



A particle method for history-dependent materials

D. Sulsky^a, Z. Chen^b, H.L. Schreyer^{c,*}

^aDepartment of Mathematics, ^bNew Mexico Engineering Research Institute, ^cDepartment of Mechanical Engineering,
University of New Mexico, Albuquerque, NM 87131, USA

Received 10 June 1993

Revised manuscript received 15 February 1994

Abstract

A broad class of engineering problems including penetration, impact and large rotations of solid bodies causes severe numerical problems. For these problems, the constitutive equations are history dependent so material points must be followed; this is difficult to implement in a Eulerian scheme. On the other hand, purely Lagrangian methods typically result in severe mesh distortion and the consequence is ill conditioning of the element stiffness matrix leading to mesh lockup or entanglement. Remeshing prevents the lockup and tangling but then interpolation must be performed for history dependent variables, a process which can introduce errors. Proposed here is an extension of the particle-in-cell method in which particles are interpreted to be material points that are followed through the complete loading process. A fixed Eulerian grid provides the means for determining a spatial gradient. Because the grid can also be interpreted as an updated Lagrangian frame, the usual convection term in the acceleration 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 so that mesh tangling is avoided while material variables are tracked through the complete deformation history. Example solutions in two dimensions are given to illustrate the robustness of the proposed convection algorithm and to show that typical elastic behavior can be reproduced. Also, it is shown that impact with no slip is handled without any special algorithm for bodies governed by elasticity and strain hardening plasticity.

1. Introduction

The particle-in-cell (PIC) method represents a fluid by Lagrangian mass points, called particles, moving through a computational grid. The 'classical' PIC method [1] is partially Lagrangian in that only a mass and position is attributed to each particle. The procedure is highly successful in tracking contact discontinuities and in modeling highly distorted flow. To reduce the amount of numerical dissipation, a 'full-particle' formulation called FLIP has been developed in which each particle is attributed all the properties of the fluid, including momentum and energy [2, 3]. It has been shown [4] that, with the use of a consistent mass matrix, kinetic energy is conserved by the mappings that are required between particles and grid vertices (nodes). This further reduction in dissipation is attained at the expense of inverting a new mass matrix for each time step. For comparable accuracy of results, the use of a large time step seems to compensate for the cost of inverting the consistent mass matrix in comparison with the alternative procedure of using a small time step and a diagonal mass matrix.

Recently, Sulsky and Brackbill [5] have extended FLIP to handle elastic bodies and elastic bodies in contact with a fluid. Because the positions of the particles are forced to be single valued through mappings inherent in the method, interpenetration cannot occur. In effect, the procedure automatically

* Corresponding author.

provides a non-slipping contact algorithm between two bodies or any two media with different path-independent constitutive relations.

So far, FLIP has been used for materials governed by history independent constitutive equations for which it is natural to apply the constitutive equation at grid nodes. In this paper, we choose to invoke the constitutive equation at the material points. Since each material point is followed, history-dependent variables, such as plastic strain and strain-hardening parameters, can be associated with the material point for the complete evolution of the problem. In addition, the equations are presented in the format of a weak formulation consistent with the finite element method which ensures that the tangent stiffness matrix (if it were to be determined) is symmetric. The formulation actually follows that of many current dynamic finite element codes in that a stiffness matrix is never computed and the internal force vector is determined by sweeping over the elements.

To illustrate the usefulness of the new approach, several sample problems in two dimensions are included. The first is one of large rotation of a rigid body to show that the dissipation normally present with Eulerian codes is not present. Next, the fundamental vibration mode for an elastic body is reproduced. One great advantage of the method is its application to impact. It is shown that elastic impact can be handled with no special algorithm for interfaces. Because the velocity field is forced to be single valued, the algorithm is actually one of no slip. Plastic behavior is also included to illustrate the fact that history-dependent variables are easily accommodated.

2. Governing equations

To present the method in the simplest possible context, the formulation is limited to small deformations although large translations are allowed. Therefore there is no need to select from among the various strain and stress tensors and the gradient operators associated with large-deformation theory.

Let \mathbf{x} denote the position of a material point at time, t . The position of the same material point at $t = 0$ is \mathbf{X}^0 , i.e. if \mathbf{x} is considered to be a function of \mathbf{X}^0 and t , then $\mathbf{x}(\mathbf{X}^0, 0) = \mathbf{X}^0$. The velocity, \mathbf{v} , is the derivative of \mathbf{x} with respect to t , denoted $\mathbf{v} = \dot{\mathbf{x}}$.

Let \mathbf{L} denote the gradient of velocity with respect to the spatial variable, \mathbf{x} . Then

$$\mathbf{L} = \mathbf{v}\nabla = \mathbf{D} + \mathbf{W}, \quad (2.1)$$

in which the rate of deformation, \mathbf{D} , is the symmetric part of \mathbf{L} and the vorticity, \mathbf{W} , is the skew-symmetric part. Frequently, a strain tensor, \mathbf{e} , is defined such that $\dot{\mathbf{e}} = \mathbf{D}$. Such a strain tensor is often used for computational convenience, as it is here, even though there is no correspondence with physical measurements, as exists, for example, with Lagrangian or logarithmic strain tensors.

The tangent modulus, \mathbf{T} , is a fourth-order tensor defined such that

$$\dot{\boldsymbol{\sigma}} = \mathbf{T} : \dot{\mathbf{e}}, \quad (2.2)$$

in which $\boldsymbol{\sigma}$ is the stress tensor. For linearly elastic materials, \mathbf{T} is just the elasticity tensor, \mathbf{T}^E . For elastic-plastic materials, suppose that the material strain is decomposed into elastic and plastic parts,

$$\mathbf{e} = \mathbf{e}^e + \mathbf{e}^p. \quad (2.3)$$

If \mathbf{M} is the evolution function for plastic strain rate, then

$$\dot{\mathbf{e}}^p = \dot{\lambda} \mathbf{M}. \quad (2.4)$$

The rate of the monotonically increasing parameter, λ , must satisfy the consistency condition, $\dot{f} = 0$, where the yield function f is a function of $\boldsymbol{\sigma}$, and stress hardening (and softening) parameters. The yield function is chosen such that $f < 0$ denotes elastic behavior, $f = 0$ denotes yielding, and $f > 0$ is not permitted. Let the gradient of f with respect to $\boldsymbol{\sigma}$ be the tensor, \mathbf{N} . Then the tangent modulus is

$$T = T^E - \frac{1}{A} (T^E : M) \otimes (N : T^E) \quad (2.5)$$

$$A = N : T^E : M - H$$

in which H is a hardening modulus that depends on the specific form of the yield function. For hardening, perfect plasticity and softening, H is positive, zero and negative, respectively. The symbol, \otimes , denotes the tensor product.

If ρ denotes the mass density, then the specific stress, σ^s , is defined such that

$$\rho \sigma^s = \sigma. \quad (2.6)$$

The specific stress proves to be useful in the weak form of the equation of motion. The specific stress requires the use of a 'specific' tangent modulus, T^s , defined such that

$$\dot{\sigma}^s = T^s : \dot{\epsilon}. \quad (2.7)$$

For small deformations, the mass density can be taken as constant so it follows that

$$\rho T^s = T. \quad (2.8)$$

If b denotes the body force per unit mass and a is the acceleration, then the equation of motion is

$$(\rho \sigma^s) \cdot \nabla + \rho b = \rho a, \quad a = \dot{v}. \quad (2.9)$$

In addition, the stress is symmetric, $\sigma = \sigma^T$. For the proposed numerical algorithm, the specific stress will be taken as a function of x and t to form the Eulerian part of the formulation whereas the velocity will be considered a function of X^0 and t . The latter is a Lagrangian formulation which implies that the acceleration does not contain the convection term which causes a significant amount of numerical error if a purely Eulerian approach is used. Convection is handled in a separate step by mapping quantities from the material points to the grid. The separation of the calculation into a Lagrangian phase and a convective phase is also the basis of ALE methods [6, 7]. A comprehensive review of Lagrangian and Eulerian schemes together with a host of other topics including contact algorithms has recently been provided by Benson [8].

In the finite element literature associated with solid mechanics, a mixed Lagrangian–Eulerian formulation is also quite common and is sometimes called an 'updated Lagrangian' scheme. In an updated Lagrangian approach, the position, X , of the material points at the end of the time step is used as the reference configuration for the subsequent step. By contrast, a 'fully Lagrangian' approach uses X^0 as the independent variable. However, in the fluid dynamics literature, the term Lagrangian often refers to the method defined here as updated Lagrangian.

