

Caveats on the Implementation of the Generalized Material Point Method

O. Buzzi¹, D. M. Pedroso² and A. Giacomini¹

Abstract: The material point method (MPM) is a numerical method for the solution of problems in continuum mechanics, including situations of large deformations. A generalization (GMPM) of this method was introduced by Bardenhagen and Kober (2004) in order to avoid some computational instabilities inherent to the original method (MPM). This generalization leads to a method more akin of the Petrov-Galerkin procedure. Although it is possible to find in the literature examples of the deduction and applications of the MPM/GMPM to specific problems, its detailed implementation is yet to be presented. Therefore, this paper attempts to describe all steps required for the explicit implementation of the material point method, including its generalization. Moreover, some caveats during the implementation are addressed. For example, the setting up of boundary conditions and the steps for the computation of values at nodes and material points are discussed. The influences of the time and space discretization are also verified, basing on numerical analyses. Two strategies for the update of stress, known as update stress first (USF) and update stress last (USL) are numerically investigated. It is shown that both the order for the computation of boundary conditions and the way that the grid values are extrapolated have high impact on the accuracy of the solution. The complete 3D algorithm is detailed and summarized in order to make easier the implementation of the GMPM/MPM.

Keyword: Material point method, MPM, generalized material point method, GMPM, boundary conditions, update stress first, update stress last.

1 Introduction

A goal of solid mechanics is the understanding of the mechanical behaviour of structures by means of the study of forces and displacements. However, as the material deforms, a rational approach for the internal response must also be considered. The continuum mechanics is such a theory that provides the framework for the representation of the internal response by defining the stress and strain concepts (Eringen, 1967; Malvern, 1969). Three requirements for the definition of a problem within the continuum mechanics are the balance of momentum, the kinematics of deformation, and the constitutive relations. These lead to a boundary value problem which, depending on the complexity of the geometry and materials under analysis, can not be solved analytically.

The material point method (MPM) (Sulsky, Chen, and Schreyer, 1994; Sulsky, Zhou, and Schreyer, 1995; Sulsky and Schreyer, 1996; Sulsky, Schreyer, Peterson, Kwok, and Coon, 2007) is one numerical solution method to the continuum boundary value problem with a range of features, for instance: a) can cope with finite deformation problems; b) avoidance of mesh tangling; c) ability to advect material properties without numerical diffusion or artificial mixing; d) automatic no-slip contact characteristics; and e) easy definition of the geometry. Several applications of this method to the simulation of a range of problems, including stress propagation, dynamic fracture, multiscale simulations, mesh refinement, among others, can be found in the literature (Bardenhagen, Guilkey, Roessig, Brackbill, Witzel, and Foster, 2001; Shen and Chen, 2005; Guo and Nairn, 2006; Ma, Lu, Wang, Roy, Hornung, Wissink, and Komanduri, 2005; Ma, Liu, Lu, and Komanduri, 2006; Ma, Lu, Wang, Hornung, Wissink, and Komanduri, 2006; Ma, Lu,

¹ Centre for Geotechnical and Materials Modelling, The University of Newcastle, Newcastle NSW 2308 Australia.

² Golder Geomechanics Centre, School of Engineering, The University of Queensland, Brisbane QLD 4072 Australia.

and Komanduri, 2006).

Nonetheless, the computer implementation of this method requires carefully designed steps in order to guarantee that first the method works properly and second a good accuracy and some efficiency can be achieved during simulations. Three main aspects must be considered: a) the establishment of the boundary conditions; b) the extrapolation and interpolation of particles quantities to grid nodes and vice-versa; and c) the order for the update of stress: before or after the solution for the discrete balance of momentum. The first one is discussed here, while the second one is studied in details by Bardenhagen and Kober (2004). The last one is discussed in Bardenhagen (2002).

Due to computational instabilities, a generalization known as the generalized interpolation material point method, here referred to as GMPM, was developed by Bardenhagen and Kober (2004) using a sort of Petrov-Galerkin procedure and, thus, resulting in a method more akin of meshless method such as the meshless local Petrov-Galerkin (MLPG) method (Atluri and Zhu, 1998). Like in the MPM, the GMPM relies on an underlying grid at least for the solution of the discrete governing equations. Therefore, since the grid can be interpreted as an updated Lagrangian frame, the method is also a mix of (arbitrary) Eulerian and Lagrangian approaches (Sulsky, Chen, and Schreyer, 1994; Sulsky, Zhou, and Schreyer, 1995; Sulsky and Schreyer, 1996; Wieckowski, Youn, and Yeon, 1999).

This paper discusses some potential pitfalls that may arise during the explicit implementations of the (generalized) material point method. In addition, the two procedures for updating the stress: a) at advance (USF), or b) lastly (USL) are studied in terms of numerical accuracy and efficiency.

2 Notation

Tensor notation is employed here. The order of every entity is indicated by adding the correspondent number of primes “ \prime ” to their symbol. For example, vectors, which are entities of first order,

are indicated by:

$$\mathbf{v}_i = v_i \mathbf{e}_i \otimes \mathbf{e}_j \quad (\text{Orthonormal Cartesian System OCS}) \quad (1)$$

in which v_i are the Cartesian components of the vector in a system of reference with the orthonormal bases vectors \mathbf{e}_i . The dyadic product between two vectors is defined as:

$$\mathbf{T}_i = \mathbf{u} \otimes \mathbf{v}_i = u_i v_j \mathbf{e}_i \otimes \mathbf{e}_j \quad (\text{OCS}) \quad (2)$$

in which \mathbf{T}_i is a second order tensor. The derivatives of vectors and tensors are given by

$$\frac{d\mathbf{v}_i}{dx_j} = \frac{\partial v_i}{\partial x_j} \mathbf{e}_i \otimes \mathbf{e}_j \quad (3)$$

$$\frac{d\mathbf{\sigma}}{dx} = \frac{\partial \sigma_{ij}}{\partial x_k} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \quad (4)$$

rendering a second and a third order tensor, respectively.

Nodes quantities are denoted by using “n” in the subscripts and material points, or particles, by using “p” in the subscripts. The term “particles” and “material points” are used interchangeably, even though the term “particles” may lead to a discrete interpretation, which is not the case here.

For the graphics, the following convention is selected: the material points are represented by black dots and the nodes by void dots.

3 Generalized Material Point Method

The material point method (MPM) solves the variational form of the conservation of momentum by means of the discretization of the continuum media into particles or material points (Sulsky, Chen, and Schreyer, 1994; Sulsky, Zhou, and Schreyer, 1995). By leaving the masses of these particles unchanged, the conservation of mass is implicitly satisfied. The conservation of energy is not considered in this method. In addition to the discretization via particles, an underlying grid is used to compute the solution for the conservation of momentum and update the particles state. Therefore two sets of interpolation/extrapolation

functions of position (x) are required: a) the particle characteristic functions $\chi_p(x)$; and b) the grid shape functions $S_n(x)$.

In the original MPM (Sulsky, Chen, and Schreyer, 1994; Sulsky, Zhou, and Schreyer, 1995), the particle characteristic functions are of the form:

$$\chi_p(x) = \delta(x - x_p)V_p \quad (5)$$

in which δ corresponds to the Dirac delta function, x_p is the particle position, and V_p is the particle volume.

A generalized form of the material point method (GMPM) was introduced by Bardenhagen and Kober (2004) where any particle characteristic functions can be adopted. Depending on the selection for these functions, the method provides an additional degree of smoothness to the solution. On the other hand, the original MPM may suffer of numerical noise, mainly when material points cross some cells in the computational grid (Bardenhagen and Kober, 2004).

This paper discusses the GMPM, whereas the MPM can be directly recovered by selecting Eq. 5 as particle characteristic function (χ_p).

As presented by Bardenhagen and Kober (2004), the derivation of the GMPM starts with the variational form of the conservation of momentum:

$$\begin{aligned} \int_A w_i \bullet t_i dA + \int_V \rho w_i \bullet b_i dV - \int_V \frac{dw_i}{dx} : \sigma_{ii} dV \\ = \int_V \rho w_i \bullet a_i dV \end{aligned} \quad (6)$$

in which w_i , t_i , b_i , and a_i are vectors representing the weighting functions, tractions, body forces, and accelerations, respectively, and σ_{ii} is the stress tensor. ρ is a scalar indicating the density field.

The density, stress, and acceleration continuum fields are discretized into material points by means of the particle characteristic functions according to:

$$\rho(x) = \sum_p \frac{m_p \chi_p(x)}{V_p} \quad (7)$$

$$\sigma_{ii}(x) = \sum_p \sigma_{ii,p} \chi_p(x) \quad (8)$$

$$\rho(x) a_i(x) = \sum_p \frac{\dot{q}_p \chi_p(x)}{V_p} \quad (9)$$

where m_p are the particle masses and \dot{q} indicates the rate of momentum.

The weighting functions and their derivatives with respect to position are discretized according to a computational grid by means of:

$$w_i(x) = \sum_n S_n(x) w_{in} \quad (10)$$

$$\frac{dw_i(x)}{dx} = \sum_n w_{in} \otimes G_n(x) \quad (11)$$

in which:

$$G_n(x) = \frac{dS_n(x)}{dx} \quad (12)$$

By substituting Eqns. 7-12 into Eq. 6, the discrete governing equations are obtained:

$$f_n^e - f_n^i = \dot{q}_n \quad (13)$$

where the external forces are:

$$f_n^e = \int_A S_n t_i dA + \sum_p m_p b_i \bar{S}_{np}, \quad (14)$$

the internal forces are:

$$f_n^i = \sum_p V_p \sigma_{ii,p} \bullet \bar{G}_{inp}, \quad (15)$$

and the rate of momenta evaluated at grid nodes are:

$$\dot{q}_n = \sum_p \dot{q}_p \bar{S}_{np} \quad (16)$$

In Eqns. 14-16, \bar{S}_{np} and \bar{G}_{inp} are weighting and gradient-weighting functions, respectively, and are given by (see details in Bardenhagen and Kober, 2004):

$$\bar{S}_{np}(x) = \frac{1}{V_p} \int_{V^*} S_n(x) \chi_p(x) dV \quad (17)$$

$$\bar{G}_{inp}(x) = \frac{1}{V_p} \int_{V^*} G_n(x) \chi_p(x) dV \quad (18)$$

where $V^* = V \cap V_p$ denotes the current support of the particle characteristic functions. Note that both functions depend implicitly on the grid nodes positions x_n and particle position x_p .

4 Aspects of implementation

In this paper, the explicit solution in time is considered, where the time step is given by Δt .

