Journal of Computational Physics 229 (2010) 7819–7833

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

# Distribution coefficient algorithm for small mass nodes in material point method

Xia Ma, Paul T. Giguere, Balaji Jayaraman, Duan Z. Zhang *

*Theoretical Division, Fluid Dynamics and Solid Mechanics Group (T-3, B216), Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

ARTICLE INFO

*Article history:*
Received 25 February 2010
Received in revised form 14 June 2010
Accepted 25 June 2010
Available online 30 June 2010

*Keywords:*
Material point method
Multiphase flow
Fluid-structure interaction
Large deformation

ABSTRACT

When using the time explicit material point method to simulate interaction of materials accompanied by large deformations and fragmentation, one often encounters a numerical instability caused by small node mass, because acceleration on a mesh node is obtained by dividing the total force on the node by the mass of the node. When the material points are in the far sides of the cells containing the node, typically happening near material interfaces, the node mass can be very small leading to artificially large acceleration and then numerical instability. For the case of small material deformations, this instability is typically avoided by placing the material points away from cell boundaries. For cases with large deformations, with the exception of initial conditions, there is no control on locations of the material points. The instability caused by small mass nodes is often encountered. To avoid this instability tiny time steps are usually required in a numerical calculation.

In this work, we present a numerical algorithm to treat this instability. We show that this algorithm satisfies mass and momentum conservation laws. The error in energy conservation is proportional to the second order of the time step, consistent with the explicit material point method. Numerical implementation of the algorithm is described. Numerical examples show effectiveness of the algorithm.



## 1. Introduction

One of the significant advantages of the material point method (MPM) is its capability of tracking material interfaces. However, direct application of the material point method often leads to instability or introduces significant noise in the calculation because of the small mass on nodes near the material interface. Numerically, this instability is a consequence of the time lag of the stress in a time explicit calculation. Although this instability can in principle be eliminated in a time implicit calculation, an implicit material point method capable of considering interaction of materials with significantly different constitutive relations is yet to be developed, especially when the material point method is used in combination with an Eulerian method to study multiphase flows or multimaterial interactions. Despite many disadvantages, time explicit methods may still be an effective choice in cases with complicated constitutive relations and with material speeds comparable to the speed of sound of the material. The main objective of the current paper is to obtain a numerical algorithm to overcome the instability caused by small node mass.

Loosely speaking, in a numerical calculation of an elastic problem using a mesh based method, the elastic body is approximated by a network of springs and mass points at mesh nodes. The smallest elastic wave time scale in the discretized system is of order $\sqrt{m/K}$, where $m$ is the mass on a node and $K$ is the spring constant. The mass $m$ and the spring constant $K$ are

* Corresponding author. Tel.: +1 505 665 4428; fax: +1 505 665 5926.
*E-mail address:* dzhang@lanl.gov (D.Z. Zhang).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.06.041

functions of the mesh or computational cells used to approximate the elastic body. To explain the unique difficulties of the material point method, in this introduction we assume the aspect ratios (or $\Delta y/\Delta x, \Delta z/\Delta x$) of mesh cells are of order 1. This assumption is not needed in the rest of the discussion in the present paper. With this assumption, the mass $m$ is of order $\rho(\Delta x)^3$, where $\rho$ is the density of the elastic body. In a Lagrangian method the mass $m$ does not vary significantly during the calculation, and the spring constant is of order $E\Delta x$, where $E$ is the Young's modulus. Therefore the time scale $\sqrt{m/K} = \Delta x/c$ does not vary significantly during the calculation, where $c$ is the speed of sound in the elastic body. In an Eulerian method or an arbitrary Lagrangian Eulerian (ALE) method, stresses are advected together with the mass of the material, therefore the spring constant is proportional to the node mass and the time scale $\sqrt{m/K} = \Delta x/c$ is also kept close to a constant. In both of these cases, the elastic wave time scales of the discretized system do not vary significantly during the calculation.

In the material point method, however, while the spring constant $K$ is still of order $E\Delta x$, the node mass is calculated as the sum of the products of the shape function and the mass of the material point. When all material points are far away from a mesh node, the shape function approaches zero as does the node mass. Therefore the lower limit of the time scale $\sqrt{m/K}$ is zero in the material point method. To ensure numerical stability, in a time explicit calculation the time step $\Delta t$ has to be a fraction of the smallest elastic wave time scale in the system. Therefore if a proper algorithm is not used for these nodes, the time step could become too small for an time explicit scheme to make meaningful progress, or the calculation could become unstable. This numerical instability often happens near a material interface, when a material point enters a cell that does not contain particles in the previous time step. This instability is related to the cell crossing instability studied in [1], but the causes of the instabilities are different. The instability discussed in the present paper is caused by small mass on the nodes; while the instability studied in [1] is caused by the discontinuity in the gradients of the shape functions in the material point method.

We also note that the reduction of the time step because of the small mass nodes does not increase the accuracy of the calculation, because the shortest period or highest frequency waves resolved in a calculation are determined by the largest value of $\Delta x/c$, not the smallest one, for a given mesh. The reduction of the time step because of the small mass nodes is purely to satisfy the stability requirement without a benefit to the accuracy of the calculation.

The main objective of the present paper is to introduce an algorithm to overcome this difficulty in the material point method by restoring the smallest time scale to $\Delta x/c$ in the discretized system, so that the required time step for stability is not reduced.

Our algorithm is combined with the method [2] of satisfying the continuity constraint developed for the coupled calculation using the ALE method and the material point method. Such a coupled calculation is especially advantageous in problems of fluid-structure interactions. The fluid phase can be calculated using the ALE method, and the solid structure can be calculated using the material point method. The interactions of the fluid and solid phases are computed using the recently developed averaged equations for continuous multiphase flows [3]. The method of satisfying the continuity constraint in the coupled ALE and MPM calculation requires the use of the Lagrangian velocity to update stresses, which is a preferred method of calculating stresses from the point of view of numerical stability [4] in a time explicit calculation.

The effectiveness of this algorithm for small mass nodes is demonstrated by several examples in the last section of this paper. The numerical results of these examples are compared to analytical solutions or experiment data. Significant improvements in the time step size are confirmed.

To facilitate our discussion and to further illustrate the issues discussed in the present paper, we list basic steps of the material point method in the following section without derivation. These steps are not new; their derivation can be found in many published papers [2,5,6]. We include these steps in the present paper because they are useful in the discussion of the introduced numerical algorithm.

## 2. Material point method for multiphase systems

The governing equations for multimaterial interactions used in the present paper are obtained from the ensemble phase averaging technique [3]. For phase $k$ material, let $\boldsymbol{u}_k$ be the average velocity, $\theta_k$ the volume fraction, $\rho_k^0$ the average material density, $\boldsymbol{\sigma}_k$ the average stress tensor, $\boldsymbol{b}_k$ the specific (per unit mass) body force applied by a source external to the system, and $\boldsymbol{f}_k$ the phase interaction force at location $\boldsymbol{x}$ and time $t$. The averaged momentum equation can be written as

$$\theta_k\rho_k^0\frac{d\boldsymbol{u}_k}{dt} = \nabla\cdot[\theta_k(\boldsymbol{\sigma}_k + P\boldsymbol{I})] - \theta_k\nabla P + \theta_k\rho_k^0\boldsymbol{b}_k + \theta_k\boldsymbol{f}_k, \tag{1}$$

after setting the auxiliary stress $\boldsymbol{\sigma}_{Ak} = -P\boldsymbol{I}$ in Eq. (15) of [3], where $P$ is an auxiliary pressure in the system, and $\boldsymbol{I}$ is the identity tensor. Except for flows in porous media [7,8], the choice of the auxiliary stress $\boldsymbol{\sigma}_{Ak} = -P\boldsymbol{I}$ provides many conveniences in modeling multiphase flows, especially disperse multiphase flows. Although the pressure effect in the first and the second terms on the right hand side can be simplified as $P\nabla\theta_k$, the momentum equation in form (1) is especially convenient for its numerical implementation in the material point method, as shown in Section 6.2.

By introducing macroscopic density $\rho_k = \theta_k\rho_k^0$, stress $\boldsymbol{s}_k = \boldsymbol{\sigma}_k + P\boldsymbol{I}$, and force density $\rho_k\boldsymbol{G}_k = \rho_k\boldsymbol{b}_k - \theta_k\nabla P + \theta_k\boldsymbol{f}_k$, we can simplify (1) as

$$\rho\frac{d\boldsymbol{u}}{dt}=\rho\left(\frac{\partial\boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right)=\nabla\cdot(\theta\boldsymbol{s})+\rho\boldsymbol{G}, \tag{2}$$

where we have omitted subscript $k$ denoting the phases, since our discussion before Section 6 is not related to phase interactions. We will restore these subscripts in that section.