3. Mixed weak form of governing equations

The governing equations are summarized as follows:

$$\begin{aligned} (\rho \sigma^s) \cdot \nabla + \rho b &= \rho a \\ \dot{\sigma}^s &= T^s : \dot{\epsilon} \\ \dot{\epsilon} &= \frac{1}{2}[(v \nabla) + (v \nabla)^T]. \end{aligned} \quad (3.1)$$

Suppose each of these equations is multiplied by the weighting functions, w , ρW , and ρW^* , in turn, and an integration over the current configuration, Ω , is performed. After the use of the divergence theorem for the first equation, the resulting set of equations is

$$\begin{aligned}
\int_{\Omega} \rho [\mathbf{w} \cdot \mathbf{a} + \boldsymbol{\sigma}^s : \mathbf{w} \nabla] dV &= \int_{\Omega} \rho \mathbf{w} \cdot \mathbf{b} dV + \int_{\partial\Omega} \mathbf{w} \cdot \boldsymbol{\tau} dS, \\
\int_{\Omega} \rho \mathbf{W} : [\dot{\boldsymbol{\sigma}}^s - \mathbf{T}^s : \dot{\mathbf{e}}] dV &= 0, \\
\int_{\Omega} \rho \mathbf{W}^* : \left\{ \dot{\mathbf{e}} - \frac{1}{2} [(\mathbf{v} \nabla) + (\mathbf{v} \nabla)^T] \right\} dV &= 0.
\end{aligned} \tag{3.2}$$

Here, $\boldsymbol{\tau}$ denotes the prescribed part of the traction on the surface $\partial\Omega$. Differentials of volume and surface are denoted by dV and dS , respectively.

In (3.2), the momentum equation is to be solved on a grid, whereas the second two equations are to be evaluated at material points in the PIC formulation. Invoking constitutive equations at material points simplifies the treatment of history-dependent variables.

With a particular choice of basis functions, the numerical procedure can be described in the finite element framework as follows. First consider the material points. Suppose the domain of the body in the original configuration is composed of the subdomains, Ω_p^0 , $P = 1, \dots, N$ as illustrated in Fig. 1. Associated with each subdomain is a reference material point, as defined by the position vector, \mathbf{X}_p^0 , and a mass, M_p . With time, these subdomains deform into the subdomains, Ω_p , with the reference material points located at \mathbf{X}_p and with the same mass. From compatibility, these material subdomains remain simply connected but, in general, the shapes of the subdomains will be unknown. Nevertheless, as shown below, the material points, \mathbf{X}_p , can be tracked.

Over the material subdomains, define piecewise constant basis functions U_p such that $U_p = 1$ for all points on Ω_p ; otherwise $U_p = 0$. It follows that $U_i(\mathbf{X}_j) = \delta_{ij}$. Suppose these basis functions are used to represent the functions \mathbf{W} , \mathbf{W}^* , $\boldsymbol{\sigma}^s$, and \mathbf{e} in the weak form of the equations. For example,

$$\boldsymbol{\sigma}^s = \sum_{P=1}^N \boldsymbol{\sigma}_P^s U_P, \quad \mathbf{e} = \sum_{P=1}^N \mathbf{e}_P U_P, \tag{3.3}$$

where $\boldsymbol{\sigma}_P^s$ and \mathbf{e}_P are the time-dependent stress and strain variables which are taken to be constant over Ω_p .

Dirac delta functions are used as basis functions for the mass density:

$$\rho = \sum_{P=1}^N M_P \delta[\mathbf{x} - \mathbf{X}_P]. \tag{3.4}$$

Then, with the argument that the generalized variables \mathbf{W}_p and \mathbf{W}_p^* are arbitrary, the last two weak equations become

$$\begin{aligned}
\dot{\boldsymbol{\sigma}}_P^s &= \mathbf{T}^s(\mathbf{X}_P) : \dot{\mathbf{e}}_P, \\
\dot{\mathbf{e}}_P &= \frac{1}{2} [(\mathbf{v} \nabla) + (\mathbf{v} \nabla)^T]_{\mathbf{X}_P},
\end{aligned} \tag{3.5}$$

in which the subscript, \mathbf{X}_P , in the expression for the strain rate indicates that the function in the square brackets is evaluated at the material point. In other words, the stress and strain rates are evaluated at

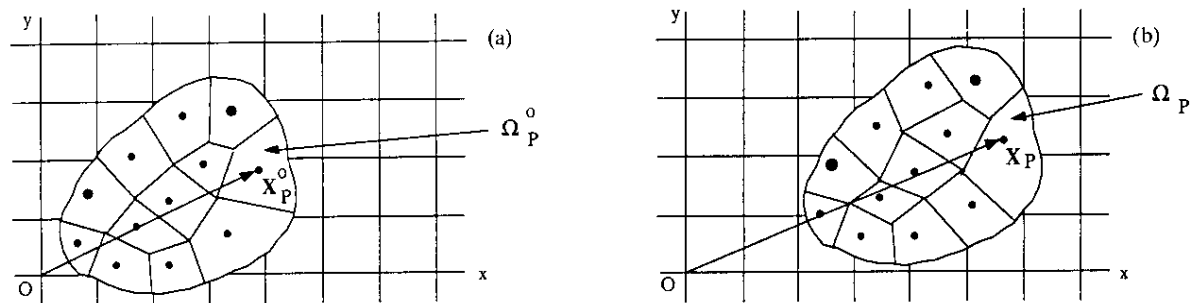


Fig. 1. Sketch of typical computational grid and material elements. (a) Initial configuration, (b) final configuration.

those material points which will be tracked as part of the computational procedure. In many finite element codes, stress and strain rates are determined at element centers or at Gauss points. Here, the tracked material points may appear at arbitrary points within an element.

Now, consider basis functions for the grid. The remaining variables \mathbf{w} , \mathbf{v} and \mathbf{a} must be continuous at least in the limit as the spatial mesh size goes to zero. Suppose a computational grid is constructed of elements which are used to form nodal basis functions $N_i(\mathbf{x})$ associated with spatial points \mathbf{x}_i with $i = 1, \dots, n$. Then the conventional finite element representations for the continuous variables are

$$\mathbf{w} = \sum_{i=1}^n \mathbf{w}_i N_i(\mathbf{x}), \quad \mathbf{v} = \sum_{i=1}^n \mathbf{v}_i(t) N_i(\mathbf{x}), \quad \mathbf{a} = \sum_{i=1}^n \mathbf{a}_i(t) N_i(\mathbf{x}), \quad (3.6)$$

in which \mathbf{w}_i , \mathbf{v}_i and \mathbf{a}_i denote the nodal vectors for the respective functions. Introduce the mapping matrix, $[S]$, whose components, S_{pi} , are values of the nodal basis functions at the current locations of the material points. $[S]$ can be thought of as a stochastic matrix in the sense that all entries are positive or zero and each row sum is one. Also define the set of gradient vectors, \mathbf{G}_{pi} , which represents the gradient of each basis function at the current locations of the material points. The components of these matrices are

$$S_{pi} = N_i(\mathbf{X}_p) \\ \mathbf{G}_{pi} = N_i \nabla|_{\mathbf{X}_p}. \quad (3.7)$$

Consider the weak form of the equation of motion given as the first equation in (3.2). With the use of (3.4), (3.6) and (3.7), the first and second terms in the equation become

$$\int_{\Omega} \rho \mathbf{w} \cdot \mathbf{a} \, dV = \sum_{i,j=1}^n \mathbf{w}_i m_{ij} \cdot \mathbf{a}_j, \quad m_{ij} = \sum_{p=1}^N M_p S_{ip}^T S_{pj}, \quad (3.8)$$

$$\int_{\Omega} \rho \boldsymbol{\sigma}^s : \mathbf{w} \nabla \, dV = - \sum_{i=1}^n \mathbf{w}_i \cdot \mathbf{f}_i^{\text{int}}, \quad \mathbf{f}_i^{\text{int}} = - \sum_{p=1}^N \mathbf{G}_{ip}^T M_p \cdot \boldsymbol{\sigma}_p^s. \quad (3.9)$$

