the notation, the gradient of the nodal basis function evaluated at $\mathbf{X}_p^k$ is denoted by

$$\mathbf{G}_{ip}^k = \nabla N_i(\mathbf{x})|_{\mathbf{x}=\mathbf{X}_p^k} \tag{28}$$

in the expression for the internal forces. Eq. (25) has a form similar to that obtained by traditional finite element schemes; however, the mass matrix m_{ij}^k varies with time and therefore must be computed each time step.

The complete algorithm consists of the following steps, an initialization phase where information is transferred from the material points to a grid; a Lagrangian phase where the equations of motion are solved in an updated Lagrangian frame on the grid; followed by a convective phase where the particles are updated and the grid is redefined.

In order to solve the discrete momentum equation, information must be gathered from the material points for the initialization phase. The mass matrix must be formed, and the internal force accumulated at the nodes using the specific stresses at the material points and the gradient of the nodal basis functions. Since there are generally more material points than grid points, a weighted least squares approach is used to determine nodal velocities from the velocities at the material points to initialize each time step. The weighting is the mass of the material point. The result is the following equation which must be solved for the nodal velocities, $\mathbf{v}_j^k$,

$$\sum_{j=1}^{N_n} m_{ij}^k \mathbf{v}_j^k = \sum_{p=1}^{N_p} M_p \mathbf{V}_p^k N_i(\mathbf{X}_p^k). \tag{29}$$

During the Lagrangian phase of the computation, the discrete momentum equation Eq. (25) is solved. The solution in a Lagrangian frame means that the nonlinear convective terms that are troublesome in purely Eulerian calculations do not appear in the formulation. Once the nodal accelerations are computed, the velocity can be updated at the nodes,

$$\mathbf{v}_i^L = \mathbf{v}_i^k + \Delta t \mathbf{a}_i^k, \tag{30}$$

where the superscript L denotes the value of the nodal velocity at the end of the Lagrangian step and Δt is the current time increment, $\Delta t = t^{k+1} - t^k$. During this phase of the computation, the nodes are assumed to move according to this computed velocity,

$$\mathbf{x}_i^L = \mathbf{x}_i^k + \Delta t \mathbf{v}_i^L. \tag{31}$$

These updates for position and velocity are also consistent with the representations (15), (17) and (18).

The above representations also imply a consistent update for the particle properties. Update particle position and velocity according to,

$$\mathbf{X}_p^L = \mathbf{X}_p^k + \Delta t \sum_{i=1}^{N_n} \mathbf{v}_i^L N_i(\mathbf{X}_p^k), \tag{32}$$

$$\mathbf{V}_p^L = \mathbf{V}_p^k + \Delta t \sum_{i=1}^{N_n} \mathbf{a}_i^k N_i(\mathbf{X}_p^k). \tag{33}$$

During this Lagrangian step, each element is assumed to deform in the flow of material so that points in the interior of the element move in proportion to the motion of the nodes, as given by the representation using nodal basis functions. Since nodal basis functions are used to map the nodal velocity continuously to the interior of the element, the positions of the material points are updated by moving them in a single-valued, continuous velocity field. Similarly, the velocity of a material point is updated by mapping the nodal accelerations to the material point position. Because the velocity field is single-valued, interpenetration of material is precluded. This feature of the algorithm allows simulations of impact and penetration without the need for a special contact algorithm [17].

For history-dependent materials, it has been convenient to carry strain and stress, as well as history variables along with the material points. Applying constitutive equations at material points allows easy evaluation and tracking of history-dependent variables. It also allows computations with multiple materials to be performed easily since each material point retains its identity (material properties) throughout the computation. Another difference between Eq. (25) and traditional finite element formulations is that the material points, where constitutive equations are applied, move from one element to another rather than remain at the center or Gauss points of an element. To describe the stress update in its simplest form, strain increments are obtained from gradients of the nodal velocities evaluated at the material point positions. Then, given a strain increment at a material point, along with current values of history variables and material parameters, standard routines are used to evaluate the stress increment and update history variables. The internal forces at the nodes are then calculated directly from the stress at the material points.

The strain increment at $\mathbf{X}_p^L$ is

$$\Delta \mathbf{e}_p = \frac{\Delta t}{2} \sum_{i=1}^{N_n} \left\{ \mathbf{G}_{ip}^k \mathbf{v}_i^L + (\mathbf{G}_{ip}^k \mathbf{v}_i^L)^T \right\}. \tag{34}$$

The stress increment, $\Delta\sigma_p^s$ at $\mathbf{X}_p^L$ is obtained from the strain increment using standard constitutive equation routines. These routines divide $\Delta\mathbf{e}_p$ into elastic and plastic contributions (see Appendix)

$$\Delta \mathbf{e}_p = \Delta \mathbf{e}_p^e + \Delta \mathbf{e}_p^{pl}. \tag{35}$$

The elastic strain increment is used to update the total elastic strain and stress for the material point

$$\mathbf{e}_p^{e,L} = \mathbf{e}_p^{e,k} + \Delta \mathbf{e}_p^e, \tag{36}$$

$$\sigma_p^{s,L} = \sigma_p^{s,k} + \Delta\sigma_p^s. \tag{37}$$

At this point in the computational cycle, the material points are completely updated and carry the complete solution. During the convective phase, the material points are held fixed and the computational grid is redefined. The grid can be chosen in any convenient manner, for example adaptive grids can be used to resolve sharp gradients or interfaces. A particularly simple choice is the regular square mesh that is used in our numerical examples. The new grid points are denoted $\mathbf{x}_i^{k+1}$. Note that if $\mathbf{x}_i^k$ corresponds to vertices of a square mesh, then $\mathbf{x}_i^L$ generally does not. If $\mathbf{x}_i^{k+1}$ is defined also to represent a regular grid, then this motion of the grid $\mathbf{x}_i^{k+1} - \mathbf{x}_i^L$, with the material points held fixed, models convection. More generally, any motion of the grid relative to the material points models convection. Note that the value of $\mathbf{x}_i^L$ is never used explicitly in the calculation, so it need not be computed. Thus, if a regular grid is used each step, the nodal positions are set initially and these initial values are retained throughout the computation.

