Journal of Computational Physics **180,** 383–403 (2002)
doi:10.1006/jcph.2002.7103

Energy Conservation Error in the Material Point Method for Solid Mechanics

S. G. Bardenhagen[1]

Department of Mechanical Engineering, University of Utah, Salt Lake City, Utah, 8412
E-mail: bard@golden.mech.utah.edu

Received January 22, 2001; revised April 3, 2002; published online June 19, 2002

The material point method (MPM) for solid mechanics conserves mass and momentum by construction, but energy conservation is not explicitly enforced. Material constitutive response and internal energy are carried on discrete points (material points), while the governing equations are solved on an overlying grid. The constitutive response (and internal energy) may be updated at the beginning or end of a numerical time step without affecting mass or momentum conservation properties. Both versions of the algorithm have been applied in the literature. Here energy conservation on the material points is investigated and found to depend strongly on the version of the algorithm used. The energy error is found and partitioned into two terms, one of which is of definite sign (and dissipative). The other term is indefinite, and of opposite sign for the two algorithmic variations. It is shown analytically for a single-material-point free-vibration example that one version of the algorithm is strictly dissipative. For the other version the error terms cancel each other out and energy is conserved. The same trends are borne out in numerical solutions of free axial vibration of continuum bars as the wavelength of the vibrational mode begins to approach the computational cell size. The dissipative algorithm may be described as tending to damp out unresolved modes. For resolved modes, both algorithms give identical results, with no perceptible energy error or dissipation. It is suggested that the dissipative algorithm is a better choice in general, as the damping is consistent with the accuracy of the solution.

Key Words: stability and convergence of numerical methods; error analysis; basic methods in solid mechanics; linear vibrations.

1. INTRODUCTION

The material point method (MPM) is one of the latest developments in several decades of particle-in-cell (PIC) methods, originally used to model highly distorted fluid flow [11].

[1] Presently: Group T-14, MS B214, Los Alamos National Laboratory, Los Alamos, NM, 87545. E-mail: bard@lanl.gov.

0021-9991/02 $35.00

Subsequent developments advanced the understanding of the algorithm and brought modifications to reduce numerical diffusion [7, 9]. Fundamental aspects of PIC methods include the interpolation of information between grid and particles and precisely which solution variables will be ascribed to the grid and which to the particles. Several variants have been tried, with a general trend toward keeping more properties on particles. Most recently, the method has been applied to solid mechanics, where the ability of the particles, or material points, to advect naturally Lagrangian state variables has been exploited in MPM [14, 16].

MPM is well suited for solid mechanics, where it is natural to have a reference state and properties associated with specific locations in the reference state. Bodies deform according to continuum mechanics constitutive models which generate stress based on both the history and the current mechanical state. These models are often complex and require the storage of "internal variables" representing the history-dependent state. Lagrangian material points allow easy implementation of these constitutive models. Versions of hyperelasticity, hypoelasticity, plasticity, and viscoelasticity have been implemented. MPM has found application in the solution of a wide variety of problems, including mantle convection [12], silo discharge [17], membrane stretching [18], landfill settlement [19], elastic vibrations [14], collisions [4, 14–16], and the response of granular material [1–3].

While the MPM algorithm conserves mass and momentum by construction, energy conservation is not explicitly enforced. The internal energy is naturally calculated on material points, where the constitutive response is evaluated. However, the double interpolation in going from material point to grid and back again complicates the determination of the algorithm's properties for material point variables, in particular the material point internal energies. Errors in energy conservation on the material points due to discretization via MPM algorithms are examined here.

2. THE MATERIAL POINT METHOD

The MPM algorithm is presented here for completeness of the presentation, for consistency of notation, and to highlight two possible algorithmic variations. It has been described in greater detail elsewhere [14, 16].

2.1. Algorithm Overview

MPM is a PIC technique in which Lagrangian bodies are discretized by (Lagrangian) material points which carry all material properties and state. This includes constitutive parameters (including internal variables), stress, strain, velocity, and temperature. In short, everything required to specify the current state and advance the solution. The MPM algorithm also requires an overlying grid. The governing equations are solved on the grid, providing a computational savings as well as a regular, structured grid on which to apply solution techniques. Advantages of the MPM algorithm include the absence of mesh-tangling problems and error-free advection of material properties via the motion of the material points.

The material point algorithm is an interplay between the material points and the overlying grid. A time step begins with all information carried by the material points. Mass and momentum are interpolated to the grid such that the total mass and momentum on the material points and the grid are equal. Forces, both internal due to material stress and external due to applied loads, are applied to the grid, where the conservation of momentum is solved and the grid deforms. Changes in momenta are interpolated back to the material

points, resulting in global conservation of momentum, and providing updated material point velocities. Material point positions are updated consistent with the motion of the grid. The material point stress states must also be updated, and this may be done in either the undeformed grid configuration (at the beginning of the time step) or in the deformed one (at the end of the time step). It is at this stage that the internal energy is updated. Finally the deformed grid is discarded and the next time step begins with the lying down of a new grid.

2.2. The MPM Algorithm

For the derivations in the following section, it is useful to write down the specific steps in the MPM algorithm. The one-dimensional interpolation function associated with grid vertex v is the bilinear "hat" function, denoted $S_v(x)$,

$$\begin{aligned} S_v(x) &= 1 - |x - x_v|/l_x \quad \text{if } -l_x \le x - x_v \le l_x, \\ &= 0 \qquad\qquad\qquad\quad \text{else,} \end{aligned} \tag{1}$$

where the vertex coordinant is x_v and l_x is the grid spacing. Note that for any x, summing over all vertices gives the identity $\sum_v S_v(x) = 1$. Higher dimensional interpolation functions are created using products of hat functions. In three dimensions

$$S_v(\mathbf{x}) = S(x_1)S(x_2)S(x_3). \tag{2}$$

Higher dimensional interpolation functions retain the normalization

$$\sum_v S_v(\mathbf{x}) = 1, \tag{3}$$

as in the one-dimensional case. The form of the interpolation function in Eqs. (1) and (2) reflect the use of a rectangular mesh with the same vertex locations at the beginning of each time step.

The MPM algorithm begins with the interpolation of material point data to the overlying grid. Data corresponding to the beginning of a time step are denoted by superscript 0, and that at the end of the time step by superscript 1. Material point quantities are denoted with subscript p, and grid quantities with subscript v. For example, $\mathbf{x}_p^0$ is the position of material point p at the beginning of the time step. The shorthand $S_{vp} = S_v(\mathbf{x}_p^0)$ is used throughout this manuscript. First material point mass and momentum are interpolated to the grid. The grid mass, m_v, is interpolated from surrounding material points,

$$m_v = \sum_p S_{vp} m_p, \tag{4}$$

where m_p denotes material point masses. Updating the material point positions consistent with the grid deformation results in the values of the interpolation functions being Lagrangian invariants of the motion during a time step [6, 7]. Because, in addition, the mass of the material points is fixed, the grid masses are constant throughout the time step, and superscripts may be omitted. Derivation of the discrete momentum conservation equations naturally leads to a consistent mass matrix (Eq. (25)) on the grid [9, 14, 16]. The lumped-mass (diagonal) matrix, Eq. (4), may be obtained from the consistent one by summing rows.

In practice lumped masses are usually used in the discrete solution to spare computational expense and avoid matrix inversion difficulties. The analysis in the following section considers only lumped-mass algorithms. The grid momenta, $\mathbf{p}_v$ are interpolated from surrounding material points

$$\mathbf{p}_v^0 = \sum_p S_{vp}\mathbf{p}_p^0, \tag{5}$$

where $\mathbf{p}_p$ denote material point momenta.