Three phases are necessary for the explicit implementation of the GMPM (or MPM): a) initialization of the underlying grid; b) solution of the discrete balance of momentum at grid nodes; and c) update of the material points state. However, two intermediary steps must also be considered: i) stress update of the material points; and ii) setting up of boundary conditions. The order of these two steps in the algorithm affects the accuracy and efficiency of the method. The first step (i) can be implemented before or after the computation of the internal forces, and, thus, before or after the discrete solution of the balance of momentum. Bardenhagen (2002) explains the difference in the solution using these two approaches for step (i), where the methods known as update stresses first (USF) and update stresses last (USL) are analysed using an energy point of view. In this paper, attention is focused on the numerical characteristics of both algorithms. The second step (ii) can be done by zeroing some terms on the computation grid, for the case of essential boundary conditions given *a priori*. Nonetheless, this last step (ii) must be called in the right position inside the explicit loop.

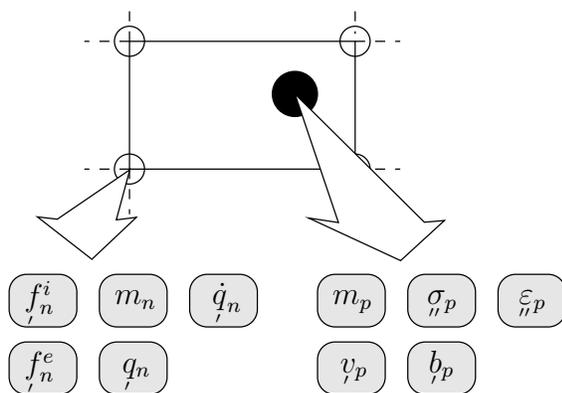


Figure 1: Entities computed at grid nodes (n) and required at material points (p).

Not every mechanical entity is required at the material points and grid nodes at the same time. Ac-

tually, the mass (m_p), velocity (v_p), strain (ϵ_{np}), stress (σ_{np}), and body masses (b_p) are recorded at material points. The mass (m_n), momentum (q_n), internal force (f_n^i), external force (f_n^e), and rate of momenta (\dot{q}_n) are computed at grid nodes (see Fig. 1). Clearly, the position of nodes (x_n) and material points (x_p) must be known or computed every time. The velocity of nodes (v_n) is only necessary temporarily, during the stress update, as discussed later, and hence does not have to be stored in the computer memory.

The algorithm that considers the first approach (USF) for the update of stresses is illustrated in Fig. 2 and is explained as follows. For each time step, clear the grid values (see Fig. 1), calculate the weighting and gradient-weighting functions, and then:

1. Compute node mass from particle masses
2. Compute node momentum from particle masses and velocities
3. Compute node velocity from grid mass and momentum
4. Compute particle strains from grid velocity
5. Update particle stress from particle strains
6. Compute internal forces from particle stresses
7. Compute external forces from particle body masses
8. Compute the rate of momentum for grid nodes
9. Update grid momentum from the rate of momentum at nodes
10. Update particle velocities from the rate of momentum at nodes
11. Update particle positions from the updated grid momentum

In this algorithm (USF - Fig. 2), the steps (1), (2), and (3) correspond to the phase a) initialization of the underlying grid, the steps (4) and (5) correspond to the intermediary step (i) update of particle stresses, the steps (6), (7) and (8) correspond to the phase b) solution of the discrete momentum balance, and the steps (9), (10), and (11) correspond to the phase c) update of particles states.

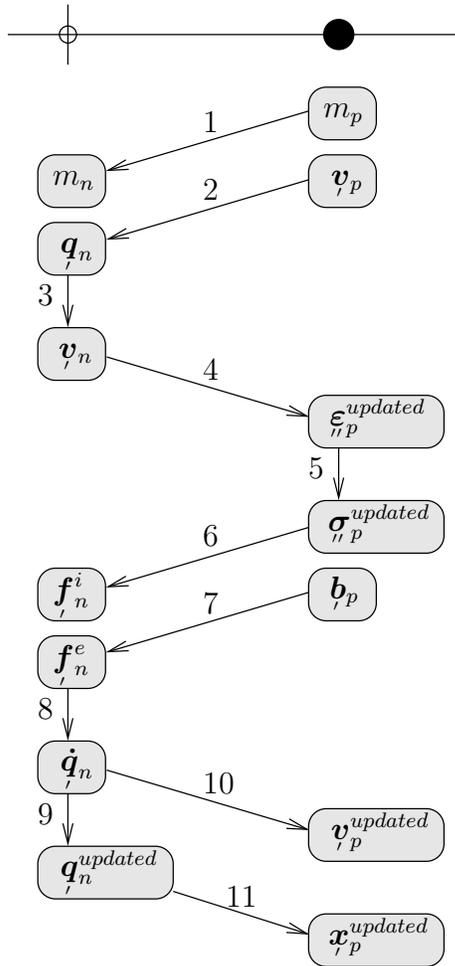


Figure 2: Update stress first (USF): entities computed at grid nodes (n) and material points (p).

The algorithm which considers the second approach (USL) for the update of stresses is illustrated in Fig. 3, where the steps are: for each time step, clear the grid values, calculate the weighting and gradient-weighting functions, and then:

1. Compute grid mass from particle masses
2. Compute grid momentum from particle masses and velocities
3. Compute internal forces from particle stresses
4. Compute external forces from particle body masses
5. Compute the rate of momentum for grid nodes

6. Update grid momentum from the rate of momentum at nodes
7. Update particle velocities from the rate of momentum at nodes
8. Update particle positions from the updated grid momentum
9. Compute grid velocity from grid mass and momentum
10. Compute particle strains from grid velocity
11. Update particle stress from particle strains

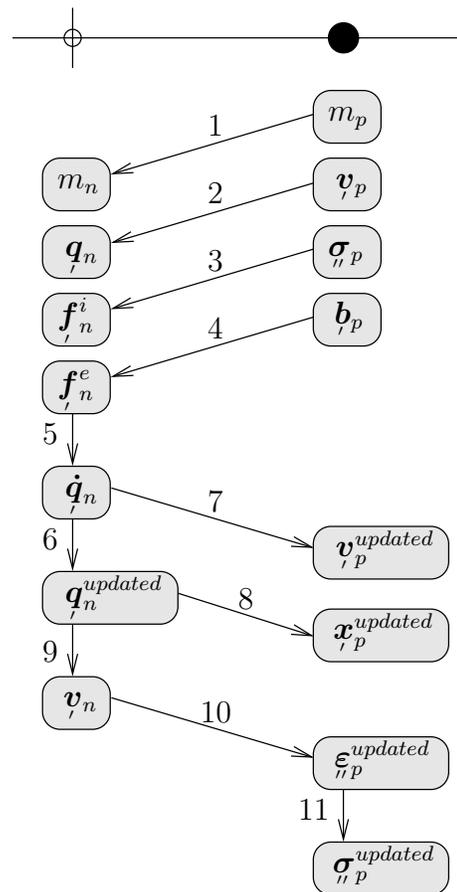


Figure 3: Update stress last (USL): entities computed at grid nodes (n) and material points (p).

In the USL algorithm (Fig. 3), the steps (1) and (2) correspond to the first phase, the steps (3), (4), and (5) correspond to the second phase, and the steps (6), (7), (8) correspond to the third phase. In this approach (USL), the steps (9), (10), and (11) are the ones for the intermediary step (i).

A complete study between the two approaches, USF and USL, is given by Bardenhagen (2002). Here, the focus is on the detailed implementation of the algorithm, considering the computation of the interpolation functions, setting up of boundary conditions, and numerical accuracy. Only the explicit version of the MPM is considered here, while details about the implicit implementation can be found elsewhere, for example in Guilkey and Weiss (2003).

4.1 Structured grid

Since the grid can have any shape, a structured grid can be adopted, resulting in great convenience, for example, when tracking the location of material points. This tracking must be done for each time step, before computing the weighting and weighting-gradient coefficients.

During the initialization phase, the values at the nodes can be accumulated from the values at the points; however it is more convenient to loop over all material points and add the contributions from each point to the surrounding nodes. This approach of implementation can also facilitate a parallel implementation of the method, considering that each point needs to add only to the surrounding nodes of the grid. Therefore, for example, each processor will have to record only those grid nodes in the domain where the tracked particles would contribute to.

In higher dimensions, 2D and 3D, it is common to adopt grid shape functions defined as product of two nodal tent functions (see e.g Bardenhagen and Kober, 2004; Daphalapurkar, Lu, Coker, and Komanduri, 2007), according to:

$$S_n(\vec{x}) = S_n^x(x) \cdot S_n^y(y) \cdot S_n^z(z) \quad (19)$$

The same can be adopted for the particle characteristics functions:

$$\chi_p(\vec{x}) = \chi_p^x(x) \cdot \chi_p^y(y) \cdot \chi_p^z(z) \quad (20)$$

Due to the definition of “weighting” and “gradient-weighting” functions (Eq. 17 and Eq. 18) in the GMPM, the grid support of these functions will range more points than in the MPM. Actually, in the 2D MPM, each particle

contributes to 4 nodes, while in the 2D GMPM, each particle contributes to at most 16 nodes. For three-dimensional problems, in the MPM, each particle contributes to 8 nodes, while in the GMPM, each particle contributes to 64 nodes. This can be observed, for instance, in Fig. 4 where all nodes 7-10, 13-16, 19-22, and 25-28 will receive contributions from particle “p” in the 2D GMPM, while only nodes 14,15,20, and 21 will get any contribution in the 2D MPM.

