International Journal of Impact Engineering 36 (2009) 272–282

Contents lists available at ScienceDirect

International Journal of Impact Engineering

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

Comparison study of MPM and SPH in modeling hypervelocity impact problems[☆]

S. Ma, X. Zhang[*], X.M. Qiu

School of Aerospace, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 28 June 2007
Received in revised form 28 May 2008
Accepted 2 July 2008
Available online 10 July 2008

Keywords:
Meshfree
Smoothed particle hydrodynamics
Material point method
Hypervelocity impact

ABSTRACT

Due to the high nonlinearities and extreme large deformation, the hypervelocity impact simulation is a challenging task for numerical methods. Meshfree particle methods, such as the smoothed particle hydrodynamics (SPH) and material point method (MPM), are promising for the simulation of hypervelocity impact problems. In this paper, the material point method is applied to the simulation of hypervelocity impact problems, and a three-dimensional MPM computer code, MPM3D, is developed. The Johnson–Cook material model and Mie–Grüneisen equation of state are implemented. Furthermore, the basic formulations of MPM are compared with SPH, and their performances are compared numerically by using MPM3D and LS-DYNA SPH module.

This study shows that the material point method is an efficient and promising method for simulating the hypervelocity impact problems. MPM possesses many prominent features. The formulation of MPM is simple and similar to the traditional finite element method (FEM). Spatial derivatives are calculated based on a regular computational grid in MPM, so that the time consuming neighbor searching, which is compulsory in most meshfree methods, is not required. The approximation of field variables and their spatial derivatives is efficiently evaluated using the information of only 8 grid nodes in three-dimensional problem, and the shape functions exactly satisfy the constant and linear consistency. The boundary conditions can be applied in MPM as easily as in FEM, and contact algorithm can be efficiently implemented whose cost is linear in the number of bodies. Because the same regular computational grid can be used in all time steps, the time step size keeps constant in MPM simulation.

1. Introduction

The numerical simulations of hypervelocity impact problems are of great interest in many engineering applications, such as shield design for spacecraft protection. Because extremely large deformations and material fracture are involved in the process of hypervelocity impact, they are challenging tasks for computer simulation.

There are two fundamental frames for describing the motion of material: the Eulerian description and the Lagrangian description. In the Lagrangian description, the mesh is embedded in and deforms with the material domain. It presents no convective effects so that the boundary conditions at free surfaces, moving boundaries and material interfaces are automatically imposed. It is ideal for history-dependent material. However, if the deformation is very large, as in hypervelocity impact and metal forming, mesh distortion and element entanglement become the significant limitations of the Lagrangian description. Contrary to the Lagrangian description, material flows through a grid fixed in space in the Eulerian description. It completely avoids element distortions, but there are still difficulties in tracking the deformation history of material and dealing with the material interfaces. The dissipation problem associated with mass flux between adjacent elements also arises in the Eulerian method. There are also some mixed methods to strengthen the advantages of these two descriptions and to avoid their disadvantages, such as the arbitrary Lagrangian–Eulerian (ALE) [1]. Unfortunately, the convection term still exists in the ALE formulation, which may cause numerical difficulties. In addition, it still remains a challenging task to design an efficient and effective mesh moving algorithm to maintain mesh regularity for three-dimensional complicated material domain.

Recently developed meshfree methods [2–5] use a set of discrete points to construct trial functions, so that the difficulties associated with mesh distortion can be avoided or alleviated. Nevertheless, most of the meshfree methods suffer from higher computational cost and the accuracy of some meshfree methods is still dependent on the node regularities to some extent. Therefore, only a few of them perform well in hypervelocity impact problems,

[☆] Supported by National Natural Science Foundation of China (10472052) and the Science Foundation of Computational Physics, IAPCM, China.

[*] Corresponding author. Tel./fax: +86 10 62782078.
E-mail address: xzhang@tsinghua.edu.cn (X. Zhang).

0734-743X/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijimpeng.2008.07.001

such as the smoothed particle hydrodynamics (SPH) [6–8], material point method (MPM) [9,10], hybrid particle-element method [11–13], conversion of distorted elements into particles method [14–16].

Meshfree methods and particle methods can be classified into the same group of methods. Particle methods in nature are well suited for modeling extremely large deformation problems with failure and fracture, so they are superior to the Lagrangian grid-based methods in the simulation of hypervelocity impact problems.

Smoothed particle hydrodynamics (SPH) [6–8] is one of the earliest meshfree Lagrangian particle methods. SPH was first proposed by Lucy [17] and Gingold and Monaghan [18] in 1977 to solve astrophysical problems in three-dimensional open space and has been extensively studied and extended to dynamics response with material strength as well as dynamic fluid flows with large deformations. SPH and its improved versions have been successfully applied to the hypervelocity impact simulations and become one of the most popular and powerful meshfree methods in this area. Because of its good performance, several commercial softwares, such as AUTODYN [19], PAM-SHOCK [20] and LS-DYNA [21], have incorporated SPH into their solvers.

Material point method (MPM) [9,10], which is also a particle method, is an extension of the FLIP particle-in-cell [22] method to solid mechanics. In MPM, material domain is discretized by a group of points, termed as particles or material points. These Lagrangian particles carry all material information and track the deformation history. The momentum equations are solved on a predefined background grid, which can be fixed in space or arbitrarily defined, and provide a Eulerian description of the material domain. With characteristic of both Lagrangian description and Eulerian description, MPM is a good choice for modeling hypervelocity impact problems which involve extreme large deformation.

Although MPM is an extension of FLIP particle-in-cell method, it is closely related to the Lagrangian finite element method (FEM). MPM can be viewed as a special Lagrangian FEM in which the particles rather than the Gauss points serve as the quadrature points. For small deformation problems, the MPM is equivalent to the FEM using Gauss points at the same locations as those of material points in each cell. Bear this in mind, Zhang et al. [23] proposed an explicit material point finite element method, which provides an uniform formulation for the MPM and FEM, and extended it to the hypervelocity impact simulation.

In this paper, the material point method is applied to the simulation of hypervelocity impact simulation, and a three-dimensional material point method code, MPM3D, is developed with FORTRAN 90 to simulate various hypervelocity impact problems. The Johnson–Cook material model and Mie–Grüneisen equation of state are implemented. The basic formulation and features of MPM are compared with SPH from following aspects: neighbor searching, approximation functions, consistency of shape functions, tensile instability, time integration, boundary conditions and contact algorithm. The performance of MPM is compared numerically with SPH by using the MPM3D and LS-DYNA SPH module.

The remaining part of the paper is organized as follows. The governing equations are briefly summarized in Section 2. The basic formulations of MPM and SPH in modeling hypervelocity problems are presented in Sections 3 and 4, respectively. The basic formulation and features of MPM and SPH are compared in detail in Section 5, and their performances are numerically investigated in Section 6. Section 7 draws some conclusions.

2. Governing equations

In the updated Lagrangian description, the material is governed by the momentum equations [21]

$$\sigma_{ij,j} + \rho f_i = \rho \ddot{u}_i \ \forall x_i \in V \tag{1}$$

subject to the traction boundary conditions

$$\sigma_{ij} n_j = t_i(t) \ \forall x_i \in \Gamma_t \tag{2}$$

and the displacement boundary conditions

$$u_i(X_\alpha, t) = d_i(t) \ \forall x_i \in \Gamma_d \tag{3}$$

where V is the current material domain, Γ_t and Γ_d are, respectively, the boundary portions of V prescribed with traction and displacement, σ_{ij} is the Cauchy stress, ρ is the current density, f_i is the body force density, $\ddot{u}_i$ is the acceleration, the comma denotes covariant differentiation and n_j is the unit outward normal to the boundary.

The conservation equation of mass is stated as

$$\dot{\rho} + \rho v_{i,i} = 0 \tag{4}$$

The energy equation is given by

$$\dot{E} = J\sigma_{ij}\dot{\varepsilon}_{ij} = Js_{ij}\dot{\varepsilon}_{ij} - Jp\dot{\varepsilon}_{kk} \tag{5}$$

where J is the determinant of the deformation gradient matrix $F_{ij} = \partial x_i / \partial X_j$, E is the energy per unit initial volume. $\dot{\varepsilon}_{ij}$ is the strain rate, s_{ij} and p represent the deviatoric stresses and pressure, i.e. $s_{ij} = \sigma_{ij} + p\delta_{ij}$.