Material point strains may now be calculated and the stresses updated (or the stress update may be done at the end of the time step). The grid velocities, $\mathbf{v}_v$, may be calculated by dividing momentum by mass, i.e.,

$$\mathbf{v}_v^0 = \mathbf{p}_v^0/m_v. \tag{6}$$

Because material point strains (and the stress update) may be calculated either at the beginning or end of the time step, it is important to distinguish between them. The strain increment calculated from velocities $\mathbf{v}_v^0$ is denoted by $\Delta\boldsymbol{\epsilon}_p^0$, where

$$\Delta\boldsymbol{\epsilon}_p^0 = \frac{1}{2}\sum_v \left(\nabla S_{vp}\mathbf{v}_v^0 + \mathbf{v}_v^0\nabla S_{vp}\right)\Delta t, \tag{7}$$

∇S_{vp} is shorthand for $\nabla S_v(\mathbf{x}_p^0)$, and ∇ is the gradient operator. Note that the strain increment is simply $\mathbf{D}_p\Delta t$, where $\mathbf{D}_p$ is the symmetric part of the velocity gradient for material point p, i.e., the rate of deformation tensor. The Cauchy stress update is

$$\sigma_p^1 = \sigma_p^0 + \Delta\sigma_p\left(\Delta\boldsymbol{\epsilon}_p^0\right), \tag{8}$$

where σ_p^0 is the Cauchy stress tensor at the beginning of the time step. For large rotations an objective rate of stress could be incorporated into Eq. (8), but it is omitted here for simplicity.

Regardless of whether the stress has been updated, forces are calculated on the grid using the material point stresses via

$$\mathbf{f}_v^{int} = -\sum_p \nabla S_{vp}\cdot\sigma_p V_p, \tag{9}$$

where V_p are material point volumes and $\cdot$ denotes the vector inner product. Contributions from body forces and boundary tractions are included in an "external force" term $\mathbf{f}_v^{ext}$. At this point conservation of momentum on the grid gives the grid point accelerations, $\mathbf{a}_v$,

$$m_v\mathbf{a}_v = \mathbf{f}_v^{int} + \mathbf{f}_v^{ext}, \tag{10}$$

which determine the deformation of the grid over the time step. Grid velocities at the end of the time step are then

$$\mathbf{v}_v^1 = \mathbf{v}_v^0 + \mathbf{a}_v\Delta t. \tag{11}$$

Material point positions and velocities are updated consistent with the deformation of the grid,

$$\mathbf{x}_p^1 = \mathbf{x}_p^0 + \sum_v S_{vp}\mathbf{v}_v^1 \Delta t, \tag{12}$$

$$\mathbf{v}_p^1 = \mathbf{v}_p^0 + \sum_v S_{vp}\mathbf{a}_v \Delta t. \tag{13}$$

Interpolating only changes in position and velocity provides for less numerical diffusion than previous PIC methods, as discussed in Refs. [7, 9].

If the stress is not updated at the beginning of the time step, it may be updated at the end. In this case the strain increment is calculated from velocities $\mathbf{v}_v^1$ and denoted by $\Delta\boldsymbol{\epsilon}_p^1$,

$$\Delta\boldsymbol{\epsilon}_p^1 = \frac{1}{2}\sum_v \left(\nabla S_{vp}\mathbf{v}_v^1 + \mathbf{v}_v^1 \nabla S_{vp}\right)\Delta t. \tag{14}$$

It should be noted that calculation of $\mathbf{v}_v^1$ from the updated grid momenta can result in numerical errors when grid masses are small. Numerical problems can be eliminated by interpolating updated material point velocities, $\mathbf{v}_p^1$, to the grid for use in the strain-increment calculation, as described in [16]. The Cauchy stress update is

$$\sigma_p^1 = \sigma_p^0 + \Delta\sigma_p\left(\Delta\boldsymbol{\epsilon}_p^1\right) \tag{15}$$

and completes the algorithm for the time step. At this point the deformed grid is discarded and a new time step begins with an interpolation of values from the material points to a new grid.

3. ENERGY CONSERVATION PROPERTIES OF MPM ALGORITHMS

Of interest in this section is the calculation of the change in kinetic and internal energies, as well as any discretization error, during a time step. Some energy properties, associated with interpolation between grid and particles, were determined previously for the FLIP algorithm [7, 9], where kinetic energies on the grid and particles were compared. In the FLIP algorithm material properties were interpolated to the grid in order to calculate internal energy there for *elastic* materials. While it is possible in theory to interpolate constitutive parameters to the grid and then evaluate constitutive response and change in internal energy there, in practice this is difficult for inelastic constitutive models with internal variables. The MPM algorithm avoids these difficulties by evaluating material response on the material points and only interpolating to the grid the resulting internal forces. This provides a general framework for the implementation of inelastic constitutive models without the need to construct material state variables on the grid.

Of interest is the change in energy of a system during a time step in the absence of heat transfer to the system, i.e., isothermal or adiabatic solid mechanics. Because the material points carry the complete solution at both the beginning and the end of a time step, with the grid serving only to provide kinematic updates, the view taken here is that the system of interest is the collection of material points. This view is also motivated by the fact that only the material points carry constitutive information in MPM, suggesting the material points

are the natural places to calculate the internal, or strain, energy. In this setting, and in the absence of external work on the system, the total energy change on the material points over a time step, $\Delta\mathcal{E}_{points}$, should be zero but may in fact have error, $\Delta\mathcal{E}_{error}$, associated with solving the discrete equations,

$$\Delta\mathcal{E}_{points} = \Delta\mathcal{KE}_{points} + \Delta\mathcal{SE}_{points} = \Delta\mathcal{E}_{error}, \tag{16}$$

where $\Delta\mathcal{KE}_{points}$ is the change in kinetic energy on the material points and $\Delta\mathcal{SE}_{points}$ is the change in strain energy on the material points. It is found advantageous to divide the error into two contributions, one due to interpolation of kinetic energies from grid to material points, $\Delta\mathcal{E}_{interpolation}$, and another due to everything else, and denoted as the algorithm error, $\Delta\mathcal{E}_{algorithm}$,

$$\Delta\mathcal{E}_{error} = -\Delta\mathcal{E}_{interpolation} - \Delta\mathcal{E}_{algorithm}, \tag{17}$$

where

$$\Delta\mathcal{E}_{interpolation} = \Delta\mathcal{KE}_{grid} - \Delta\mathcal{KE}_{points}, \tag{18}$$

$$\Delta\mathcal{E}_{algorithm} = -\Delta\mathcal{KE}_{grid} - \Delta\mathcal{SE}_{points}, \tag{19}$$

and $\Delta\mathcal{KE}_{grid}$ is the change in kinetic energy on the grid. This subdivision is useful in determining the sign of the incremental error terms, and thus whether or not they are dissipative. With the sign convention adopted, $\Delta\mathcal{E}_{interpolation}$ and $\Delta\mathcal{E}_{algorithm}$ are dissipative when positive.

First the interpolation error is found. The change in kinetic energy is defined as expected on the grid by

$$\Delta\mathcal{KE}_{grid} = \frac{1}{2}\sum_v m_v \mathbf{v}_v^1 \cdot \mathbf{v}_v^1 - \frac{1}{2}\sum_v m_v \mathbf{v}_v^0 \cdot \mathbf{v}_v^0, \tag{20}$$

and similarly on the material points

$$\Delta\mathcal{KE}_{points} = \frac{1}{2}\sum_p m_p \mathbf{v}_p^1 \cdot \mathbf{v}_p^1 - \frac{1}{2}\sum_p m_p \mathbf{v}_p^0 \cdot \mathbf{v}_p^0. \tag{21}$$

It is convenient to work with changes in velocity rather than acceleration. The change in velocity at a grid vertex, $\Delta\mathbf{v}_v$, is

$$\Delta\mathbf{v}_v = \mathbf{a}_v \Delta t, \tag{22}$$

and that on a material point is found from interpolation (as in Eq. (13)),

$$\Delta\mathbf{v}_p = \sum_v S_{vp} \Delta\mathbf{v}_v. \tag{23}$$

From Eqs. (11), (13), (18), and (20–23), it has been shown [7] that an error of order Δt^2 in kinetic energy is caused by interpolating velocities from material points to grid and back again, i.e.,

$$\Delta\mathcal{E}_{interpolation} = \frac{1}{2}\left(\sum_v m_v \Delta\mathbf{v}_v \cdot \Delta\mathbf{v}_v - \sum_{vv'} m_{vv'} \Delta\mathbf{v}_v \cdot \Delta\mathbf{v}_{v'}\right), \tag{24}$$

where the full mass matrix [9], $m_{vv'}$, is defined by

$$m_{vv'} = \sum_p m_p S_{vp} S_{v'p}. \tag{25}$$

Further, it was shown [7, 9] that the interpolation error is positive semidefinite, indicating that the algorithm is dissipative from Eq. (17). This result applies to the MPM algorithms as well, where kinetic energy interpolation is handled in the same way.