In the material point method, we use both an Eulerian mesh and Lagrangian material points. There is a shape function $S_i$ for each mesh node $i$. Each material point $p$ is assigned a mass $m_p$. The momentum equation is discretized as [2]

$$\sum_{j=1}^{N}m_{ij}\frac{d\boldsymbol{u}_j}{dt}\approx-(\theta\boldsymbol{s},\nabla S_i)+\sum_{j=1}^{N}m_{ij}\boldsymbol{G}_j+\int_{\partial\Omega}\theta\boldsymbol{s}\cdot\boldsymbol{n}S_i(\boldsymbol{x})dS, \tag{3}$$

where the subscripts denote node number, $N$ is the total number of nodes, $S_i$ is the shape function of node $i$, $(\theta\boldsymbol{s},\nabla S_i)$ is the inner product of $\theta\boldsymbol{s}$ and $\nabla S_i$, $\partial\Omega$ is the boundary of the computational domain $\Omega$, and $m_{ij}$ denotes the $i$th row and the $j$th column element of the mass matrix calculated as

$$m_{ij}=\int_{\Omega}\rho S_i(\boldsymbol{x})S_j(\boldsymbol{x})dv\approx\sum_{p=1}^{N_p}m_pS_i(\boldsymbol{x}_p(t))S_j(\boldsymbol{x}_p(t)). \tag{4}$$

In the material point method, the inner product $(\theta\boldsymbol{s},\nabla S_i)$ is approximated as

$$(\theta\boldsymbol{s},\nabla S_i)\approx\sum_{p=1}^{N_p}v_p\boldsymbol{s}_p\nabla S_i(\boldsymbol{x}_p), \tag{5}$$

where $v_p$ is the volume associated with the material point $p$, $\boldsymbol{s}_p$ is the stress $\boldsymbol{s}$ evaluated at the location $\boldsymbol{x}_p$ of the material point, $S_i(\boldsymbol{x}_p)$ is the shape function of node $i$ at the material point, and $N_p$ is the total number of material points.

Eq. (3) is a system of coupled linear equations for $d\boldsymbol{u}_j/dt$. In an explicit material point method, to avoid solving this system of coupled equations, we note that $m_{ij}$ is non-zero only for the nodes ($j$'s) that are within the support (non-zero region) of the shape function $S_i$. Since these nodes are in the vicinity of node $i$, $d\boldsymbol{u}_j/dt$ can be approximated by $d\boldsymbol{u}_i/dt$ with a spatial discretization error of $O[(\Delta x)^d]$, where $d=1$ if function $d\boldsymbol{u}(\boldsymbol{x},t)/dt$ is continuous in space or node $i$ is a boundary node; and $d=2$ if $d\boldsymbol{u}(\boldsymbol{x},t)/dt$ is smooth (first order differentiable) in space. Similarly, with the same order error, we can also approximate the body force $\boldsymbol{G}_j$ by $\boldsymbol{G}_i$. With this approximation, the linear equations can be decoupled and be written as

$$m_i\frac{d\boldsymbol{u}_i}{dt}\approx-(\theta\boldsymbol{s},\nabla S_i)+m_i\boldsymbol{G}_i+\int_{\partial\Omega}\theta\boldsymbol{s}\cdot\boldsymbol{n}S_i(\boldsymbol{x})dS, \tag{6}$$

where

$$m_i=\sum_{j=1}^{N}m_{ij}=\int_{\Omega}\rho S_i dv=\sum_{p=1}^{N_p}m_pS_i(\boldsymbol{x}_p). \tag{7}$$

This approximation is equivalent to approximating the matrix elements $m_{ij}$, $(i,j=1,2,\ldots,N)$, by a diagonal matrix in which the diagonal elements are the sum of the elements in the corresponding rows of the original matrix,

$$m_{ij}\approx m_i\delta_{ij}. \tag{8}$$

Therefore this approximation is called the lumped mass matrix approximation and is known to cause artificial energy dissipation of order $(\Delta x)^2$ [9].

With the acceleration $d\boldsymbol{u}_i/dt$ calculated from either (3) or (6) the updated Lagrangian velocity on a node is calculated as

$$\boldsymbol{u}_i^L=\boldsymbol{u}_i^n+\frac{d\boldsymbol{u}_i}{dt}\Delta t, \tag{9}$$

where superscript $L$ indicates a Lagrangian step, superscript $n$ indicates the value at time step $n$ and $\Delta t$ is the time step. The change $\boldsymbol{u}_i^L-\boldsymbol{u}_i^n$ of velocity is interpolated to material points to update the velocity at a material point,

$$\boldsymbol{u}_p^{n+1}=\boldsymbol{u}_p^n+\sum_{i=1}^{N}(\boldsymbol{u}_i^L-\boldsymbol{u}_i^n)S_i\left(\boldsymbol{x}_p^n\right). \tag{10}$$

To prevent numerical diffusion, we interpolate the difference $(\boldsymbol{u}_i^L-\boldsymbol{u}_i^n)$ to the material points, not the value $\boldsymbol{u}_i^L$. The value of the shape function is evaluated at time level $n$, because this is a Lagrangian step. In this step the coordinate system moves and deforms with the material.

The location $\boldsymbol{x}_p$ of material point $p$ is updated using the average of the Lagrangian velocity $\boldsymbol{u}_i^L$ and the velocity $\boldsymbol{u}_i^n$ at the beginning of the time step, interpolated to the material point,

$$\boldsymbol{x}_p^{n+1}=\boldsymbol{x}_p^n+\frac{1}{2}\Delta t\sum_{i=1}^{N}(\boldsymbol{u}_i^L+\boldsymbol{u}_i^n)S_i\left(\boldsymbol{x}_p^n\right). \tag{11}$$

To update the node velocity for time level $n+1$, we use the following relation [2] between quantities on nodes and on material points. For any given time $t$,

$$\sum_{j=1}^{N} m_{ij}(t)\boldsymbol{u}_j(t) \approx \sum_{p=1}^{N_p} m_p \boldsymbol{u}_p(t) S_i\big(\boldsymbol{x}_p(t)\big). \tag{12}$$

Eq. (12) is a set of coupled equations for $\boldsymbol{u}_j$ at the nodes. Again because of the local support of the shape functions, by approximating $\boldsymbol{u}_j$ with $\boldsymbol{u}_i$, we have

$$m_i \boldsymbol{u}_i \approx \sum_{p=1}^{N_p} m_p \boldsymbol{u}_p S_i(\boldsymbol{x}_p). \tag{13}$$

The volume $V_{pi}$ of the material represented by the material points at node $i$ can be calculated as

$$V_{pi} \approx \sum_{p=1}^{N_p} v_p S_i(\boldsymbol{x}_p). \tag{14}$$

The material density $\rho_i^0$ on node $i$ is calculated as the mass $m_i$ divided by the volume $V_{pi}$ of the material at the node,

$$\rho_i^0 \approx \frac{m_i}{\sum_{p=1}^{N_p} v_p S_i(\boldsymbol{x}_p)}. \tag{15}$$

With the velocity at the mesh node updated using (12) or (13) and the material density updated using (15), for single phase systems, one can then use the constitutive relation or equation of state to calculate the stress of the material at material points. Once the stress is calculated, one can repeat the procedures listed above as the next time step.

For multiphase or multimaterial systems, we need to enforce the continuity constraint: the sum of the volume fractions over all phases equals unity. The volume fraction of a material represented by material points is calculated as

$$\theta_i = \frac{V_{pi}}{V_i} \approx \frac{1}{V_i} \sum_{p=1}^{N_p} v_p S_i(\boldsymbol{x}_p), \tag{16}$$

where

$$V_i = \int_\Omega S_i \, dv, \tag{17}$$

is the volume associated with node $i$. The difference between this volume and the node volume in a finite volume method is of order $(\Delta x)^2$ for meshes without sudden change in cell sizes. Therefore for a reasonable mesh the difference can be neglected [2].

To prevent error accumulations, the continuity constraint should not be enforced directly, and an alternative, but equivalent constraint should be used [2]. The combination of the alternative method with the algorithm for small mass nodes introduced in the present paper is described in Section 6. For single phase systems, the material point method described above satisfies the alternative constraint automatically to the second order of the time step.

## 3. Distribution coefficients for small mass nodes

In the material point method, for node $\ell$, if the material points are located at the far sides of the cells with $\ell$ as a node, the shape function $S_\ell(\boldsymbol{x}_p)$ and the masses $m_{\ell j}$ and $m_\ell$ calculated using (4) and (7) are small, while the gradient of the shape function and therefore the force $(\theta \boldsymbol{s}, \nabla S_i)$ in (3) or (6) are not necessarily small. If these equations are used directly to calculate $d\boldsymbol{u}_\ell/dt$, large change in the value of $\boldsymbol{u}_\ell$ can lead to instability in a time explicit calculation. This instability happens frequently in a region with a large gradient of the volume fraction. Typically this region is near a material interface. In such cases, Eqs. (3) and (6) are in fact a statement of a boundary condition on the interface, because this is how the traction continuity condition is derived on a material interface. In the region of large gradient of volume fraction, the error in the material point method is of order $\Delta x$. Within this order of error, we can transfer a part of the force $(\theta \boldsymbol{s}, \nabla S_\ell)$ acting on node $\ell$ that has a small mass to neighbor nodes where the mass is not small. We only need to transfer force $(\theta \boldsymbol{s}, \nabla S_\ell)$ on the right hand sides of (3) and (6), not the other terms, because the other terms are proportional to the node volume fraction, which is proportional to the node masses. For this transfer, we introduce a distribution coefficient $C_{\ell i} (\geqslant 0)$ that transfers the force from node $\ell$ to node $i$ and write the right hand sides of (3) and (6) as

$$\boldsymbol{f}_i = -\sum_{\ell=1}^{N} C_{\ell i}(\theta \boldsymbol{s}, \nabla S_\ell) + \sum_{j=1}^{N} m_{ij} \boldsymbol{G}_j + \int_{\partial\Omega} \theta \boldsymbol{s} \cdot \boldsymbol{n} S_i(\boldsymbol{x}) dS. \tag{18}$$

In this way some of the force acting on node $\ell$ is transferred to node $i$. If the mass on node $i$ is small, the distribution coefficient $C_{\ell i}$ should be small to reduce the acceleration of the node. If node $\ell$ has sufficient mass, the force transfer should not take place, or the distribution coefficient should be set to $\delta_{\ell i}$($\delta_{\ell i} = 1$, if $\ell = i$, and $\delta_{\ell i} = 0$, otherwise).