The stresses at time t^{k+1} can be obtained by

$$\sigma_{ij}^{k+1} = \sigma_{ij}^{k} + \dot{\sigma}_{ij}^{k}\Delta t \tag{6}$$

where $\dot{\sigma}_{ij}$ is the material time derivative of the stress, and given by

$$\dot{\sigma}_{ij} = \sigma_{ij}^{\nabla} + \sigma_{jl}\omega_{il} + \sigma_{il}\omega_{jl} \tag{7}$$

in which $\omega_{ij} = (1/2)(\dot{x}_{i,j} - \dot{x}_{j,i})$ is the spin tensor, and σ_{ij}^{∇} is the Jaumann stress rate and determined from the strain rate $\dot{\varepsilon}_{ij} = (1/2)(\dot{x}_{i,j} + \dot{x}_{j,i})$ by a constitution model. Johnson–Cook material model [24] is implemented in MPM3D code, in which the equivalent tensile flow stress is expressed as

$$\sigma_y = (A + B\varepsilon^n)\left(1 + C\ln\dot{\varepsilon}^*\right)\left(1 - T^{*m}\right) \tag{8}$$

where ε is equivalent plastic strain, $\dot{\varepsilon}^* = \dot{\varepsilon}/\dot{\varepsilon}_0$ is the dimensionless plastic strain rate for $\dot{\varepsilon}_0 = 1.0\ \text{s}^{-1}$ and $T^* = (T - T_{\text{room}})/(T_{\text{melt}} - T_{\text{room}}) \in [0, 1]$ is the homologous temperature. A, B, n, C and m are the material constants. Temperature is computed by assuming that the 90% of the energy dissipated due to plastic work goes to raise the temperature.

The pressure is updated by an equation of state. The Mie–Grüneisen equation of state [25] is implemented in MPM3D, in which the pressure is updated by

$$p = p_H\left(1 - \frac{\gamma\mu}{2}\right) + \gamma\rho E \tag{9}$$

$$p_H = \begin{cases} a_0\mu + b_0\mu^2 + c_0\mu^3 & \mu > 0 \\ a_0\mu & \mu < 0 \end{cases} \tag{10}$$

The subscript H refers to the Hugoniot curve, $\mu = \rho/\rho_0 - 1$ is used to represent the compression and γ is the Grüneisen parameter. The constants a_0, b_0, and c_0 in Eq. (10) are related to C and S in shock wave velocity and particle velocity relation $U_s = C + SU_p$ through a Taylor's expansion of the Hugoniot curve. $a_0 = \rho_0 C^2, b_0 = a_0[1 + 2(S-1)], c_0 = a_0[2(S-1) + 3(S-1)^2]$.

3. Material point method

The material domain is discretized by a finite number of particles in MPM, as illustrated in Fig. 1. Each particle carries all of

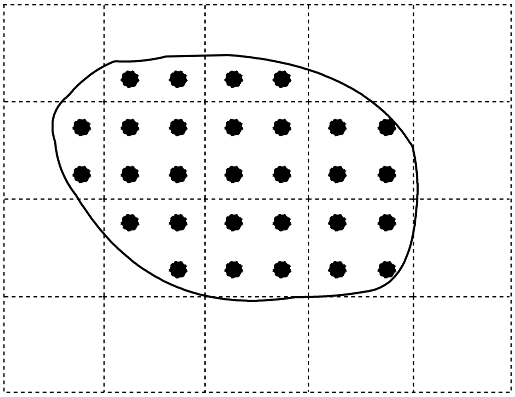

Fig. 1. Solid line denotes the boundary of material domain, solid dot denotes material point and dash lines are background computational mesh.

material variables, such as the mass, position, velocity, strain and stress. Instead of evaluating the momentum equations on particles as in the SPH, the momentum equations are evaluated on the predefined background grid nodes. The solution process of MPM is divided into two phases. In the first phase of solution, particles are rigidly attached to the background grid and they deform with the grid. After obtaining the kinematic solution on the grid nodes, they are mapped back to the particles to update their positions and velocities. The deformed grid is discarded in the subsequent time step and a new regular grid is used to avoid mesh distortion. Generally speaking, the same fixed regular grid can be used in all time steps.

The fundamental formulation of MPM can be obtained from the weak form of momentum equations and traction boundary condition as:

$$\delta\Pi = \int_V \rho\ddot{u}_i\delta u_i \mathrm{d}V + \int_V \rho\sigma^s_{ij}\delta u_{i,j}\mathrm{d}V - \int_V \rho f_i\delta u_i \mathrm{d}V - \int_{\Gamma_t} t_i\delta u_i \mathrm{d}\Gamma = 0 \tag{11}$$

where $\sigma^s_{ij} = \sigma_{ij}/\rho$ is the specific stress and δu_i is the virtual displacement. The displacement boundary conditions have been assumed to be satisfied a priori.

Since the movement of particles represents the deformation of physical domain and mass is carried by particles, the mass conservation is automatically satisfied in MPM.

Because the particles are rigidly attached to the computational grid, the particle displacement u_{pi} and its derivatives $u_{pi,j}$ can be obtained by mapping their grid point values u_{gi} and $u_{gi,j}$ to the particle using the standard finite element shape functions of the grid, namely

$$u_{pi} = \sum_{g=1}^{8} N_{pg}u_{gi} \tag{12}$$

$$u_{pi,j} = \sum_{g=1}^{8} N_{pg,j}u_{gi} \tag{13}$$

where $N_{pg} = N_g(x_{pi})$. In this paper, 8-point hexahedron grid is used as the background grid so that the shape function is given by

$$N_g = \frac{1}{8}(1+\xi\xi_g)\left(1+\eta\eta_g\right)(1+\zeta\zeta_g), \quad g = 1,2,\cdots 8 \tag{14}$$

if the particle (ξ, η, ζ) is inside the hexahedron, where ξ_g, η_g and ζ_g take on their nodal values (±1, ±1, ±1) at the grid node g. If the particle is outside the hexahedron, $N_g = 0$.

The material mass is lumped at particles, hence the density ρ at point x_i can be approximated as

$$\rho(x_i) = \sum_{p=1}^{n_p} M_p\delta(x_i - x_{pi}) \tag{15}$$

where x_{pi} denotes the coordinate of pth particle in ith direction.

Substituting Eqs. (12), (13) and (15) into the weak form (11), and invoking the arbitrariness of δu_{hi} yields

$$\dot{p}_{hi} = f^{\mathrm{int}}_{hi} + f^{\mathrm{ext}}_{hi}, \quad h = 1,2,\cdots,n_g \tag{16}$$

where n_g is the number of grid nodes,

$$p_{hi} = \sum_{g=1}^{n_g} m_{hg}\dot{u}_{gi} \tag{17}$$

is the momentum of hth grid point in the i direction, $m_{hg} = \sum_{p=1}^{n_p} M_pN_{ph}N_{pg}$ is the mass matrix. If the lumped mass matrix $m_h = \sum_{p=1}^{n_p} M_pN_{ph}$ is used, the momentum p_{hi} can be simplified as

$$p_{hi} = m_h\dot{u}_{hi} \tag{18}$$

Moreover,

$$f^{\mathrm{int}}_{hi} = -\sum_{p=1}^{n_p} V_pN_{ph,j}\sigma_{pij} \tag{19}$$

and

$$f^{\mathrm{ext}}_{hi} = \sum_{p=1}^{n_p} M_pN_{ph}f_{pi} + \sum_{p=1}^{n_p} M_pN_{ph}t^s_{pi}h^{-1} \tag{20}$$

are the internal force and external force, $t^s_{pi} = t_{pi}/\rho$ is the specific traction, h is the thickness of the boundary layer, $\sigma_{pij} = \sigma_{ij}(x_p)$ and $f_{pi} = f_i(x_p)$.

The Lagrangian formulation is used in MPM so that the acceleration in Eq. (16) does not contain the convection term which can cause significant numerical error in purely Eulerian approaches.