Now the second term in Eq. (17), the algorithm error, is considered. First the internal, or strain energy $\mathcal{SE}$, must be defined,

$$\mathcal{SE}(t) = \int_v \rho e \, dv, \tag{26}$$

where $\rho(\mathbf{x}, t)$ is the density, $e(\mathbf{x}, t)$ is the internal energy, and the integration is over the current configuration. The first law of thermodynamics, in its purely mechanical form [10] (in the absence of heat transfer), states that $\rho de/dt = \sigma : \mathbf{D}$, where $\sigma(\mathbf{x}, t)$ is the Cauchy stress, $\mathbf{D}(\mathbf{x}, t)$ is the rate of deformation tensor as discussed in conjunction with Eq. (7), and : is a tensor inner product, i.e., $\sigma : \mathbf{D} = \sum_{i=1}^{3} \sum_{j=1}^{3} \sigma_{ij} D_{ij}$ in three dimensions. Using the first law and a transport theorem, the rate of change of the strain energy is

$$\frac{d\mathcal{SE}}{dt} = \int_v \sigma : \mathbf{D} \, dv, \tag{27}$$

where the integration is over the current volume of the system, v. This becomes a sum over material points in accordance with the spatial discretization

$$\frac{d\mathcal{SE}}{dt} = \sum_p \sigma_p : \mathbf{D}_p V_p. \tag{28}$$

An approximation to the total change in strain energy on the material points over a time step, $\Delta\mathcal{SE}_{points}$, is found by substituting the Taylor series expansions

$$\sigma_p(t) = \sigma_p^0 + \dot{\sigma}_p(t - t_0) + O(t - t_0)^2, \tag{29a}$$

$$V_p(t) = V_p^0 + \dot{V}_p(t - t_0) + O(t - t_0)^2 \tag{29b}$$

and integrating over a time step. Here a superimposed dot indicates derivation with respect to time. This gives

$$\Delta\mathcal{SE}_{points} = \sum_p \frac{1}{2}\left(\sigma_p^0 V_p^1 + \sigma_p^1 V_p^0\right) : \Delta\epsilon_p + O(\Delta t^3). \tag{30}$$

For infinitesimal deformations the volume is unchanged and

$$\Delta\mathcal{SE}_{points} = \sum_p \frac{\sigma_p^0 + \sigma_p^1}{2} : \Delta\epsilon_p V_p + O(\Delta t^3). \tag{31}$$

This form is often used more generally for computational efficiency, using only the volume at a particular time during a time step to update the strain energy, which is first-order accurate in Δt.

For infinitesimal deformations Eq. (31) is exact for hypoelastic materials, for which the tangent modulus is constant and the expansion Eq. (29a) has no quadratic or higher order terms. Using this approximation to the change in the internal energy rather than that used previously [15] results in accuracy of energy accounting throughout to order Δt^2 on the material points.

Equation (20) may be rewritten, using Eqs. (11) and (22), as

$$\Delta \mathcal{KE}_{grid} = \sum_v m_v \Delta \mathbf{v}_v \cdot \left(\mathbf{v}_v^0 + \frac{1}{2} \Delta \mathbf{v}_v \right). \tag{32}$$

At this point, from Eqs. (9), (10), and (22), it is noted that for $\mathbf{f}_v^{ext} = 0$,

$$\Delta \mathbf{v}_v = -\frac{1}{m_v} \sum_p \nabla S_{vp} \cdot \boldsymbol{\sigma}_p V_p \Delta t. \tag{33}$$

Substituting Eq. (33), switching the order of summation, and substituting Eqs. (7) and (14), the change in kinetic energy on the grid, Eq. (32), may be rewritten as

$$\Delta \mathcal{KE}_{grid} = -\frac{1}{2} \sum_p \boldsymbol{\sigma}_p : \left(\Delta \boldsymbol{\epsilon}_p^0 + \Delta \boldsymbol{\epsilon}_p^1 \right) V_p. \tag{34}$$

From Eqs. (30) and (34), $\Delta \mathcal{E}_{algorithm}$ becomes

$$\Delta \mathcal{E}_{algorithm} = \frac{1}{2} \sum_p \boldsymbol{\sigma}_p : \left(\Delta \boldsymbol{\epsilon}_p^0 + \Delta \boldsymbol{\epsilon}_p^1 \right) V_p - \sum_p \frac{\boldsymbol{\sigma}_p^0 + \boldsymbol{\sigma}_p^1}{2} : \Delta \boldsymbol{\epsilon}_p V_p. \tag{35}$$

This second contribution to energy conservation error is also of order Δt^2, but may be of either sign, and is strongly dependent on when the stress state is updated. Specifically, $\boldsymbol{\sigma}_p$, appearing in the first term on the right hand side of Eq. (35), and $\Delta \boldsymbol{\epsilon}_p$, appearing in the second term, depend on whether the stress is updated at the beginning or at the end of the algorithm. The two possibilities are considered separately.

3.1. Update Material Point Stresses First (USF) Algorithm

As outlined in the previous section on the MPM algorithm, the stress on the material points may be updated based on the strain increment, $\Delta \boldsymbol{\epsilon}_p^0$, calculated from the initial material point velocities interpolated to the grid, $\mathbf{v}_v^0$ (Eq. (7)). This option is referred to subsequently as the update-stress-first (USF) algorithm. For this case $\boldsymbol{\sigma}_p^1$ is given by Eq. (8), and the grid velocity increment is determined from the updated stress,

$$\Delta \mathbf{v}_v = -\frac{1}{m_v} \sum_p \nabla S_{vp} \cdot \boldsymbol{\sigma}_p^1 V_p \Delta t. \tag{36}$$

The change in kinetic energy on the grid is then, from Eq. (34),

$$\Delta \mathcal{KE}_{grid} = -\frac{1}{2} \sum_p \boldsymbol{\sigma}_p^1 : \left(\Delta \boldsymbol{\epsilon}_p^0 + \Delta \boldsymbol{\epsilon}_p^1 \right) V_p, \tag{37}$$

and the change in strain energy of the material points is given from Eq. (30), using strain increment $\Delta\boldsymbol{\epsilon}_p^0$,

$$\Delta\mathcal{SE}_{points} = \sum_p \frac{\boldsymbol{\sigma}_p^0 + \boldsymbol{\sigma}_p^1}{2} : \Delta\boldsymbol{\epsilon}_p^0 V_p. \tag{38}$$

Finally, the algorithm error is

$$\Delta\mathcal{E}_{algorithm} = \frac{1}{2}\sum_p \left(\boldsymbol{\sigma}_p^1 : \Delta\boldsymbol{\epsilon}_p^1 - \boldsymbol{\sigma}_p^0 : \Delta\boldsymbol{\epsilon}_p^0\right) V_p. \tag{39}$$

The error term is the difference between two strain energy-like terms.