The term m_{ij} denotes a component of the mass matrix associated with the computational grid and $\mathbf{f}_i^{\text{int}}$ is the internal force vector associated with node i . Nodal vectors for the body force field, \mathbf{b}_i , and the surface traction are defined as a natural consequence of the volume and surface integrals involving applied forces:

$$\int_{\Omega} \rho \mathbf{w} \cdot \mathbf{b} \, dV = \sum_{i=1}^n \mathbf{w}_i \cdot \mathbf{b}_i, \quad \mathbf{b}_i = \sum_{p=1}^N S_{ip}^T M_p \mathbf{b}_p, \\ \int_{\partial\Omega} \mathbf{w} \cdot \boldsymbol{\tau} \, dS = \sum_{i=1}^n \mathbf{w}_i \cdot \hat{\boldsymbol{\tau}}_i, \quad \hat{\boldsymbol{\tau}}_i = \int_{\partial\Omega} N_i \boldsymbol{\tau} \, dS. \quad (3.10)$$

Alternatively, since the body force and surface traction are explicitly given, the conventional finite element form can be used in which these functions are evaluated at the grid nodes. The external force vector, $\mathbf{f}_i^{\text{ext}}$, is defined to be

$$\mathbf{f}_i^{\text{ext}} = \mathbf{b}_i + \hat{\boldsymbol{\tau}}_i. \quad (3.11)$$

The components of \mathbf{w}_i are arbitrary except for those points where components of the displacement are prescribed. With the understanding that the constraints on the displacement field are invoked, the weak form of the equation of motion yields

$$\sum_{j=1}^n m_{ij} \mathbf{a}_j = \mathbf{f}_i^{\text{int}} + \mathbf{f}_i^{\text{ext}}, \quad i = 1, \dots, n. \quad (3.12)$$

The set of equations given by (3.5) and (3.12) are similar in form to those obtained by conventional finite element schemes. For example, the internal and external force vectors associated with nodes are developed by sweeping over elements at each time step. The components of physical vectors are

arranged sequentially to form a vector of scalar components. The constitutive equation subroutines are also traditional.

However, there are important differences between the present algorithm and the conventional finite element approach. First, the mass matrix, m_{ij} , varies with time and therefore must be computed at each time step. To simplify calculations, as in the conventional finite element procedure for transient problems, a diagonal mass matrix can be employed in which each diagonal term consists of the row sum of m_{ij} . Second, the gradient, stress and strain are evaluated at material points that can move from one element to another rather than remain at the center or at the Gauss points of an element. Third, the use of the point mass representation for the mass density results in the appearance of point masses in expressions for the internal and external force vectors. Fourth, the specific stress is used to provide a convenient form of the equation of motion in which the stress is evaluated only at the material points.

4. The convective phase

A great advantage of this approach is that the computational grid can be chosen for convenience. For example, the grid can be kept fixed in contrast to updated Lagrangian schemes in which elements can become severely distorted and even entangled. However, fixing the grid in space implicitly means that material points cross grid lines and the convection associated with material motion must be included in the computational procedure. With a procedure initiated by Brackbill and Ruppel [2], the convection phase is handled by mapping the velocity field based on values at the material points to values at the nodes of the computational grid. The procedure is described in this section.

Once the accelerations at the grid nodes are determined from (3.12), explicit time integration gives values for the nodal velocity vectors. This integration is carried out as if the grid were an updated Lagrangian frame so that convection terms are not required. Information obtained during this Lagrangian step is then transferred to the material points to update their properties. When the material points move, they transport material properties assigned to them without error. With the use of information carried by the material points, the solution can be reconstructed on any grid. In contrast to methods that continue to use the current updated Lagrangian frame, the freedom to choose the grid means mesh entanglement can be avoided. In the numerical examples of Section 7, information is mapped from material points to a uniform Eulerian grid to begin each time step. The mappings between the grid and material points are detailed below.

If the representation for the velocity given by (3.6) is evaluated at a material point, X_p , the result is

$$\mathbf{v}_p = \sum_{i=1}^n \mathbf{v}_i N_i(X_p). \quad (4.1)$$

Eq. (4.1) maps the velocity on the grid to material-point locations. The use of the nodal basis functions assures single-valuedness and continuity of the velocity field.

Let the vector of N terms, $\{V\}$, denote one component (the x -component, say) of the velocity for all material points. Similarly, let the vector of n terms, $\{v\}$, represent the same component of velocity at the grid nodes. Then with the use of the mapping matrix $[S]$ with components, S_{pi} , defined in (3.7), each component of the above equation can be given in matrix form as

$$\{V\} = [S]\{v\}. \quad (4.2)$$

Explicit time integration is used to obtain the updated position of each material point with the use of these components of the velocity vector. Strain can also be updated by using the gradient of this velocity field evaluated at the current locations of the material points. Then stress is obtained from the constitutive equation.

The convective phase consists of mapping the velocity back from the updated material points to the grid points. Because $[S]$ is rectangular, the procedure is not straightforward. The approach used in FLIP can be interpreted as using weighted least squares to determine the nodal velocities from the velocities at the material point. The weighting consists of the diagonal matrix $[M_D]$ formed from the

point masses, M_p , associated with the material points. The result is the following equation which must be solved for $\{v\}$:

$$[m]\{v\} = [S]^T[M_D]\{V\}, \quad (4.3)$$

where the components of $[m]$ are m_{ij} , the same grid mass matrix, (3.8), that appears in the equation of motion. However, we also use the diagonal form of the grid mass matrix for computation efficiency in obtaining $\{v\}$.

Burgess et al. [4] have shown that this particular formulation for mapping velocity from material points to grid nodes implies that kinetic energy, linear momentum and angular momentum are conserved. Kinetic energy is conserved provided the consistent mass matrix, $[m]$, is used with grid quantities. The result of using a diagonal mass matrix is some dissipation of kinetic energy that has been quantified by Brackbill and Ruppel [2], Brackbill et al. [3] and Burgess et al. [4]. The same analysis applies in the context of this study.

5. Generation of material points

The method followed here is an adaptation of a procedure used by Brackbill and Ruppel [2]. Suppose the computational grid is constructed in a convenient manner to cover the potential domain for the boundary value problem. Unless there is some reason to do otherwise, choose square elements. Suppose further that the initial configuration for each material is defined analytically by a set of relations

$$g_\alpha(X_p^0) \leq 0. \quad (5.1)$$

Each function, g_α , might describe one segment of a surface. Voids can also be described within the framework. The current implementation in 2-D allows combinations of straight line segments, circles and ellipses but more general forms can be used.

Each material region is discretized by prescribing the number of material points and their locations within each element. A loop is performed over the computational grid with trial material points in each element. If the trial point does not satisfy the inequalities (5.1) for that material region, then the trial point is discarded. Otherwise, the material point is added to an array associated with a specific constitutive equation and the initial values of parameters for that material are stored. Initial values consist of position, velocity, and mass which is determined based on mass density and number of material points per cell. No attempt is made to allocate partial masses to account for the fact that material boundaries can pass through a cell. The effect of not allocating partial masses can be assessed by performing a convergence study with respect to changes in the number of material points assigned to each cell.

The result is a mesh and material point generation scheme that is remarkably general and easy to implement. In effect, all that is needed is a description of the region in a form given by (5.1) for each material type, the size of each element (cell) in a regular grid, and the number of material points to be assigned initially to each element.

6. Numerical algorithm

For the time integration, let the time step be s and the discrete time be $t^k = ks$ with $k = 0, 1, 2, \dots$. The diagonal mass matrix $[M_D]$ associated with the material points is based on the initial discretization and is fixed for all time, i.e. the mass of a material point does not change.

Suppose the following parameters are known at t^k : (i) the mapping matrix, $[S]^k$; (ii) the gradient matrix, $[G]^k$; (iii) the diagonal form of the grid mass matrix, $[m_D]^k$; (iv) the grid nodal values of each component of velocity, $\{v\}^k$; (v) each component of the position of the material points, $\{X\}^k$; (vi) each

component of the internal force vector, $\{f^{int}\}$; and (vii) each component of the external force vector, $\{f^{ext}\}$.

In the following description, only one of the components (x , y and z for three dimensions) of the matrices $[G]^k$, $\{v\}^k$, $\{X\}^k$, $\{f^{int}\}$ and $\{f^{ext}\}$ will be used to simplify the presentation.

The algorithm consists of the following steps:

- (1) Determine the acceleration, $\{a\}^k$, at the grid nodes from the equation of motion (3.12):

$$[m_D]^k \{a\}^k = \{f^{int}\}^k + \{f^{ext}\}^k. \quad (6.1)$$

- (2) Use an explicit time integrator to obtain the velocity, $\{v^*\}^{k+1}$, of points located at the grid nodes:

$$\{v^*\}^{k+1} = \{v\}^k + s\{a\}^k. \quad (6.2)$$

- (3) Obtain the velocity gradients at the material points. There are nine such combinations in three dimensions of which a typical expression is $[G]^k \{v^*\}^{k+1}$.