Since the material points do not move during this convective phase, material point properties have the same value at the end of the convective phase as they had at the end of the Lagrangian phase,

$$\mathbf{X}_p^{k+1} = \mathbf{X}_p^L, \qquad \mathbf{V}_p^{k+1} = \mathbf{V}_p^L, \qquad \mathbf{e}_p^{k+1} = \mathbf{e}_p^L \quad \text{and} \quad \sigma_p^{s,k+1} = \sigma_p^{s,L}. \tag{38}$$

This completes the computational cycle. A new cycle is begun using the information carried by the material points to initialize nodal values on a new grid.

The principal modification for solid mechanics of the original FLIP algorithm [3,5] is the treatment of internal forces. The original algorithm for fluids forms gradients of the nodal velocity given by Eq. (29) to obtain a strain rate for an element and then constitutive equations are applied element-by-element to compute the stress. The internal force then comes from taking the divergence of the element stress. A similar approach has also been used for linear elastic solids [8,13] where there is also no history dependence in the constitutive equations. By contrast, the current method makes greater use of the particles to maintain current values of history variables and material parameters; and constitutive equations are evaluated at material points rather than at element centers.

4. Numerical implementation

With the algorithm described in the last section, if a single material point crosses into an element and is close to the element boundary, then the value of a nodal basis function identified with a node at the opposite side of the element may be small. However, the internal force vector involves the gradient of the basis function, which does not approach zero for points near the element boundary. The result is that computed accelerations at outer nodes can occasionally be unphysical and lead to material points separating from the main body. One approach to this problem is to detect small nodal masses and to set the corresponding internal force at the node to zero if the mass is below a certain cutoff value. Although this approach is quite successful, it is difficult to justify the choice of the cutoff criterion and sometimes material points still separate. However, with a simple change to the order of arithmetic operations, the need to apply the cutoff is eliminated.

The key to reordering operations is to work with momentum instead of velocity as much as possible, thus avoiding divisions by nodal masses. Define the nodal momentum at step k,

$$\mathbf{p}_i^k = \sum_{j=1}^{N_n} m_{ij}^k \mathbf{v}_j^k, \tag{39}$$

and the momentum for a material point

$$\mathbf{P}_p^k = M_p \mathbf{V}_p^k. \tag{40}$$

Eq. (29) can be rewritten

$$\mathbf{p}_i^k = \sum_{p=1}^{N_p} \mathbf{P}_p^k N_i(\mathbf{X}_p^k), \tag{41}$$

and solution of the momentum equation Eq. (25) provides the momentum update

$$\mathbf{p}_i^L = \mathbf{p}_i^k + \Delta t \left(\mathbf{f}_i^{\text{int},k} + \mathbf{f}_i^{\text{ext},k}\right), \tag{42}$$

where the internal force is accumulated from the particles as before, Eq. (26). The particle update becomes

$$\mathbf{X}_p^L = \mathbf{X}_p^k + \Delta t \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} (m_{ij}^k)^{-1} \mathbf{p}_j^L N_j(\mathbf{X}_p^k), \tag{43}$$

$$\mathbf{V}_p^L = \mathbf{V}_p^k + \Delta t \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} (m_{ij}^k)^{-1} \mathbf{f}_j^{\text{int},k} N_j(\mathbf{X}_p^k), \tag{44}$$

where $(m_{ij}^k)^{-1}$ denotes the inverse of the mass matrix.

Velocity gradients are needed for the strain increment. The Lagrangian grid velocity can be obtained from the updated particle velocity by solving the equation

$$\sum_{j=1}^{N_n} m_{ij}^k \mathbf{v}_i^L = \sum_p M_p \mathbf{V}_p^L N_i(\mathbf{X}_p^k) \tag{45}$$

for $\mathbf{v}_i^L$. Mathematically, there is no difference between the algorithm in this section and the last, if the consistent mass matrix is invertible; however, the numerical properties are improved. In Eq. (43) and Eq. (44), the multiplication of momenta and internal forces by the nodal basis function prior to division by the mass has the effect of balancing the numerator and denominator in the case of small nodal masses.

In practice, in order to simplify computations, a lumped mass matrix is used instead of the consistent mass matrix. The lumped mass matrix is the diagonal matrix with each diagonal entry being the corresponding row sum of the consistent version. Accordingly, the lumped masses are

$$m_i^k = \sum_{p=1}^{N_p} M_p N_i(\mathbf{X}_p^k). \tag{46}$$

Matrix inversions become trivial if a lumped mass matrix is used, at the cost of introducing a small amount of numerical dissipation [8].