In practice the fourth-order tangent stiffness tensor, $\mathbf{L}$, is used to advance the stress over a time step, i.e., Eq. (8) is rewritten as

$$\boldsymbol{\sigma}_p^1 = \boldsymbol{\sigma}_p^0 + \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^0. \tag{40}$$

In general the tangent stiffness tensor is state dependent and varies from time step to time step. Substituting (40) into (39) gives

$$\Delta\mathcal{E}_{algorithm} = \frac{1}{2}\sum_p \left(\boldsymbol{\sigma}_p^1 : \left(\Delta\boldsymbol{\epsilon}_p^1 - \Delta\boldsymbol{\epsilon}_p^0\right) + \Delta\boldsymbol{\epsilon}_p^0 : \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^0\right) V_p. \tag{41}$$

Using the definition of the strain increments, Eqs. (7) and (14), along with Eq. (36), the algorithm error becomes

$$\Delta\mathcal{E}_{algorithm} = -\frac{1}{2}\sum_v m_v \Delta\mathbf{v}_v \cdot \Delta\mathbf{v}_v + \frac{1}{2}\sum_p \Delta\boldsymbol{\epsilon}_p^0 : \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^0 V_p. \tag{42}$$

Finally the total error increment, Eq. (17), using Eqs. (24) and (42) is found to be

$$\Delta\mathcal{E}_{error} = \frac{1}{2}\sum_{vv'} m_{vv'} \Delta\mathbf{v}_v \cdot \Delta\mathbf{v}_{v'} - \frac{1}{2}\sum_p \Delta\boldsymbol{\epsilon}_p^0 : \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^0 V_p \tag{43}$$

for the USF algorithm. Written this way, $\Delta\mathcal{E}_{error}$ takes a form which is suggestive of a trade-off between an incremental change in kinetic energy and an incremental change in strain energy. As such, these terms might be expected to cancel each other out.

3.2. Update Material Point Stresses Last (USL) Algorithm

The other option, equally valid from an algorithmic standpoint, is to update the stress on the material points at the end of the time step, using the strain increment, $\Delta\boldsymbol{\epsilon}_p^1$, calculated from the updated material point velocities interpolated to the grid, $\mathbf{v}_v^1$ (Eq. (14)). This option is referred to subsequently as the update-stress-last (USL) algorithm. For this case $\boldsymbol{\sigma}_p^1$ is given by Eq. (15), but the grid velocity increment is determined from the initial stress,

$$\Delta\mathbf{v}_v = -\frac{1}{m_v}\sum_p \nabla S_{vp} \cdot \boldsymbol{\sigma}_p^0 V_p \Delta t. \tag{44}$$

The change in kinetic energy on the grid is then, from Eq. (34),

$$\Delta \mathcal{KE}_{grid} = -\frac{1}{2}\sum_p \boldsymbol{\sigma}_p^0 : \left(\Delta\boldsymbol{\epsilon}_p^0 + \Delta\boldsymbol{\epsilon}_p^1\right) V_p, \tag{45}$$

and the change in strain energy of the material points uses the strain increment $\Delta\boldsymbol{\epsilon}_p^1$,

$$\Delta \mathcal{SE}_{points} = \sum_p \frac{\boldsymbol{\sigma}_p^0 + \boldsymbol{\sigma}_p^1}{2} : \Delta\boldsymbol{\epsilon}_p^1 V_p. \tag{46}$$

Finally the algorithm error is given by

$$\Delta \mathcal{E}_{algorithm} = \frac{1}{2}\sum_p \left(\boldsymbol{\sigma}_p^0 : \Delta\boldsymbol{\epsilon}_p^0 - \boldsymbol{\sigma}_p^1 : \Delta\boldsymbol{\epsilon}_p^1\right) V_p. \tag{47}$$

The error term has exactly the opposite sign as for the USF algorithm, Eq. (39).

It is clear then that when the material point stresses are updated has a strong impact on the sign of the incremental error term $\Delta\mathcal{E}_{algorithm}$. If $\Delta\mathcal{E}_{algorithm}$ were of one sign for one choice of the algorithm (starting from the same conditions at the beginning of a time step), it would be of the opposite sign for the other choice.

It should be noted at this point that in fact the error terms are not exact opposites of each other in the two algorithms. There is steady divergence on account of the grid velocity updates, Eqs. (36) and (44), which use updated and initial stresses, respectively. In addition, updating material point stresses also results in updates to material point volumes for finite deformations. The corrections are small. If one chooses for convenience to define Eq. (38) using the updated volume V_p^1 and Eq. (46) using volume V_p^0, Eqs. (39) and (47) still follow, but with V_p^1 and V_p^0 appearing, respectively.

As for the USF algorithm, in practice the stress is advanced using the tangent stiffness tensor, $\mathbf{L}$. Equation (15) may be rewritten as

$$\boldsymbol{\sigma}_p^1 = \boldsymbol{\sigma}_p^0 + \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^1. \tag{48}$$

Substituting (48) into (47) gives

$$\Delta \mathcal{E}_{algorithm} = \frac{1}{2}\sum_p \left(\boldsymbol{\sigma}_p^0 : \left(\Delta\boldsymbol{\epsilon}_p^0 - \Delta\boldsymbol{\epsilon}_p^1\right) - \Delta\boldsymbol{\epsilon}_p^1 : \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^1\right) V_p. \tag{49}$$

Using the definition of the strain increments, Eqs. (7) and (14), along with Eq. (44), the algorithm error becomes

$$\Delta \mathcal{E}_{algorithm} = \frac{1}{2}\sum_v m_v \Delta\mathbf{v}_v \cdot \Delta\mathbf{v}_v - \frac{1}{2}\sum_p \Delta\boldsymbol{\epsilon}_p^1 : \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^1 V_p. \tag{50}$$

Finally the total error increment, Eq. (17), using Eqs. (24) and (50) is found to be

$$\Delta \mathcal{E}_{error} = -2\Delta\mathcal{E}_{interpolation} - \frac{1}{2}\sum_{vv'} m_{vv'} \Delta\mathbf{v}_v \cdot \Delta\mathbf{v}_{v'} + \frac{1}{2}\sum_p \Delta\boldsymbol{\epsilon}_p^1 : \mathbf{L} : \Delta\boldsymbol{\epsilon}_p^1 V_p \tag{51}$$

for the USL algorithm. Written this way, $\Delta\mathcal{E}_{error}$ takes a form similar to that in Eq. (43), but with an additional dissipative term. If Eq. (43) were found to be zero, the error for the USL algorithm would be strictly dissipative and with an absolute value equal to twice the interpolation error, $\Delta\mathcal{E}_{interpolation}$.

Both versions of the algorithm have found application in the literature. The USL algorithm is presented and applied in [14–17]. Elastic vibration calculations, with results consistent with those presented in the following section, are presented in [14, 15], although the internal energy was calculated differently and the error was not computed. The USF algorithm is used in [1, 3, 4].

4. APPLICATIONS

To experiment numerically with the two versions of the MPM algorithm easily, a one-dimensional MPM code was written. Its modular design allows the material point stress state to be easily updated either at the beginning (USF) or end (USL) of the algorithm. A classical example of a conservative system is free vibration, two examples of which are considered here. To determine the accuracy of the numerical results, it is useful to have exact solutions to compare against. Analytical solutions can be derived for free vibration of elastic material for the two cases considered here. The first is a single-material-point problem, for which the MPM algorithm equations can be solved analytically in the limit $\Delta t \to 0$. The second is the axial vibration of a continuum bar, which can then be discretized and solved approximately using MPM.

4.1. Single-Material-Point Vibration

The initial setup is one material point, with velocity v_0, in a cell with vertices at $x = 0$ and L (see Fig. 1). Boundary conditions are imposed on the grid and demand that both the vertex velocity and the acceleration at $x = 0$ be zero throughout the simulation. The interpolation function for the vertex at $x = L$ is also shown in Fig. 1. It is the only interpolation function needed, on account of the null values of the interpolants at the left vertex. A hypoelastic constitutive model is used for the material points, i.e.,

$$\dot{\sigma} = E\dot{\epsilon}, \tag{52}$$

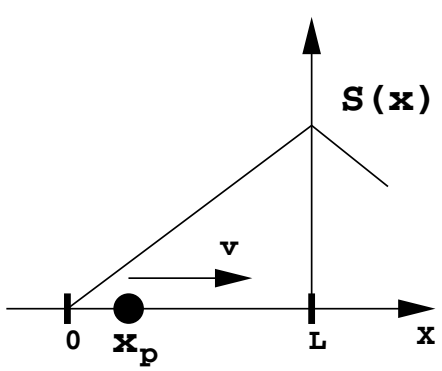

FIG. 1. Initial configuration for the single-material-point problem. The material point is shown as a filled circle, and grid vertices are indicated with thick black ticks marks. The interpolation function at $x = L$ is also shown.

where $\dot{\epsilon} = dv/dx$ and modulus E is a constant. This material response allows the equations to be easily solved analytically.