With a structured grid, it is possible to define a reference node (see Fig. 4) and to loop over all the contributing nodes. For the MPM, the number of this node can be computed using the following expression:

$$\begin{aligned} n_p^* = & \text{trunc} \left(\frac{x_p - x_{min}}{\Delta x} \right) \\ & + \text{trunc} \left(\frac{y_p - y_{min}}{\Delta y} \right) N_x \\ & + \text{trunc} \left(\frac{z_p - z_{min}}{\Delta z} \right) N_x N_y \end{aligned} \quad (21)$$

where x_{min} , y_{min} and z_{min} are the minimum grid nodes coordinates; x_p , y_p and z_p are the coordinates of the material point. The *trunc* function returns the entire value of a number. The terms with N_x (number of nodes along the x direction) and N_y (number of nodes along the y direction) can be dropped in case of 1D or 2D simulations, respectively. For the GMPM, the reference node is in one cell further from the cell where the material point under consideration is located (see Fig. 4), since the support of the interpolation function is bigger. In this case, the reference node can be found using the following expression:

$$\begin{aligned} n_p^* = & \text{trunc} \left(\frac{x_p - x_{min}}{\Delta x} \right) - 1 \\ & + \left[\text{trunc} \left(\frac{y_p - y_{min}}{\Delta y} \right) - 1 \right] N_x \\ & + \left[\text{trunc} \left(\frac{z_p - z_{min}}{\Delta z} \right) - 1 \right] N_x N_y \end{aligned} \quad (22)$$

In the USF approach, for each time-step, the weighting values are used 3 times and the weighting-gradient values are used 2 times. Therefore, if memory is not a limitation, these values can be computed once at the beginning of the

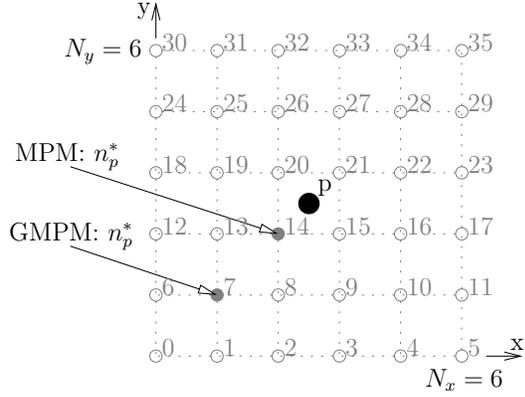


Figure 4: Structured grid and reference nodes.

time-step at stored in two arrays. For the 3D situation using the GMPM, each point will need 64 float points for the weighting values and 3×64 for the weighting-gradients, which may lead to a memory demanding implementation.

In the algorithm presented here, the implementation with pre-allocated arrays for the weighting and weighting-gradients is adopted. These arrays are:

$$S_{pr} = \bar{S}_{np}(x_n, x_p) \quad (23)$$

and

$$G_{rpr} = \bar{G}_{rnp}(x_n, x_p) \quad (24)$$

where r corresponds to the range of nodes that each material point have some contribution ($1 \leq r \leq 2^{N_{DIM}}$ for the MPM, and $1 \leq r \leq 4^{N_{DIM}}$ for the GMPM) and n can be found from this range by means of Eq. 21 or Eq. 22.

4.2 First phase: points to grid

The two principal entities required at grid nodes are the mass (m_n) and the momentum (q_n). During the initialization step, an extrapolation of these values from the particles to the grid nodes must be carried out. One approach is to use the weighting functions according to:

$$m_n = \sum_p \bar{S}_{np} m_p \quad (25)$$

and

$$q_n = \sum_p \bar{S}_{np} m_p v_p \quad (26)$$

Note that the momentum on the particles is not required explicitly in this implementation, since the momentum on the nodes is computed directly from the mass and velocity on the particles (material points). This point will be discussed in section 5.4.

Additionally, in this step, the essential boundary conditions must be considered. In this case, for each constrained direction, the correspondent component of the momentum at nodes is cleared:

$$q_k^{\text{fixed-nodes}} = 0 \quad \{k \in D_{\text{constrained}}\} \quad (27)$$

in which $D_{\text{constrained}}$ is the set of all constrained directions (degrees of freedom - DOFs). The order that this step is implemented in the algorithm is important in terms of accuracy. For example, if the velocities are set equal to zero for the fixed nodes during the stress-update, the boundary conditions will not be considered during the update of nodes position.

4.3 Second phase: discrete solution

The solution of the discrete governing equations is quite simple in both the explicit MPM and GMPM. This is accomplished by subtracting the external forces from the internal forces and updating the nodes momentum according to Eq. 13. The caveat here is that the update of the nodes momentum must be done after the consideration of the essential boundary conditions. These are set simply by zeroing the components of the rate of momentum of the fixed nodes, for each constrained direction (DOF):

$$\dot{q}_k^{\text{fixed-nodes}} = 0 \quad \{k \in D_{\text{constrained}}\} \quad (28)$$

This phase can be viewed as an updated Lagrangian procedure, since the nodes are actually moving, at least temporarily and until the material points (or particles) are updated.

If the update stress first (USF) is selected, the strains and stresses in the material points must be updated at this stage, where the strain increment

can be calculated from the (grid nodes) velocity gradient by means of a volume weighted average over each particle, according to:

$$\Delta \varepsilon_{np} = \Delta t \sum_n \frac{1}{2} \left(\gamma_n \otimes \bar{G}_{np} + \bar{G}_{np} \otimes \gamma_n \right) \quad (29)$$

The increment of stress can then be calculated by any conventional stress-update algorithm. For non-linear constitutive laws, the explicit schemes based on the embedded-Runge-Kutta of second order can be adopted (see, for example, Sloan, 1987; Sloan and Booker, 1992; Sloan, Abbo, and Sheng, 2001; Pedroso, Sheng, and Sloan, 2008) resulting in a convenient algorithm due to the automatic substepping technique.

4.4 Third phase: grid to points

As discussed in Bardenhagen and Kober (2004), since there are not necessarily unique relationships between points and nodes, an weighted averaged approach have to be selected in order to extrapolate (back) the solution from grid nodes to the material points. To this end, the position and velocity of the particles are also updated by using the weighting functions, according to:

$$x_p = x_p + \Delta t \sum_n \frac{\bar{S}_{np} q_n}{m_n} \quad (30)$$

$$v_p = v_p + \Delta t \sum_n \frac{\bar{S}_{np} \dot{q}_n}{m_n} \quad (31)$$

It is important to note that care must be taken with grid nodes that have no mass (herein referred to as phantom nodes), or that have a small mass as compared to a tolerance M_{TOL} , because the denominator in Eqns. 30 and 31 for the conversion of node momentum to node velocity.

As discussed by Chen and Brannon (2002), particle interpenetration is precluded due to the use of nodes momentum in Eq. 30.

4.5 Complete algorithm

For each time-step, the grid values must be cleared, in other words, the previous grid is discarded, and the interpolation values are computed again. Then the three phases, a) points to grid,

```

! Initialize material points
m_p, v_p, \varepsilon_{np}, \sigma_p, b_p, x_p
! Range of contributions (shape functions support)
if (GMPM) R_contr. = [0, 1, 2, 3] else R_contr. = [0, 1]
! Run explicit update
while (t < t_f)
  ! 1) Discard previous grid
  m_n = 0, q_n = 0, f_n^i = 0, f_n^e = 0, \dot{q}_n = 0
  ! 2) Compute interpolation values
  for (p in 1 to N_particles)
    Compute: n_p^* ! Ref. nodes (Eq. 21 or Eq. 22)
    r = 1 ! Position in the S and G arrays
    for (i, j, k in R_contr. \times R_contr. \times R_contr.)
      n = n_p^* + i + jN_x + kN_xN_y
      S_{pr} = \bar{S}_{np}(x_n, x_p)
      G_{pr} = \bar{G}_{np}(x_n, x_p)
      r = r + 1
    end
  end
  ! 3) Initialize grid state (mass and momentum)
  for (p in 1 to N_particles)
    r = 1
    for (i, j, k in R_contr. \times R_contr. \times R_contr.)
      n = n_p^* + i + jN_x + kN_xN_y
      m_n = m_n + S_{pr}m_p
      q_n = q_n + S_{pr}m_pv_p
      if (n is fixed) q_n = 0 ! Fix nodes
      r = r + 1
    end
  end
  ! 4) Update strain and stress
  if (USF) call UpdateStrainsAndStresses
  ! 5) Compute internal and external forces
  for (p in 1 to N_particles) r = 1
    for (i, j, k in R_contr. \times R_contr. \times R_contr.)
      n = n_p^* + i + jN_x + kN_xN_y
      f_n^i = f_n^i + V_p \sigma_p \bullet G_{pr}
      f_n^e = f_n^e + m_p b_p S_{pr}
      if (p has tractions) f_n^e = f_n^e + \int_A S_n t dA
      r = r + 1
    end
  end
end

```

Figure 5: Explicit MPM/GMPM algorithm.

b) discrete solution, and c) grid to points are repeated for each time step. The main loop in the explicit algorithm can be organized into 8 steps, as shown in details in Fig. 5-Fig. 7:

1. Discard previous grid
2. Compute interpolation values
3. Initialize grid state
4. Update strain and stress (USF)
5. Compute internal and external forces
6. Compute rate of momentum and update nodes
7. Update material points
8. Update strain and stress (USL)

Note that steps 4 and 8 are exclusive.

```

! 6) Compute rate of momentum and update
nodes
for (n in 1 to Nnodes)
  q̇n = fne - fni
  if (n is fixed) q̇n = 0 ! Fix nodes
  qn = qn + q̇nΔt
end
! 7) Update material points (position and ve-
locity)
for (p in 1 to Nparticles)
  r = 1
  for (i, j, k in Rcontr. × Rcontr. × Rcontr.)
    n = np* + i + jNx + kNxNy
    if (mn > MTOL)
      xp = xp + ΔtSprqn/mn
      vp = vp + ΔtSprq̇n/mn
    end
  r = r + 1
end
! 8) Update strain and stress
if (USL) call UpdateStrainsAndStresses
! Update time
t = t + Δt
end

```

Figure 6: Explicit MPM/GMPM algorithm (cont).

```

! Update strain and stress
for (p in 1 to Nparticles)
  r = 1, Δε = 0
  for (i, j, k in Rcontr. × Rcontr. × Rcontr.)
    n = np* + i + jNx + kNxNy
    if (mn > MTOL) vn = qn/mn else vn = 0
    Δε = Δε + 0.5(vn ⊗ Gpr + Gpr ⊗ vn)Δt
  r = r + 1
end
εp = εp + Δε
call Update σp for Δε
end

```

Figure 7: Update strains and stresses.

5 Features and caveats

To illustrate the features and to discuss further the caveats during the implementation of the GMPM and MPM, two simple problems in one-dimensional space are analysed. Only a numerical study is carried out here. These are the single-material-point vibration problem and the axial vibration of a continuum bar, as discussed in Bardenhagen (2002) using the original MPM, but here they are also analysed using the generalized material point method (GMPM). Both of these problems have exact analytical solutions, thus the accuracy can be easily investigated.

Fig 8 illustrates the single-material-point vibration, while the vibration of a continuum bar is solved using many material points distributed in this bar.

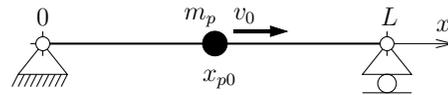


Figure 8: Single-material-point vibration. The bar is represented by a single point initially located at $x_{p0} = L/2$, which has an initial velocity v_0 .