- (4) The strain rate at material points, \dot{e}_p^{k+1} , is formed from the components of the velocity gradient tensor, as in (3.5).

- (5) The stress at material points is updated with the increment obtained from the constitutive equation subroutine. History-dependent variables such as plastic strain and strain-hardening parameters are also updated for each material point.

- (6) Map to obtain velocities and accelerations at the material points, as in (4.1):

$$\{V^*\}^{k+1} = [S]^k \{v^*\}^{k+1}, \quad \{A\}^k = [S]^k \{a\}^k. \quad (6.3)$$

- (7) Use an explicit time integrator to update components of the position vector and velocity of the material points,

$$\{X\}^{k+1} = \{X\}^k + s\{V^*\}^{k+1}, \quad \{V\}^{k+1} = \{V\}^k + s\{A\}^k. \quad (6.4)$$

- (8) Determine new mapping and gradient matrices as in (3.7):

$$[S]^{k+1} = \{N\}(X^{k+1}), \quad [G]^{k+1} = \{N\} \nabla|_{X^{k+1}}. \quad (6.5)$$

- (9) The internal force vector is determined in terms of the current stress and gradient matrices, as in (3.9).

- (10) Update the consistent mass matrix associated with the grid,

$$[m]^{k+1} = [S^T]^{k+1} [M_D] [S]^{k+1}. \quad (6.6)$$

Sum rows to obtain the diagonal version, $[m_D]^{k+1}$.

- (11) Map back to obtain velocities at the grid nodes, i.e. solve

$$[m_D]^{k+1} \{v\}^{k+1} = [S^T]^{k+1} [M_D] \{V^{k+1}\} \quad (6.7)$$

for $\{v\}^{k+1}$. This step corresponds to (4.3) where the diagonal form of the grid mass matrix has been substituted in the left-hand side of the equation.

- (12) The cycle for one time step is complete; go to (1).

We want to emphasize that the matrices used to describe the procedure are never formed. Rather, quantities are accumulated by sweeping over elements as done in many finite element programs. The details are not included because the procedure is well established in the finite element literature.

7. Numerical examples

7.1. Rotation test

A model problem that is often used to illustrate properties of convection algorithms is the continuity equation describing transport of the non-diffusive scalar quantity $\Psi(\mathbf{x}, t)$,

$$\frac{d\Psi}{dt} = \frac{\partial\Psi}{\partial t} + \mathbf{v} \cdot (\Psi \nabla) = -\Psi(\mathbf{v} \cdot \nabla). \quad (7.1)$$

In this model problem, the velocity field $\mathbf{v}(\mathbf{x}, t)$ is prescribed. A common test is to prescribe a velocity field corresponding to rigid body motion. One example used by Smolarkiewicz [8] is that of a cylinder with a sharp notch cut out. The function $\Psi(\mathbf{x}, 0)$ is assigned the height of the cylinder and the velocity is prescribed to be one of rigid body rotation in the plane perpendicular to the unit vector \mathbf{k} in the z -direction,

$$\mathbf{v}(\mathbf{x}, t) = \Omega \mathbf{k} \times (\mathbf{x} - \mathbf{x}_c), \quad (7.2)$$

in which \mathbf{x}_c is the fixed center of rotation. The velocity field is divergence free so after one full rotation Ψ should be the same as its initial value. Typical convection algorithms on Eulerian grids artificially diffuse Ψ and/or produce ripples in the solution [8].

In FLIP, $\Psi_p = \Psi(\mathbf{X}_p^0, 0)$ is assigned to the material points. Since the velocity field is divergence free, Ψ is a Lagrangian invariant; so Ψ_p is fixed and transported by the particles without error.

The top view of the cylinder, computational grid and material points are shown in Fig. 2(a). The cylinder has radius 0.15 and height 4.0 so that initially $\Psi_p = 4.0$. The computational domain is a square with side length 1.0. The angular velocity is $\Omega = 0.1$ and the components of \mathbf{x}_c are (0.5, 0.5, 0). The grid has 51 nodes in each coordinate direction and the time step is $s = 0.1$. One full rotation corresponds to 628 time steps. Initially, four material points were assigned to each element where Ψ is non-zero. This formulation contains fewer grid points than the number used by Smolarkiewicz [9], but because of the material points, the computational effort is greater. However, even when the number of elements is reduced, our results are unchanged.

The algorithm for this test problem differs slightly from the algorithm in Section 6. There is no acceleration and the velocity in Step 2 is prescribed rather than computed:

$$\{\mathbf{v}^*\}^{k+1} = \Omega \mathbf{k} \times (\mathbf{x}^{k+1/2} - \mathbf{x}_c), \quad (7.3)$$

where $\mathbf{x}^{k+1/2}$ is the time-centered position, $\frac{1}{2}(\mathbf{x}^k + \mathbf{x}^{k+1})$, and

$$\{\mathbf{x}\}^{k+1} = \{\mathbf{x}\}^k + s\{\mathbf{v}^*\}^{k+1}. \quad (7.4)$$

After the velocity on the grid is prescribed, it is mapped to the material points according to Step 6 and the positions of the material points are updated as in Step 7.

The result of applying the algorithm is shown in Fig. 2(b) which is a plot of the computed surface $\Psi(\mathbf{x}, t)$ after one full revolution. The only source of numerical error is the mapping of the velocity from the grid to the material points at each step and the discrete-time integration algorithm used to update the positions of the material points. The errors associated with the process are negligible compared with the errors associated with convection algorithms on Eulerian grids.

In the discretized equation of motion combined with a time integrator, if there is no acceleration and if the solution for the velocity can be represented exactly by the element shape functions, then the numerical solution will be exact. A particular case is uniform translation, another standard test problem in which $\mathbf{v}(\mathbf{x}, t)$ is a prescribed constant. Numerical illustrations are given in some of the following examples.

7.2. Vibrating solid elastic cylinder

Consider an infinitely long, solid cylinder of radius $R = 0.6$ cm. For copper, the mass density is $\rho = 8.92$ kg/m³, and the elastic parameters are Young's modulus, $E = 126$ GPa, and Poisson's ratio, $\nu = 0.35$. The cylinder will vibrate in its fundamental mode which is a Bessel function of order one with

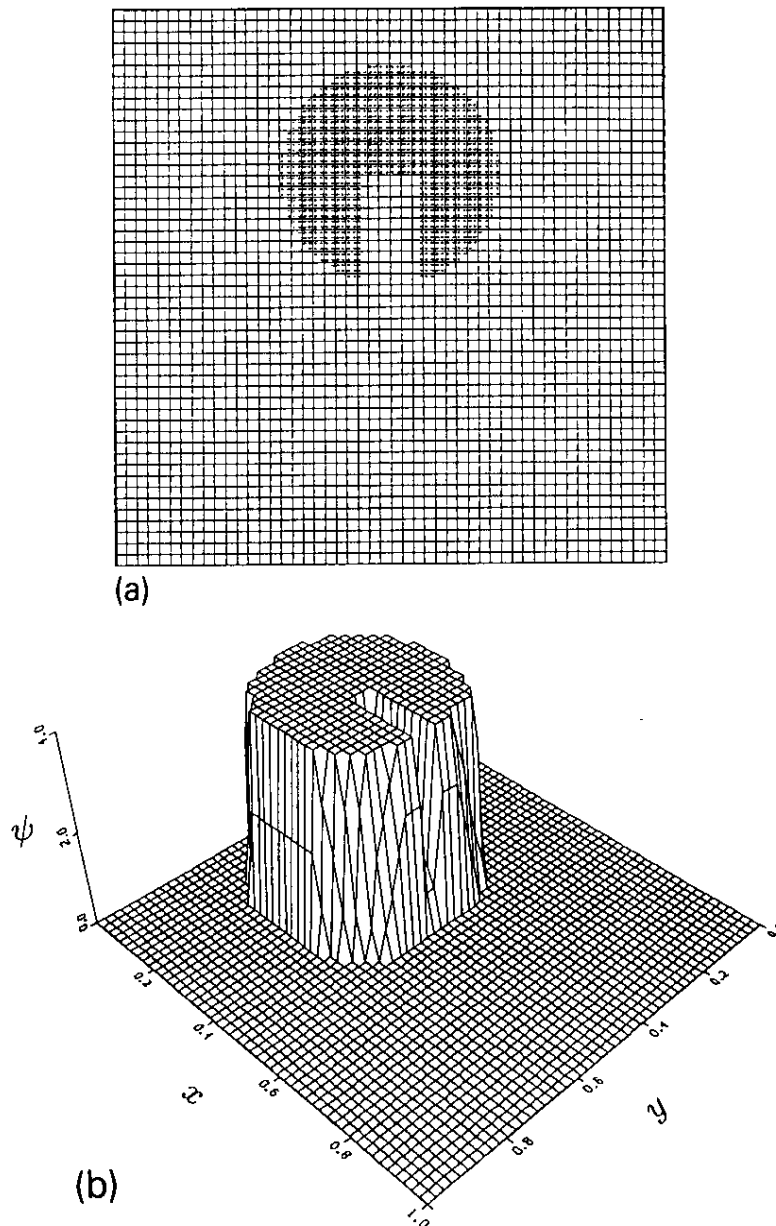


Fig. 2. Notched cylinder subjected to rigid body rotation. (a) Initial configuration of material points and grid (top view), (b) surface after one revolution.