In the limit $\Delta t \to 0$ there is no difference between the USL and USF algorithms. The steps in either algorithm give the following updates for material point variables, where the subscript p has been omitted because only one material point is considered. The material point strain update, Eq. (7) or (14), becomes

$$\dot{\epsilon} = \frac{v}{L}. \tag{53}$$

The stress update is given by Eq. (52), and the material point velocity update is, from Eqs. (13), (22), and (33),

$$\dot{v} = -\frac{\sigma V}{mL}. \tag{54}$$

These equations may be combined to give

$$\ddot{v} + \frac{E}{\rho L^2} v = 0, \tag{55}$$

where $\rho = m/V$.

This equation may be easily solved using the initial conditions $v(0) = v_0$, $\dot{v}(0) = 0$ to give

$$v(t) = v_0 \cos(\omega t), \tag{56}$$

where $\omega = \sqrt{E/\rho}/L$ and the density is assumed to remain constant. The solution corresponds to harmonic oscillation with frequency ω. Because of the way material point positions are updated, the position is not a simple integration of Eq. (56). Rather, the algorithm specifies, from Eq. (12),

$$\dot{x} = S(x)v. \tag{57}$$

This equation may be integrated, along with initial condition $x(0) = x_0$, to give

$$x(t) = x_0 \exp\left[\frac{v_0}{L\omega} \sin(\omega t)\right]. \tag{58}$$

For $v_0/L\omega \ll 1$, Eq. (58) is approximately a sinusoidal oscillation. It can easily be shown that the analytical solution corresponds to oscillatory material point stress and strain, such that the sum of the kinetic and strain energies is constant, and energy is exactly conserved on the material point.

Energy conservation on the material point is approximate for the discrete algorithm, and the error terms are of order Δt^2. A numerical example illustrates how the errors depend on the details of the algorithm. Solutions were obtained with material point masses equal to 1, volumes equal to 1, and moduli equal to $4\pi^2$. The cell size is $L = 1$, and initial conditions are $x_0 = 0.5$, $v_0 = 0.1$. Any set of consistent units suffice. The material point volume, and hence density, remain constant and equal to unity in the calculations. The elastic wave speed in the problem, c, is $\sqrt{E/\rho} = 2\pi$, and a time step of $0.1L/c$ (10% of the CFL limit on the

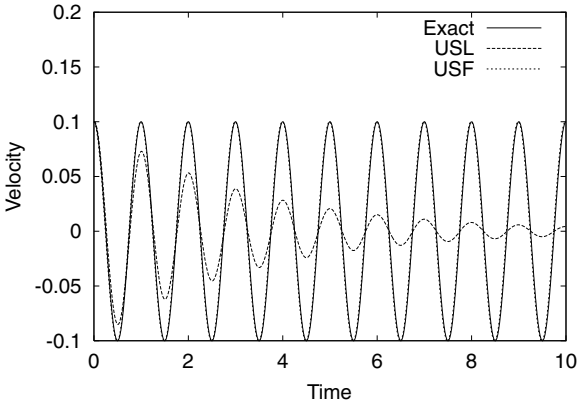

FIG. 2. Velocity during oscillation for both variations of the MPM algorithm. The exact solution is shown for reference.

explicit time step size) gives the results seen in Fig. 2, where the material point velocity is shown. The exact solution and that using the USF algorithm are indistinguishable. The strong dissipation seen for the USL algorithm can be understood by examining the energy error terms. At this point, it is simply noted that the time step is sufficiently small to very accurately capture the period of oscillation of the analytical solution ($2\pi/\omega = 1$) using either algorithm. It should be noted, however, that this solution is a poor approximation to the axial vibration of a continuum bar (see the following section), as the period of oscillation is overestimated by nearly 60%.

Numerical results for the energies are depicted in Fig. 3. Strain energy, kinetic energy, and their sum, the total energy, are depicted. The analytical solution conserves total energy, at a value equal to the initial kinetic energy of the system, 0.005. The USL results are strongly dissipative, and the USF results appear conservative on average. The oscillations are caused by the strain energy leading the kinetic energy in the USL algorithm, and by lagging in the USF algorithm. It would be natural to update the strain energy at half time steps, but both algorithms update the strain energy at the end of the time step, where the kinetic energy is calculated, for simplicity. The strain energy leads in the USL algorithm because the updated value of the strain is used, while the opposite is the case for the USF algorithm. Oscillations and damping consistent with these results are seen in [14], in two-dimensional elastic vibration calculations, where the USL algorithm is used.

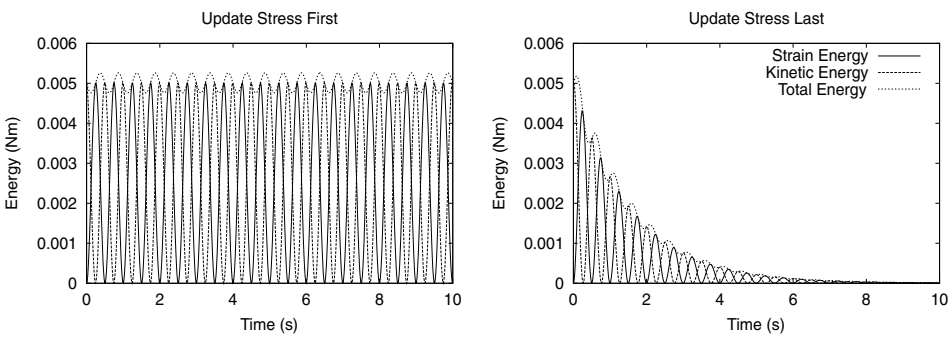

FIG. 3. Oscillation energies for both variations of the MPM algorithm. The legend applies to both algorithms.

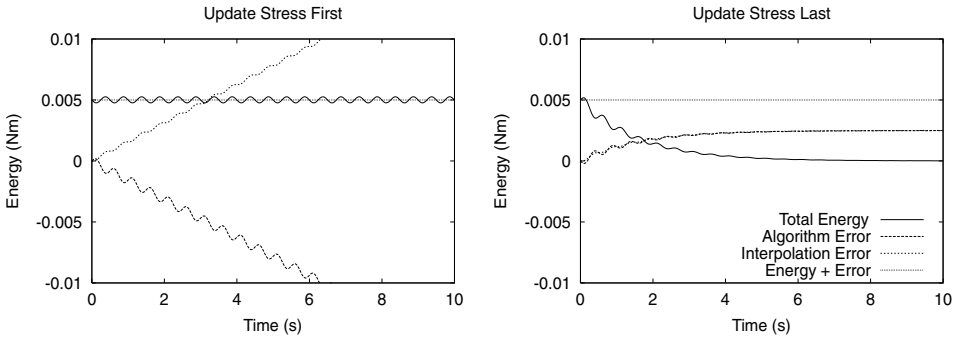

FIG. 4. Oscillation energy and error terms for both variations of the MPM algorithm. The legend applies to both algorithms.

The numerical solution error terms are depicted in Fig. 4, along with the total energy and the sum of the three. The sum is constant and equal to the initial energy of the system. The difference between the algorithms is seen in the algorithm errors for the two cases. For the USF algorithm, the algorithm error is generally negative, supplying energy to the system. However, as always, the interpolation error is positive, dissipating energy. The two errors miraculously cancel, on average, creating an apparently conservative system overall. This adds credence to the postulate made in conjunction with Eq. (43), that the total error for the USF algorithm is the sum of incremental kinetic and strain energy terms. While the total error is not identically zero, the fact that the total energy oscillates about the energy + Error indicates that it is zero on average. The USL algorithm is quite different, with both errors positive and dissipative, resulting in the rapid damping out of the motion. Here again there is evidence that the incremental kinetic energy and incremental strain energy terms cancel in Eq. (51). While not identical, the algorithm error and the interpolation error oscillate about the same mean value and tend toward the same asymptote, indicating that the total error is dissipative, with absolute value equal to twice the interpolation error on average.