The distribution coefficient introduced above provides a way to control the numerical values of the accelerations on the small mass nodes. We have so far only discussed general requirements for the distribution coefficients. Their specification is found by studying their effects on conservation laws and time scales in the discretized system.

## 4. Conservation laws

Since the force distribution coefficient introduced above does not affect the way the node mass is calculated, the mass conservation properties are not altered from the original material point method, which can be shown to satisfy the mass conservation law by summing (7) over $i$, and noting that mass of the material points are fixed in the calculation without a phase change.

### 4.1. Conservation of momentum

In a system without the boundary force, replacing the right hand side of (3) with (18), the momentum change at node $i$ from $t^n$ to $t^{n+1}$ can be calculated as

$$\sum_{j=1}^{N} m_{ij}^n \Delta \boldsymbol{u}_j = -\left(\theta \boldsymbol{s}, \sum_{\ell=1}^{N} C_{\ell i} \nabla S_\ell\right) \Delta t + \sum_{j=1}^{N} m_{ij}^n \boldsymbol{G}_j \Delta t, \tag{19}$$

where $\Delta \boldsymbol{u} = \boldsymbol{u}^L - \boldsymbol{u}^n$, and $\Delta t = t^{n+1} - t^n$ is the time step. Multiplying both sides of (10) by $m_p$, and then using (7), we find

$$\sum_{p=1}^{N_p} m_p \boldsymbol{u}_p^{n+1} = \sum_{p=1}^{N_p} m_p \boldsymbol{u}_p^n + \sum_{i=1}^{N} m_i^n \Delta \boldsymbol{u}_i^n. \tag{20}$$

The first term on the right hand side of (20) and the left hand side are the values of the total momentum before and after the time advancement, based on the material points. By summing (12) or (13) over all nodes, we can find that the total momentum calculated based on the material points is exactly the same as the total momentum calculated based on nodes regardless of whether the lumped mass matrix approximation is used.

Summing (19) over all nodes, noting $m_{ij}^n = m_{ji}^n$, we find

$$\sum_{i=1}^{N} m_i^n \Delta \boldsymbol{u}_i^n = -\left(\theta \boldsymbol{s}, \sum_{\ell=1}^{N} \nabla S_\ell \sum_{i=1}^{N} C_{\ell i}\right) \Delta t + \sum_{i=1}^{N} \sum_{j=1}^{N} m_{ij}^n \boldsymbol{G}_j \Delta t. \tag{21}$$

Substituting (21) into (20), noting $m_{ij}^n = m_{ji}^n$ and using (7), we find

$$\sum_{p=1}^{N_p} m_p \boldsymbol{u}_p^{n+1} = \sum_{p=1}^{N_p} m_p \boldsymbol{u}_p^n + \sum_{j=1}^{N} m_j^n \boldsymbol{G}_j \Delta t - \left(\theta \boldsymbol{s}, \sum_{\ell=1}^{N} \nabla S_\ell \sum_{i=1}^{N} C_{\ell i}\right) \Delta t. \tag{22}$$

According to the definition of $\boldsymbol{G}$ in (2), $\sum_{j=1}^{N} m_j^n \boldsymbol{G}_j$ is the force external to the material acting on node $j$. The second term on the right hand side of (22) is the total impulse from the force external to the material. If the sum of the force distribution coefficients over the second index $i$, $\sum_{i=1}^{N} C_{\ell i}$, is a constant independent of node $\ell$, the last term in (22) vanishes, because the sum of the shape function gradients is zero. The change in the total momentum is then caused by external force only. As mentioned in the last section, if node $\ell$ has sufficient mass, $C_{\ell i} = \delta_{\ell i}$, and $\sum_{i=1}^{N} C_{\ell i} = 1$. In order for $\sum_{i=1}^{N} C_{\ell i}$ to be a constant independent of $\ell$, we must have

$$\sum_{i=1}^{N} C_{\ell i} = 1 \tag{23}$$

for all nodes, including the nodes with small masses. According to (22), if (23) is satisfied, the conservation of momentum is preserved with the force distribution coefficients.

### 4.2. Energy conservation

The total kinetic energy based on material points at time step $n$ can be calculated as

$$K_p^n = \frac{1}{2} \sum_{p=1}^{N_p} m_p \left(\mathbf{u}_p^n\right)^2. \tag{24}$$

The difference of the kinetic energy during a time advancement is then

$$K_p^{n+1}-K_p^n=\frac{1}{2}\sum_{p=1}^{N_p}m_p\left[(\mathbf{u}_p^{n+1})^2-(\mathbf{u}_p^n)^2\right]=\frac{1}{2}\sum_{p=1}^{N_p}m_p\left(\mathbf{u}_p^{n+1}+\mathbf{u}_p^n\right)\cdot\left(\mathbf{u}_p^{n+1}-\mathbf{u}_p^n\right)=\sum_{p=1}^{N_p}m_p\left(\mathbf{u}_p^n+\frac{1}{2}\Delta\mathbf{u}_p^n\right)\cdot\Delta\mathbf{u}_p^n$$
$$=\sum_{j=1}^{N}\sum_{p=1}^{N_p}m_p\mathbf{u}_p^nS_j(\boldsymbol{x}_p^n)\Delta\mathbf{u}_j^n+\frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N}m_{ij}^n\Delta\mathbf{u}_i^n\cdot\Delta\mathbf{u}_j^n, \tag{25}$$

where $\Delta\mathbf{u}_p^n=\mathbf{u}_p^{n+1}-\mathbf{u}_p^n$ and $\Delta\mathbf{u}_i^n=\mathbf{u}_i^L-\mathbf{u}_i^n$. The last step of (25) is a result of (10) and (4). Using (12) and noting $m_{ij}=m_{ji}$, we have

$$K_p^{n+1}-K_p^n=\sum_{j=1}^{N}\sum_{i=1}^{N}m_{ji}^n\left(\mathbf{u}_i^n+\frac{1}{2}\Delta\mathbf{u}_i^n\right)\cdot\Delta\mathbf{u}_j^n=\sum_{i=1}^{N}\frac{\mathbf{u}_i^L+\mathbf{u}_i^n}{2}\cdot\sum_{j=1}^{N}m_{ij}^n\Delta\mathbf{u}_j^n. \tag{26}$$

Energy transport in a multiphase system is affected by phase interactions not only through exchange forces but also through related stresses. It is beyond the scope of this paper to study energy transfer between phases. The numerical properties of energy conservation with the force distribution coefficients can be understood by considering an elastic body in a vacuum subject to a potential body force $\boldsymbol{b}$, such as gravity. In this case, the auxiliary pressure can be set to zero, resulting in $\boldsymbol{s}=\boldsymbol{\sigma}$ and $\boldsymbol{G}=\boldsymbol{b}$. Using (4), (5) and (19), after exchanging the orders of summations, we have

$$K_p^{n+1}-K_p^n=-\sum_{p=1}^{N_p}v_p\boldsymbol{\sigma}_p:\nabla\hat{\boldsymbol{u}}^{n+1/2}(\boldsymbol{x}_p^n)\Delta t+\sum_{p=1}^{N_p}m_p\sum_{j=1}^{N}\boldsymbol{b}_jS_j(\boldsymbol{x}_p)\cdot\left[\sum_{i=1}^{N}\frac{\mathbf{u}_i^L+\mathbf{u}_i^n}{2}S_i(\boldsymbol{x}_p)\right]\Delta t, \tag{27}$$

where $\nabla\hat{\boldsymbol{u}}^{n+1/2}$ is the half time velocity gradient calculated as

$$\nabla\hat{\boldsymbol{u}}^{n+1/2}(\boldsymbol{x}_p^n)=\sum_{\ell=1}^{N}\frac{\hat{\mathbf{u}}_\ell^L+\hat{\mathbf{u}}_\ell^n}{2}\nabla S_\ell(\boldsymbol{x}_p^n), \tag{28}$$

with the velocity $\hat{\boldsymbol{u}}_\ell$ defined by

$$\hat{\mathbf{u}}_\ell^L=\sum_{i=1}^{N}C_{\ell i}\mathbf{u}_i^L,\quad \hat{\mathbf{u}}_\ell^n=\sum_{i=1}^{N}C_{\ell i}\mathbf{u}_i^n. \tag{29}$$