If a lumped mass matrix is used, there is one difference between the algorithm of this section and the last. The velocities in Eq. (45), whose gradients are used in the constitutive equations, are altered. To see the difference, substitute the lumped masses for the consistent mass matrix in Eqs. (45), (44), (25), (29), and rewrite the result in the following form

$$\begin{aligned} m_i^k \mathbf{v}_i^L &= \sum_{p=1}^{N_p} M_p \mathbf{V}_p^L N_i(\mathbf{X}_p^k) \\ &= \sum_{p=1}^{N_p} M_p \mathbf{V}_p^k N_i(\mathbf{X}_p^k) + \sum_{p=1}^{N_p} M_p \left\{ \Delta t \sum_{j=1}^{N_n} \mathbf{f}_j^{\text{int},k} N_j(\mathbf{X}_p^k)/m_j^k \right\} N_i(\mathbf{X}_p^k) \\ &= m_i^k \mathbf{v}_i^k + \Delta t \sum_{j=1}^{N_n} m_{ij}^k \mathbf{f}_j^{\text{int},k}/m_j^k \\ &= m_i^k \mathbf{v}_i^k + \Delta t \sum_{j=1}^{N_n} m_{ij}^k \mathbf{a}_j^k . \end{aligned} \tag{47}$$

Comparing this last equation with Eq. (30) shows that the accelerations used to update the nodal velocities during the Lagrangian phase are smoothed by multiplication with the mass matrix.

5. Numerical examples

The first example problem involves a hollow elastic cylinder, under the assumption of plane strain, impacting a rigid wall. The material parameters are chosen so that the cylinder has properties similar to a tennis ball. The inner and outer radii of the cylinder are 3 cm and 4 cm, respectively. The Young's modulus $E = 0.073$ GPa, Poisson's ratio $\nu = 0.4$ and mass density $\rho_0 = 1.01$ g/cm^3 result in a uniaxial wave speed of $c_0 = \sqrt{E/\rho_0} = 852$ m/s. The cylinder has an initial horizontal speed of 50 m/s = $0.06 c_0$. The calculation is performed in a 2-D Cartesian coordinate system using square elements with a side length of $\Delta x = 0.1$ cm and a time step of $\Delta t = 1$ μs. Initially, there are 4 material points per element in the ring.

Fig. 1 shows the result of the computation. Fig. 1a is the initial configuration of material points in which each particle is given an initial velocity pointing to the right. Fig. 1b shows the first impact with the wall and the following frames (c–h) show subsequent rebound and then impact with the opposite wall. This problem provides a challenging test of the code because during impact the outer edge of the cylinder is in compression and the inner edge is in tension, and these stresses must subsequently be propagated correctly around the thin ring. The large deformations of the ring also provide a stringent test. Swegle [18] originally suggested this problem to demonstrate unphysical fracturing that is possible in an SPH code. Using SPH, the cylinder shattered on impact. SPH was subsequently shown to be unstable in tension [19], accounting for this behavior. Using the original formulation of our method, Section 3, it is difficult to pick an appropriate cutoff value for this problem, but the modified formulation given in Section 4 runs successfully.

The next example is a classical cylinder-impact test. Taylor [20] studied the problem of a steel rod impacting a rigid surface at high velocity as a means for determining the dynamic yield point of metals. In this example, the calculation is performed in 2-D axisymmetric, cylindrical coordinates. Fig. 2 is a sketch of the initial configuration, showing an elastoplastic 6061-T6 aluminum cylinder with initial density ρ_0 = 2700 kg/m^3, Young's modulus $E = 78.2$ GPa, Poisson's ratio $\nu = 0.3$, and a Von Mises yield stress $Y_0 = 0.29$ GPa. An initial

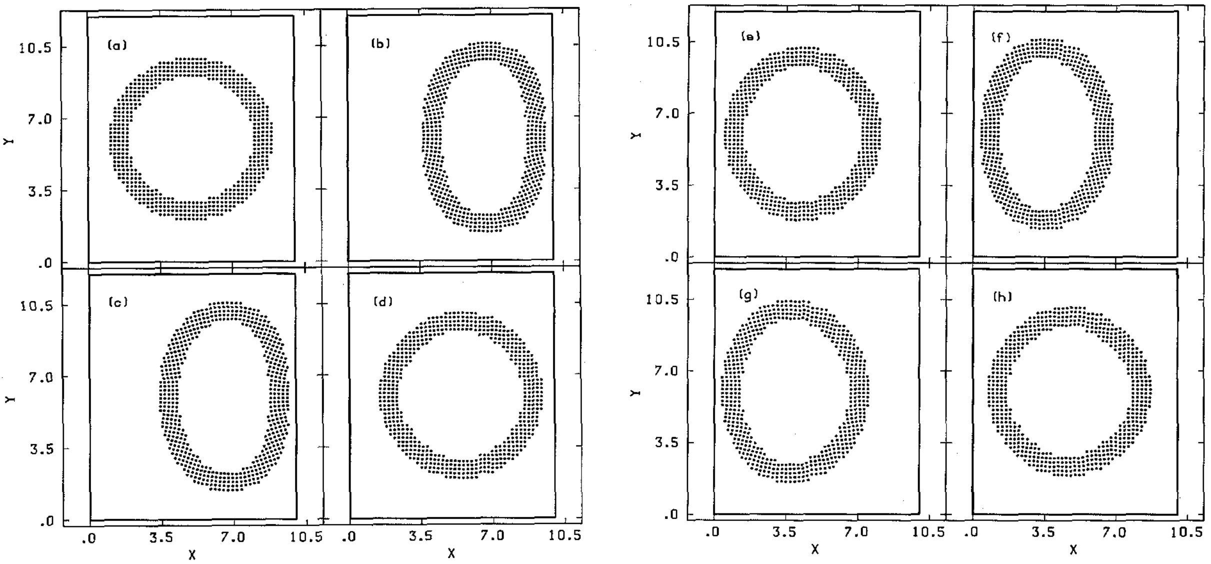

Fig. 1. Impact of an elastic cylinder on a rigid wall. The material points making up the initial configuration of the cylinder in (a) are given a velocity to the right. Subsequent frames are at intervals of 500μs. Frames (b–d) show the first impact and rebound, and frames (e–h) show the second impact with the opposite wall.