The sign of the algorithm error terms can be found analytically for this simple problem. Assuming the exact solution for the velocity and stress at the beginning of a time step ($t = t_0$), the stress is

$$\sigma_p^0 = \frac{E}{\omega}\frac{v_0}{L}\sin(\omega t_0), \tag{59}$$

and the strain increment, $\Delta\epsilon_p^0$, is

$$\Delta\epsilon_p^0 = \frac{v_0}{L}\cos(\omega t_0)\Delta t. \tag{60}$$

The updated stress, σ_p^1, and strain increment, $\Delta\epsilon_p^1$, are found using the USF and USL algorithms. Evaluating the incremental algorithm error terms, and integrating over one period to eliminate periodic fluctuations, for the USF algorithm, from Eq. (41),

$$\int_0^{\frac{2\pi}{\omega}} \Delta\mathcal{E}_{algorithm}\, dt = \frac{1}{2}E\frac{v_0^2}{L^2}\Delta t^2\left(\frac{\pi}{\omega} - \int_0^{\frac{2\pi}{\omega}} \frac{\sin^2(\omega t)}{S_{vp}}\, dt\right)V_p \le 0. \tag{61}$$

For the USL algorithm, from Eq. (49),

$$\int_0^{\frac{2\pi}{\omega}} \Delta\mathcal{E}_{algorithm}\, dt = \frac{1}{2} E \frac{v_0^2}{L^2} \Delta t^2 \left(\int_0^{\frac{2\pi}{\omega}} \frac{\sin^2(\omega t)}{S_{vp}}\, dt - \frac{\pi}{\omega} \right) V_p \geq 0. \tag{62}$$

In both cases the sign of the error is determined using the fact that $S_{vp} \leq 1$, from Eq. (1).

It should be noted that, consistent with either algorithm, momentum is conserved in these calculations. An oscillating and translating system will continue to translate with no loss in total momentum. The motion relative to the center of mass (which is all the motion in the above examples) is damped out quickly by the USL algorithm, but not by the USF algorithm. Multiple material point discretizations of a continuum are considered next to examine the energy properties for more general cases.

4.2. Axial Vibration of a Continuum Bar

Under infinitesimal displacements, the equations for free axial vibration of a uniform continuous bar may be solved using separation of variables [13]. The bar is found to oscillate in modes, which are dependent on the boundary conditions. The case considered here is the analogy to the single-material-point problem. One end ($x = 0$) of the bar is fixed, and the other ($x = L$) is free, where L is the length of the bar.

The modes of vibration, ϕ_n, are

$$\phi_n(x) = \sin \beta_n x, \quad \beta_n = \frac{2n-1}{2} \frac{\pi}{L}, \quad n = 1, 2, \ldots. \tag{63}$$

For the examples here, L is taken to be an integral number of computational cells. The frequencies of oscillation, ω_n, are related to eigenvalues, β_n, by

$$\omega_n = \beta_n c, \tag{64}$$

where $c = \sqrt{E/\rho}$, as before. A general solution for any initial conditions may found as a Fourier sine series for the displacement, $u(x, t)$,

$$u(x, t) = \sum_{n=1}^{\infty} (A_n \sin \omega_n t + B_n \cos \omega_n t)\, \phi_n(x), \tag{65}$$

where A_n and B_n are constants determined from the initial conditions. However, it is possible to excite specific modes using initial conditions which are multiples of the natural modes of the system. Only mode n is excited using initial conditions $u(x, 0) = 0$, $v(x, 0) = \dot{u}(x, 0) = v_0 \sin \beta_n x$; i.e., the solutions are

$$u(x, t) = \frac{v_0}{\omega_n} \sin \omega_n t \sin \beta_n x, \tag{66}$$

$$v(x, t) = v_0 \cos \omega_n t \sin \beta_n x, \tag{67}$$

where v_0 is the amplitude of the initial velocity.

The response depicted in Fig. 5 was elicited for both versions of the MPM algorithm using mode 1($n = 1$) initial conditions. The velocity of the center of mass is shown for a 50 material point discretization, with two material points per cell. The cell length is $\Delta x = 1$,

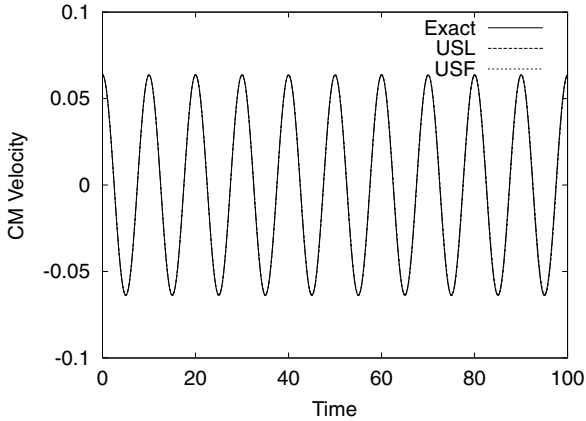

FIG. 5. First-mode oscillation center-of-mass velocity for both variations of the MPM algorithm. The exact solution is shown for reference.

$L = 25$. The modulus $E = 100$ and density $\rho = 1$, giving the wave speed $c = 10$. The velocity amplitude $v_0 = 0.1$, and a time step of $0.1\Delta x/c$ (10% of the CFL limit on the explicit time step size) are used in the calculations. As in the previous example, any set of consistent units may be used.

The exact solution for a continuum is shown for comparison. The period of vibration is $2\pi/\omega_1 = 10$, and the wavelength of the first mode is $2\pi/\beta_1 = 4L$. The center of mass motion $v_{cm}(t)$ is found from the exact solution by taking the first mass-weighted moment of the velocity,

$$v_{cm}(t) = \int_0^L \rho(x)v(x,t)\,dx \Big/ \int_0^L \rho(x)\,dx = \frac{v_0}{\beta_n L}\cos(\omega_n t). \tag{68}$$

All center-of-mass velocities overlie one another in Fig. 5.

Figure 6 depicts total material point energies for both algorithms for mode 1 excitation. No perceptible difference can be discerned between the two algorithms, and very little error is found. Conspicuously absent is the strong dissipation seen previously for the USL algorithm. Figure 7 depicts total energies and error terms for both algorithms for mode 1 excitation. Again no perceptible differences are seen. For the first mode, the interpolation error is essentially zero, and the algorithm error results in barely perceptible oscillations.

To investigate further, higher modes are excited. Figure 8 depicts exact and numerical solutions for the fifth-mode center-of-mass velocity. The exact solution has period $2\pi/\omega_5 = 10/9$ and modal wavelength $4L/9$. The numerical solutions are run for 10 periods of oscillation, as for the first mode. For this mode there is some error in the calculated frequencies, resulting in the period of the numerical solutions being about 5% too large. Additionally, dissipation is evident in the USL algorithm results.

Figure 9 depicts energies for both algorithms for mode 5 excitation. Results are reminiscent of the single-material-point problem. The USF algorithm appears conservative on average, and the dissipative nature of the USL algorithm is evident. Figure 10 depicts total energies and error terms for both algorithms for mode 5 excitation. The trends established in the single-material-point problem are again exhibited. For this mode, with a wavelength

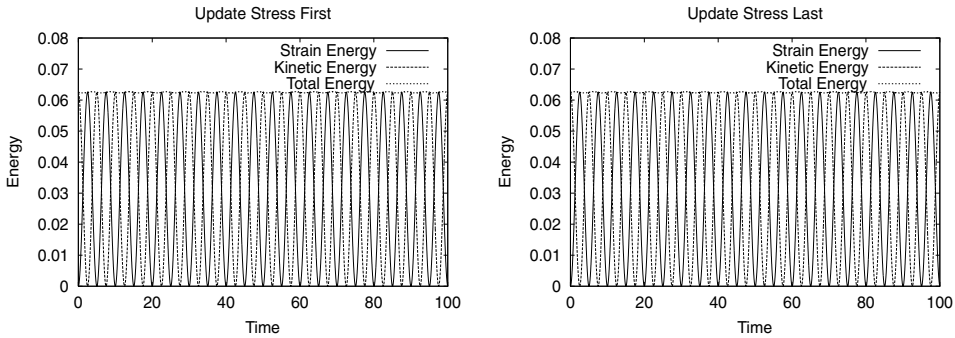

FIG. 6. First-mode oscillation energies for both variations of the MPM algorithm.