Using the central difference scheme, the momentum equations on gride nodes can be integrated as

$$p^{k+1}_{hi} = p^k_{hi} + \left(f^{\mathrm{int}}_{hi} + f^{\mathrm{ext}}_{hi}\right)\Delta t \tag{21}$$

where the superscripts "k" and "$k+1$" denote the values at time t^k and t^{k+1}, respectively. The particle velocities and positions are updated by

$$x^{k+1}_{pi} = x^k_{pi} + \sum_{h=1}^{n_g} \frac{p^{k+1}_{hi}}{m^k_h}N^k_{ph}\Delta t \tag{22}$$

$$v^{k+1}_{pi} = v^k_{pi} + \sum_{h=1}^{n_g} \frac{f^k_{hi}}{m^k_h}N^k_{ph}\Delta t \tag{23}$$

where $f^k_{hi} = f^{\mathrm{int}}_{hi} + f^{\mathrm{ext}}_{hi}$.

In order to update stress, the strain rate $\dot{\varepsilon}_{pij}$ is required to be calculated at the particles using updated velocity as

$$\dot{\varepsilon}_{pij} = \frac{1}{2}\sum_{h=1}^{8}\left(v^{k+1}_{hi}N^k_{ph,j} + v^{k+1}_{hj}N^k_{ph,i}\right) \tag{24}$$

where velocities at grid nodes are mapped back from the updated particle velocities by $v_{hi}^{k+1} = \sum_{p=1}^{n_p} M_p v_{pi}^{k+1} N_{ph}^{k} / m_h^k$.

In MPM, there is no interaction between material particles separated by grid cells. Fracture can be simulated roughly without special treatment [23]. In order to obtain more precise simulation, fracture model can be added into the code. One of the failure models in LS-DYNA [21] was implemented in the MPM3D in which particles will fail if tensile pressure or plastic strain exceed the limit, i.e. $p < p_{\min}$ or $\varepsilon^p > \varepsilon^p_{\max}$. Once failure has occurred, pressure may never be negative and the components of deviatoric stress are set zero for all the time so that the failed particle can only carry compression loads.

4. Smoothed particle hydrodynamics

Similar to MPM the material domain V in SPH is also represented by a finite number of particles that carry individual mass m_J and occupy individual space $\Delta V_J = m_J/\rho_J$, in which ρ_J is the density of Jth particle. The function $u_i(x)$ is approximated by the kernel approximation (integral function representation) as

$$u_i(x) \simeq u_i^h(x) = \int_V u_i(\bar{x}) W(x - \bar{x}, h) \mathrm{d}V_{\bar{x}} \tag{25}$$

where W is the kernel function, which should be normalized and compactly supported, h is the smoothing length defining the influence domain of the kernel function W. W should become Delta function when h approaches zero.

The kernel approximation for the divergence $u_{i,i}(x)$ is given by

$$u_{i,i}(x) \simeq u_{i,i}^h(x) = \int_V u_{i,\bar{i}}(\bar{x}) W(x - \bar{x}, h) \mathrm{d}V_{\bar{x}} \tag{26}$$

where the subscript '$\bar{i}$', denotes differentiation with respect to $\bar{x}_i$, namely, $u_{i,\bar{i}} = \sum_{i=1}^{3} \partial u_i / \partial \bar{x}_i$. Integrating by parts and using the divergence theorem, Eq. (26) can be reduced to

$$u_{i,i}^h(x) = -\int_V u_i(\bar{x}) W_{,\bar{i}}(x - \bar{x}, h) \mathrm{d}V_{\bar{x}} = \int_V u_i(\bar{x}) W_{,i}(x - \bar{x}, h) \mathrm{d}V_{\bar{x}} \tag{27}$$

for those points whose support domain is inside the problem domain. In Eq. (27), $W_{,\bar{i}}(x - \bar{x}, h) = -W_{,i}(x - \bar{x}, h)$.

After discretizing the material domain V by a set of particles, Eqs. (25) and (27) can be evaluated by

$$u_i^h(x) = \sum_{J=1}^{N} W(x - x_J, h) \frac{m_J}{\rho_J} u_{iJ} \tag{28}$$

$$u_{i,i}^h(x) = \sum_{J=1}^{N} W_{,i}(x - x_J, h) \frac{m_J}{\rho_J} u_{iJ} \tag{29}$$

where the subscript J denotes the quantities associated with the Jth particle, $u_{iJ} = u_i(x_J)$ is the displacement of the Jth particle in x_i direction, and N is the number of particles within the support domain of the point x.

According to Eqs. (28) and (29), the function $u_i(x)$ and its divergence $u_{i,i}(x)$ can be approximated at Ith particle as

$$u_i^h(x_I) = \sum_{J=1}^{N} W_{IJ} \frac{m_J}{\rho_J} u_{iJ} \tag{30}$$

$$u_{i,i}^h(x_I) = \sum_{J=1}^{N} W_{IJ,i} \frac{m_J}{\rho_J} u_{iJ} \tag{31}$$

where

$$W_{IJ} = W(x_I - x_J) \tag{32}$$

$$W_{IJ,i} = W_{,i}(x - x_J)|_{x = x_I} \tag{33}$$

The derivatives of the field functions can also be approximated as [6]:

$$u_{i,i}^h(x_I) = \frac{1}{\rho_I} \left[\sum_{J=1}^{N} m_J (u_{iJ} - u_{iI}) W_{IJ,i} \right] \tag{34}$$

$$u_{i,i}^h(x_I) = \rho_I \left[\sum_{J=1}^{N} m_J \left(\frac{u_{iJ}}{\rho_J^2} + \frac{u_{iI}}{\rho_I^2} \right) W_{IJ,i} \right] \tag{35}$$

In the above two equations, the field functions appear in the term of paired particles.

In SPH method, the momentum Eq. (1), continuity Eq. (4) and energy Eq. (5) are evaluated on particles. Applying SPH approximation to these equations will result in the SPH formulations [6–8]:

$$\ddot{u}_{iI} = \sum_{J=1}^{N} m_J \left(\frac{\sigma_{Iij}}{\rho_I^2} + \frac{\sigma_{Jij}}{\rho_J^2} \right) W_{IJ,i} \tag{36}$$

$$\dot{\rho}_I = \sum_{J=1}^{N} m_J (v_{iI} - v_{iJ}) W_{IJ,i} \tag{37}$$

$$\dot{e}_I = \frac{1}{2} \sum_{J=1}^{N} m_J \left(\frac{p_I}{\rho_I^2} + \frac{p_J}{\rho_J^2} \right) (v_{iI} - v_{iJ}) W_{IJ,i} + \frac{1}{\rho_I} s_{Iij} \dot{\varepsilon}_{Iij} \tag{38}$$

for body force free case, where $\sigma_{Iij} = \sigma_{ij}(x_I), s_{Iij} = s_{Iij}(x_I), \varepsilon_{Iij} = \varepsilon_{ij}(x_I)$, and $e = E/\rho J$ is the internal energy per unit mass.

Libersky [25,26] extended SPH to treat the dynamics response of solids in 1990. Because of the Lagrangian and particle nature in SPH, it is easily to handle large deformations and is well suited in the simulation of hypervelocity impact problems [27,28].

After SPH was extended to solid dynamics, tensile instability [29,30] was exposed. It is more frequently encountered in solid because there are usually no tensile stress in astrophysical or general fluid applications. If the materials are under tensile stress in SPH calculation, particles are tend to clumped together and voids may be formed. Sometimes, this instability could cause numerical fracture which is unreal. Several remedies have been proposed to avoid the tensile instability.

Nowadays, SPH and its improved versions are among the most popular and powerful methods in hypervelocity impact simulations.

5. Comparison between MPM and SPH

SPH and MPM are both meshfree particle methods, and are both well suited for the simulation of hypervelocity impact problems. However, they possess different advantages and disadvantages. In this section, MPM and SPH are compared in detail.

5.1. Neighbor searching

All numerical methods for solving partial differential equations (PDEs) require a nodal connectivity to evaluate the spatial derivatives of the field variables. The nodal connectivity is constructed based on finite elements in FEM, and constructed by searching neighbor nodes in meshfree methods. Although the term 'meshless methods' has been widely used to name the methods in which the approximation functions are constructed based on a collection of discrete points instead of on a predefined mesh/grid, they are not truly meshless. In these methods, nodal connectivity is still essential for calculating the spatial derivatives of the approximation functions. However, the nodal connectivity in these methods might be changed at each time step, rather than fixed as in the grid-based method like the finite element method. Therefore, the term 'meshfree methods' is more appropriate to name these methods.