For an elastic material, we have $\boldsymbol{\sigma}=\rho^0\partial u/\partial\boldsymbol{\varepsilon}$, where $u$ is the specific (per unit mass) elastic potential energy, $\rho^0$ is the microscopic density, and $\boldsymbol{\varepsilon}$ is the strain of the material. The total elastic potential energy $U_p$ in the system can be approximately calculated using material points as

$$U_p=\sum_{p=1}^{N_p}m_pu_p. \tag{30}$$

Differentiating (30) with respect to time and noting $\boldsymbol{\sigma}=\rho^0\partial u/\partial\boldsymbol{\varepsilon}$ for an elastic material and $v_p=m_p/\rho_p^0$, we find

$$\left(U_p^{n+1}-U_p^n\right)=\sum_{p=1}^{N_p}v_p\boldsymbol{\sigma}_p:\dot{\boldsymbol{\varepsilon}}_p\Delta t+O[(\Delta t)^2]. \tag{31}$$

Noting that the stress tensor $\boldsymbol{\sigma}_p$ is symmetric, comparing (27) with (31) we find that, if the strain rate $\dot{\boldsymbol{\varepsilon}}_p$ is calculated as the symmetric part of $\nabla\hat{\boldsymbol{u}}^{n+1/2}(\boldsymbol{x}_p)$ using (28), we have

$$K_p^{n+1}+U_p^{n+1}=K_p^n+U_p^n+\sum_{p=1}^{N_p}m_p\boldsymbol{b}_p\cdot\mathbf{u}_p^{n+1/2}\Delta t+O[(\Delta t)^2], \tag{32}$$

where

$$\boldsymbol{b}_p=\sum_{j=1}^{N}\boldsymbol{b}_jS_j(\boldsymbol{x}_p), \tag{33}$$

is the specific (per unit mass) body force at the location $\boldsymbol{x}_p$ of material point $p$, and

$$\mathbf{u}_p^{n+1/2}=\sum_{\ell=1}^{N}\frac{\mathbf{u}_\ell^L+\mathbf{u}_\ell^n}{2}S_\ell(\boldsymbol{x}_p), \tag{34}$$

is the half time velocity at the location $\boldsymbol{x}_p$ of the material point. The third term on the right hand side of (32) represents the work done by the body force.

If the body force $m_p\boldsymbol{b}_p$ is the gradient of a potential $\phi$, that is $m_p\,\boldsymbol{b}_p=-\nabla\phi(\boldsymbol{x}_p)$, then the change in the potential for each material point can be calculated as $\phi(\boldsymbol{x}_p^{n+1})-\phi(\boldsymbol{x}_p^n)=-m_p\boldsymbol{b}_p(\boldsymbol{x}_p^{n+1}-\boldsymbol{x}_p^n)$. Therefore when the displacement of the material point is calculated as $\boldsymbol{x}_p^{n+1}-\boldsymbol{x}_p^n=\mathbf{u}_p^{n+1/2}\Delta t$, relation (32) implies that conservation of the total energy, including this force

potential, is within an error of order $(\Delta t)^2$. In other words, to ensure the error in total energy conservation is within the order of the time step squared, the strain rate or the velocity gradient used for stress calculation is required to be calculated using the node velocity weighted by the force distribution coefficients as in (28) and (29), and displacements of the material points are required to be calculated using the unweighted velocity.

We note that the second order error proportional to the time step $\Delta t$ comes from (31). If the volume $v_p$, the stress $\boldsymbol{\sigma}_p$ and the strain rate are evaluated at the half time between $t^n$ and $t^{n+1}$, the error can be reduced to $O[(\Delta t)^3]$. This will require an implicit material point method. In an explicit material point method, we can update the stress before or after the Lagrangian velocity is calculated. If the stress is updated before the Lagrangian velocity is calculated, then the strain rate has to be calculated using the velocity $\hat{\boldsymbol{u}}^n$. This is equivalent to approximating $\hat{\boldsymbol{u}}^L$ with $\hat{\boldsymbol{u}}^n$ in (28). For convenience in computing material interactions as discussed in Section 6, we choose to update stress after the Lagrangian velocity is calculated. This way of updating stress has been shown to have better stability properties [10,11]. In this way, the calculated stress is used in the next time step. To prevent further time lag in stress calculation, the strain rate used to calculate the stress is computed from Lagrangian velocity $\hat{\boldsymbol{u}}^L$ alone. This is equivalent to replacing $\hat{\boldsymbol{u}}^n$ with $\hat{\boldsymbol{u}}^L$ in (28). This way of calculating stress is different from an implicit method, because such calculated stress is used at the next time step, instead of iterating it in the current time step as in an implicit method.

### 4.3. Energy dissipation due to lump sum of the mass matrix

In most implementations of the material point method, the lumped mass matrix approximation is used to avoid solving coupled equations. We show that the use of distribution coefficients with this approximation does not cause energy dissipation in addition to that which is originally associated with the approximation.

We note that $\Delta \boldsymbol{u}_i^n \cdot \Delta \boldsymbol{u}_j^n$ in (25) can be written as

$$\Delta \boldsymbol{u}_i^n \cdot \Delta \boldsymbol{u}_j^n = \frac{1}{2}\left[(\Delta \boldsymbol{u}_i^n)^2 + (\Delta \boldsymbol{u}_j^n)^2\right] - \frac{1}{2}\left[\Delta \boldsymbol{u}_i^n - \Delta \boldsymbol{u}_j^n\right]^2. \tag{35}$$

Substituting this relation into (25), using $m_{ij} = m_{ji}$, and (7) and (13) instead of (12) for the lumped mass matrix approximation, we find

$$K_p^{n+1} - K_p^n = \sum_{i=1}^{N}\left(\mathbf{u}_i^n + \frac{1}{2}\Delta \boldsymbol{u}_i^n\right) m_i \Delta \boldsymbol{u}_i^n - D, \tag{36}$$

where

$$D = \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N} m_{ij}^n \left[\Delta \boldsymbol{u}_i^n - \Delta \boldsymbol{u}_j^n\right]^2 = O[(\Delta x)^{2d}(\Delta t)^2] \geqslant 0, \tag{37}$$

because the velocity increments, $\propto \Delta t$, between neighboring nodes differ by $O[(\Delta x)^d]$, and $m_{ij} \geqslant 0$.

Using (19) with $m_{ij} = m_i \delta_{ij}$, and (28) and (31), we find

$$K_p^{n+1} + U_p^{n+1} = K_p^n + U_p^n + \sum_{p=1}^{N_p} m_p \boldsymbol{b}_p \cdot \boldsymbol{u}_p^{n+1/2} \Delta t - D + O[(\Delta t)^2]. \tag{38}$$

Comparing this relation to (32) there is an extra term $D$ resulting only from the approximation of lumping the mass matrix. This term causes a numerical energy dissipation of order $(\Delta x)^{2d}(\Delta t)^2$. This error is smaller than the error of $O[(\Delta t)^2]$ caused by time discretization for sufficiently fine meshes.

## 5. Time scales affected by distribution coefficients

In the previous section we show that to ensure momentum conservation, the distribution coefficients have to satisfy (23). To ensure energy conservation the velocities used to compute the strain rate and then the stress need to be weighted by the distribution coefficient, while the velocities used to advance positions of the material points should not be weighted. In this section, we use a simple example to study the effects of these coefficients on the time scales in the system, and then to find a way to specify the coefficients.

We consider an one-dimensional elastic problem with only one computational cell and one material point. The computational cell has two nodes. Node 0 is located at $x = 0$ and node 1 is located at $\Delta x$. In this cell the material point, with mass $m_p$, is located at $x_p$. This problem can be thought of as a situation encountered in the motion of an elastic bar when the material point at the end of the bar enters a new cell. Suppose that elastic bar loses most of its translation momentum and oscillates in this position with a small amplitude. As in most implementations of the material point method, we use a lumped mass matrix in this problem. We further assume the velocity at node 0 is zero.

Let $E$ be the Young's modulus of the elastic bar, and $\lambda_1$ be the displacement of the material on node 1 relative to that on node 0. In this one-dimensional example, the "volume" of the material point $v_p = \Delta x$. The mass of the material point is $m_p = \rho^0 \Delta x$. The shape function of node 1 is $S(x) = x/\Delta x$, and the mass on node 1 is $m_1 = m_p S(x_p) = \rho^0 x_p$. With the original mate-

rial point method, the strain is calculated as $\varepsilon = \lambda_1/\Delta x$, and the stress at the material point is $\sigma_p = E\lambda_1/\Delta x$. Without external forces, using (19) with the lumped mass matrix approximation, we have $C_{10} = 0$ and $C_{11} = 1$ for the original material point method. The momentum equation for node 1 becomes

$$m_1 \frac{d^2\lambda_1}{dt^2} = -\frac{E}{\Delta x}\lambda_1. \tag{39}$$

In this equation the mass $m_1$ is a function of the material point position $x_p$, which is a function of time. For small amplitude motions, we can neglect such time dependency of $m_1$ and solve the equation. The solution has a time period of $2\pi\sqrt{m_1\Delta x/E} = 2\pi\sqrt{\rho^0 x_p \Delta x/E}$. If $x_p$ is very small, the time period is also small. A small time step is needed to perform the numerical calculation to prevent a numerical instability. Often the time step is impractically small, and the calculation fails to progress.