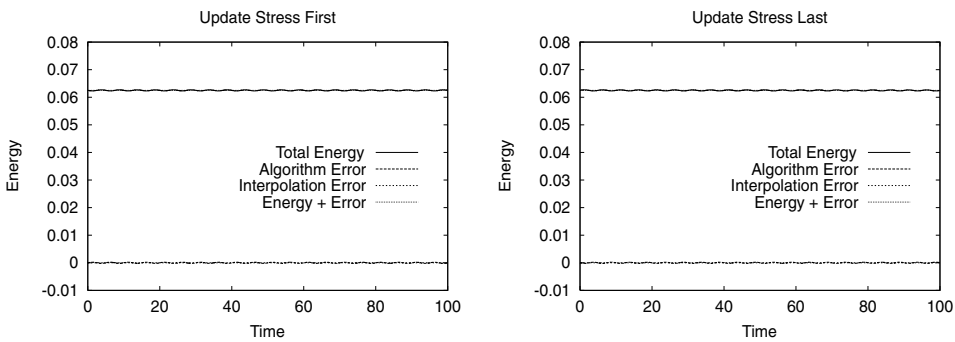

FIG. 7. First-mode oscillation energy and error terms for both variations of the MPM algorithm.

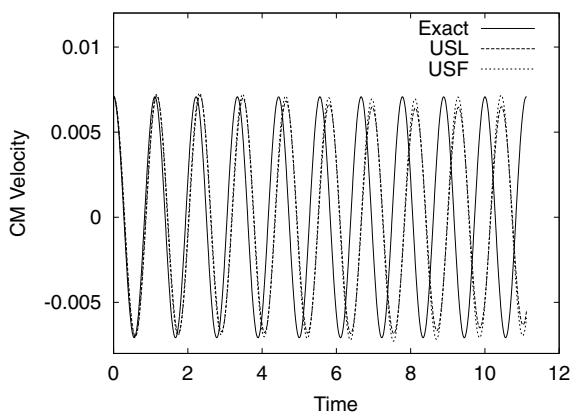

FIG. 8. Fifth-mode oscillation center-of-mass velocity for both variations of the MPM algorithm. The exact solution is shown for reference.

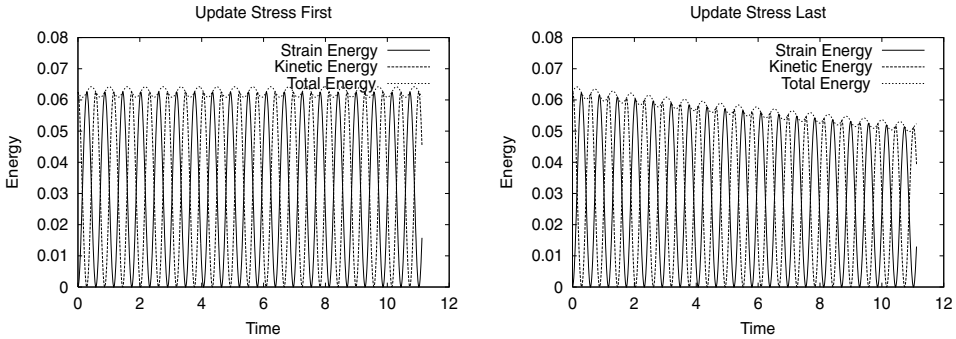

FIG. 9. Fifth-mode oscillation energies for both variations of the MPM algorithm.

of approximately 11 computational cells, the error is associated with difficulty in resolving the mode on the grid. Increasing the resolution decreases the error. This behavior has been seen with the FLIP algorithm as well [5, 8].

To further test the algorithms, the tenth mode is excited. Figure 11 depicts exact and numerical solutions for the tenth-mode center-of-mass velocity. The exact solution has period $2\pi/\omega_{10} = 10/19$ and modal wavelength $4L/19$. The numerical solutions are run for 10 periods of oscillation, as before. For this mode there is more error in the calculated frequencies, resulting in the period of numerical solutions being about 20% too large. Significant dissipation is evident in the USL algorithm results.

Figure 12 depicts energies for both algorithms for mode 10 excitation. Results are reminiscent of those for mode 5, but with larger error oscillations for both algorithms. The USF algorithm appears conservative on average, and the USL algorithm is strongly dissipative. Figure 13 depicts total energies and error terms for both algorithms for mode 10 excitation. Again the trends established in the single-material-point problem are exhibited. For this mode the wavelength is approximately five computational cells, and there is error in resolving the mode on the grid. In fact there is error associated with resolving the initial conditions, and longer runs indicate longer wavelength modes have also been excited.

The free-vibration numerical results given in this section may be considered converged in the sense that decreases in time step magnitude do not change the calculated periods

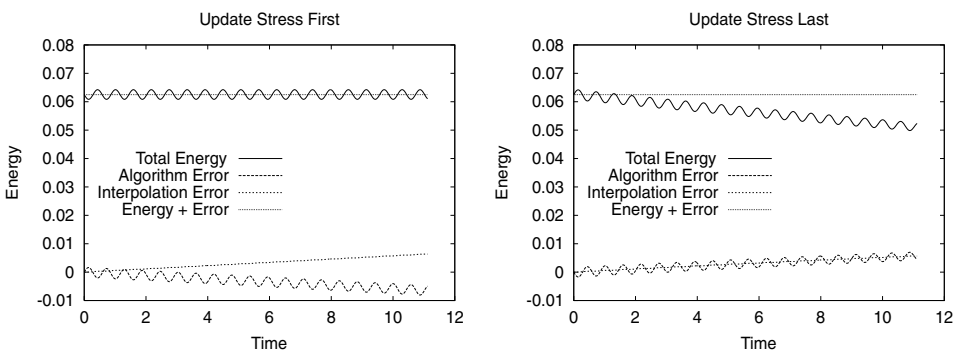

FIG. 10. Fifth-mode oscillation energy and error terms for both variations of the MPM algorithm.

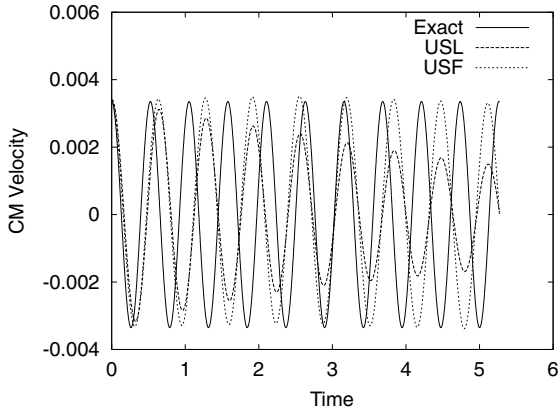

FIG. 11. Tenth-mode oscillation center-of-mass velocity for both variations of the MPM algorithm. The exact solution is shown for reference.

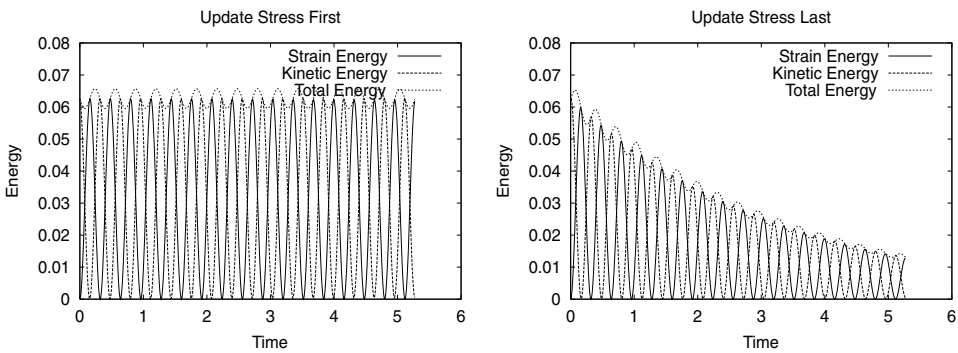

FIG. 12. Tenth-mode oscillation energies for both variations of the MPM algorithm.