For the single-material-point vibration problem, the bar has Young's modulus equal to $E = 4\pi^2$ and length equal to $L = 1$. The point, represented

by a black dot, is originally located at $x_{p0} = L/2$ and has an original velocity v_0 . The grid is made of two nodes located at $x = 0$ and $x = L$ and represented by circles. In this problem, there is no gravity. The solution in this case is given by:

$$v(t) = v_0 \cos(\omega t) \quad (32)$$

for the velocity and

$$x(t) = x_0 \exp \left[\frac{v_0}{L\omega} \sin(\omega t) \right] \quad (33)$$

for the position, where $\omega = \frac{1}{L} \sqrt{\frac{E}{\rho}}$ and the density is considered constant (equal to 1).

For the vibration of a continuum bar, the Young's modulus adopted is $E = 10$, the length is $L = 1$ and the analytical solution depends now on the mode of vibration (Meirovitch, 1967). Here, only the first mode ($n = 1$) is considered. Then, the solution is

$$v(x, t) = v_o \cos(\omega_1 t) \sin(\beta_1 x) \quad (34)$$

for velocities and

$$u(x, t) = \frac{v_o}{\omega_1} \sin(\omega_1 t) \sin(\beta_1 x) \quad (35)$$

for displacements, where $\beta_1 = \frac{\pi}{2L}$ and $\omega_1 = \frac{\pi}{2L} \sqrt{\frac{E}{\rho}}$. The subscript 1 refers to the first mode of vibration. In this problem, however, the initial conditions are also dependent of the position and have to be set according to:

$$v(x, 0) = v_o \sin(\beta_1 x) \quad (36)$$

for velocities and

$$u(x, 0) = 0 \quad (37)$$

for displacements.

5.1 USF versus USL

The two approaches, USL and USF, are evaluated for the solution of the single-material-point vibration problem. Here, the solution is achieved using both the MPM and the GMPM. The results are presented in Figs. 9-12. Figs. 9 and 10 presents the results computed using the MPM, for each approaches USF and USL, respectively and Figs. 11

and 12 presents the results using the GMPM with the USF and USL, respectively. In each figure, the numerical and analytical values of displacement and velocity are plotted as a function of time. The strain, kinematic and total energy are also shown in order to assess the conservation of energy.

The results obtained with the MPM and USF (Fig. 9) are reasonably accurate when comparing with the analytical solutions. The results obtained with the MPM and USL (Fig. 10) exhibits a higher dissipation of energy, leading to a less accurate solution both in terms of velocities and displacement. These results are in accordance with those presented by Bardenhagen (2002).

For the GMPM, the behaviour in terms of energy dissipation is the same as for the MPM (Figs. 11 and 12). However, the accuracy in the displacements is lower in the case of this single-material-point problem. In the case of the vibration of a continuum bar, as the number of material points increase, the solution using the GMPM becomes much better. This is illustrated in Figs. 13-14. In Fig. 13 the USF approach is used, where no energy dissipation can be easily observed, and in Fig. 14, this dissipation can be observed in the solution using the USL approach, in a similar trend as with using the MPM for the solution of the single-material-point vibration problem.

5.2 Influence of the time step

As in every numerical method, spatial and time discretization play a key role on the accuracy of the results. To assess the convergence features of the MPM and GMPM as these discretizations are improved, the problem of single-material-point vibration is numerically solved where the following error measure is defined:

$$\varepsilon = \frac{\|v_a - v_n\|}{1 + \|v_a\|} \quad (38)$$

in which v_a is the analytical solution for the velocity at the centre of mass while v_n is the numerical solution. Fig. 15 shows the evolution of the computed error with respect to time for two different time steps $\Delta t = 0.1$ and $\Delta t = 0.0001$. As expected, decreasing the time step, the accuracy improves considerably.

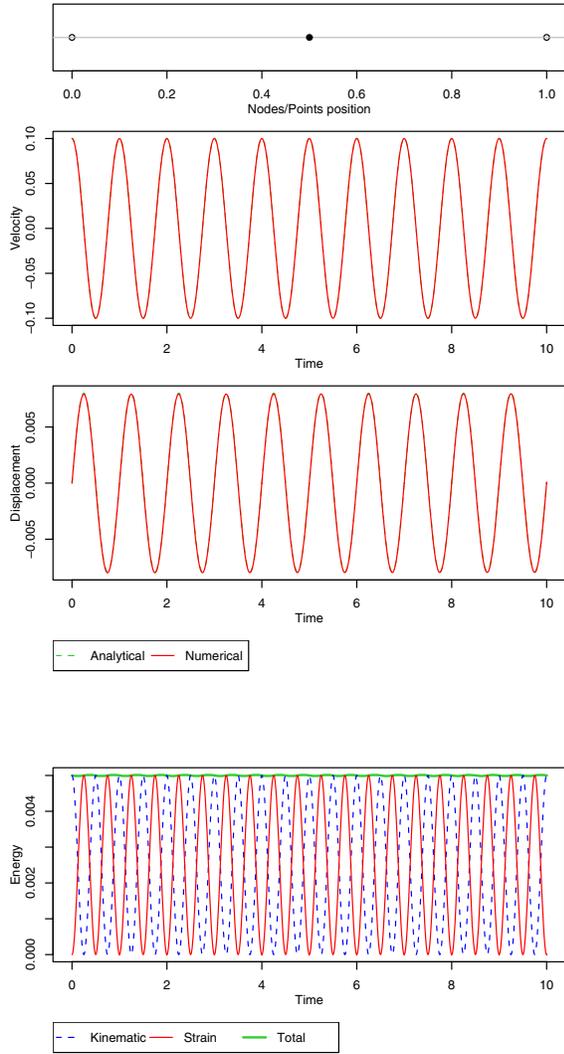


Figure 9: Numerical and analytical results of the single-material-point vibration problem with the USF (update stress first) approach. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. MPM.

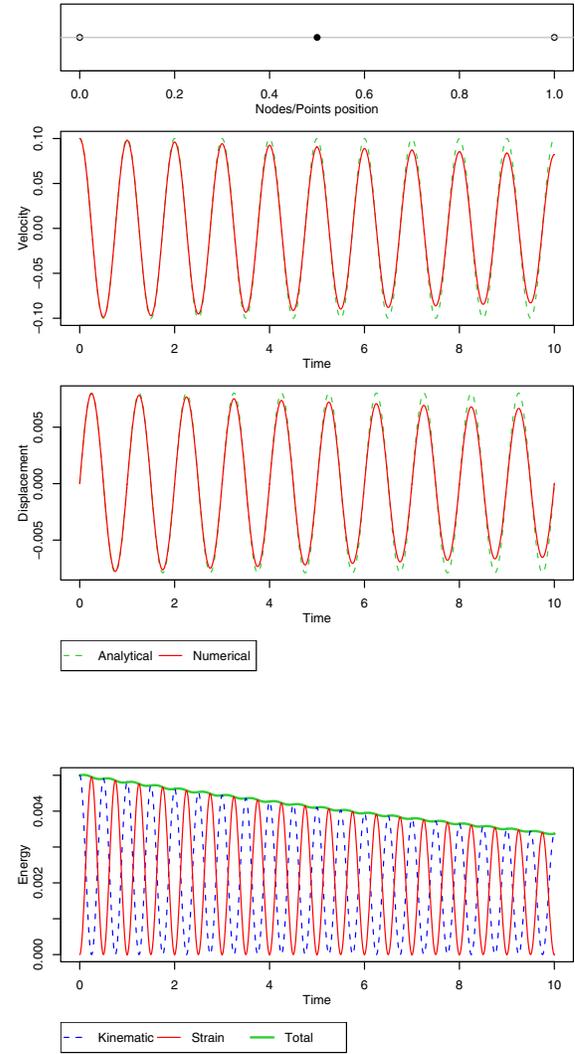


Figure 10: Numerical and analytical results of the single-material-point vibration problem with the USL (update stress last) approach. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. MPM.

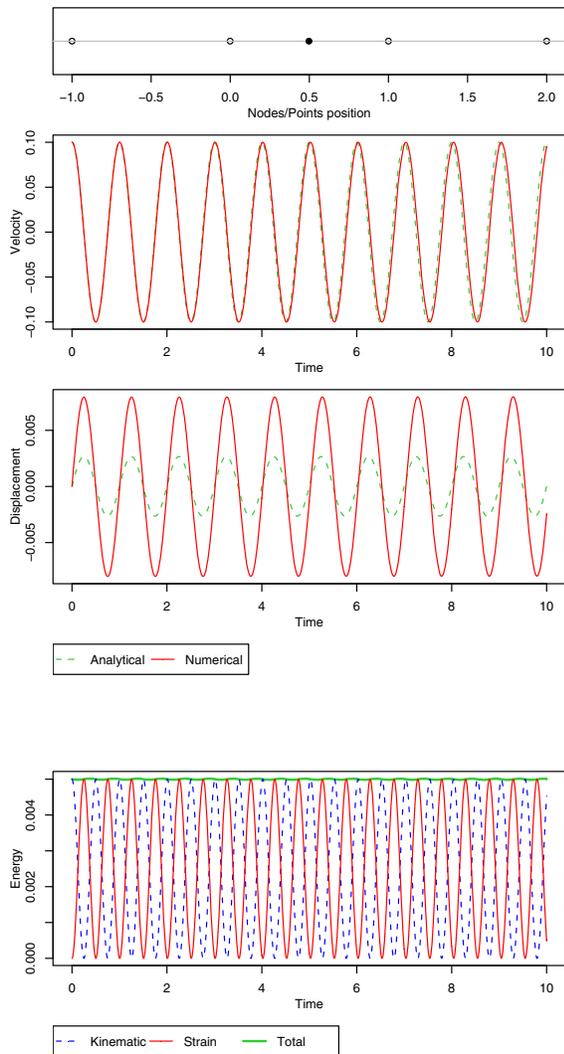


Figure 11: Numerical and analytical results of the single-material-point vibration problem with the USF (update stress first) approach. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. GMPM.