With the algorithm for small mass nodes introduced in the present paper, when $m_1$ is smaller than a mass $m_\epsilon$, the strain rate is calculated as $C_{11}d\lambda_1/dt$. Neglecting the small position change of the material point, the strain can be calculated as $C_{11}\lambda_1/\Delta x$, and then stress is $\sigma_p = C_{11}E\lambda_1/\Delta x$. Noting $C_{01} = 0$ in this case, the momentum equation for node 1 becomes

$$m_1 \frac{d^2\lambda_1}{dt^2} = -C_{11}^2 \frac{E}{\Delta x}\lambda_1, \tag{40}$$

and the time period of the solution becomes $2\pi\sqrt{m_1\Delta x/(C_{11}^2 E)}$. If we choose $C_{11} = \sqrt{m_1/m_p}$, then the time period becomes $2\pi\sqrt{m_p\Delta x/E} = 2\pi\Delta x/c$, where $c$ is the sound speed. For this time period, the corresponding wave length is $2\pi\Delta x$. This is in agreement with the minimum wave length that can be solved reliably using mesh size of $\Delta x$ in a mesh based calculation. The time step required to solve this case is then independent of the position of the material point. This simple example implies that to preserve reasonable time steps in solving a momentum equation, the force distribution coefficient $C_{\ell\ell}$ should be proportional to $\sqrt{m_\ell}$.

For this reason, in solving the momentum equation, we distribute the force on a small mass node to surrounding nodes proportionally to the square root of the mass at a recipient node. That is

$$C_{\ell i} = \frac{\sqrt{m_i}}{\sum_{(j,\ell)}\sqrt{m_j}}, \tag{41}$$

where $\sum_{(j,\ell)}$ denotes the summation over node $\ell$ and its neighbors. These coefficients satisfy (23).

A similar approach can be used to prevent numerical instabilities caused by small thermal capacities at nodes. The corresponding distribution coefficients for thermal flux can also be derived in a similar manner.

## 6. Coupling material interactions

Since the material point method is computationally more expensive than a typical arbitrary Lagrangian Eulerian method, it is often preferred to use the ALE method for fluid phases, and only use MPM for solid phases, or for phases with history dependent constitutive relations. A method enabling such use of coupled ALE and MPM calculation is described in [2]. In order to correctly calculate material interactions, the algorithm described above needs to be incorporated with schemes for handling time scales from exchange forces between phases, and a method for satisfying the continuity requirement. The mathematical foundation of the method is described in our previous paper [2]. In that paper we also outlined the steps of the numerical implementation for the method for satisfying the continuity requirement. To couple the method with the present algorithm, we need to make several subtle but important modifications to the implementation. We now describe our new implementation.

In each time step, advancement of the velocities for all the phases uses a prediction and correction method, and is divided into four major steps. We now describe each of them in the following subsections.

### 6.1. Stress acceleration

In the first step, we only consider the effects of $\nabla\cdot[\theta_k(\boldsymbol{\sigma}_k + P\boldsymbol{I})]$ and the body force term on the right hand side of (1). In this step, for a phase calculated using the ALE method, the Lagrangian velocity is first updated in the typical ALE fashion [12]. For phases calculated using the material point method, we use a lumped mass matrix. The Lagrangian velocity is first calculated using (19), with $m_{ij} = m_i\delta_{ij}$ and $\boldsymbol{G}_j = \boldsymbol{b}_{kj}$, the specific body force for the phase $k$ material at node $j$.

The velocities calculated in the first step do not account for the pressure gradient. More importantly, the velocities so obtained do not usually satisfy the continuity constraint, which is often expressed as requiring that the sum of the updated volume fractions equals unity in the multiphase system. Therefore these velocities need to be corrected.

### 6.2. Continuity requirement and pressure acceleration

The volume fraction $\theta_k$ of phase $k$ is calculated as $\theta_k = \rho_k/\rho_k^0$, where $\rho_k$ is the macroscopic density, and $\rho_k^0$ is the material density. The macroscopic density satisfies the mass conservation equation [3]

$$\frac{\partial \rho_k}{\partial t} + \nabla \cdot (\boldsymbol{u}_k \rho_k) = \rho_{kc}^0 \dot{\phi}_k, \tag{42}$$

where $\dot{\phi}_k$ represents rate of volume gain due to phase changes and $\rho_{kc}^0$ is the material density of the material gained or lost due to phase change. The material density satisfies the evolution equation [3]

$$\theta_k \left( \frac{\partial \rho_k^0}{\partial t} + \boldsymbol{u}_k \cdot \nabla \rho_k^0 \right) = -\theta_k \rho_k^0 \langle \nabla \cdot \boldsymbol{u}_k \rangle + (\rho_{kc}^0 - \rho_k^0) \dot{\phi}_k, \tag{43}$$

with

$$\langle \nabla \cdot \boldsymbol{u}_k \rangle = \alpha_k \nabla \cdot \boldsymbol{u}_k + B_k, \tag{44}$$

where $0 \leqslant \alpha_k \leqslant 1$ is related to the morphology of the phase [3]. In (44), the first term, $\alpha_k \nabla \cdot \boldsymbol{u}_k$, is directly related to the macroscopic motions of the material; the second term $B_k$ represents interaction of materials. For modeling material interactions [3], we require that the rate of pressure change caused by $B_k$ is the same in all phases.

The first term in (44) can be easily calculated. The following procedures are taken to calculate $B_k$. We first advance the macroscopic density $\rho_k$ at a node using (42) for all phases through advection operations. For a phase represented by material points, such calculated macroscopic density is used to compute the material strain, strain rate and the pressure, or the negative of the isotropic component of the stress tensor; therefore, as discussed in Section 4, the velocity field used in this advection should be the velocity weighted by the force distribution coefficients as defined in (29) to ensure total energy conservation within an error of order $(\Delta t)^2$. Such weighted node velocities are first used to calculate the face velocities on a control volume, and then to calculate the divergence of the mass flux $\rho_k \boldsymbol{u}_k$, which is used to update the macroscopic density $\rho_k$ on a node. For a phase represented by material points, this velocity and the macroscopic density are only used in this step; they are eventually overwritten at the end of the time step as described in Section 6.4. Since distribution coefficients are only used for phases represented by material points, for a phase calculated using the ALE method, distribution coefficients and hence the weighted velocity are not defined. For these phases the usual velocity field is used in the advection.

To update the microscopic density $\rho_k^0$ on a mesh node, we first calculate an interim material density $\rho_k^{0*}$ using (43). In this calculation, $\langle \nabla \cdot \boldsymbol{u}_k \rangle$ in (43) is calculated without $B_k$ in (44). The term $\boldsymbol{u}_k \cdot \nabla \rho_k^0$ is calculated as $\nabla \cdot (\rho_k^0 \boldsymbol{u}_k) - \rho_k^0 \nabla \cdot \boldsymbol{u}_k$ with the divergences calculated in the typical manner of a finite volume method, namely, as the sum of the fluxes out of the surfaces of the control volume divided by the volume of the control volume. We also calculate a pressure $P_k^*$, defined as the negative of the isotropic component of the stress tensor, corresponding to the interim density $\rho_k^{0*}$ through the equation of state or the constitutive relations of the material. In these calculations, again, the velocity is the distribution coefficient weighted velocity for the phases represented by material points, and is the usual velocity for phases using the ALE method.

With the calculated macroscopic density and the interim material density, an interim volume fraction $\theta_k^* = \rho_k / \rho_k^{0*}$ for phase $k$ can be calculated. Such calculated volume fractions do not, in general, satisfy the continuity condition. To enforce the continuity condition, we then change pressures in all the phases by the same amount $\Delta P$. This pressure change leads to changes in the material densities and the volume fractions of all the phases. It can be shown [3] that this density change $\rho_k^{0L} - \rho_k^{0*} = -\rho_k^{0*} B_k \Delta t$, as called for in (44). As described in [2], because of unavoidable numerical errors related to the volume fraction calculation in the material point method, continuity condition $\sum_{k=1}^M \theta_k = 1$ should not be enforced directly to avoid undesired accumulation of numerical errors. To correctly enforce the continuity condition in the material point method the following equation is used to solve for the common pressure change $\Delta P$.

$$\sum_{k=1}^{M} \left[ \frac{\rho_k}{\rho_k^0 (p_k^* + \Delta P)} \right] - \sum_{k=1}^{M} \theta_k^n + \boldsymbol{u}_m \cdot \nabla \sum_{k=1}^{M} \theta_k \Delta t = 0, \tag{45}$$

where $M$ is the total number of phases or materials in the system, superscript $n$ denotes the value at the beginning of the time step, and $\boldsymbol{u}_m$ is the mixture velocity. In the sense of a weak solution, this equation is equivalent [2] to $\sum_{k=1}^M \theta_k = 1$. In solving (45), because of numerical errors, the condition $\sum_{k=1}^M \theta_k = 1$ should not be assumed, and the mixture velocity $\boldsymbol{u}_m$ should be calculated as

$$\boldsymbol{u}_m = \sum_{k=1}^{M} \theta_k \boldsymbol{u}_k \Big/ \sum_{k=1}^{M} \theta_k, \tag{46}$$

to prevent undesired accumulation of numerical error [2].