a frequency of $\omega^e = 1.73 \times 10^6$ Hz for plane strain. The cylinder is subjected to a uniform radial strain of 2% so that only the first mode is excited.

Again, four material points per element are used. Results for various time steps and element sizes are given in the form of energy plotted as a function of time in Fig. 3. Dotted lines denote elastic strain energy, dashed lines represent kinetic energy, and solid lines the sum. Fig. 3(a) shows the results for a time step of $s = 0.2 \mu_s$ ($\omega^e s = 0.35$) and a square mesh spacing of $h = 0.80 \text{ cm} = 0.133R$. The predicted frequency is 1.68×10^6 Hz. A measure of numerical dispersion is $\omega^d = \omega/\omega^e$ which for this case is $\omega^d = 0.97$. If the amplitude for total energy is approximated with an expression of the type $E = E_0 \exp(-\gamma\omega t/2\pi)$ then γ can be considered a measure of numerical dissipation. For this case, $\gamma = 0.121$.

The results of a convergence study for consecutive halving of both the time step and element size are given in Table 1. Both dissipation and dispersion are reduced to an insignificant amount with mesh and

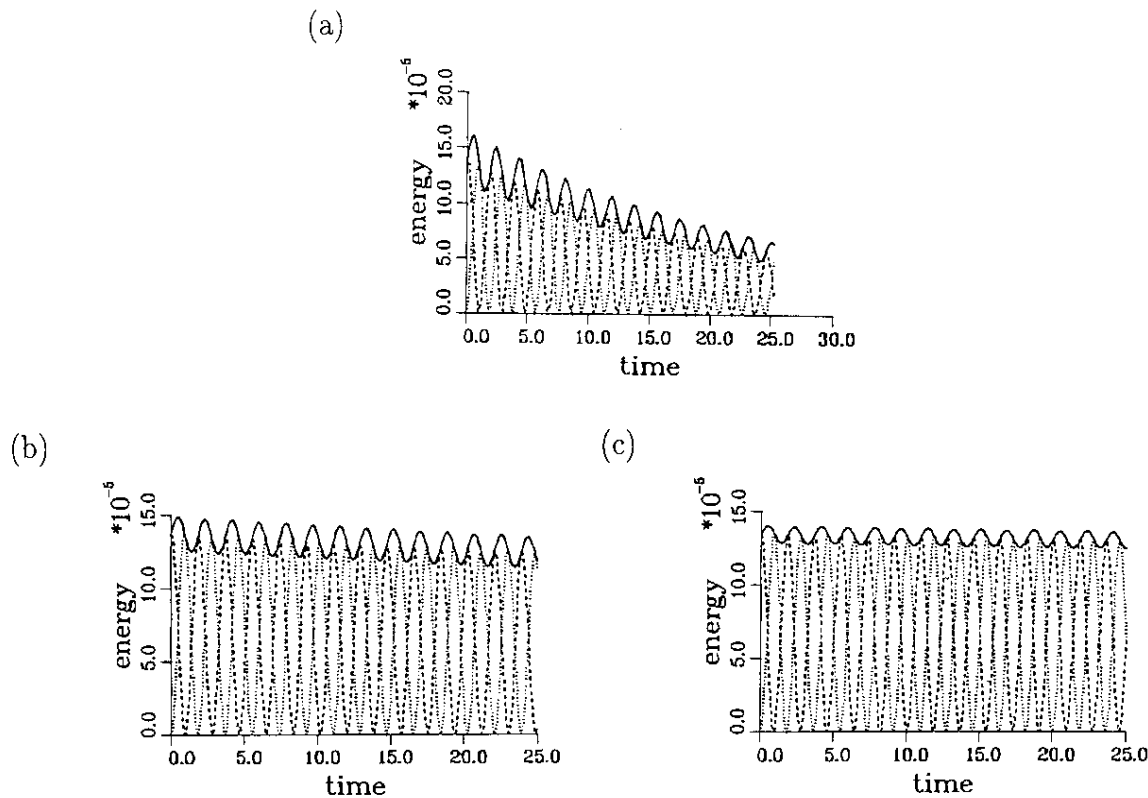


Fig. 3. Energy plots for vibrating solid-elastic cylinder. (a) $h = 0.8$ cm and $s = 0.2 \mu_s$, (b) $h = 0.$ cm and $s = 0.1 \mu_s$, (c) $h = 0.2$ cm and $s = 0.05 \mu_s$.

time step refinement. This problem was also studied previously [4, 5], with similar results. The difference between our algorithm and the one in these references is that we apply constitutive equations at the material points rather than at grid nodes.

7.3. Impact of two elastic bodies

Fig. 4 shows the impact of two identical elastic disks, with Young's modulus $E = 1000$, Poisson's ratio $\nu = 0.3$ and density $\rho = 1000$. The disks start in the lower left and upper right corners with initial velocities of $(0.1, 0.1)$ and $(-0.1, -0.1)$, respectively. The grid is uniform with square elements of side $h = 0.05$ and the time step is $s = 0.001$. In Fig. 4(a), the disks have already travelled some distance through the grid. The displacement of each disk makes an angle of 45° with respect to either grid line. As mentioned previously, there is no error in the numerical solution associated with a uniform translation of an object through a grid. Fig. 4(b) shows the distortion that results when the disks are in contact during impact under the assumption of plane strain. Figs. 4(c) and 4(d) show how the disks rebound and translate in the opposite direction after impact has occurred.

Energy plots are given in Fig. 5(a). All of the initial energy is kinetic energy (dashed line). The

Table 1
Results of convergence study for vibrating solid cylinder

s (μ_s)	$s\omega^c$	h (cm)	ω^d	γ
0.2	0.35	0.08	0.97	0.121
0.1	0.17	0.04	0.99	0.0126
0.05	0.087	0.02	1.00	0.0041

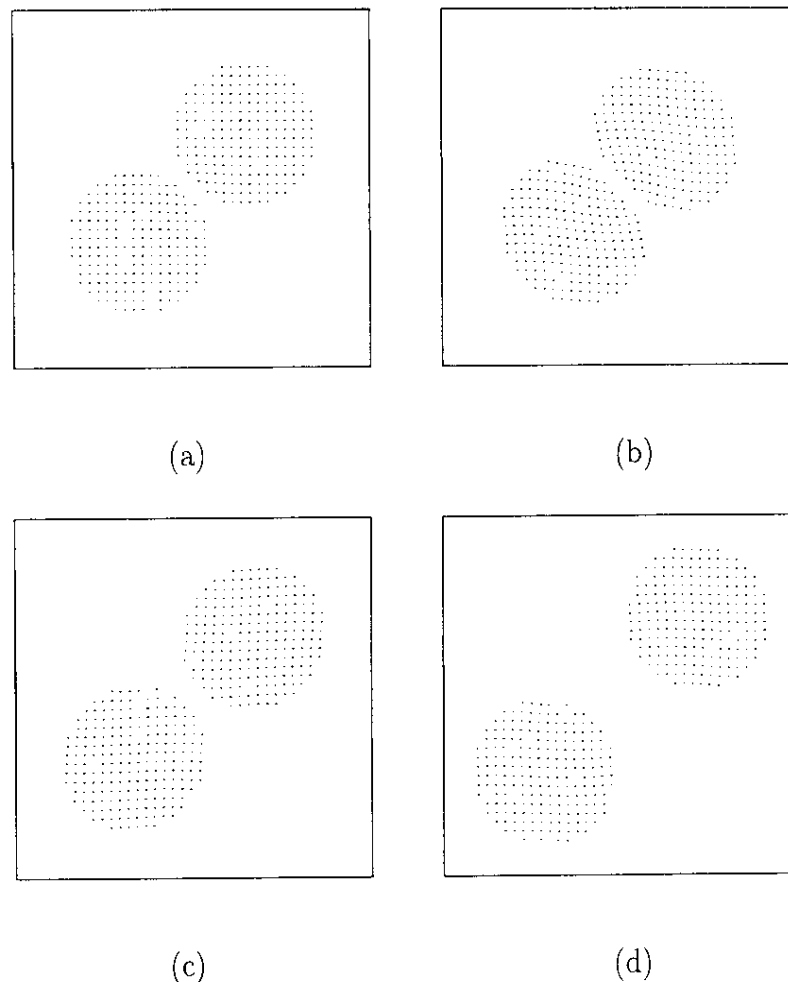


Fig. 4. Positions of disks at various times for elastic impact. (a) $t = 1.0$, (b) $t = 1.5$, (c) $t = 2.0$, (d) $t = 2.5$.