downward velocity of 373 m/s is assigned to each material point. The grid is 15 × 33 square elements with a side length of 0.0782 cm and there are 4 material points per element making up the cylinder. The initial length of the cylinder L_0 = 2.346 cm is six times the initial radius R_0 = 0.391 cm. An associated flow rule is used with a yield function of the form

$$F = \bar{\sigma} - Y, \tag{48}$$

where $\bar{\sigma}$ is the second invariant of the stress deviator

$$\bar{\sigma} = (\tfrac{3}{2}\sigma^d : \sigma^d)^{1/2}. \tag{49}$$

For perfect plasticity, $Y = Y_0$.

The problem is made dimensionless by scaling lengths using R_0, velocity using the uniaxial wave speed $c_0 = \sqrt{E/\rho_0} = 5382$ m/s, stress using E, density using ρ_0, and time using R_0/c_0. The configuration of material points at a dimensionless time of 3100 (about 23 μs) under the assumption of perfect plasticity is shown in Fig. 3. At this point, all of the plastic deformation has taken place and the only motion is due to residual vibrations. The final length of the cylinder is found to be L_f = 1.42 cm, the final radius at the rigid boundary is R_1 = 0.94 cm, and the radius 0.25 cm up from the impact surface is R_2 = 0.50 cm.

The same computation is performed by Predebon et al. [11] using their Eulerian wavecode, CSQ, and a Lagrangian code called HEMP. A summary of their results and ours are given in Table 1. In the Lagrangian calculation, the cylinder has 5 radial zones and 30 axial zones with the same 0.0782 cm square mesh. For the Eulerian calculation, the zone size is also the same but the grid is 25 × 32. As with FLIP, the grid is defined for regions into which material will flow. Although there is close agreement between the quantities measured in Table 1, the overall deformation is somewhat different. In [11] comparisons of the deformed shape show a smooth surface in the HEMP calculation, qualitatively similar to the surface in Fig. 3, while the CSQ calculation shows a surface bulge roughly halfway up the cylinder. Using a finer zoning with CSQ eliminates the bulge,

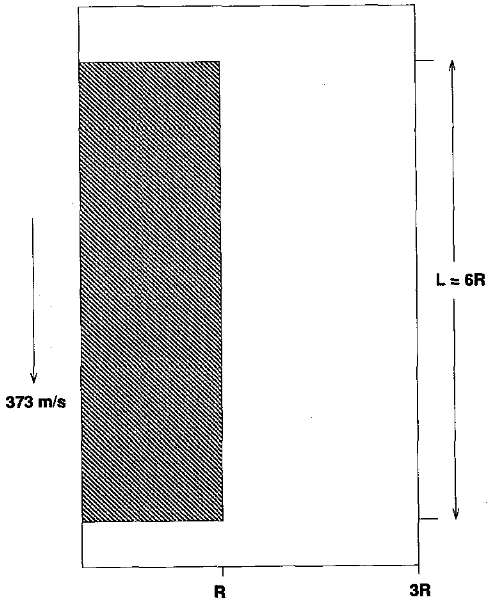

Fig. 2. In a 2-D axisymmetric calculation, an elastoplastic aluminum cylinder is given an initial downward velocity of 373 m/s. The len; of the cylinder is initially six times the initial radius.

Table 1
Perfect plasticity

	L^f (cm)	R_1 (cm)	R_2 (cm)
FLIP	1.42	0.94	0.50
HEMP	1.479	0.898	0.582
CSQ	1.477	0.826	0.585
CSQ-fine	1.433	0.935	0.569

Table 2
Strain hardening

	L^f (cm)	R_1 (cm)	R_2 (cm)
FLIP	1.63	0.74	0.49
HEMP	1.652	0.742	0.553
CSQ	1.605	0.720	0.555
CSQ-fine	1.584	0.795	0.558

but there is more radial flow than in the HEMP calculation. The shape determined using FLIP seems closest the fine-zoned CSQ calculations.

Predebon et al. [11] also report results for a computation including strain-hardening where the flow stress given by

$$Y = Y_0(1 + 125\epsilon^{pl})^{0.1}. \qquad (5$$

The value of Y_0 is the same as above, and ϵ^{pl} is the plastic strain invariant

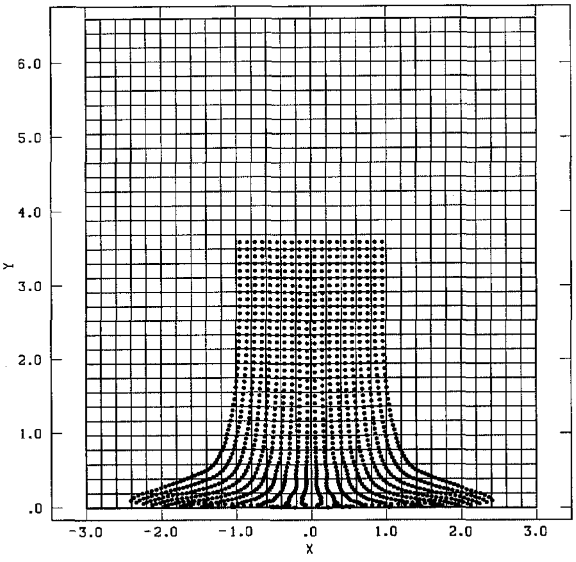

Fig. 3.