Similar to the most meshfree methods, the approximation of field variables and their spatial derivatives in SPH is calculated based on a current local set of arbitrarily distributed particles, see Eqs. (28) and (29). Therefore, it is required in SPH to search the nearest neighbor particles of a particle to define nodal connectivity. Neighbor particles for a given particle may change with time in hypervelocity impact problems due to the large deformation, so that the neighbor searching has to be carried out in every time step. Several efficient search schemes, such as tree-like method [31–33] and box algorithm [25,34,35], are available, usually the complexity of searching algorithms can reach order of $O(N \log N)$. Neighbor searching in SPH is still time consuming, especially for larger scale problems.

On the contrary, the nodal connectivity in MPM is defined based on the computational grid. The same regular computational grid is usually used in all time steps in MPM, so that the nodal connectivity does not change with time. It can be seen from Eq. (13) that the derivatives of the displacement of the pth particle are calculated from their nodal values at the grid nodes of the hexahedron in which the particle is located. It is trivial to determine the index of hexahedron and its grid nodes. The index of the hexahedron in which the pth particle is located can be obtained efficiently from the following FORTRAN statements

```
NumCellx = int((SpanX(2)-SpanX(1))/DCell + 0.5)
NumCelly = int((SpanY(2)-SpanY(1))/DCell + 0.5)
NumCellxy = NumCellx * NumCelly
ix = ceiling((xp-SpanX(1))/DCell)
iy = ceiling((yp-SpanY(1))/DCell)
iz = ceiling((zp-SpanZ(1))/DCell)
InWhichCell = (iz - 1) * NumCellxy
+ (iy - 1) * NumCellx + ix
```

where `xp`, `yp` and `zp` are the coordinates of the pth particle, `DCell` is the size of the hexahedron, `SpanX`, `SpanY` and `SpanZ` are the minimum and maximum $x-$, $y-$ and $z-$ coordinates of the computational grid, `InWhichCell` is the index of the hexahedron.

After obtaining the index of the hexahedron, the grid nodes used to calculate the derivatives of the field variables can be retrieved from a array without any extra costs. Therefore, the computational cost required in SPH to define the nodal connectivity is much higher than that required in MPM.

5.2. Approximation functions

One of the most significant advantages of SPH is that it does not require a pre-defined mesh or grid for the purpose of both approximation construction and integration calculation. The continuous integral representations of a function and its derivatives, Eqs. (25) and (27), are converted to the discretized summations over all particles within the support domain of the point x, as shown in Eqs. (28) and (29) or (30) and (31). To ensure the accuracy and the numerical stability, the support domain should be sufficiently large, so that sufficient particles (sampling points for integration) are included in the summation.

Compared with SPH, the traditional finite element approximation is used in MPM to evaluate the approximation of field variables and their derivatives. As shown in Eqs. (12) and (13), the approximation functions are constructed on grid nodes and particle values are calculated by using the information on the grid nodes of the hexahedron in which the particle is located. The summation in Eqs. (12) and (13) is taken over 8 grid nodes, while the summation in Eqs. (30) and (31) has to be taken over N neighbors of a particle. In three-dimensional hypervelocity impact analysis, N is usually greater than 8. As a consequence, the computational cost required to evaluate the approximation of field functions and their derivatives in SPH is higher than that in MPM.

Bardenhagen and Kober [36] proposed a generalization of MPM named generalized interpolation material point (GIMP) method, in which the approximation function could have continuous derivative (in contrast to C^0 continuous finite element shape function used in MPM). This method increases the computational cost of MPM, but benefits much in precision and stability as discussed in the following sections. Fig. 2 illustrates the typical approximation functions of SPH, MPM and GIMP.

Because the same regular computational grid can be used in all time steps, the mesh distortion and element entanglement associated with the traditional finite element method are avoided in MPM.

It worth noting that density approximation

$$\rho(x_i) = \sum_{J=1}^{N} m_J W_J(x_i) \tag{39}$$

is usually preferred in SPH simulation [27]. In this way, the density field is smoothed by the kernel function. The particle in both SPH and MPM carries mass. The difference is that the kernel function $W_J(x_i)$ in Eq. (39) is replaced by δ function in MPM as shown in Eq. (15).

5.3. Consistency of shape functions

For a numerical method to converge, the shape function employed must satisfy a required degree of consistency. Unfortunately, due to the unbalanced particles contributing to the summation in Eq. (28), the constant consistency condition

$$\sum_{J=1}^{N} W(x - x_J, h)\frac{m_J}{\rho_J} = 1 \tag{40}$$

and the linear consistency condition

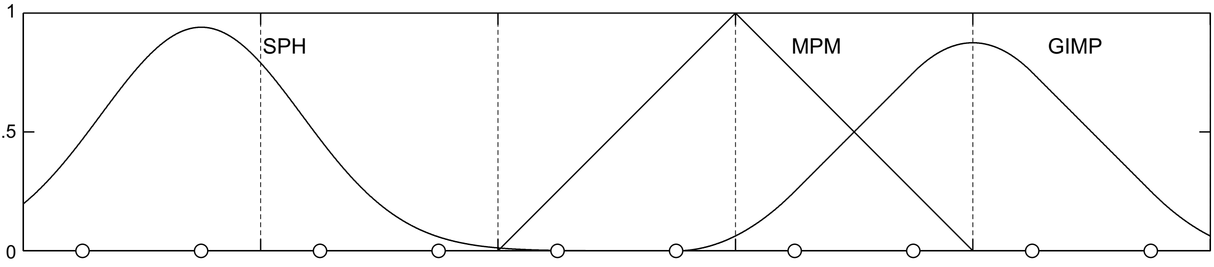

Fig. 2. Typical approximation functions of SPH, MPM and GIMP in one dimension. White circles indicate the particles.

$$\sum_{J=1}^{N}\left(x-x_J\right)W(x-x_J,h)\frac{m_J}{\rho_J}=0 \quad (41)$$

are not satisfied in SPH for the following two cases:

- Particles at or near the boundary of the problem domain so that the support domain intersects with the boundary, even for regular particle distribution.
- Irregularly distributed particles, even for the interior particles whose support domains are not truncated.

Although there are different ways [27,37] to restore the consistency condition, they require much more additional computational cost. Furthermore, they may lead to some problems in simulating hydrodynamic problems [7], such as causing negative density and negative energy that can result in a breakdown of the entire computation.

With the use of density approximation (39), the approximation function is reformed to Shepard function. The constant consistency is satisfied and simulation results can be improved, but the linear consistency condition remains unsatisfied under the two cases mentioned above.

The constant and linear consistency conditions are satisfied exactly in MPM because the traditional finite element shape functions are used. As for GIMP, the shaped function has similar characteristics as kernel function. Usually evenly spaced grid node is used, so the constant and linear consistency are satisfied in most of the computational domain except a thin layer near the boundary as discussed above.

5.4. Instability

SPH converts the integration over the support domain into a summation only over a finite number of particles. Similar to the node integration for the element-free Galerkin method (EFG) [38], insufficient sampling points for integration may result in numerical instability. When particles are under tensile stress state, the motion of the particles become unstable, which may lead to particle clumping or complete blowup in the computation [29]. Several means have been proposed to improve the tensile instability, such as the special smoothing functions [39], the corrective smoothed particle method (CSPH) [40], the artificial force [41] and stress points [42,43].

Different from SPH, the grid nodes in MPM serve as field nodes to construct the approximation functions of the field variables, whereas the material points serve as sampling points for integration. Usually, the number of material points is much greater than that of the grid nodes, so that the numerical instability arisen from insufficient sampling points for integration is avoided in MPM.

In some MPM simulations, numerical noises, especially in stress or velocity results, can be observed when the material points move across the cell boundaries during deformation. The noise was identified as cell crossing noise. This noise may be due to the abrupt change of node mass when particle cross the cell boundary or C^0 continuous feature of the finite element shape function. To suppress the noise, Bardenhagen and Kober [36] proposed a general mathematical framework named the generalized interpolation material point method (GIMP), which is a good alternative for those problems in which the cell crossing noise bothers much.