kinetic energy decreases during impact and then is mostly recovered after separation. The strain energy (dotted line) attains its maximum value at the point of maximum deformation during impact and then decreases to a nominal value associated with free vibration of the disks after impact. The strain energy is small but not zero at discrete times after impact. The fact that a zero value is not achieved can be

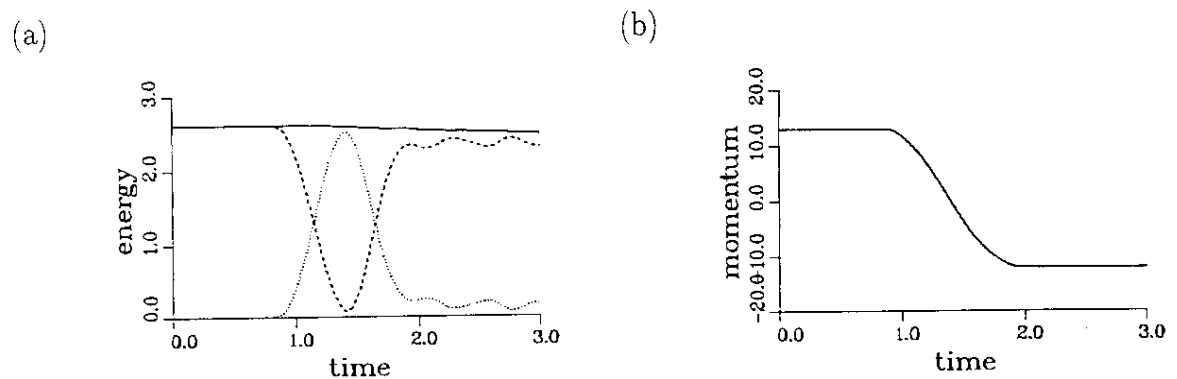


Fig. 5. Energy and momentum plots for elastic impact. (a) Energy as a function of time, (b) momentum as a function of time.

attributed to the activation of several modes which do not exhibit zeros at the same time. The total energy (solid line) decreases slightly with time indicating some numerical dissipation.

For the disk located initially in the lower left corner, the x -component of momentum is plotted as a solid line in Fig. 5(b). As expected, the momentum switches sign after impact. Also plotted is the y -component of momentum (dotted line) which is indistinguishable from the x -component both before and after impact.

7.4. Bouncing bar

Consider a rectangular elastic bar of dimensions $0.2 \times 0.4\sqrt{2}$ moving with a uniform initial velocity of $(-0.1, -0.1)$. A corner of a bar strikes an elastic layer 0.25 thick. For both materials, $E = 1000$, $\nu = 0.3$, and $\rho = 1000$. Initially, four particles per square element with size $h = 0.05$ are assigned as indicated in Fig. 6(a) and the time step is $s = 0.005$. The subsequent configurations are shown in the series of particle plots given in Fig. 6.

When one corner of the bar strikes the layer, the bar rebounds and starts to rotate. A second corner of the bar then strikes the layer. Then the second rebound velocity distribution implies that the bar will not strike the layer again.

Velocity plots on the grid are shown in Fig. 7 for the same configurations given by the particle plots in Fig. 6. Localized velocities at the points of impact are shown in the layer. All points in the layer are actually excited but velocities below a certain level were excluded from the plot.

7.5. Impact of two inelastic bodies

The calculation in Section 7.3 is repeated using the same parameters except that the disks are now considered to be elastic-plastic solids. The model is von Mises plasticity with a yield function, f , and strain hardening function, H , given by

$$f = \bar{\sigma} - H, \quad H = c_1 + c_2 \bar{e}, \quad (7.5)$$

in which $\bar{\sigma}$ is the second invariant of the stress deviator, $(\frac{3}{2} \boldsymbol{\sigma}^d : \boldsymbol{\sigma}^d)^{1/2}$, and \bar{e} is the plastic strain path invariant $\int (\frac{2}{3} \dot{\mathbf{e}}^p : \dot{\mathbf{e}}^p)^{1/2} dt$. In the numerical calculation, $c_1 = 100$ (0.1E) and $c_2 = 400$. The positions of the material points are very similar to those shown in Fig. 4 for the elastic impact and are not plotted. A small difference appears after impact, where a part of each disk is flattened and remains permanently distorted after separation. The amount of separation is not as large as that shown in Fig. 4(d) for the corresponding elastic case.

The energy plot of Fig. 8(a) shows a large dissipation in the total energy (solid line) after impact because of plastic dissipation. After impact, the kinetic energy (dashed line) oscillates about a lower value than existed before the impact and the elastic strain energy (dotted line) does not achieve a value of zero for any time. In addition to the argument concerning the activation of several modes as a reason for non-zero strain energy, there is now the additional factor that residual plastic strains can prevent the elastic strain (and stress) from achieving a zero state even in the unloaded condition.

As in the elastic case, x - and y -components of momentum (solid and dotted lines, respectively) for the lower-left disk switches sign after impact as indicated in Fig. 8(b). However, the magnitude of each component now significantly decreases after impact. Again, the x - and y -components are indistinguishable.

7.6. Impact of an elastic disk with a strain-hardening disk

Similar to the problems of Section 7.3 and 7.5, two disks are allowed to impact. However, now one disk is elastic (lower left) and one is elastic-strain hardening plastic (upper right). The properties, computational grid, initial velocities and time step used in the previous examples are used here as well. However, here the initial velocity is chosen to make an angle of 30° with respect to the horizontal coordinate and the initial separation of the centers is 0.55 which is slightly smaller than the initial