For a time explicit scheme, in solving for $\Delta P$ from (45), the macroscopic density $\rho_k$ is not a function of $\Delta P$, and the equation can be solved in pointwise manner without the need for information from the neighbor nodes. For a time implicit scheme, the macroscopic density $\rho_k$ is related to the pressure gradient, therefore information from neighbor nodes is needed.

In this way of enforcing the continuity constraint, we have calculated the updated pressure $P_k = p_k^* + \Delta P$ and the Lagrangian material density $\rho_k^{0L} = \rho_k^0 (p_k^* + \Delta P)$, for all the phases, on mesh nodes. Since the auxiliary pressure $P$ in (1) is the pressure of one of the phases in the system, the pressure gradient $\nabla P$ in (1) can be calculated. By adding $-\nabla P \Delta t / \rho_k^0$ to the Lagrangian velocity obtained in Section 6.1, we account for the effect of this pressure gradient. This updated Lagrangian velocity, denoted as $\boldsymbol{u}_k^{(2)}$, satisfies

$$\theta_k\rho_k^0\frac{\boldsymbol{u}_k^{(2)}-\boldsymbol{u}_k^n}{\Delta t}=\nabla\cdot\left[\theta_k\left(\boldsymbol{\sigma}_k^n+P^n\boldsymbol{I}\right)\right]+\theta_k\rho_k^0\boldsymbol{b}_k-\theta_k\nabla P, \tag{47}$$

where superscript $n$ denotes the value of the last time step, which is also the value at the beginning of this time step. The effects of the first two terms on the right hand side are accounted for in Section 6.1. Pressures $P^n$ and $P$ are different by $O(\Delta t)$, which can be neglected. The algorithm described here takes the advantage of the form of the momentum equation (1). As mentioned in Section 2, if the pressure effect were written as $P\nabla\theta_k$, we would have caused an acceleration $(P-P^n)\nabla\theta_k/(\theta_k\rho_k^0)$ due to the gradient in the volume fraction, even in an equilibrium system, in which all the deviatoric stresses are zero, and the pressures for all phases are the same and uniform in space. Although the acceleration is small, $O(\Delta t)$, it is unphysical. In the form of (1) and the implementation described in the present paper, the stress $\boldsymbol{\sigma}_k$ and the pressure are calculated at the same time. When the system is at equilibrium, all the stresses equal $-P^n\boldsymbol{I}$, and the first term on the right hand sides of (1) and (47) vanishes exactly. Given $\nabla P=\boldsymbol{0}$, without the body force, the velocity change $\boldsymbol{u}_k^{(2)}-\boldsymbol{u}_k^n$ is exactly zero.

### 6.3. Accounting for phase interaction force

Comparing (47) with (1), the phase interaction force $\boldsymbol{f}_k$ is missing from (47). To add the effect of this force, we approximate $d\boldsymbol{u}_k/dt$ in (1) by $(\boldsymbol{u}_k^L-\boldsymbol{u}_k^n)/\Delta t$, and subtract (47) from the resulting equation, to find

$$\rho_k^0\left(\boldsymbol{u}_k^L-\boldsymbol{u}_k^{(2)}\right)=\boldsymbol{f}_k(\boldsymbol{u}^L,\boldsymbol{u}^n)\Delta t. \tag{48}$$

The phase interaction force $\boldsymbol{f}_k$ is a function of the Lagrangian velocities of all phases in the system, denoted by $\boldsymbol{u}^L$, and the velocities $\boldsymbol{u}^n$ at the beginning of the time step. This equation is a coupled system among all phases in the calculation, but it is a pointwise equation, and does not involve quantities on neighbor nodes. The final Lagrangian velocity $\boldsymbol{u}_k^L$ is solved from (48).

### 6.4. Time step completion

After these three steps, we have the Lagrangian velocities on the nodes. These velocities are used in (10) and (11) to advance the velocities and positions of the material points. To advance the stress on material points, we calculate the velocity gradient at the location of the material point using (28). The velocity used in this calculation is obtained from the force distribution coefficient weighted Lagrangian velocities around the nodes according to (29). Since such calculated stress is used in the next time step, to prevent time lag in the stress, as explained in Section 4, we replace $\hat{\boldsymbol{u}}_k^n$ by $\hat{\boldsymbol{u}}_k^L$ in (28). Before this velocity gradient is used to update the stress tensor, it is modified by subtracting $\ln(\rho_k^{0L}/\rho_k^{0*})/\Delta t\boldsymbol{I}=-B_k\boldsymbol{I}+O(\Delta t)$ from it to account for the effect of $B_k$ in (44).

For the phases represented by the material points, the time-advanced quantities, such as the velocity $\boldsymbol{u}_k^{n+1}$, the material density $\rho_k^{0\,n+1}$, and the macroscopic density $\rho_k^{n+1}=\theta_k^{n+1}\rho_k^{0\,n+1}$, stored on the mesh nodes are overwritten using the quantities at material points through relations (13), (15) and (16) with shape functions calculated at the time-advanced positions for the material points. For the phases calculated using the ALE method, advection is then performed to remap the Lagrangian quantities to nodes [12].

## 7. Numerical examples

The numerical procedures described in Section 6 have been implemented in the code CartaBlanca. In this section we present five examples obtained with the code to illustrate the usefulness of the distribution coefficient algorithm for small mass nodes.

The first example considers one-dimensional translation of four material points in air. The material points represent a porous elastic body with material density $\rho^0=2.0$ g/cm$^3$, Young's modulus $E=10$ GPa and volume fraction 0.5. The computational domain consists of ten cells with $\Delta x=0.1$ cm in an interval $0\leqslant x\leqslant 1$. Four material points are initially placed in the 5th and 6th cells at positions $x_p=0.425, 0.475, 0.525, 0.575$. In these cells air fills the volume left by the solid to ensure the sum of the volume fractions of the solid and air is unity. The initial velocity of both materials is 10 m/s. Initial pressures of the solid material and the air are set to be one atmosphere. There is no initial deviatoric stress on the solid. On the left boundary of the computational domain, we specify densities of both materials corresponding to their values at one atmosphere. On this boundary a constant inflow velocity of 10 m/s is specified for both materials. On the right boundary we specify a constant pressure of one atmosphere. The sound speed in the solid is $c=\sqrt{E/\rho^0}=2.24$ km/s. We turn on the algorithm for small mass nodes when the node mass is less then 4% of the largest mass of neighbor nodes, corresponding to an $80\%(=1-\sqrt{4\%})$ reduction of time step based on the largest mass node. If we allow the Courant number to be one, the maximum allowed time step is $\Delta t=0.2\Delta x/c=8.93\times10^{-8}$ s. In this calculation, $\Delta t=8.92\times10^{-8}$ s is used. As expected, the results show pressure of the air, the stress in the solid and the velocities of both phases remain constant throughout the calculation of 5000 time steps. At the end of the calculation three material points translated to positions $x_p=0.871$, 0.921, 0.971, and the fourth material point, originally at $x_p=0.575$, left the computational domain. To ensure the calculation

is indeed stable within the range of time steps allowed by the Courant limitation, we decreased the time step by a factor of ten to $\Delta t = 8.92 \times 10^{-9}$ s and increased the number of time steps to 50000. The results are the same. In both cases, no sign of instability is observed. In contrast to these results, we also performed a calculation without the distribution coefficient algorithm. The calculation lost its stability when $\Delta t = 1.5 \times 10^{-8}$ s was used.

The second example calculates vibration of a one-dimensional bar in air. In this example both the solid material and air have the same properties as described in the first example. The region containing the solid bar contains 1% air. The solid bar is fixed at the left end, and the right end is free. The computational domain ranges from $x = 0$ to $x = 1$ cm, and is divided into 80 cells with $\Delta x = 0.0125$ cm. There are 81 material points representing the solid material. The material points are located at positions $x_p = (3/500 + n_p/2)\Delta x$, $n_p = 0,\ldots,80$. In this distribution of material points, the last material point is located at $x_p = 0.500075$ cm, slightly beyond node 41 at 0.5 cm as shown in Fig. 1. The mass at node 42 is $7.43 \times 10^{-5}$ g, a small number compared to the mass of node 41 (0.0186 g). As an initial condition, linear initial velocity of the solid material is set as $u_x = 100x$ cm/s. The subsequent velocity is governed by the wave equation.