5.5. Time integration

Explicit integration is used in both SPH and MPM, so the critical time step which assure the stability can be obtained from the Courant–Friedrichs–Lewy (CFL) condition as

$$\Delta t_{\mathrm{cr}}=\frac{h}{c} \quad (42)$$

where c is the wave speed of material, h is a characteristic length of element. The critical time step is determined by the smallest smoothing length [42] in SPH, and determined by the background mesh size [44] in MPM. Since the same background mesh is used in all time steps, the constant time step can be used in the whole process of computation in MPM. However, the smoothing length h in SPH varies with particle distances which decrease with the material compression. Therefore, the critical time step may decrease significantly in the process of computation in SPH, especially in extremely large deformation problems. Consequently, SPH requires much more time steps than MPM for the simulation of the same hypervelocity impact problem.

5.6. Boundary conditions

It should be noted that Eq. (27) is not valid for points near or on the boundary because at which the residual boundary term

$$\int_S u_i(\overline{x})W(x-\overline{x},h)n_i\mathrm{d}S_{\overline{x}} \quad (43)$$

in the integrating by parts does not vanish. Therefore, the enforcement of boundary conditions is not trivial as that in the traditional finite element method. The early fluid dynamic uses of SPH either did not require boundary conditions or only required simple conditions. Meanwhile, the approximations in SPH do not have the property of strict interpolants so that they are not equal to

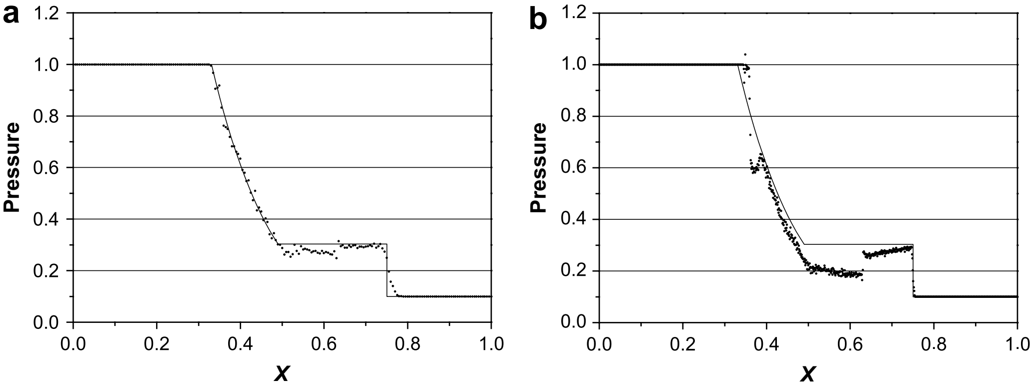

Fig. 3. Pressure profile of analytical solution (solid line) and MPM simulation (dots) at time $t = 0.143$. (a) 200 cells, (b) 1000 cells.

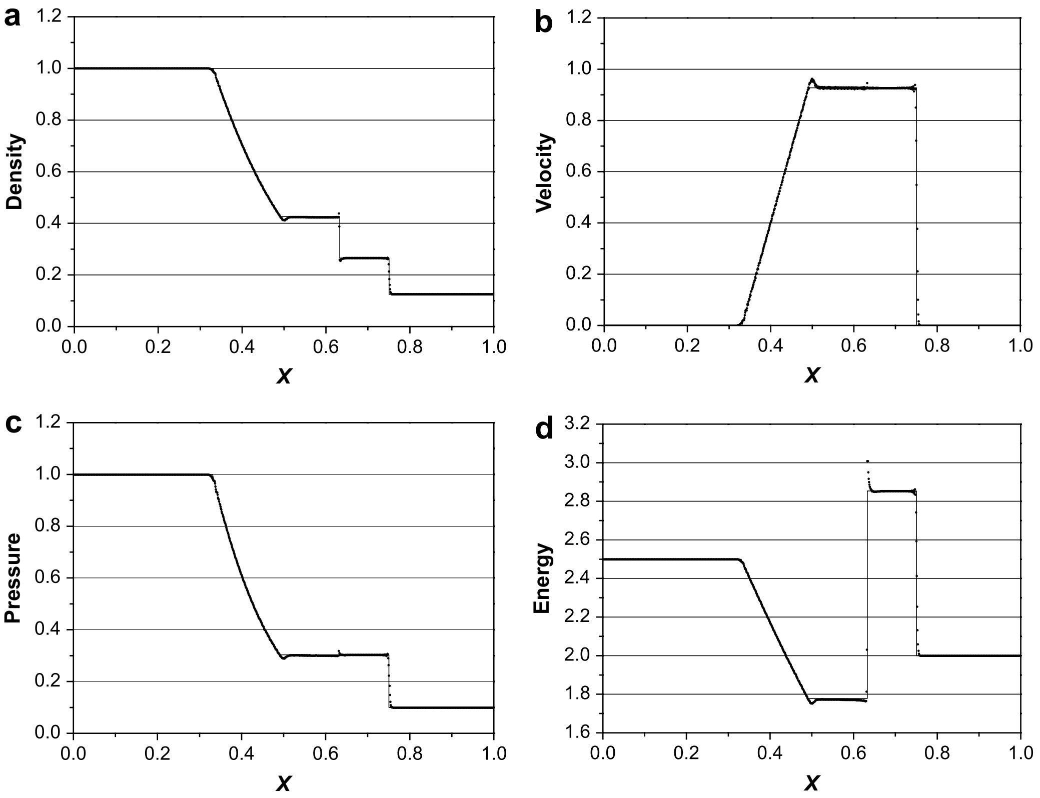

Fig. 4. Profiles of density, velocity, pressure and energy obtained by analytical solution (solid line) and GIMP simulation using 1000 cells (dots).

Table 1
Material properties

ρ (kg/m^3)	E (GPa)	ν	A (MPa)	B (MPa)	n	C
8930	117	0.35	157	425	1.0	0.0

the particle value of the dependent variable. This complicates the enforcement of the essential boundary conditions.

Recently some improvements have been proposed to treat the boundary conditions. Campbell [45] addressed the SPH boundary condition problem for equations involving gradients, and treated the boundary conditions by including the residual boundary term, Eq. (43), in the integration by parts when evaluating the original kernel integral involving gradients. An alternative way to treat the boundary conditions is to use virtual particles [46] or ghost particles [25,27,47].

The MPM formulation is derived from the weak form (11), hence the boundary conditions at the free surface are satisfied automatically without any special treatments. The implementation of traction boundary conditions requires choosing a suitable set of material points to represent the boundary layer, see Eq. (20). Because the standard finite element shape functions are used in MPM, the essential boundary conditions can be easily applied on the background grid nodes in the same way as that in the traditional finite element method. However, as for GIMP method, which has smoother shaped function and larger influence region for each grid node, enforcement of the essential boundary conditions also became a problem to be dealt with. Analogous to SPH, ghost grid nodes may be used to ensure exact satisfaction of essential boundary conditions.

5.7. Contact algorithm

SPH can be used in impact problems without any special treatments to contact boundary condition. Contact was handled automatically by treating particles from both contact body equally when finding neighbor particles. Since SPH does not require the velocity field to be single-valued, a degree of penetration and mixing may occur at the contact surface. To solve this problem, Campbell et al. [48] developed a particle-to-particle contact algorithm by using penalty method.