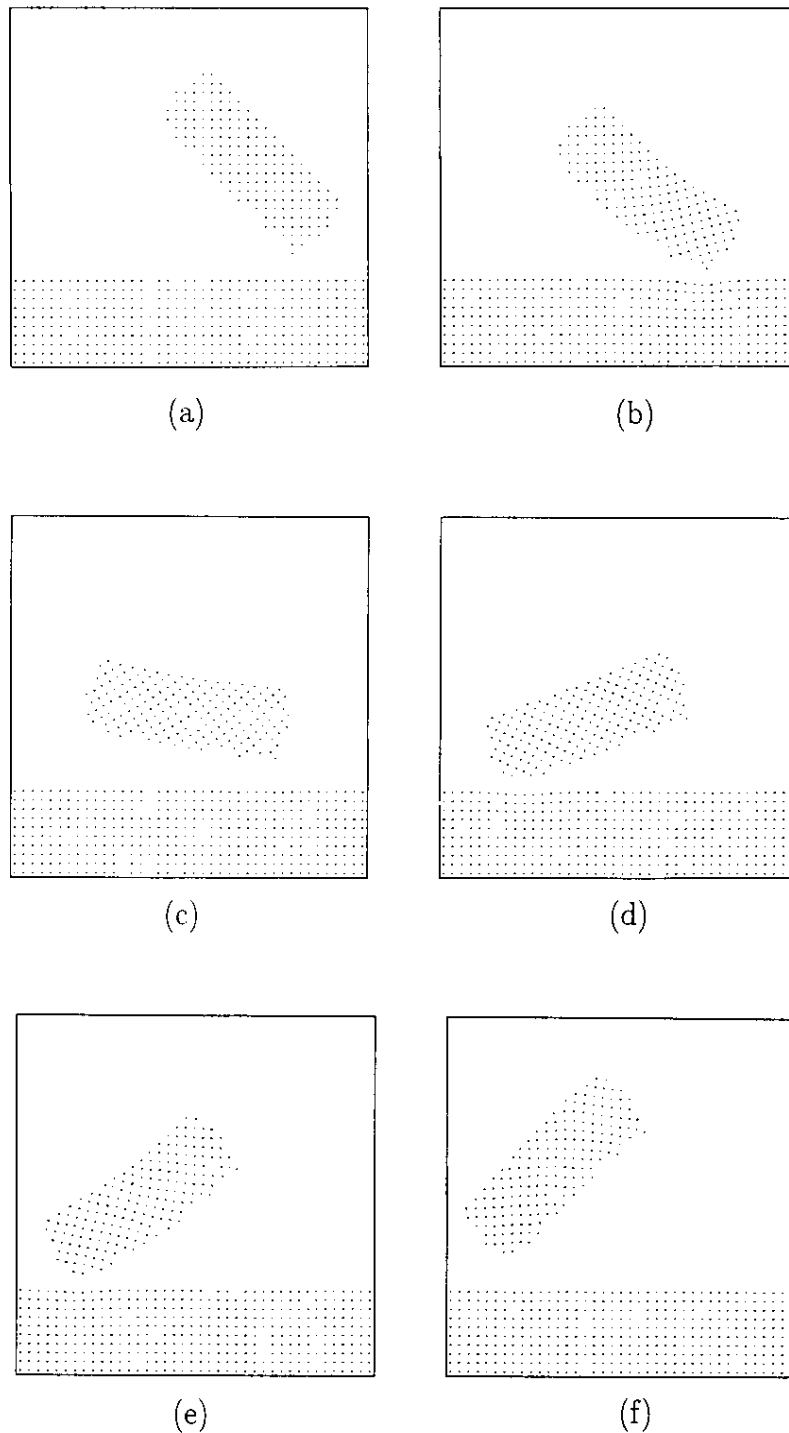


Fig. 6. Elastic impact of a bar on a layer (particle plots). (a) $t = 0.0$, (b) $t = 1.0$, (c) $t = 2.0$, (d) $t = 3.0$, (e) $t = 4.0$, (f) $t = 5.0$.

separation of 0.71 used for the previous examples. Particle plots are given in Fig. 9 and show a slight flattening of one side of the plastic disk after impact. The energy plot of Fig. 10(a) shows results intermediate to those of the problems in Sections 7.3 and 7.5. The components of momentum are shown in Figs. 10(b) and 10(c) for the elastic and plastic disks, respectively. Because of the direction of

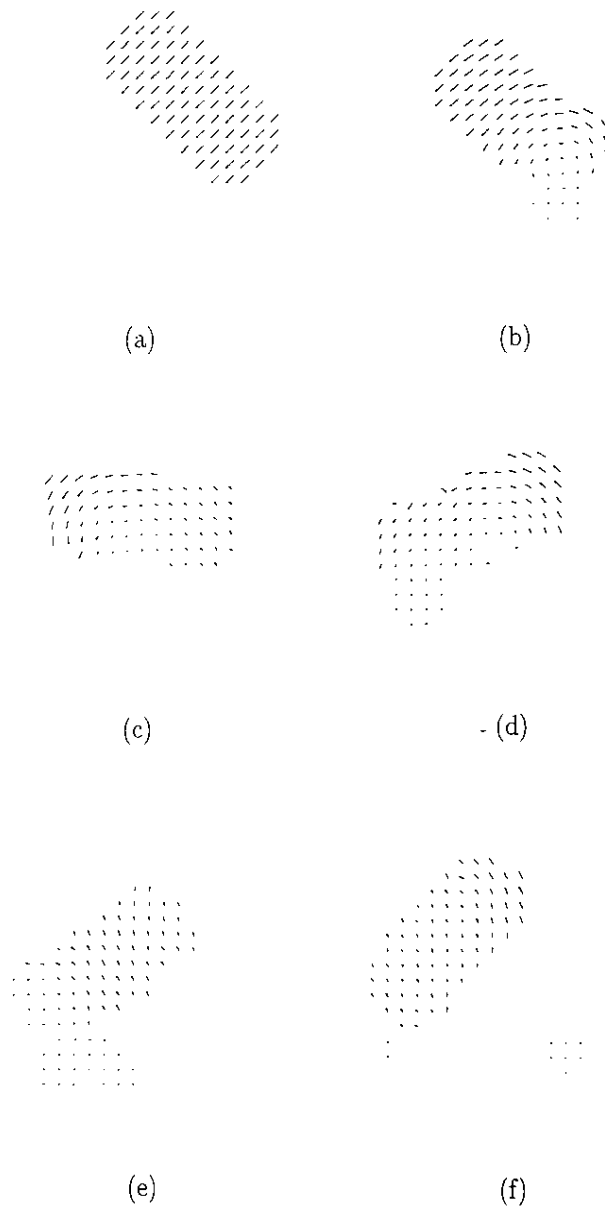


Fig. 7. Elastic impact of a bar on a layer (velocity plots). (a) $t = 0.0$, (b) $t = 1.0$, (c) $t = 2.0$, (d) $t = 3.0$, (e) $t = 4.0$, (f) $t = 5.0$.

motion, the x and y components of momentum are not equal. Furthermore, the relative magnitude of the two components of momentum switches after impact as expected.

8. Conclusion

With an extension of the particle-in-cell method, a numerical algorithm has been developed that combines several desirable features. In effect, the grid can be interpreted as an updated Lagrangian one so that the diffusion associated with the numerical evaluation of convective derivatives in a Eulerian description is not present. Material points are followed so that interpolation for history-dependent

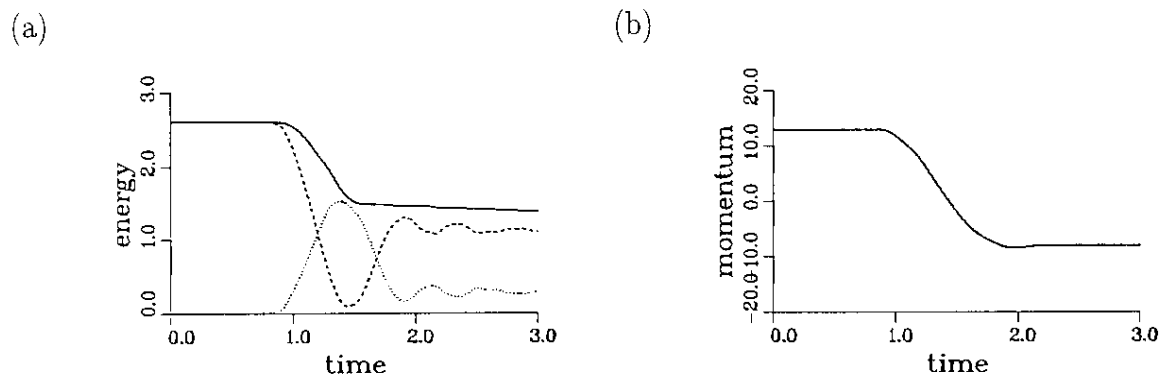


Fig. 8. Energy and momentum plots for elastic-plastic impact. (a) Energy as a function of time, (b) momentum as a function of time.

variables is not required. Furthermore, the symmetry associated with the weak formulation and the formation of an internal force vector generally associated with the finite element method have been retained. The additional cost of the method over existing finite element algorithms appears in the form