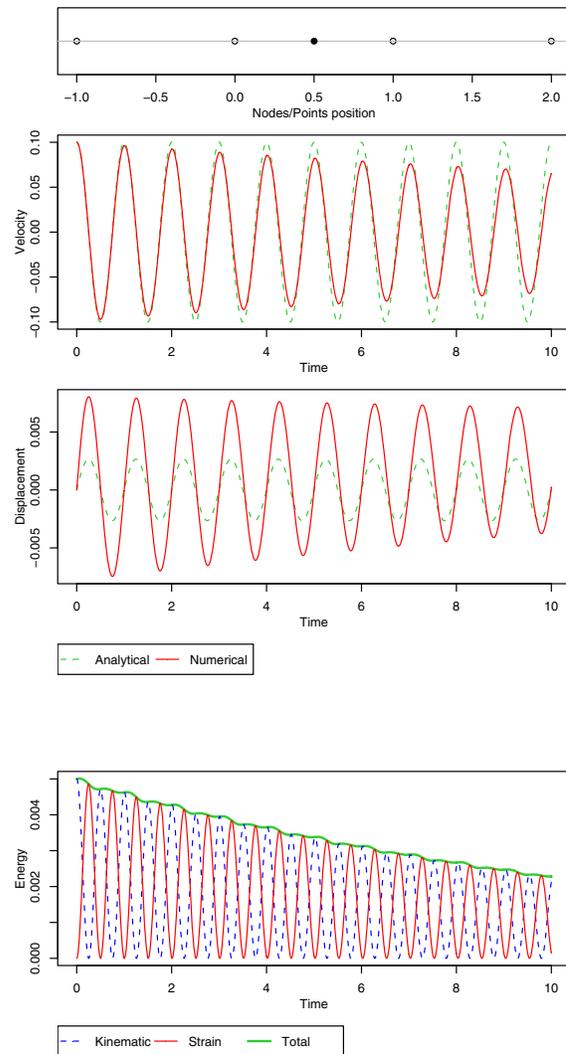


Figure 12: Numerical and analytical results of the single-material-point vibration problem with the USL (update stress last) approach. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. GMPM.

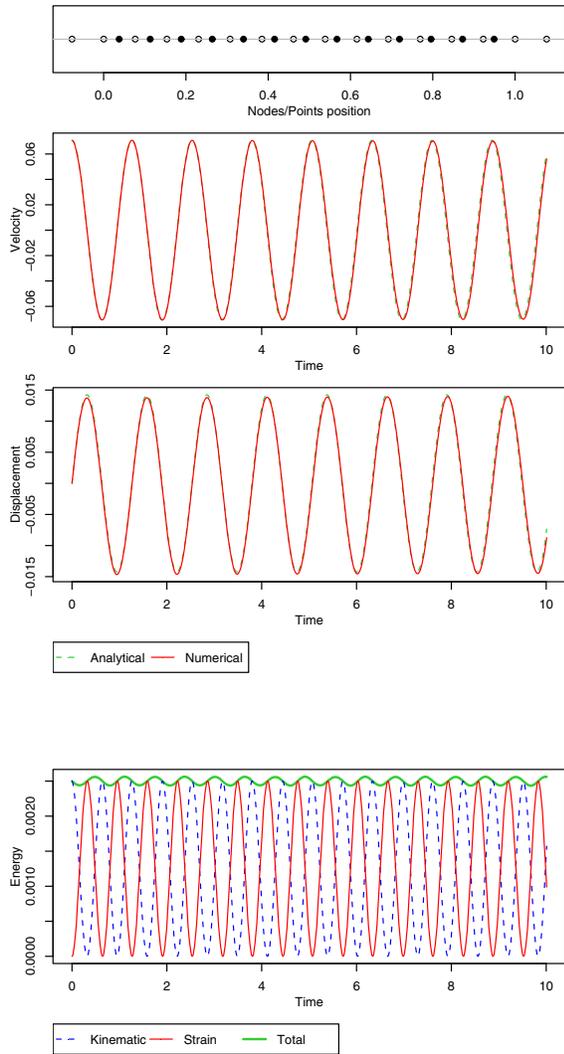


Figure 13: Numerical and analytical results of the vibration of a continuum bar problem (13 material points and 14 nodes) with USF (update stress first) approach. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. GMPM

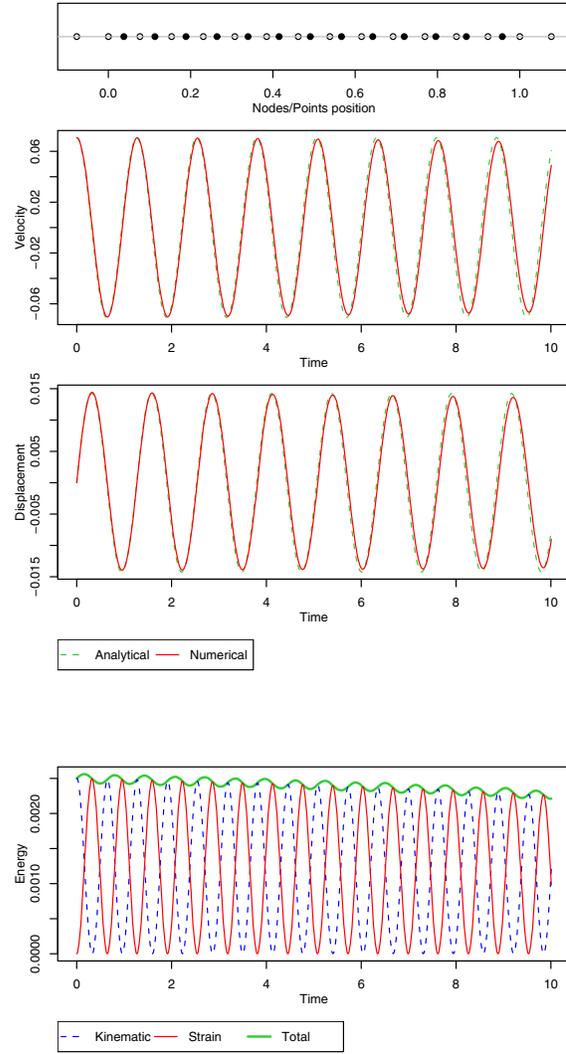


Figure 14: Numerical and analytical results of the vibration of a continuous bar problem (13 material points and 14 nodes) with the USL (update stress last) approach. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. GMPM

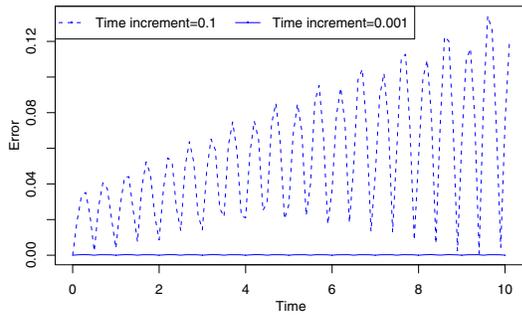


Figure 15: Evolution of the numerical error on the velocity at the centre of mass *versus* time for the single-material-point vibration problem. Simulations run using the MPM, USF and two different time increments $\Delta t = 0.1$ and $\Delta t = 0.0001$.

Additionally, in all results presented so far, a slight oscillation can be noticed on the evolution of total energy. To reduce this oscillation, the time step can be reduced. For example, instead of using $\Delta t = 0.001$ as before, the ten-times smaller increment $\Delta t = 0.0001$ can be considered, resulting in a better accuracy (see Fig. 16). In Fig. 16, different time steps are adopted, in which it is possible to observe that the oscillations tend to vanish with smaller time steps. Moreover, as the time step decreases, it is possible to observe that the dissipation on the total energy computed with the USL approach also decreases.

5.3 Space discretization

The convergence of the solution using the MPM is further investigated by means of numerical analyses. The vibration of a continuum bar problem is solved again, where now both the number of material points and grid nodes are varied. In Fig. 17, the results using the MPM with the USF approach, 3 and 7 material points, and 2 grid nodes are presented. In this case it is possible to observe a relatively high error in the velocity at the centre of mass of the bar. Increasing only the number of material points (Fig. 17(b)) does lead

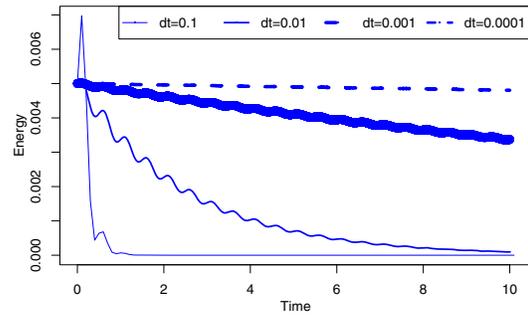


Figure 16: Influence of the time step on the total energy using the USL (update stress last) approach.

to improvement of accuracy. On the contrary, as illustrated in Fig. 18, increasing both the number of grid nodes (4 and 8) and material points (3 and 7), in order to obtain an homogeneous repartition of nodes and materials points, improves significantly the accuracy.

5.4 Extrapolation of point momentum to grid momentum

The first phase of the algorithm, after the calculation of the interpolation functions and initialization the material points, is to extrapolate the material data (mass and momentum) from the point to the grid nodes. The extrapolation of point mass does not rise any particular issue whereas interpolation of the momentum can lead to less accurate results. The correct method is to extrapolate directly the point momentum (Sulsky, Zhou, and Schreyer, 1995; Bardenhagen, 2002) using the weighting functions according to Eq. 26 ($q_n = \sum_p \bar{S}_{np} m_p v_p$). An alternative method would be to interpolate the mass and velocity on the nodes, separately, and then compute the momentum on the nodes, by means of:

$$q_n = m_n v_n = \sum_p \bar{S}_{np} m_p \sum_p \bar{S}_{np} v_p \quad (39)$$

However, as it can be seen from Eq. 39, the weighting functions are used twice during the