Unlike SPH, the velocity field is single-valued in MPM, so that the interpenetration of material is precluded without any iteration and no-slip contact condition between impinging bodies is automatically satisfied at no additional cost. To release the inherent no-slip contact constraint in MPM while avoiding interpenetration, contact/sliding/separation schemes were proposed [49–51]. The idea is that if the bodies are moving toward one another, the material points are moved in the usual center-of-mass velocity field

Table 2
Numerical results obtained by SPH and MPM in Taylor bar impact simulation

	L (mm)	D (mm)	W (mm)	$\overline{\Delta}$
Test	16.2	13.5	10.1	–
SPH1	15.4	15.6	9.9	0.075
SPH2	15.5	14.7	10.0	0.047
MPM	16.3	13.0	9.6	0.031
FEM	16.3	13.2	10.1	0.009

Table 3
Variation of time step size in Taylor bar impact simulation

	$\Delta t_{\max}$ (μs)	$\Delta t_{\min}$ (μs)	Steps
SPH1	0.120	0.118	672
SPH2	0.140	0.138	576
MPM	0.133	0.133	604
FEM	0.0398	0.0123	5486

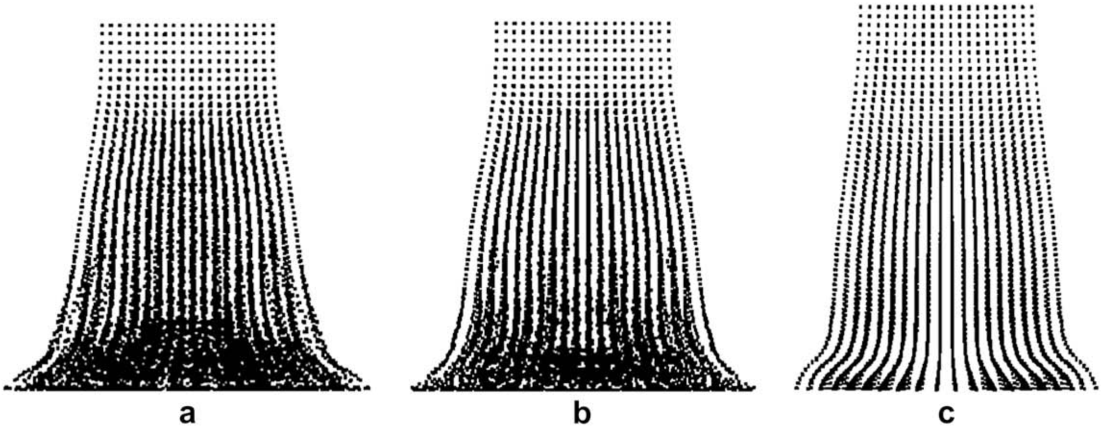

Fig. 5. Final configuration of the bar. (a) SPH1, (b) SPH2, (c) MPM.

which automatically enforces the no-penetration condition. If the bodies are moving away from one another, they move in their own velocity field which allows the separation to occur. Therefore, if the bodies are determined to be in contact, the equation of motion (16) is solved, where the internal force f_{hi}^{int} results from all material points even if they are from different bodies. If the bodies are releasing from one another, the equation of motion is solved for a separate body, namely

$$\dot{p}_{hi}^{I} = f_{hi}^{I,\text{int}} + f_{hi}^{I,\text{ext}}, \quad h = 1, 2, \cdots, n_g \tag{44}$$

where the internal force $f_{hi}^{I,int}$ results only from body *I*. The cost of these contact algorithms is linear in the number of bodies because it can be applied to all bodies with one sweep over the computational grid without iteration, so they are able to account for the interactions between a large number of bodies, such as grains [50,52].

6. Numerical examples

To compare the accuracy and efficiency of MPM and SPH, several numerical examples are presented. In the following examples, the SPH simulations are carried out by using LS-DYNA version 970 while the MPM simulations are carried out by using our MPM3D code.

6.1. Shock tube problem

Shock tube problem is often used to test the capability of a code on simulating compressible fluid. Sod's model problem [53] is regarded as a standard test. This problem consists of a shock tube where a diaphragm separates two regions which have different densities and pressures. The fluid in two regions are initially at rest. The initial values are: $\rho_1 = 1.0$, $p_1 = 1.0$, $\rho_2 = 0.125$, and $p_2 = 0.1$. At time $t > 0$ the diaphragm is broken. Then the shock and the contact interface travel at different speed. The results usually plotted at $t = 0.143$ when shock travels a distance of about 0.25.

SPH has been used to simulate shock tube problem in its early age [54]. Monaghan [55] and Inutsuka [56] improve the SPH shock simulation by introducing Riemann solver. Sigalotti et al. [57] capture the shock by using adaptive kernel estimation and get relatively good result. Usually SPH gets blurred or smooth shock profiles, but Monaghan [58] argued that real shock fronts is only a few molecular mean free path. No current method gives the width of a shock front accurately. Therefore it is more important to get pre- and post-shock values of physical variables as far as evaluating SPH is concerned.

The same argument holds for MPM too. York et al. [59] simulated shock tube problem by using MPM. Oscillation, which might be caused by cell crossing noise, was observed in their results. MPM3D can be used to simulate compressible fluid by simply ignoring deviatoric part of stress and strain and using perfect gas EOS to update pressure. For this one dimension problem, MPM3D is used by setting the solution variables constant in *y* and *z* directions. Two cases are analyzed, in which background meshes of 200 cells and 1000 cells are used, respectively. In both cases, four particles are initially placed in each cell for left high density region and two particles per cell for right low density region, so that 600 particles and 3000 particles are used, respectively. All data are plotted at grid

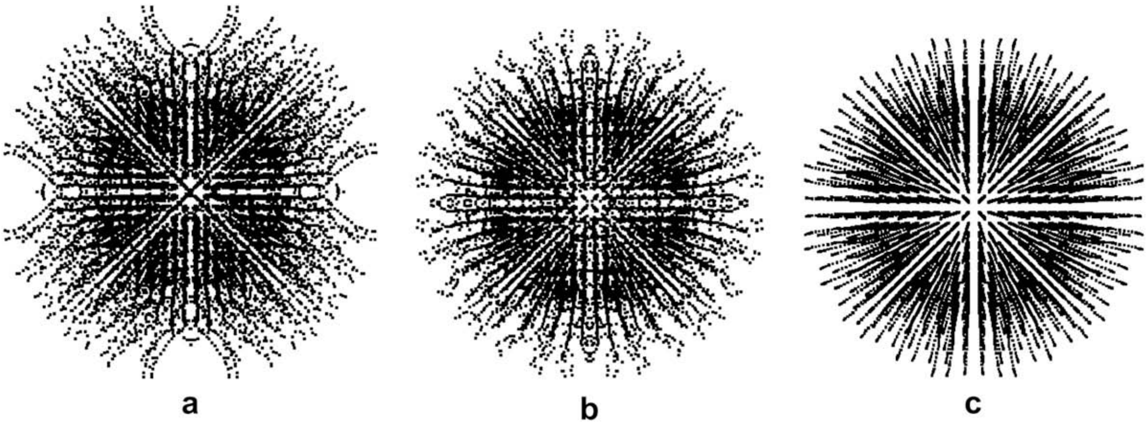

Fig. 6. Final configuration of the bar (top view). (a) SPH1, (b) SPH2, (c) MPM.

Table 4
Material properties for the aluminum

	ρ (kg/m^3)	E (GPa)	ν	σ_y (MPa)	E_t (MPa)	ε^p max	p min (MPa)
Sphere	2790	72.4	0.33	276	200	2.5	−5000
Target	2700	68.9	0.33	276	200	2.0	−4000

nodes. The pressure results of MPM with different meshs are illustrated in Fig. 3. Cell crossing noise could become an obstacle when trying to get better results by refining the mesh. Pressure profile result would deteriorate when finer mesh is used because there are more cell crossing. This problem can be well solved by using GIMP method as observed in Fig. 3. Interpolation function of GIMP is taken from Pan's article [52]. The profiles of density, velocity, pressure and specific internal energy are shown in Fig. 4 for background mesh of 1000 cells. They agree well with analytical results.

Artificial bulk viscosity is important in shock simulation. We adopted the same type viscosity with LS-DYNA [21]:

$$q = \begin{cases} c_0 \rho l_e^2 \left(\dot{\varepsilon}_{kk}\right)^2 - c_1 \rho l_e c \dot{\varepsilon}_{kk} & \dot{\varepsilon}_{kk} < 0 \\ 0 & \dot{\varepsilon}_{kk} \geq 0 \end{cases} \tag{45}$$

where l_e is characteristic length, $\dot{\varepsilon}_{kk}$ is bulk strain rate, c is sound speed, and c_0 and c_1 are dimensionless constants. The default value of c_0 and c_1 in LS-DYNA is 1.5 and 0.06. However, they are raised to 6.0 and 0.6, respectively, in this simulation to suppress the oscillation. The peak in energy profile at contact surface as shown in Fig. 4d might be caused by the artificial bulk viscosity.

Boundary condition can be applied to boundary grid nodes at two ends of shock tube directly in MPM. As for GIMP, we treat two layers of grid node as boundary nodes to ensure the exact satisfaction of boundary condition.