Fig. 2 shows the displacement of the last material point (originally at $x_p = 0.500075$ cm) calculated with and without the distribution coefficient algorithm for small mass nodes. The solid line is the analytical solution from the wave equation. Hollow squares are results calculated without the distribution coefficient algorithm using time step $\Delta t = 3.0 \times 10^{-9}$ s. The line with hollow circles shows the results without the algorithm calculated using a slightly larger time step, $\Delta t = 3.2 \times 10^{-9}$ s. For this time step size, the calculation loses its stability near the trough of the wave as shown in the figure. At the trough, the last material point moves closer to node 41; the mass at node 42 is further reduced. According to our analysis in Section 5, the required time step is also reduced. Although in this explicit calculation, the time step is limited to be less than Courant number 0.75 based on both the elastic wave speed and the material speed, the calculation still loses its stability in this case. With the distribution coefficient algorithm for small mass nodes described in the present paper, we are able to calculate this vibration problem with time step $\Delta t = 3.0 \times 10^{-8}$ s, with corresponding Courant number 0.5, about ten times the time step required without the algorithm. The difference between results obtained with this algorithm and the analytical solution is almost invisible in the figure. The fact that the amplitude of the oscillation does not decay suggests conservation of energy in the calculation.

This example was set up to show the necessity of the algorithm for small mass nodes. If the sole purpose is to solve the physical problem described here, the initial material points can be better distributed to avoid the numerical instability. However, often in a numerical calculation, especially for cases of large deformation, the locations of the material points

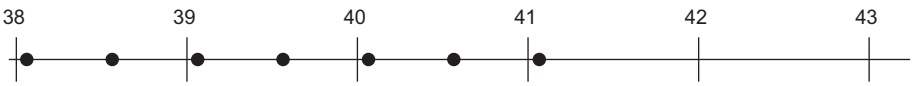


**Fig. 1.** A schematic illustration of the mesh and material points near the material interface in the one-dimensional vibration example.

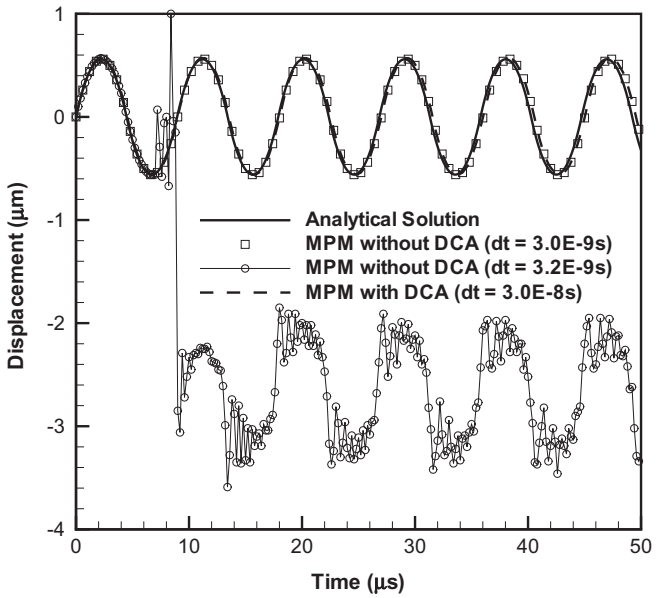


**Fig. 2.** Comparison of displacements of a vibrating one-dimensional elastic bar calculated with and without the distribution coefficient algorithm (DCA) for small mass nodes.

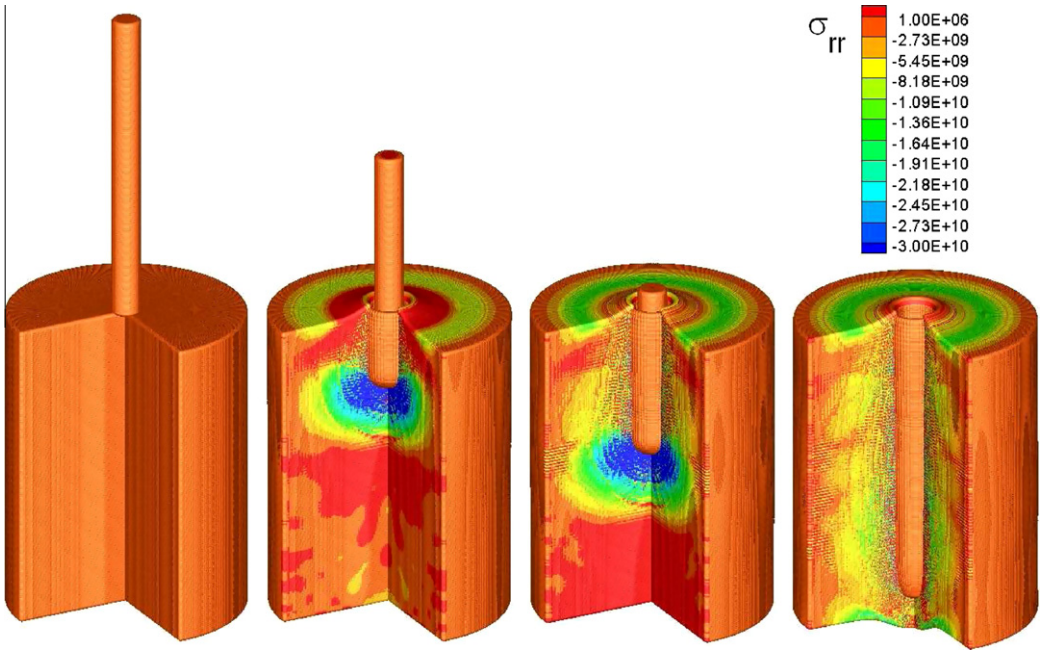


**Fig. 3.** An illustration of projectile-target interaction calculated with the material point method. Material points are colored by hoop stress.

are part of the solution. There is no way to ensure they all will remain in positions that do not cause instability. Without an algorithm for small mass nodes, numerical instabilities will occur sooner or later.

Indeed, this is what happens in the following third example. Here, we calculate interaction of a tungsten projectile and a cylinder of armor steel, as illustrated in Fig. 3. We have studied this interaction previously [2] to demonstrate the algorithms for enforcing the continuity constraint in multimaterial interactions. The calculation was done with small time step sizes. This is one of the few relevant cases for which there are available experimental data. In the present paper we calculate it again to compare the results obtained with and without the algorithm for small mass nodes.

This calculation involves three phases, air, tungsten and armor steel. Of course air has little effect in this process. We choose to include it to demonstrate the capability of handling large material density ratios using the material point method coupled with the arbitrary Lagrangian Eulerian method. In this example the tungsten rod is 5 cm in length and is shot into an armor steel block of 4.95 cm thickness. The initial speed of the tungsten rod is 1.7 km/s. Fig. 4 shows the results calculated with and without the algorithm for small mass nodes. The results are almost indistinguishable for the tail positions. For the nose positions, the differences are slightly more visible. This is because the interactions on the interface between the projectile material and the target material on the nose are more important than the interface interactions between the projectile material and air on the tail. Although there is little difference in the results, the computation time is very different. With the algorithm for small mass nodes the entire calculation can be completed (to 70 μs) within 25 min on a Dell Inspiron laptop with a 2.16 GHz Intel chip. Without the algorithm for small mass nodes to achieve numerical stability, we have to limit the time step at $\Delta t = 10^{-9}$ s, and took 6 h and 40 min to run to physical time of about 66 μs. The calculation could not reach the intended 70 μs because the time steps were reduced to $10^{-20}$ s. This behavior of the numerical solution is in agreement with our analysis above: the algorithm for small mass nodes does not increase the accuracy of the numerical solution but greatly enhances stability of the calculation. Therefore larger time steps can be used to reduce total computation time.

As mentioned in Section 3, the distribution coefficient algorithm introduced in the current paper makes approximations of $O(\Delta x)$ on accelerations at nodes near nodes with small masses. These nodes are often near the material boundaries or interfaces. To ensure the algorithm produces accurate results in cases with large ratio of boundary nodes to internal nodes, we now consider the vibration of a thin elastic spherical shell with inner radius 3.9 cm and outer radius 4.0 cm. In this example, the material density $\rho$ is 2.7 g/cm$^3$. The Young's modulus of the material is $E = 70.38$ GPa, and the Poisson's ratio is $\nu = 0.275$. In this calculation we use cylindrical coordinates. The computational domain is a larger spherical shell containing the elastic shell, with 40 cells in the radial direction and 40 cells in the zenith direction. Initially, the elastic material only occupies four middle layers of the cells along the radial direction as shown in Fig. 5. Other cells are initially empty. In this way we have plenty of room for possible large material deformation. The elastic shell is set into vibration by imposing an initial outward velocity of 50 m/s. The theoretical value of the period of vibration is 29.27 μs as calculated [13] using $\pi a\sqrt{(\rho/\mu)(1-\nu)/(1+\nu)}$, where $a = 3.95$ cm is the radius of the middle plane of the shell and $\mu = E/[2(1+\nu)]$ is the shear modulus. We numerically calculate this example with and without the distribution coefficient algorithm. At time step $\Delta t = 0.01$ μs the calculation using the distribution coefficient algorithm yields the period of vibration 29.28 μs; while the cal-