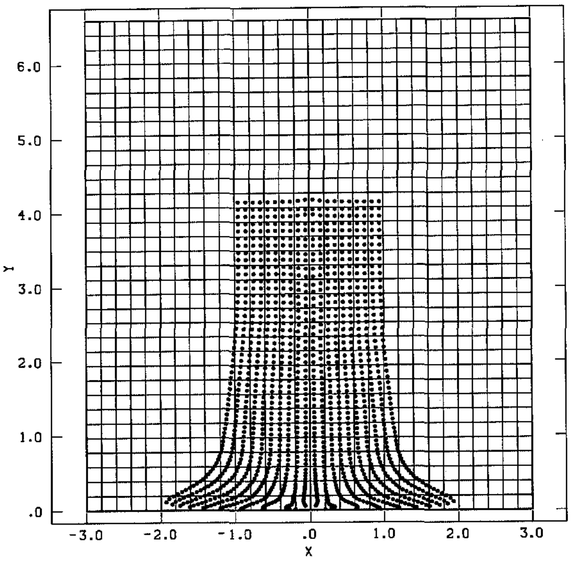

Fig. 4.

Fig. 3. Particle plots superimposed on the computational grid show the final deformation under the assumption of perfect plasticity for the cylinder described initially in Fig. 2. The final length is 61% of the initial length; and the radius at the impact surface is more than twice the original radius.

Fig. 4. Particle plots superimposed on the computational grid for a simulation corresponding to Fig. 3 except with a strain-hardening plastic cylinder. The final length is 70% of the initial length; and the radius at the impact surface is 1.9 times the original radius.

$$\epsilon^{pl} = \int (\tfrac{2}{3}\dot{\epsilon}^{pl} : \dot{\epsilon}^{pl})^{1/2} dt. \tag{51}$$

A comparison of results for the strain-hardening calculation appears in Table 2. Experiments by Wilkins and Guinan [21] measure a final length of $L_f = 1.651$ cm for an aluminum cylinder with the same initial dimensions and impact velocity. The computed final length of the cylinder agrees well with the experimentally measured value. Fig. 4 shows the configuration of material points using FLIP with strain hardening. In this case, the FLIP results seem closest to the HEMP calculations.

Contours of the plastic strain invariant are shown in Fig. 5a,b for the impacting cylinder simulated with FLIP. The contours in Fig. 5a are for perfect plasticity while those in Fig 5b include strain hardening. Plastic strains near the impact surface are similar to those obtained by HEMP, while CSQ appears to underestimate plastic flow as measured by the plastic strain invariant [11]. The FLIP calculation also shows plastic strain along the axis of the cylinder which is not reported in the HEMP or CSQ calculation. However, a computation by Wilkins and Guinan [21] using HEMP for a steel cylinder does show plastic straining at the center of the cylinder which they attribute to "cylindrical convergence". A small radial displacement of the cylinder surface becomes a large radial strain when waves focus at the axis of symmetry. These results warrant closer examination and are the subject of on-going research.

The third example consists of an elastic sphere impacting an elastic-perfectly plastic target. To use dimensionless parameters, stress is scaled by the yield stress of the target, lengths are scaled so that the computational domain has side-length one, and the density is scaled so that the dimensionless wave speed in the penetrator is unity. In the following list of material parameters, the dimensionless values are given in parentheses. The penetrator is composed of AISI 52-100 chromium steel, Young's modulus of 200 GPa (667), Poisson's ratio