6.2. *Taylor bar problem*

Taylor bar problem, which is a cylinder metal bar normally impacted against a rigid wall, is often used to validate the constitutive models in codes. There are ample experimental data which can be referred to. In this example, SPH and MPM are used to simulate the impact of a copper cylinder to a rigid wall with impact velocity of 190 m/s to compare their accuracy and efficiency. In SPH simulations, constant associated with smoothing length was set 1.2 and 1.4, respectively. The value of 1.2 is the default value used in LS-DYNA, and larger value will increase the computational time but may improve the result with more neighbors for each particle. In the following tables and figures, these two cases were referred to as SPH1 and SPH2. This problem is also simulated by using an explicit finite element code with one point Gaussian quadrature developed by us for comparison.

The initial length of the cylinder is $L_0 = 25.4$ mm and the initial diameter is $D_0 = 7.6$ mm. The particles are evenly located with initial interval of 0.38 mm. A total of 21172 particles are used in the simulation. The Johnson–Cook model is used and material properties are taken from Ref. [24] as list in Table 1.

The simulation time is 80 μs, when the kinetic energy reaches zero. Table 2 compares the numerical results obtained by SPH and MPM with the experimental results, which shows that MPM possesses better accuracy than SPH in this example and SPH with larger smoothing length results in better accuracy. The average error $\overline{\Delta}$ introduced by Johnson [24] is defined as:

$$\overline{\Delta} = \frac{1}{3}\left(\frac{|\Delta L|}{L} + \frac{|\Delta D|}{D} + \frac{|\Delta W|}{W}\right) \tag{46}$$

Table 3 compares the maximum time step $\Delta t_{\max}$, the minimum time step $\Delta t_{\min}$, time steps used in SPH and MPM simulation. In this example, time step factor α is set 0.8 for all the case.

Figs. 5 and 6 compare the final configurations of the bar obtained by using the two methods. It can be observed from Fig. 6 that the SPH algorithm suffers from numerical fracture due to tensile instability. Enlarging the smoothing length can alleviate the numerical fracture, but particle clumps may still exist. Furthermore, enlarging the smoothing length enlarges the time step size but increases the one step computational cost significantly. The total cost is also increased as a consequence.

6.3. *Debris cloud problem*

Debris cloud could result from the typical hypervelocity impact of a penetrator to a thin plate. Beissel et al. [16] have simulated

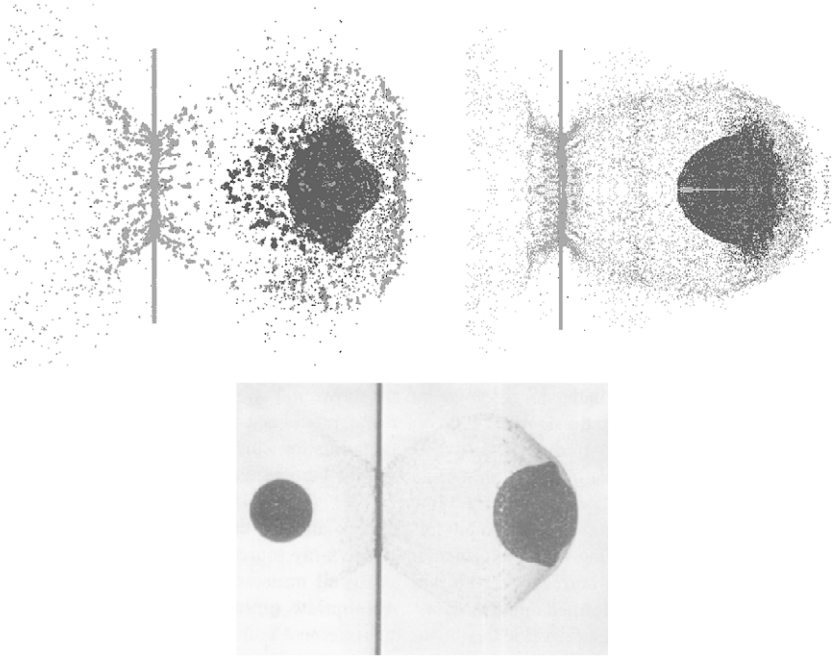

Fig. 7. Debris cloud obtained by SPH (top left), MPM (top right) and experiment [16] (bottom) at 8 µs after impact.

Table 5
Variation of time step size in debris cloud simulation

	$\Delta t_{\max}$ (μs)	$\Delta t_{\min}$ (μs)	Steps
SPH	0.0282	0.0074	1059
MPM	0.0290	0.0290	276

hypervelocity impact of aluminum spheres on aluminum thin plate using their coupled method in which distorted finite element was converted to meshfree particle. An Al–Al impact is simulated here using SPH and MPM. The sphere is of the 16 mm diameter. The target plate has the thickness of 0.8 mm and the diameter of 60 mm. The size data have slight deviation from original data but that deviation makes almost no difference to the results. The impact velocity is 6150 m/s. Particle interval is set 0.2 mm and a total of 550,848 particles are used. The linear hardening material model is used to assure the material parameters are the same in the two methods. The parameters of Grüneisen equation of state are taken from Ref. [60]: $C = 5300$ m/s, $S = 1.34$, and $\gamma_0 = 2.0$. All material parameters used are listed in Table 4.

Fig. 7 shows the computed results of debris cloud at 8 μs after impact and compares with the experimental results. The debris cloud produced by SPH simulation consists of larger fragments than by MPM. As discussed in Section 5, larger fragments in SPH results may be caused by tensile instability which could induce particles clumping. Furthermore, the configuration of the debris cloud produced by SPH simulation doesn't agree well with the experimental results. According to published literature, SPH could perform better in simulating hypervelocity impact problem with special treatments to avoid the instability and the inconsistency. Unfortunately, few of them have yet been implemented in LS-DYNA version 970.

In this simulation, GIMP makes no significant difference in debris cloud configuration than MPM. So MPM is used, which is more efficient than GIMP.

Table 5 shows that critical time step size in SPH simulation is shortened very much due to the severe compression of material. With almost the same initial time step size, SPH has to take much more time steps than MPM to finish the simulation.

7. Conclusions

A three-dimensional MPM computer code MPM3D is developed for the simulation of hypervelocity impact problems in this paper, in which the Johnson–Cook material model and Mie–Grüneisen equation of state are implemented. The basic formulations and features of SPH and MPM are compared in detail, and their performances are investigated numerically by using MPM3D code and LS-DYNA SPH module. MPM possesses many prominent features. The formulation of MPM is simple and similar to the traditional finite element method (FEM). The time consuming neighbor searching, which is compulsory in most meshfree methods like SPH and EFG, is not required in MPM. The shape functions exactly satisfy the constant and linear consistency. MPM avoid tensile instability that is annoying in SPH. The boundary conditions can be applied in MPM as easily as in FEM, and contact algorithm can be efficiently implemented whose cost is linear in the number of bodies. Because the same regular computational grid can be used in all time steps, the time step keeps constant in MPM simulation. The material point method is an efficient and promising method for simulating the hypervelocity impact problems.

References

[1] Gadala MS, Wang J. Ale formulation and its application in solid mechanics. Computer Methods in Applied Mechanics and Engineering 1998;167(1–2):33–55.

[2] Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. Computer Methods in Applied Mechanics and Engineering 1996;139(1–4):3–47.

[3] Li S, Liu WK. Meshfree and particle methods and their applications. Applied Mechanics Reviews 2002;55(1):1–34.

[4] Chen W, Hon YC. Numerical convergence of boundary knot method in the analysis of Helmholtz, modified Helmholtz, and convection-diffusion problems. Computer Methods in Applied Mechanics and Engineering 2003;192:1859–75.

[5] Liu X, Liu GR. Radial point interpolation collocation method for the solution of nonlinear poisson problems. Computational Mechanics 2005;36(4):298–306.

[6] Monaghan JJ. Smoothed particle hydrodynamics. Annual Review of Astronomy and Astrophysics 1992;30:543–74.

[7] Liu GR, Liu MB. Smoothed particle hydrodynamics: a meshfree particle method. Singapore: World Scientific; 2003.

[8] Li S, Liu WK. Meshfree particle methods. Berlin: Springer; 2004.

[9] Sulsky D, Chen Z, Schreyer HL. A particle method for history-dependent materials. Computer Methods in Applied Mechanics and Engineering 1994;118(1–2):179–96.