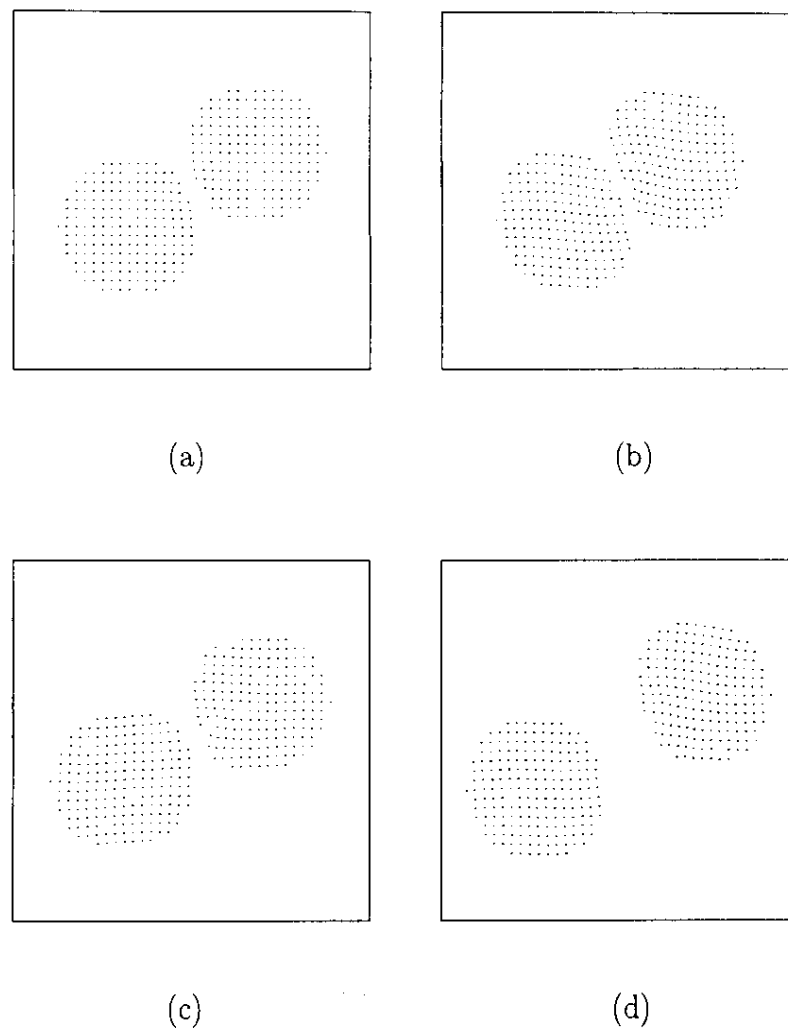


Fig. 9. Impact of disks composed of different materials. (a) $t = 0.5$, (b) $t = 1.0$, (c) $t = 1.5$, (d) $t = 2.0$.

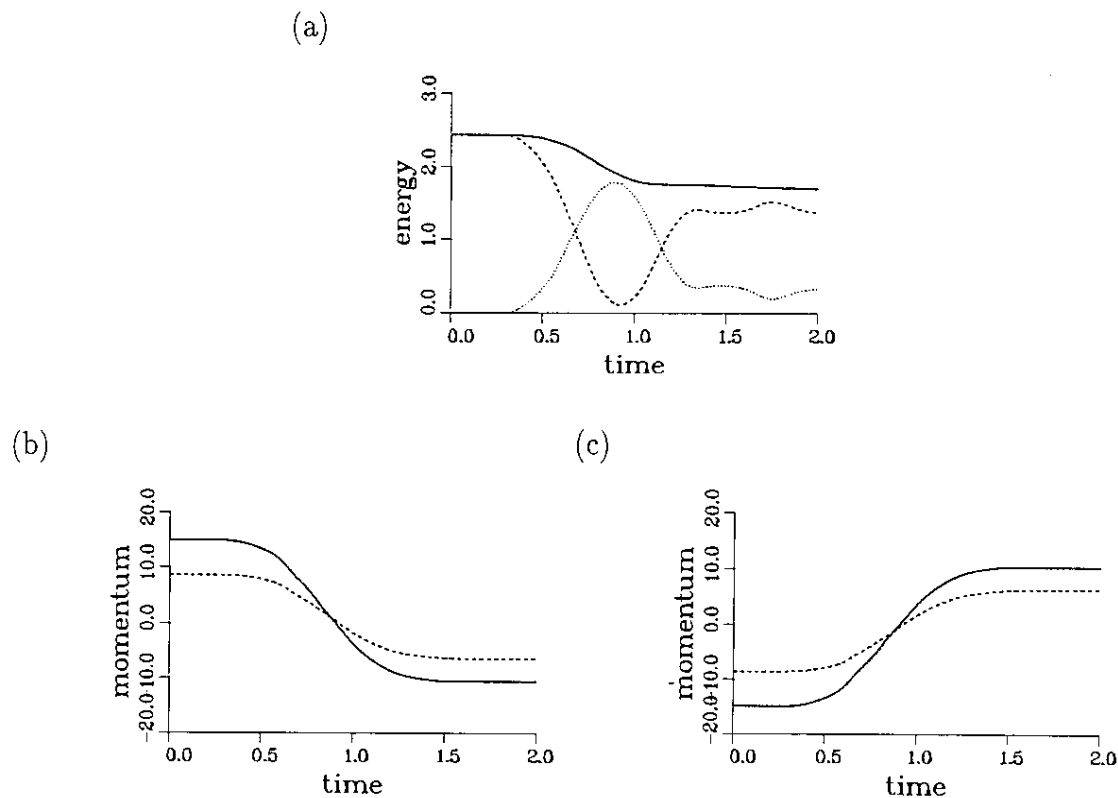


Fig. 10. Energy and momentum plots for impact of elastic and plastic disks. (a) Energy as a function of time, (b) momentum as a function of time for elastic disk, (c) momentum as a function of time for elastic-plastic disk.

of maps between the material points and the grid. The benefit is no mesh distortion and, consequently, no need to remesh in the event that large deformations occur.

An advantage of applying the constitutive equation at the particle locations is an automatic treatment of 'mixed cells'. In purely Eulerian codes, elements (cells) that contain more than one material require special treatment. In order to insure that the correct constitutive equation be applied to each material in the element, an algorithm is usually developed to partition element variables among the materials present. For impact problems, the situation arises naturally in that frequently one body remains elastic while the other is inelastic. By following material points, the appropriate constitutive equation is invoked no matter how many or what kind of material points are located within a single element.

Recently, there has been considerable research activity on a procedure called smooth particle hydrodynamics [10]. Since basis functions are constructed at each time step based on the current location of material points, the method has the advantage of being grid-free. However, as material points separate, the basis functions can become severely distorted and special algorithms to handle such cases must be constructed. The proposed approach uses a regular grid so the method is fundamentally different from smooth particle hydrodynamics and the problem with basis functions does not arise.

The sample problems with the particle method show that conventional elastic vibrations can be reproduced. For impact, the enforcement of a single-valued velocity field within any element, no matter how many material points are in that element, implies that no sliding occurs along the impact surface. Since no slideline algorithm is required the computational procedure is remarkably efficient for this class of problems.

The purpose of this work has been to demonstrate the potential benefit of combining features of the particle-in-cell method with those of the finite element method normally used for inelastic deformations of solid structures. Numerous points remain to be investigated in detail, including boundary conditions, relaxation of the no slip condition, the utility of using more or fewer material points per element, and

close observations in the material failure zone when softening occurs. Numerical solutions to a rather extensive set of model problems have shown no difficulties with locking or hourglassing but a detailed analysis of typical cells have not been performed. With a satisfactory completion of these investigations, the fact that solid–solid as well as solid–fluid interfaces can be handled so effortlessly suggests that the method will have numerous applications.

Acknowledgment

The support of this research by Sandia National Laboratories (Contract No. AC-1801) and the National Science Foundation (Contract No. DMS-9102975) is gratefully acknowledged.

References

- [1] F.H. Harlow, The particle-in-cell computing method for fluid dynamics, in: B. Adler, S. Fernbach and M. Rotenberg, eds. *Methods for Computational Physics*, Vol. 3 (1964) 319–343.
- [2] J.U. Brackbill and H.M. Ruppel, FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *J. Comput. Phys.* 65 (1986) 314–343.
- [3] J.U. Brackbill, D.B. Kothe and H.M. Ruppel, FLIP: a low-dissipation, particle-in-cell method for fluid flow, *Comput. Phys. Comm.* 48 (1988) 25–38.
- [4] 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.
- [5] D. Sulsky and J.U. Brackbill, A numerical method for suspension flow, *J. Comput. Phys.* 96 (1991) 339–368.
- [6] C.W. Hirt, A.A. Amsden and J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, *J. Comput. Phys.* 14 (1974) 227–293.
- [7] T.B. Belytschko, J.M. Kennedy and D.F. Schoeberle, Quasi-Eulerian finite element formulation for fluid–structure interaction, *J. Pressure Vessel Technol.* 102 (1980) 62–69.
- [8] D.J. Bensen, Computational methods in Lagrangian and Eulerian hydrocodes, *Comput. Methods Appl. Mech. Engrg.* 99 (1992) 235–394.
- [9] P.K. Smolarkiewicz, A fully multidimensional positive definite advection transport algorithm with small implicit diffusion, *J. Comput. Phys.* 54 (1984) 325–362.
- [10] L.D. Libersky, A.G. Petschek, T.C. Carney, J.R. 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.