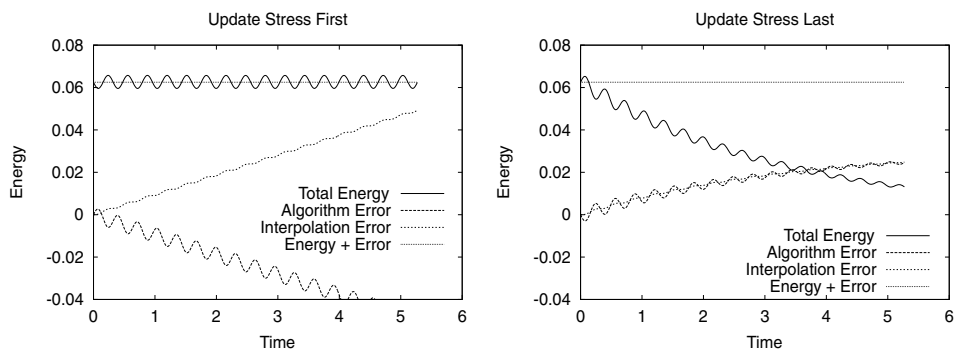

FIG. 13. Tenth-mode oscillation energy and error terms for both variations of the MPM algorithm.

of oscillation. Effectively, the numerical discretization results in the solution to a softer system when the mode is unresolved on the grid. The limiting case is given for the one-material-point problem, where the period is overestimated by almost 60%. Of course, energy errors and dissipation decrease with decreasing time step. It is also worth noting that these results are essentially unaffected by varying the number of material points per cell.

5. CONCLUSIONS

This manuscript details energy conservation properties of two variants of the MPM algorithm, where material response is determined on discrete points (material points), while the governing equations are solved on an overlying grid. Consistent with the constitutive response, the material internal energy is also carried on the material points. While this provides substantial simplification in the implementation of complex material response relations, it complicates the calculation of change in energy over a time step. Material point internal energies at the beginning and end of a time step, during which information is interpolated from material points to grid and back again, must be compared.

In the application of the MPM algorithm there is no reason *a priori* to prefer to update the material point stresses and internal energies at the beginning (USF algorithm) or end (USL algorithm) of a time step. Both versions of the algorithm conserve mass and momentum. Here the internal and kinetic energies on material points, as well as errors due to discretization, are considered. The energy accounting benefits from a more accurate calculation of the internal energy than used previously, resulting in accuracy to order Δt^2 throughout the calculations. For both algorithms it is found that the error may be partitioned into that due to interpolation plus a remainder (termed the algorithm error). The interpolation error is a more general property of PIC algorithms and is dissipative. The algorithm error is specific to the MPM algorithms and is found to depend strongly on the order of updating material point states, having opposite signs for USF and USL algorithms.

The error properties of the two algorithms are demonstrated using free-vibration examples. Numerical solutions are compared with the exact solution for a single-material-point problem, for which the discrete equations may be solved analytically. For the single-material-point problem both algorithms accurately reproduce the exact solution period of oscillation, but the USL algorithm is strongly dissipative. For this simple example the sign of the error terms may be found analytically. Both terms are of the same sign and dissipative for the USL algorithm. For the USF algorithm the error terms are of opposite sign and cancel each other out, resulting in a conservative system.

Numerical solutions are compared with the exact solution for free axial vibration of a continuum bar. It is found that both algorithms give identical results with negligible energy error or dissipation when the oscillation-mode wavelength is long relative to the cell size, i.e., the mode is resolved on the computational grid. When the mode is not resolved, both algorithms overestimate the period of oscillation with equal error, increasing as the wavelength approaches the computational cell size. The single-material-point problem may be considered the coarsest possible discretization of the axial vibration of a bar, for which the period of oscillation obtained via the MPM algorithm is in error by more than 50%. For the unresolved modes the USL algorithm is dissipative, while the USF algorithm is conservative. The USL algorithm may be described as tending to damp out unresolved

modes. The USL algorithm may be a better choice, as the damping is consistent with the accuracy of the solution.

ACKNOWLEDGMENTS

The author would like to acknowledge helpful discussions with Dr. J. U. Brackbill, Theoretical Division, Los Alamos National Laboratory. This work was supported by the U.S. Department of Energy through the Center for the Simulation of Accidental Fires and Explosions, under Grant W-7405-ENG-48.

REFERENCES

1. S. G. Bardenhagen and J. U. Brackbill, Dynamic stress bridging in granular material, *J. Appl. Phys.* **83**, 5732 (1998).
2. S. G. Bardenhagen, J. U. Brackbill, and D. Sulsky, The material-point method for granular materials, *Comput. Methods Appl. Mech. Eng.* **187**, 529 (2000).
3. S. G. Bardenhagen, J. U. Brackbill, and D. Sulsky, Numerical study of stress distributions in sheared granular material in two dimensions, *Phys. Rev. E* **62**, 3882 (2000).
4. S. G. Bardenhagen, E. N. Harstad, P. J. Maudlin, G. T. Gray, and J. C. Foster, Viscoelastic models for explosive binder materials, *Shock Compression Condens. Matter* 281 (1998).
5. J. U. Brackbill, The ringing instability in particle-in-cell calculations of low-speed flow, *J. Comput. Phys.* **75**, 469 (1988).
6. J. U. Brackbill, Flip-mhd: A particle-in-cell method for magnetohydrodynamics, *J. Comput. Phys.* **96**, 163 (1991).
7. 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**, 25 (1988).
8. J. U. Brackbill and H. M. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations in two dimensions, *J. Comput. Phys.* **65**, 314 (1986).
9. D. Burgess, D. Sulsky, and J. U. Brackbill, Mass matrix formulation of the FLIP particle-in-cell method, *J. Comput. Phys.* **103**, 1 (1992).
10. Y. C. Fung, *Foundations of Solid Mechanics* (Prentice Hall, Engelwood Cliffs, NJ, 1965).
11. F. H. Harlow, The particle-in-cell computing method for fluid dynamics. *Methods Comput. Phys.* **3**, 319 (1963).
12. A. Lenardic, L. Moresi, and H. Mühlhaus, The role of mobile belts for the longevity of deep cratonic lithosphere: The crumpled zone model, *Geophys. Res. Lett.* **27**, 1235 (2000).
13. L. Meirovitch, *Analytical Methods in Vibrations* (Macmillan, New York, 1967).
14. D. Sulsky, Z. Chen, and H. L. Schreyer, A particle method for history-dependent materials, *Comput. Methods Appl. Mech. Eng.* **118**, 179 (1994).
15. D. Sulsky and H. L. Schreyer, Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems, *Comput. Methods Appl. Mech. Eng.* **139**, 409 (1996).
16. D. Sulsky, S.-J. Zhou, and H. L. Schreyer, Application of a particle-in-cell method to solid mechanics, *Comput. Phys. Commun.* **87**, 236 (1995).
17. Z. Więckowski, S.-K. Youn, and J.-H. Yeon, A particle-in-cell solution to the silo discharging problem, *Int. J. Numer. Methods Eng.* **45**, 1203 (1999).
18. A. R. York, D. Sulsky, and H. L. Schreyer, The material point method for simulation of thin membranes, *Int. J. Numer. Methods Eng.* **44**, 1429 (1999).
19. S. Zhou, J. Stormont, and Z. Chen, Simulation of geomembrane response to settlement in landfills by using the material point method, *Int. J. Anal. Methods Geomech.* **23**, 1977 (1999).