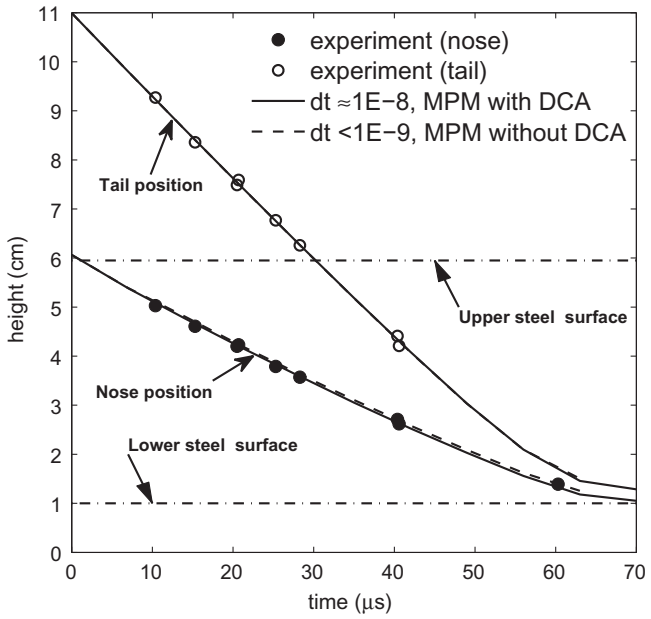


**Fig. 4.** Comparison of the projectile positions calculated with and without the distribution coefficient algorithm (DCA) for small mass nodes.

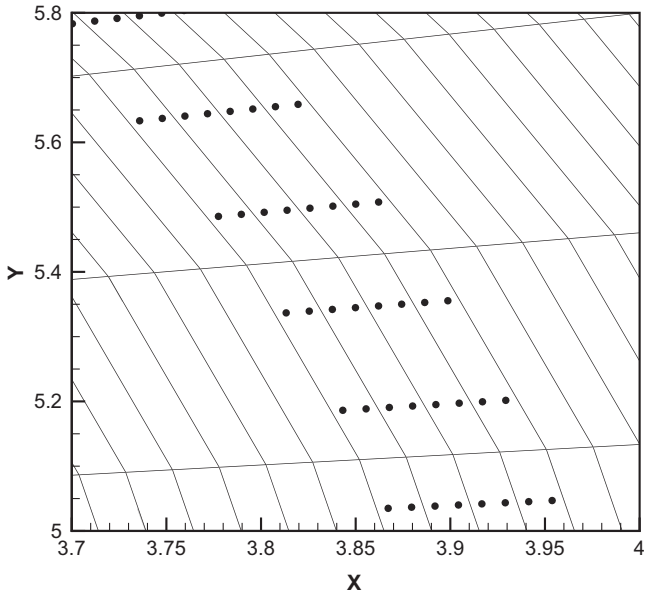


**Fig. 5.** A close-up look at the initial relation between mesh and material points in the example of the thin shell vibration.

culation without the distribution coefficient algorithm lost its stability. When the time step is reduced to $\Delta t = 0.001$ μs, both methods produce the same result 29.28 μs. The values of the kinetic energy and the total mechanical energy are plotted in Fig. 6 as functions of time for the calculations using $\Delta t = 0.001$ μs. Both methods are dissipative. In this example, the calculation with the distribution coefficient algorithm is slightly more dissipative than the original material point method. However, this is not a general trend. As an additional test to the distribution coefficient algorithm, we calculate the radial vibration of a circular copper disk under plane strain as in [6,9], the original material point method is slightly more dissipative than the method with the distribution coefficient algorithm. In these calculations, the copper disk is the same size (0.6 cm in radius ) as in [6,9]. The symmetry condition is used. The computational domain is a $1 \times 2$ cm rectangle divided

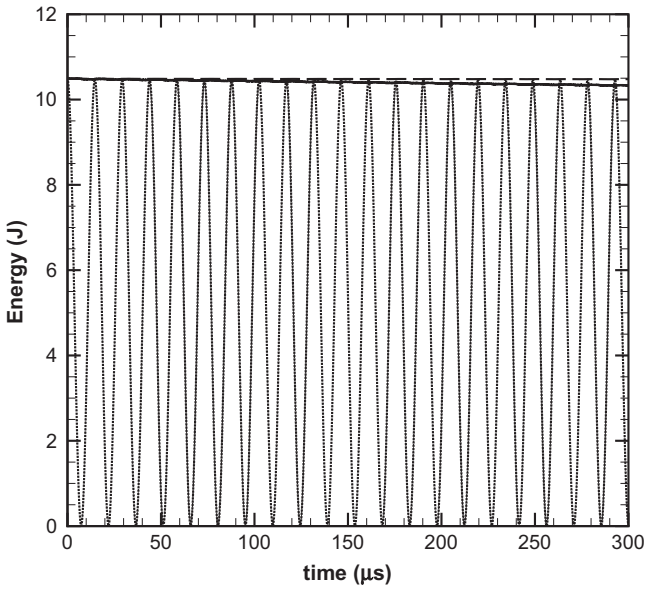


**Fig. 6.** Comparison of energies calculated with and without the distribution coefficient algorithm. The lines going up and down are kinetic energies of the shell as functions of time. In this figure, the kinetic energy difference between the calculations with and without the distribution coefficient algorithm is barely visible. The dashed line on the top is the mechanical energy calculated without the distribution coefficient algorithm, and the solid line is the mechanical energy calculated with the distribution coefficient algorithm.

into $20 \times 40$ square cells in the $x$ and $y$ directions respectively, and contains half of the disk. In both cases, the calculated period for the radial vibration is the same as the theoretical value 2.63 μs ([6,9]).

## 8. Conclusions

For cases with large deformation of materials, the distribution of material points often causes small masses on nodes near material interfaces. Such a small mass node not only leads to tiny time steps, but also often results in instability and failure of numerical simulations. A numerical algorithm for these small mass nodes is introduced in the present paper.

This algorithm distributes the force on small mass nodes to surrounding nodes. The amount of the force distributed is proportional to the square root of the mass at the recipient node. This algorithm is shown to exactly satisfy mass and momentum conservation laws. The energy conservation error is proportional to the square of the time step, as in the original material point methods.

Numerical examples show this numerical algorithm is very effective and can significantly increase the allowed size of the time steps, therefore significantly reducing total computation time. The introduced algorithm is a numerical scheme to prevent numerical instability; it does not increase or reduce the accuracy of the numerical solution.

In the present paper we derived the algorithm for the momentum equations. Similar stability issues appear in solving energy equations on nodes with small thermal capacities. A similar approach can be used to enhance the numerical stability in those calculations.

## Acknowledgments

The authors would like to acknowledge Dr. Rick M. Rauenzahn for many constructive discussions. This work was performed under the auspices of the United States Department of Energy.

## References

[1] S.G. Bardenhagen, E.M. Kober, The generalized interpolation material point method, Int. J. Numer. Meth. Eng. 5 (1) (2004) 477.
[2] D.Z. Zhang, Q. Zou, W.B. VanderHeyden, X. Ma, Material point method applied to multiphase flows, J. Comput. Phys. 227 (2008) 3159.
[3] D.Z. Zhang, W.B. VanderHeyden, Q. Zou, Nely T. Padial-Collins, Pressure calculations in disperse and continuous multiphase flows, Int. J. Multiphase Flow. 33 (2007) 86.
[4] S.G. Bardenhagen, J.U. Brackbill, D. Sulsky, The material-point method for granular materials, Comput. Method Appl. M. 187 (2000) 529.
[5] D. Sulsky, S.-J. Zhou, H.L. Schreyer, Application of a particle-in-cell method to solid mechanics, Comput. Phys. Commun. 87 (1995) 236.
[6] S.J. Cummins, J.U. Brackbill, An implicit particle-in-cell method for granular materials, J. Comput. Phys. 180 (2002) 506.
[7] D. Yang, R.P. Currier, D.Z. Zhang, Ensemble phase averaged equations for multiphase flows in porous media. Part 1: The bundle-of-tubes model, Int. J. Multiphase Flow 35 (2009) 628.
[8] D.Z. Zhang, Ensemble phase averaged equations for multiphase flows in porous media. Part 2: A general theory, Int. J. Multiphase Flow 35 (2009) 640.

[9] D. Burgess, D. Sulsky, J.U. Brackbill, Mass matrix formulation of the FLIP particle-in-cell method, J. Comput. Phys. 103 (1992) 1.
[10] S.G. Bardenhagen, Energy conservation error in the material point method for solid mechanics, J. Comput. Phys. 180 (2002) 383.
[11] O. Buzz, D.M. Pedroso, A. Giacomini, Caveats on the implementation of the generalized material point method, CMES 1 (1) (2008) 1.
[12] A. Prosperetti, G. Tryggvason, Computational Methods for Multiphase Flow, Cambridge University Press, 2006 (Chapter 10).
[13] A.E.H. Love, A Treatise on the Mathematical Theory of Elasticity, fourth ed., Dover Publications, New York, 1927 (Chapter 8).