[10] Sulsky D, Zhou SJ, Schreyer HL. Application of a particle-in-cell method to solid mechanics. Computer Physics Communications 1995;87(1–2):236–52.

[11] Rabb RJ, Fahrenthold EP. Numerical simulation of oblique impact on orbital debris shielding. International Journal of Impact Engineering 1999;23(1):735–44.

[12] Fahrenthold EP, Horban BA. An improved hybrid particle-element method for hypervelocity impact simulation. International Journal of Impact Engineering 2001;26(1–10):169–78.

[13] Park YK, Fahrenthold EP. A kernel free particle-finite element method for hypervelocity impact simulation. International Journal for Numerical Methods in Engineering 2005;63(5):737–59.

[14] Johnson GR, Beissel SR, Stryk RA. An improved generalized particle algorithm that includes boundaries and interfaces. International Journal for Numerical Methods in Engineering 2002;53(4):875–904.

[15] Johnson GR, Stryk RA. Conversion of 3d distorted elements into meshless particles during dynamic deformation. International Journal of Impact Engineering 2003;28(9):947–66.

[16] Beissel SR, Gerlach CA, Johnson GR. Hypervelocity impact computations with finite elements and meshfree particles. International Journal of Impact Engineering 2006;33(1–12):80–90.

[17] Lucy L. A numerical approach to the testing of the fission hypothesis. Astrophysical Journal 1977;82:1013.

[18] Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. Monthly Notices of the Royal Astronomical Society 1977;181:375–89.

[19] Faraud M, Destefanis R, Palmieri D, Marchetti M. SPH simulations of debris impacts using two different computer codes. International Journal of Impact Engineering 1999;23(1):249–60.

[20] Groenenboom PHL. Numerical simulation of 2D and 3D hypervelocity impact using the SPH option in PAM-SHOCK (TM). International Journal of Impact Engineering 1997;20(1–5):309–23.

[21] Hallquist JO. LS-DYNA theoretical manual. Livermore: Livermore Software Technology Corporation; 1998.

[22] Brackbill JU, Ruppel HM. Flip: a method for adaptively zoned, particle-in-cell calculations in two dimensions. Journal of Computational Physics 1986;65:314–43.

[23] Zhang X, Sze KY, Ma S. An explicit material point finite element method for hyper-velocity impact. International Journal for Numerical Methods in Engineering 2006;66(4):689–706.

[24] Johnson GR, Holmquist TJ. Evaluation of cylinder-impact test data for constitutive model constants. Journal of Applied Physics 1988;64(8):3901–10.

[25] Libersky L, Petschek AG, Carney TC, Hipp J, Allahdadi FA. High strain lagrangian hydrodynamics – a three-dimensional SPH code for dynamics material response. Journal of Computational Physics 1993;109:67–75.

[26] Libersky LD, Petschek AG. Smooth particle hydrodynamics with strength of materials. Advances in the free-Lagrange method. Lecture Notes in Physics 1990;395:248–57.

[27] Randles PW, Libersky LD. Smoothed particle hydrodynamics: some recent improvements and applications. Computer Methods in Applied Mechanics and Engineering 1996;139(1–4):375–408.

[28] Johnson GR, Stryk RA, Beissel SR. SPH for high velocity impact computations. Computer Methods in Applied Mechanics and Engineering 1996;139(1–4):347–73.

[29] Swegle JW, Hicks DL, Attaway SW. Smoothed particle hydrodynamics stability analysis. Journal of Computational Physics 1995;116(1):123–34.

[30] Balsara DS. von Neumann stability analysis of smoothed particle hydrodynamics – suggestions for optimal-algorithms. Journal of Computational Physics 1995;121(2):357–72.

[31] Hernquist L, Katz N. Treesph: a unification of SPH with the hierarchical tree method. The Astrophysical Journal Supplement Series 1989;70:419–46.

[32] Waltz J, Page GL, Milder SD, Wallin J, Antunes A. A performance comparison of tree data structures for N-body simulation. Journal of Computational Physics 2002;178:1–14.

[33] Capuzzo-Dolcetta R, Miocchi P. A comparison between the fast multipole algorithm and the tree-code to evaluate gravitational forces in 3-D. Journal of Computational Physics 1998;143:29–48.

[34] Munjiza A, Andrews KRF. NBS contact detection algorithm for bodies of similar size. International Journal for Numerical Methods in Engineering 1998;43:131–49.

[35] Batra RC, Zhang GM. Search algorithm, and simulation of elastodynamic crack propagation by modified smoothed particle hydrodynamics (MSPH) method. Computational Mechanics 2007;40:531–46.

[36] Bardenhagen SG, Kober EM. The generalized interpolation material point method. CMES – Computer Modeling in Engineering & Sciences 2004;5(6):477–95.

[37] Johnson GR, Beissel SR. Normalized smoothed functions for SPH impact computations. International Journal for Numerical Methods in Engineering 1996;39:2725–41.

[38] Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. Computer Methods in Applied Mechanics and Engineering 1996;139:49–74.

[39] Morris JP. Analysis of smoothed particle hydrodynamics with applications. Ph.D. thesis, Monash University.

[40] Chen JK, Beraun JE, Jih CJ. An improvement for tensile instability in smoothed particle hydrodynamics. Computational Mechanics 1999;23:279–87.

[41] Monaghan JJ. SPH without a tensile instability. Journal of Computational Physics 2000;159:290–311.

[42] Dyka CT, Randles PW, Ingel RP. Stress points for tension instability in SPH. International Journal for Numerical Methods in Engineering 1997;40(13):2325–41.

[43] Randles PW, Libersky LD. Normalized SPH with stress points. International Journal for Numerical Methods in Engineering 2000;47:1445–62.

[44] Sulsky D, Kaul A. Implicit dynamics in the material-point method. Computer Methods in Applied Mechanics and Engineering 2004;193(12–14):1137–70.

[45] Campbell PM. Some new algorithms for boundary value problems in smoothed particle dynamics. DNA Report 1989. DNA-TR-88–286.

[46] Monaghan JJ. Simulating free surface flow with SPH. Journal of Computational Physics 1994;110:399.

[47] Liu MB, Liu GR, Lam KY. Investigation into water mitigations using a meshless particle method. Shock Waves 2002;12(3):181–95.

[48] Campbell J, Vignjevic R, Libersky L. A contact algorithm for smoothed particle hydrodynamics. Computer Methods in Applied Mechanics and Engineering 2000;184(1):49–65.

[49] York AR, Sulsky D, Schreyer HL. The material point method for simulation of thin membranes. International Journal for Numerical Methods in Engineering 1999;44(10):1429–56.

[50] Bardenhagen SG, Brackbill JU, Sulsky D. The material-point method for granular materials. Computer Methods in Applied Mechanics and Engineering 2000;187(3–4):529–41.

[51] Hu W, Chen Z. A multi-mesh MPM for simulating the meshing process of spur gears. Computers & Structures 2003;81(20):1991–2002.

[52] Pan XF, Xu AG, Zhang GC, Zhu JS. Generalized interpolation material point approach to high melting explosive with cavities under shock. Journal of Physics D: Applied Physics 2008;41:015401.

[53] Sod GA. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. Journal of Computational Physics 1978;27:1–31.

[54] Monaghan JJ, Gingold RA. Shock simulation by the particle method SPH. Journal of Computational Physics 1983;52:374–89.

[55] Monaghan JJ. SPH and Riemann solvers. Journal of Computational Physics 1997;136(2):298–307.

[56] Inutsuka S. Reformulation of smoothed particle hydrodynamics with Riemann solver. Journal of Computational Physics 2002;179(1):238–67.

[57] Sigalotti LDG, Lopez H, Donoso A, Sira E, Klapp J. A shock-capturing SPH scheme based on adaptive kernel estimation. Journal of Computational Physics 2006;212(1):124–49.

[58] Monaghan JJ. Smoothed particle hydrodynamics. Reports on Progress in Physics 2005;68(8):1703–59.

[59] York AR, Sulsky D, Schreyer HL. Fluid–membrane interaction based on the material point method. International Journal for Numerical Methods in Engineering 2000;48(6):901–24.

[60] Wilkins ML. Computer simulation of dynamic phenomena. Berlin: Spinger; 1999.