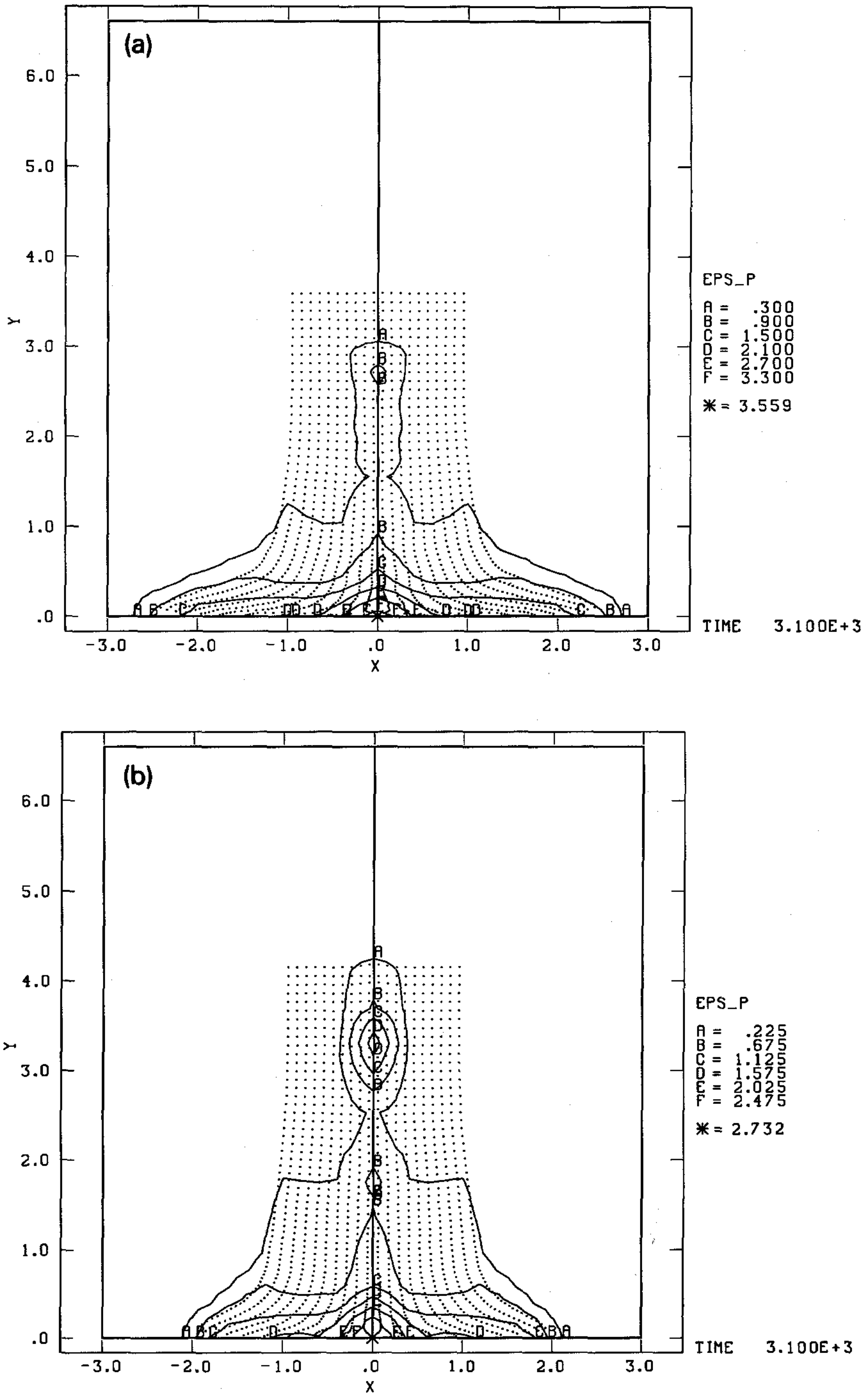

Fig. 5. Contours of the plastic strain invariant for the deformed cylinder of (a) Fig. 3 and (b) Fig. 4, show plastic strain at the base of the cylinder and along the axis of symmetry.

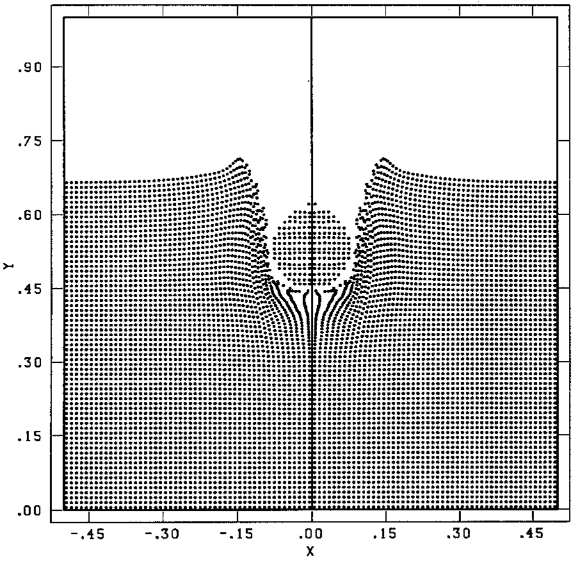

Fig. 6.

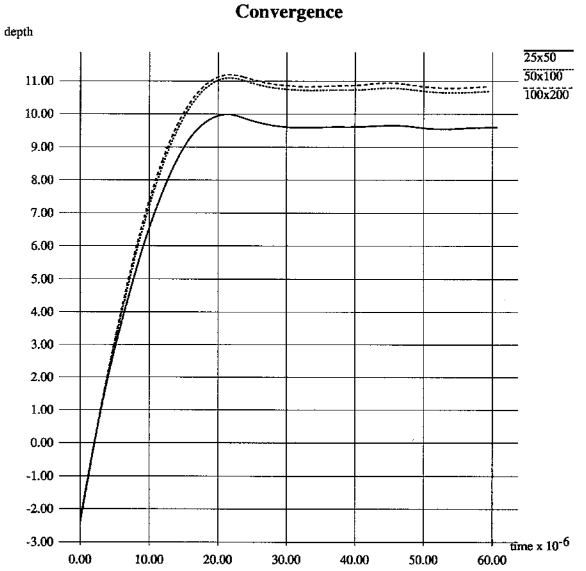

Fig. 7.

Fig. 6. Final configuration of material points for an elastic steel sphere impacting an elastic-perfectly plastic aluminum target.