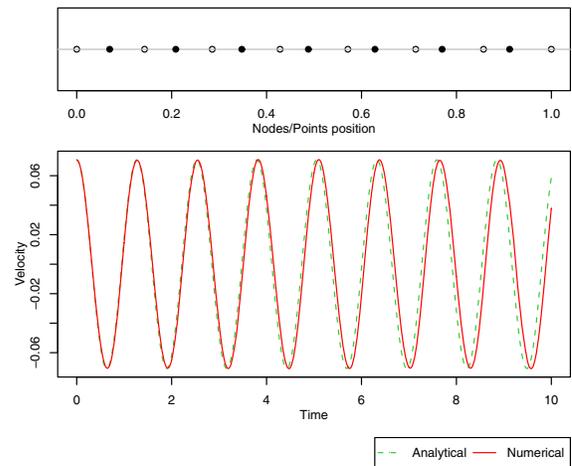
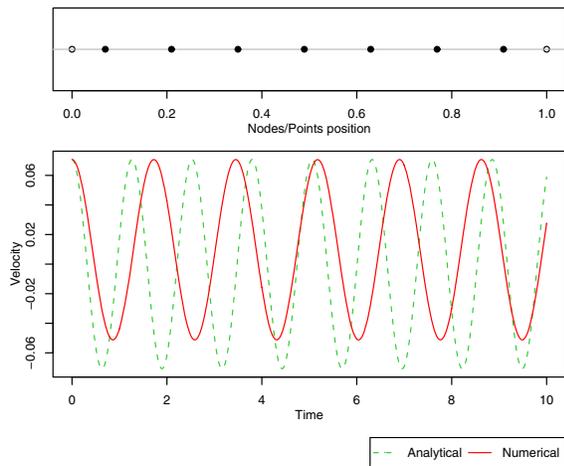
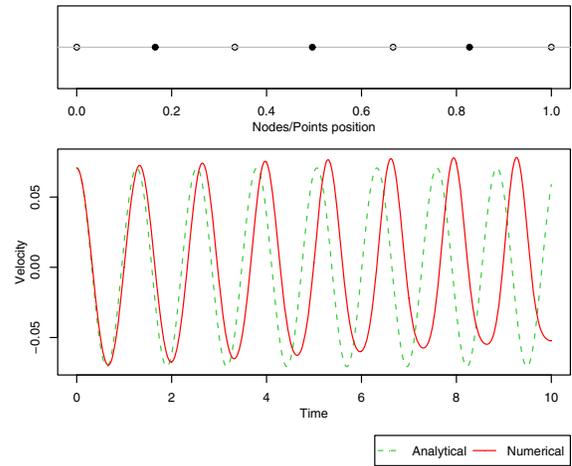
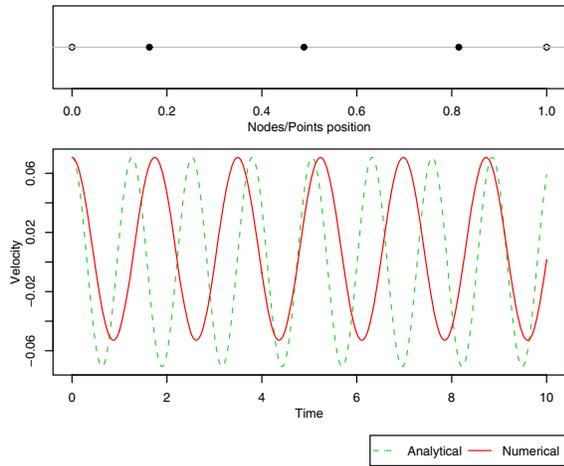


Figure 17: Effect of spatial discretization: numerical and analytical velocity of the centre of mass for the vibration of a continuous bar problem. (a) 3 material points and 2 nodes. (b) 7 material points and 2 nodes. $\Delta t = 0.001$. USF. MPM

Figure 18: Effect of spatial discretization: numerical and analytical velocity of the centre of mass for the vibration of a continuous bar problem. (a) 3 material points and 4 nodes. (b) 7 material points and 8 nodes. $\Delta t = 0.001$. USF. MPM

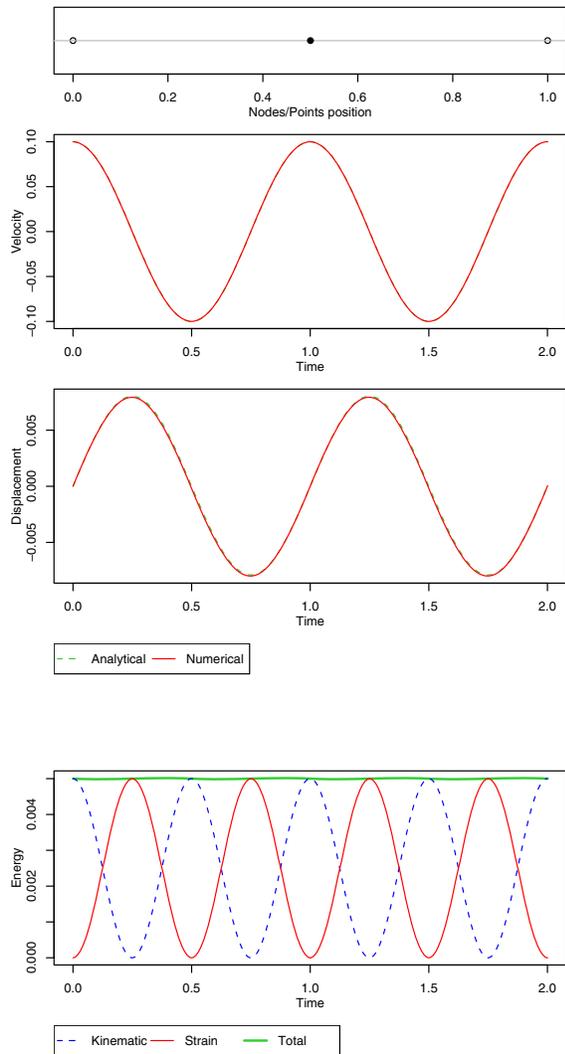


Figure 19: Effect of the extrapolation method: numerical and analytical results of the single-material-point vibration problem point with the “correct” method. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. USF.

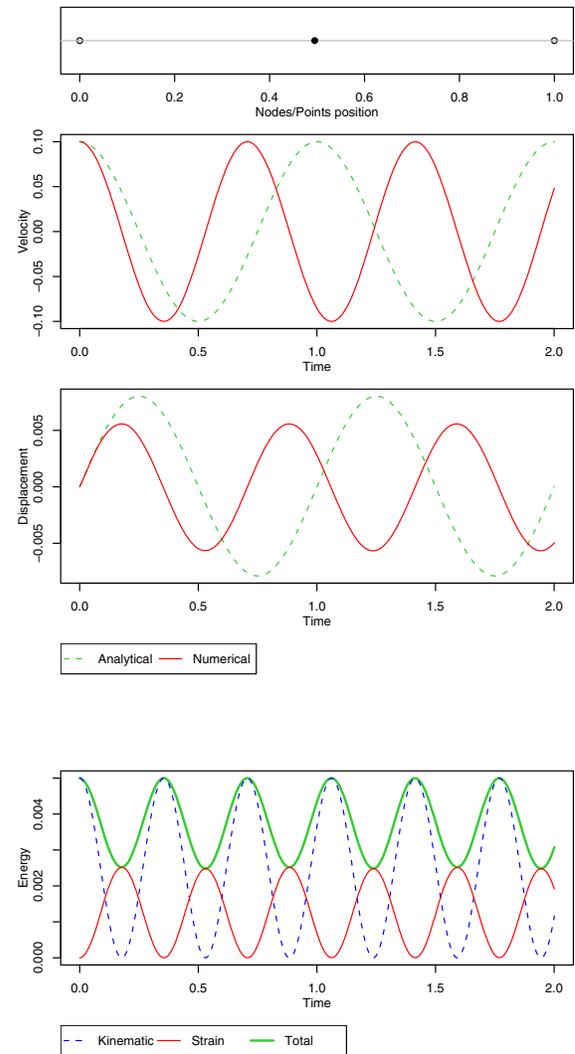


Figure 20: Effect of the extrapolation method: numerical and analytical results of the single-material-point vibration problem with the “wrong” method. (a) Velocity and displacement of centre of mass. (b) Kinematic, strain and total energy. $\Delta t = 0.001$. USF.

computation of the momentum on nodes and, thus, leading to a less accurate result (wrong). The comparison between the “correct” and “wrong” extrapolations, via numerical simulations, is illustrated in Figs. 19 and 20, respectively, where a significant loss of accuracy can be observed for the “wrong” method.

To further improve the computation of the nodes velocities from particles velocities, Wallstedt and Guilkey (2007) suggested the use of information related with the velocity gradient. Nonetheless, this improvement, named gradient enhancement, is not considered in this paper.

5.5 Boundary conditions

The implementation of boundary conditions, more specifically, constrained displacements, is not very detailed in the literature dedicated to the material point method and its generalization. Here, the influence of the application of displacement boundary conditions is investigated, where the conditions of constraints (fixities) with null displacements are applied at the grid nodes by simply zeroing the respective degrees of freedom.

For the analyses of this section, the vibration of a continuous bar problem is solved by the MPM using 7 material points and 8 nodes. In this problem, the only boundary condition to be applied is the constrained horizontal displacement at the left-most node ($x_n = 0$). To illustrate the implementation of the boundary conditions, the three phases: a) points to grid; b) discrete solution; and c) grid to points, discussed before, are sketched as flow charts in Figs. 21 and 22.

Two methods are investigated: the “correct” as illustrated in Fig. 21 and the “improper” as in Fig. 22. In the “correct” method, the BC1 and BC2 steps (see Fig. 21) assure that the momentum at the constrained node is zero, since $q_1^{updated} = q_1 + \dot{q}_1 = 0$. On the other hand, the “improper” method with the BC2 and BC3 steps (see Fig. 22) does not guarantee that the momentum on the left-most node (fixed) is zero. This is also illustrated in Figs. 23 and 24, with the results of the simulation of the vibration of a continuous bar problem using the “correct” and “improper” methods, respectively. The only way the “improper” method

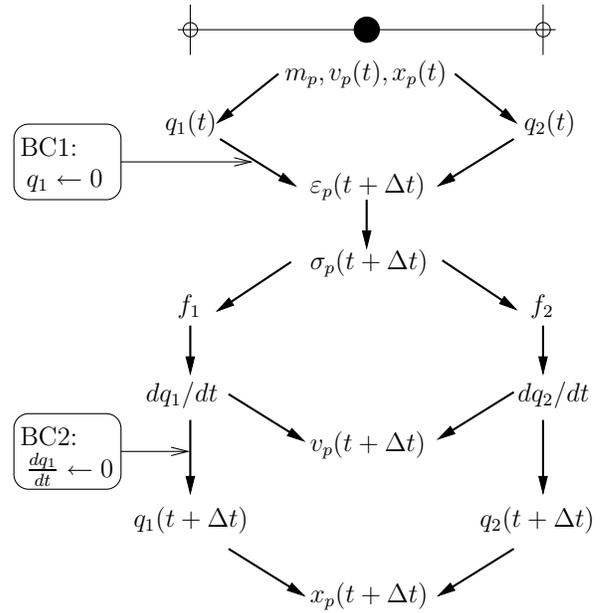


Figure 21: “Correct” method for the displacement boundary conditions. USF.