Fig. 7. Depth of penetration as a function of time for an elastic steel sphere impacting a strain-hardening plastic, aluminum target. The solid line is for a computation on a 25 × 50 mesh, the dotted line on a 50 × 100 grid and the dashed line on a 100 × 200 grid. The converged value for the penetration depth is 11 mm.

of 0.3, mass density of 7850 kg/m^3 (667) and diameter 9.53mm (0.159). The uniaxial stress wave speed is 5050 m/s (1) and the initial speed of the penetrator is 1160 m/s (0.23). The target is composed of 6061-T6 aluminum 60mm × 40mm (1.0 × 0.67), modeled as an elastic-perfectly plastic material with Young's modulus of 78.2 GPa (261), Poisson's ratio of 0.3, mass density of 2700 kg/m^3 (229), and a von Mises yield stress of 0.3 GPa (1). The uniaxial stress wave speed in the target is 5380 m/s (1.07). Fig. 6 shows the configuration of material points at maximum penetration. The penetration depth is 13 mm. The axisymmetric computational domain is 30 mm × 60 mm (0.5 × 1.0) with a uniform mesh of 25 × 50 square elements of size 1.2 mm (0.02) on a side, with 4 material points per element.

The same computation is performed by Silling [12] using the Eulerian wavecode CTH. The final penetration depth using a computational mesh of square zones with side length 0.3 mm is 20 mm. The experimentally measured depth is 21 mm. The experimental data also show that the sphere is more or less undeformed after penetration [12]. CTH uses a boundary layer algorithm to allow the Eulerian simulation of sliding solid surfaces, with or without friction. The main purpose of this algorithm is to preserve the shape of the nearly rigid penetrator as it goes through the target [12].

The FLIP algorithm automatically applies no-slip contact at the penetrator-target interface. FLIP calculations using the same mesh size as Silling predict a final depth of 18 mm; however, there is no evidence of convergence with mesh refinement. Non-convergence may be a result of ill-posedness of the problem under the assumption of perfect plasticity. As a test, an additional computation is done for a strain-hardening target. To model strain hardening, a yield function is used for the aluminum target of the form

$$F = \bar{\sigma} - Y_0(1 + a\epsilon^{pl}). \tag{52}$$

The hardening parameter has the value $a = 1$ in the numerical computations. Fig. 7 shows the depth of penetration as a function of time on 25 × 50, 50 × 100 and 100 × 200 grids. The converged depth with strain hardening is 11 mm. For the original form of the algorithm given in Section 3, convergence is not obtained [16]. Non-convergence of the original formulation is probably due to the difficulty of picking a cutoff value for this problem, particularly as the mesh is refined. This difficulty is eliminated with the algorithm described in Section 4 and used here.

6. Conclusion

An extension to solid mechanics of the FLIP particle-in-cell method is presented. The particle-in-cell method uses two representations of the continuum, one based on a collection of material points and the other based on a computational grid. The material points are followed throughout the complete loading process and provide a Lagrangian description that is not subject to mesh tangling. This feature permits constitutive equations with history-dependent variables to be applied at these material points with no requirement for mapping the history parameters from one point to another. A grid, which can be held fixed or adapted as the need arises, is used to determine spatial gradients. Since the grid is used as an updated Lagrangian frame, the nonlinear convection term associated with Eulerian formulations does not appear. With the use of maps between material points and the grid, the advantages of both Eulerian and Lagrangian schemes can be attained.

Sample problems in two-dimensions are included to illustrate the natural impact algorithm. Since material points move in a single-valued velocity field, no-slip impact between bodies, inelastic, elastic, or rigid, is handled automatically by the method without resorting to a special contact algorithm. These examples also illustrate the use of history-dependent constitutive equations. By following material points, history-dependent variables, such as the plastic strain invariant, are easily tabulated. In addition, the material points provide a natural framework for studying the interactions of multiple materials. Furthermore, the symmetry associated with the weak formulation, and the formation of the internal force vector generally associated with the finite element method, are retained.

Extensions of the current work will make other classes of problems open to analysis. The general algorithm is now sufficiently advanced that a method for handling material failure should be considered. Anisotropic features in the constitutive equations are also important, especially for problems in crystal plasticity, metal rolling and metal forming. In order to treat general conditions at the interface between materials, the no-slip condition should be relaxed to include general slip, including friction. With these extensions, a robust and general numerical method will be available for problems in large-deformation solid mechanics.

Acknowledgements

This work was partially supported by Sandia National Laboratories and Department of Energy.

Appendix A

In elastoplasticity theory, the yield surface $F = F(\sigma, \mathbf{I})$ is a function of the stress tensor σ and a vector of internal variables $\mathbf{I}$ which describe the internal state of the material. The region in stress space with $F < 0$ defines elasticity, $F = 0$ defines elastoplasticity and $F > 0$ cannot be attained. For perfect plasticity the yield surface remains fixed during plastic deformation, while expansion of the yield surface during plastic deformation models hardening.

For elastic-plastic materials the material strain is decomposed into elastic and plastic parts,

$$\mathbf{e} = \mathbf{e}^{e} + \mathbf{e}^{pl}. \tag{A.1}$$

The plastic strain rate evolves according to the evolution function, **M**,

$$\frac{d\mathbf{e}^{pl}}{dt} = \frac{d\lambda}{dt}\mathbf{M}(\sigma, \mathbf{I}). \tag{A.2}$$

The unit tensor **M** describes the direction of plastic strain. Let **N** be the unit normal to the yield surface in stress space,

$$\mathbf{N} = \frac{1}{N}\frac{\partial F}{\partial \sigma}, \qquad N = \left(\frac{\partial F}{\partial \sigma} : \frac{\partial F}{\partial \sigma}\right)^{1/2} \tag{A.3}$$

then the flow rule is called associated if **M** = **N**. The monotonically increasing parameter λ must satisfy the consistency condition, $dF/dt = 0$, which guarantees that the stress remains on the yield surface during plastic deformation. The internal variables are determined by specifying hardening rules, **H**,

$$\frac{d\mathbf{I}}{dt} = \frac{d\lambda}{dt}\mathbf{H}(\sigma, \mathbf{I}). \tag{A.4}$$