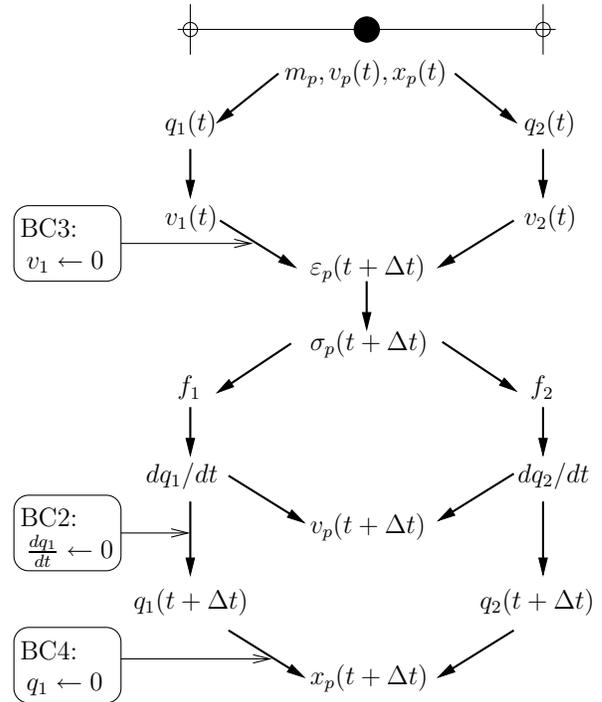


Figure 22: “Improper” method for the displacement boundary conditions for the vibration of a continuous bar problem. USF.

works properly is by adding another step BC4 (as in Fig. 22), making the algorithm a bit more complicated.

Therefore, the displacement boundary conditions at grid nodes have to be applied by zeroing the momentum at the fixed grid nodes during the initialization of the grid state (see step 3 in the algorithm of Fig. 5) and by means of the zeroing of the node momentum during the computation of the rate of momentum (see step 6 in the algorithm of Fig. 6).

5.6 2D Simulations

The application of the algorithm of Figs. 5-7 to the solution of a two-dimensional problem is presented in this section. The problem of two bouncing disks with radii equal to 0.14 is simulated (Fig. 25). The disks have initial velocities with magnitudes equal to 0.1 such as they are moving towards each other along the diagonal of a square. They have a Young's modulus equal to $E = 1.0$ and a Poisson's coefficient equal to $\nu = 0.2$.

First, the results with the GMPM of a simulation with a relatively coarse discretization, as compared with the simulation with a fine discretization presented next, is shown in Fig. 26. In this figure, the current time moment corresponds to that during the impact between the two disks. This coarse discretization corresponds to 24 material points per disk. The colormap displays the relative values of mean pressure $p = (\sigma_x + \sigma_y + \sigma_z)/3$, where it is possible to observe that the points near the contact region have higher compressive pressures. The moment after the impact for this case is presented in Fig. 27, where it is possible to observe a slight change on the shape and relative position of material points. In addition, evolution of kinetic, strain, and total energy from the beginning of the simulation to a moment after the impact is given in Fig. 28.

The same situation is also simulated with the MPM, for which the results are illustrated in Figs. 29 and 30, representing the situation during and after the impact, respectively. For this case, the change in energy is given in Fig. 31, which allows for the conclusion that the results obtained with the MPM are slightly less smooth than with

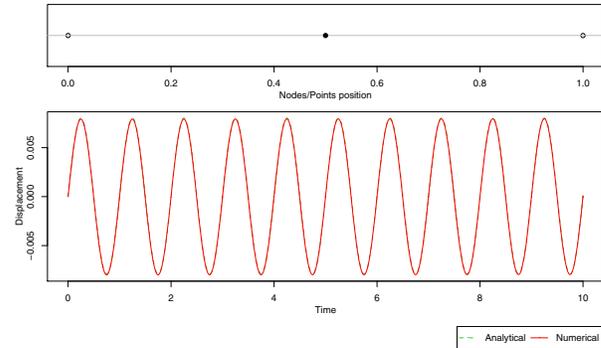


Figure 23: Vibration of a continuous bar problem: “correct” implementation of boundary conditions (BC1 & BC2). 7 material points and 8 nodes. $\Delta t = 0.001$. USF. MPM.

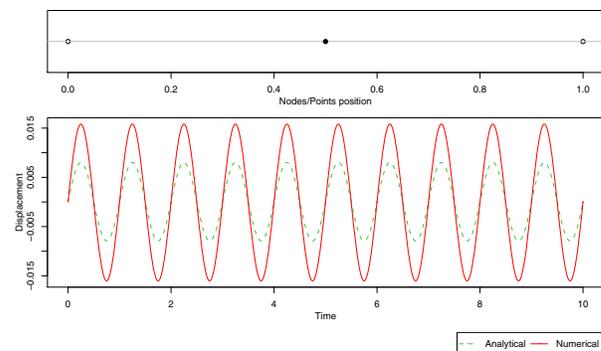


Figure 24: Vibration of a continuous bar problem: “improper” implementation of boundary conditions: BC3 & BC2 but without BC4. 7 material points and 8 nodes. $\Delta t = 0.001$. USF. MPM.

the GMPM; compare, for instance, Fig. 28 with Fig. 31, regarding the energy; and Fig. 27 with Fig. 30, regarding the shape of the material points after the impact. It is also interesting to observe that the shape of the material points simulated with the GMPM seems more akin with the results of the finer discretization, as shown in the following.

Afterwards, simulations with a finer discretization

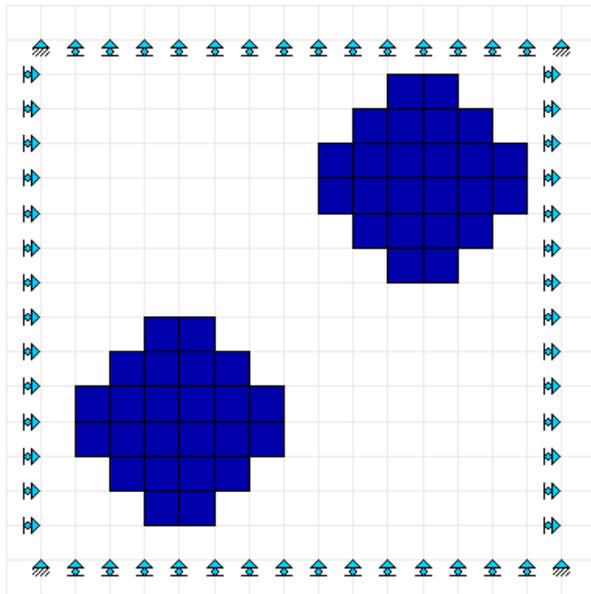


Figure 25: Bouncing disks at initial position. Coarse space discretization.

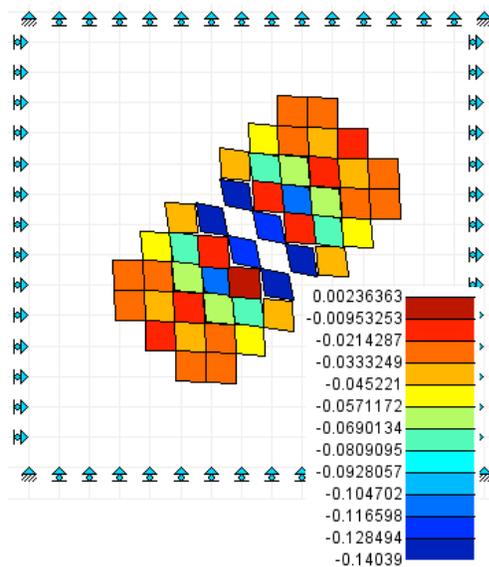


Figure 26: Disks during impact. USF. GMPM.

(392 material points per disk) are presented in order to show the influence of the results in terms of the density field and energy conservation. The density field is, actually, observed by means of the area (volume) of each material point; this is illustrated through small rectangles for each point.

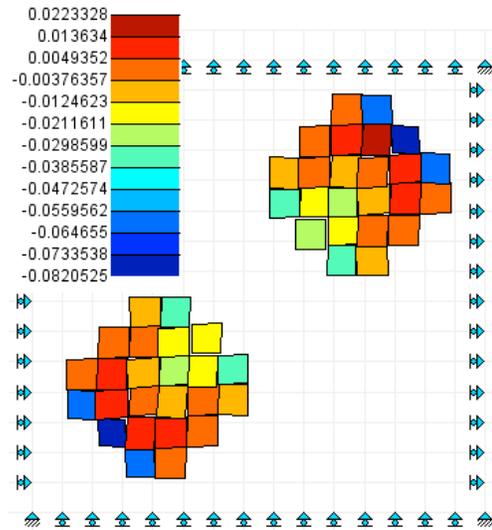


Figure 27: Disks after impact. USF. GMPM.

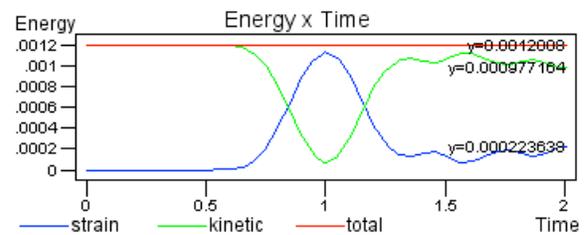


Figure 28: Energy – coarse discr. USF. GMPM.

Only results with the GMPM are presented here; however, the results simulated by the MPM with this finer discretization were quite similar with those obtained with the GMPM.

In Fig. 32, the situation during the impact is illustrated, while in Fig. 33 shows the situation after the impact. The evolution of energy for this case is given in Fig. 34, which allows for the conclusion that the refinement led to a better conservation of energy, in addition to the smoother simulation of the fields of stress and density (volume). Additionally, it can be observed that simulations using the GMPM provides a slight smoother results in terms of density field (small volumes represented by blue squares), total, kinetic, and strain energy than those using the MPM; with more material points, the dissipation of energy is smaller. By improving both the time and space discretiza-

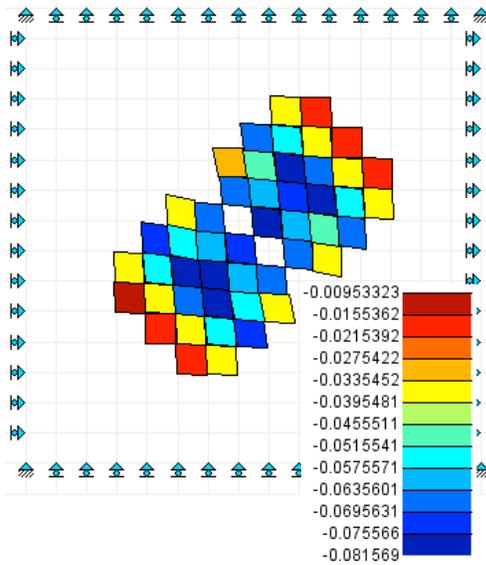


Figure 29: Disks during impact. USF. MPM.