With $\mathbf{T}^e$ the elasticity tensor, the tangent modulus has an explicit representation as

$$\mathbf{T} = \mathbf{T}^e - (\mathbf{T}^e : \mathbf{M}) \otimes (\mathbf{N} : \mathbf{T}^e)/A, \tag{A.5}$$

where

$$A = \mathbf{N} : \mathbf{T}^e : \mathbf{M} - H \tag{A.6}$$

and

$$H = \frac{1}{N}\mathbf{H} \cdot \frac{\partial F}{\partial \mathbf{I}}. \tag{A.7}$$

The tangent modulus provides a convenient symbolic notation for the nonlinear constitutive equations of elastoplasticity. However, in practice, if **T** is used in a numerical algorithm there is a tendency for the stress to drift from the yield surface. Instead, an incremental-iterative integration scheme is used to numerically solve the constitutive equations. Given an increment in strain, first assume the step is completely elastic, determine a trial stress, and evaluate F. If $F \le 0$ further calculation is not required. If $F > 0$, subincrements in plastic strain with corresponding decrements in elastic strain and stress are added to force F back to zero. The numerical algorithm is based on the secant method for finding a zero of F, with F considered to be a function of the time derivative of λ.

References

[1] S.W. Attaway, M.W. Heinstein, F.J. Mello and J.W. Swegle, Coupling of smooth particle hydrodynamics with Pronto, preprint.

[2] T. Belytschko, Y.Y. Lu and L. Gu, Element-free Galerkin methods, Int. J. Numer. Methods Eng. 37 (1994) 229–256.

[3] J.U. Brackbill and H.M. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations in two dimensions, J. Comput. Phys. 65 (1986) 314–343.

[4] J.U. Brackbill, On modeling angular momentum and vorticity in compressible fluid flow, Comput. Phys. Commun. 47 (1987) 1.

[5] J.U. Brackbill, D.B. Kothe and H.M. Ruppel, FLIP: A low-dissipation,particle-in-cell method for fluid flow, Comput. Phys. Commun. 48 (1988) 25–38.

[6] J.U. Brackbill, The ringing instability in particle-in-cell calculations of low speed flow, J. Comput. Phys. 75 (1988) 469.

[7] J.U. Brackbill, FLIP-MHD: A particle-in-cell method for magnetohydrodynamics, J. Comput. Phys. 96 (1991) 163–192.

[8] D. Burgess, D. Sulsky and J.U. Brackbill, Mass matrix formulation of the FLIP particle-in-cell method, J. Comput. Phys. 103 (1992) 1–15.

[9] L.D. Libersky, A.G. Petschek, A.G. Carney, T.C. Hipp and F. A. Allahdadi, High strain Lagrangian hydrodynamics – A three-dimensional SPH code for dynamic material response, J. Comput. Phys. 109 (1993) 67–75.

[10] W.K. Liu, Y.F. Zhang, T. Belytschko, S. Jun, J. Adee and L. Gu, Reproducing kernel particle methods, Int. J. Numer. Meths. Fluids, submitted.

[11] W.W. Predebon, C.E. Anderson and J.D. Walker, Inclusion of evolutionary damage measures in Eulerian wavecodes, Comput. Mech. 7 (1991) 221–236.

[12] S.A. Silling, An algorithm for Eulerian simulation of penetration, in: New Methods in Transient Analysis, AMD-Vol. 143, Eds. P. Smolinski, W.K. Liu, G. Hulbert and K. Tamma (The American Society of Mechanical Engineers, New York, 1992) pp. 123–129.

[13] D. Sulsky and J.U. Brackbill, A Numerical Method for Suspension Flow, J. Comput. Phys. 96 (1991) 339–368.

[14] D. Sulsky, Z. Chen and H. L. Schreyer, The application of a material-spatial numerical method to penetration, Sandia National Laboratories Technical Report, SAND91-7095 (1991).

[15] D. Sulsky, Z. Chen and H.L. Schreyer A particle method for history-dependent materials, Comput. Methods Appl. Mech. 118 (1994) 179–196.

[16] D. Sulsky and H.L. Schreyer, A particle method with large rotations applied to the penetration of history-dependent materials, in: Symposium on Advances in Numerical Simulation Techniques for Penetration and Perforation of Solids, AMD-Vol. 171 (The American Society for Mechanical Engineers, New York, 1993) pp. 95–102.

[17] D. Sulsky and H.L. Schreyer, The particle-in-cell method as a natural impact algorithm, in: Advanced Computational Methods for Material Modeling, AMD-Vol. 180, PVP-Vol. 268 (The American Society for Mechanical Engineers, New York, 1993) pp. 219–229.

[18] J.W. Swegle, SPH behavior in tension, Memo (13 August 1992).

[19] J.W. Swegle, D.L. Hicks and S.W. Attaway, Smoothed particle hydrodynamics stability analysis J. Comput. Phys. 116 (1995) 123–134.

[20] G.I. Taylor, The use of flat-ended projectiles for determining dynamic yield stress. Part I, Proc. R. Soc. London, Ser. A 194 (1948) 289–299.

[21] M.L. Wilkins and M.W. Guinan, Impact of cylinders on a rigid boundary, J. Appl. Phys. 44 (1973) 1200–1206.