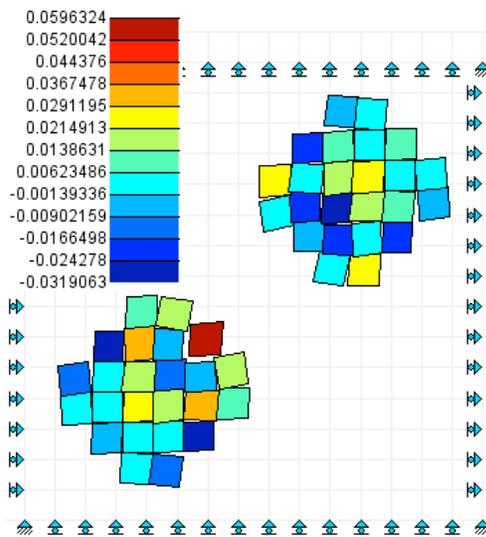


Figure 30: Disks after impact. USF. MPM.

tion, clearly, both methods will converge to the same results.

6 Conclusions

The material point method (MPM) is a numerical technique suited for the solution of large displacement problems in continuum mechanics. The generalized version, GMPM, provides a higher

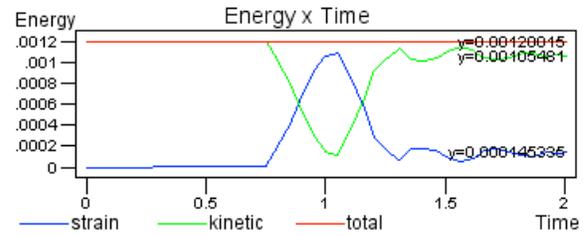


Figure 31: Energy – coarse discr. USF. MPM.

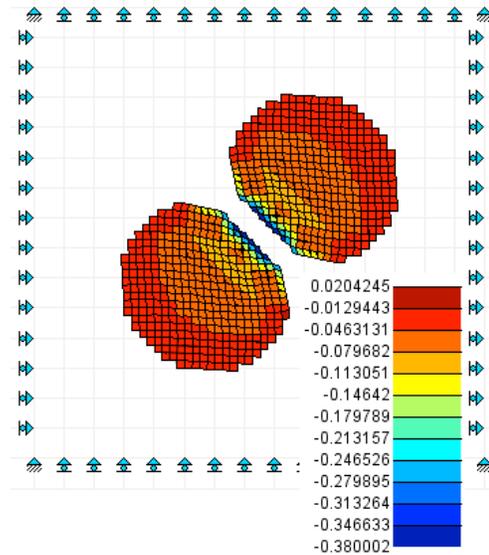


Figure 32: Bouncing disks during impact. USF. GMPM.

degree of smoothness on the computed solution. Although the method is well discussed in the literature, a straightforward presentation containing all steps necessary for its computer implementation is not available. Thus, this paper attempted to give all details required.

Among the details of the implementation of the MPM or GMPM, the setting up of essential boundary conditions, methods for the extrapolation from particles to nodes and vice-versa, and the order for the steps for stress-update were clarified and some caveats discussed.

Two approaches for the stress-update were considered, following the study by Bardenhagen (2002). It is found that the USF (update stress first) approach, which is called before the compu-

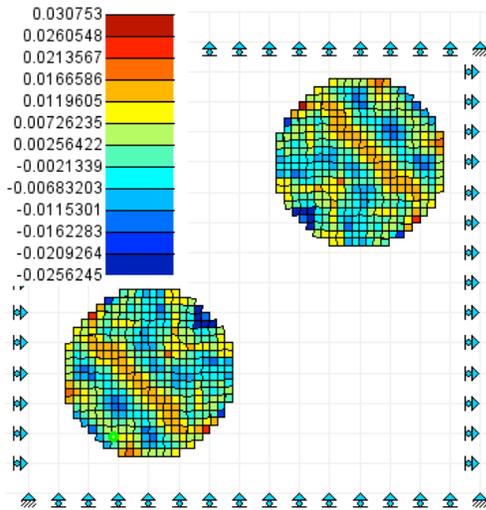


Figure 33: Bouncing disks after impact. USF. GMPM.

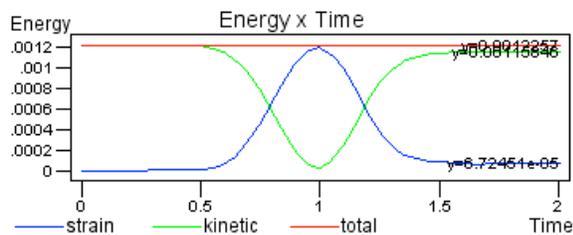


Figure 34: Energy – fine discr. GMPM. USF.

tation of the internal forces, gives a better conservation of the energy than the USL (update stress last) approach.

The essential boundary conditions at fixed nodes are applied by zeroing the DOF components of the nodes momentum during the initialization of the grid state and by means of zeroing of the rate of momentum of nodes after the computation of the internal and external forces.

The influence of the time and space discretization were also investigated. It was observed that there is an ideal balance between the number of material points and grid nodes to achieve the best accuracy with efficiency. Nonetheless, the convergence of both the MPM and GMPM was illustrated, both in 1D and 2D situations.

References

Atluri, S. N.; Zhu, T. (1998): A new meshless local Petrov-Galerkin approach in computational mechanics. *Computational Mechanics*, vol. 22, pp. 117–127.

Bardenhagen, S. G. (2002): Energy conservation error in the material point method for solid mechanics. *J. of Computational Physics*, vol. 180, pp. 383–403. <http://dx.doi.org/10.1006/jcph.2002.7103>.

Bardenhagen, S. G.; Guilkey, J. E.; Roessig, K. M.; Brackbill, J. U.; Witzel, W. M.; Foster, J. C. (2001): An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES: Computer Modeling in Engineering & Sciences*, vol. 2, no. 4, pp. 509–522.

Bardenhagen, S. G.; Kober, E. M. (2004): The generalized interpolation material point method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 5, no. 6, pp. 477–495.

Chen, Z.; Brannon, R. (2002): An evaluation of the material point method. *SAND Report, Sandia National Laboratories*, pp. 1–46. SAND2002-0482.

Daphalapurkar, N. P.; Lu, H.; Coker, D.; Komanduri, R. (2007): Simulation of dynamics crack growth using the generalized interpolation material point (GIMP) method. *Int. J. of Fracture*, vol. 143, pp. 79–102. <http://dx.doi.org/10.1007/s10704-007-9051-z>.

Eringen, A. C. (1967): *Mechanics of Continua*. John Wiley and Sons, Inc., New York, USA.

Guilkey, J. E.; Weiss, J. A. (2003): Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method. *Int. J. for Num. Methods in Eng.*, vol. 57, pp. 1323–1338. <http://dx.doi.org/10.1002/nme.729>.

Guo, Y. J.; Nairn, J. A. (2006): Three-dimensional dynamic fracture analysis using the

material point method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 16, no. 3, pp. 141–155.

Ma, J.; Liu, Y.; Lu, H.; Komanduri, R. (2006): Multiscale simulation of nanoindentation using the generalized interpolation material point (GIMP) method, dislocation dynamics (DD) and molecular dynamics (MD). *CMES: Computer Modeling in Engineering & Sciences*, vol. 16, no. 1, pp. 41–55.

Ma, J.; Lu, H.; Komanduri, R. (2006): Structured mesh refinement in generalized interpolation material point (GIMP) method for simulation of dynamic problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, no. 3, pp. 213–227.

Ma, J.; Lu, H.; Wang, B.; Hornung, R.; Wissink, A.; Komanduri, R. (2006): Multiscale simulation using generalized interpolation material point (GIMP) method and molecular dynamics (MD). *CMES: Computer Modeling in Engineering & Sciences*, vol. 14, no. 2, pp. 101–117.

Ma, J.; Lu, H.; Wang, B.; Roy, S.; Hornung, R.; Wissink, A.; Komanduri, R. (2005): Multiscale simulations using generalized interpolation material point (GIMP) method and SAMRAI parallel processing. *CMES: Computer Modeling in Engineering & Sciences*, vol. 8, no. 2, pp. 135–152.

Malvern, L. E. (1969): *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Englewood Cliffs, USA.

Meirovitch, L. (1967): *Analytical Methods in Vibrations*. Macmillan, New York.

Pedroso, D. M.; Sheng, D.; Sloan, S. W. (2008): Stress update algorithm for elastoplastic models with nonconvex yield surfaces. *Int. J. for Num. Methods in Eng.* (in press).

Shen, L.; Chen, Z. (2005): A silent boundary scheme with the material point method for dynamic analyses. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, no. 3, pp. 305–320.

Sloan, S. W. (1987): Substepping schemes for the numerical integration of elastoplastic stress-strain relations. *Int. J. for Num. Methods in Eng.*, vol. 24, pp. 893–911.

Sloan, S. W.; Abbo, A. J.; Sheng, D. (2001): Refined explicit integration of elastoplastic models with automatic error control (Erratum: 19(5/6), 594-594, 2002). *Eng. Computations*, vol. 18, no. 1/2, pp. 121–154.

Sloan, S. W.; Booker, J. R. (1992): Integration of Tresca and Mohr-Coulomb constitutive relations in plane strain elastoplasticity. *Int. J. for Num. Methods in Eng.*, vol. 33, pp. 163–196.

Sulsky, D.; Chen, Z.; Schreyer, H. L. (1994): A particle method for history-dependent materials. *Comp. Methods in App. Mech. and Eng.*, vol. 118, pp. 179–196.

Sulsky, D.; Schreyer, H. L. (1996): Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems. *Comp. Methods in App. Mech. and Eng.*, vol. 139, pp. 409–429.

Sulsky, D.; Schreyer, H. L.; Peterson, K.; Kwok, R.; Coon, M. (2007): Using the material-point method to model sea ice dynamics. *J. of Geophysical Research*, vol. 112, pp. 1–18. <http://dx.doi.org/10.1029/2005JC003329>.

Sulsky, D.; Zhou, S. J.; Schreyer, H. (1995): Application of a particle-in-cell method to solid mechanics. *Computer Physics Comm.*, vol. 87, pp. 236–252.

Wallstedt, P. C.; Guilkey, J. E. (2007): Improved velocity projection for the material point method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 19, no. 3, pp. 223–232.

Wieckowski, Z.; Youn, S. K.; Yeon, J. H. (1999): A particle-in-cell solution to the silo discharging problem. *Int. J. for Num. Methods in Eng.*, vol. 45, pp. 1203–1225.