

An explicit material point finite element method for hyper-velocity impact

X. Zhang^{1,*}, K. Y. Sze^{2,‡} and S. Ma^{1,§}

¹*School of Aerospace, Tsinghua University, Beijing 100084, China*

²*Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR, China*

SUMMARY

In this paper, an explicit material point finite element (FE) method is proposed and a computer code EMPFE-3D is developed for simulating hyper-velocity impact. The material domain is discretized by a mesh of finite elements. The momentum equations are solved on a predefined computational grid (like the material point method) in the large deformation zone, and on the FE mesh (like the traditional FE method) elsewhere. The grid may be fixed in space or moved in a predefined way. The nodes covered by the grid are treated as material particles, and the remaining nodes are treated as FE nodes. The proposed method yields the same results as the traditional FE method if the grid vanishes. On the other hand, it yields the same results as the material point method if the grid covers the entire material domain at all time steps. The method combines the advantages of Eulerian and Lagrangian descriptions of motion while eliminates their drawbacks due to element entanglement and numerical dissipation. The method is computationally efficient and can be easily implemented in an existing explicit FE code like DYNA3D. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: meshfree; material point method; hyper-velocity impact; finite element

1. INTRODUCTION

During the last decades, the finite element (FE) method has yielded significant progress and has been successfully applied to many scientific and engineering problems [1]. Two descriptions, the Lagrangian and Eulerian, are typically used. In the Lagrangian description, the FE mesh is embedded in and deforms with the material domain. It presents no convective effects and

*Correspondence to: X. Zhang, School of Aerospace, Tsinghua University, Beijing 100084, China.

†E-mail: xzhang@tsinghua.edu.cn

‡E-mail: kysze@hku.hk

§E-mail: ms02@mails.tsinghua.edu.cn

Contract/grant sponsor: National Natural Science Foundation of China; contract/grant number: 10472052

Contract/grant sponsor: National Basic Research Program of China; contract/grant number: 2004CB619304

Contract/grant sponsor: The University of Hong Kong; contract/grant number: 10205141

Received 11 May 2005

Revised 29 August 2005

Accepted 3 October 2005

it is ideal for history-dependent materials. However, if the deformation is very large, e.g. in hyper-velocity impact and metal forming, mesh distortion and element entanglement become the significant limitations of the Lagrangian description.

An alternative to the Lagrangian description is the Eulerian description in which the material flows through a grid fixed in space. It completely avoids element distortions but introduces other difficulties like appropriate representation of free boundary and tracking of the material deformation history. There are dissipation and dispersion problems associated with the mass flux between adjacent elements. In addition, many redundant elements may be needed in the Eulerian grid to enclose the entire physical space that the material domain occupies in the course of the simulation.

The arbitrary Lagrangian–Eulerian (ALE) formulation [2, 3] alleviates many of the drawbacks in the traditional Lagrangian and Eulerian descriptions. In ALE, the computational mesh is continuously moved independently of material deformation to optimize element shapes and describe the boundaries accurately. However, the convective terms still pose some problems. Furthermore, designing an efficient and effective mesh-moving algorithm for complicated 3D problems remains a challenging task.

In recent years, much effort has been devoted to the development of meshfree methods and more than 10 different meshfree methods have been developed [4–9]. In meshfree methods, the trial functions are constructed through a set of discrete points instead of a mesh. Thus, the difficulties arising from mesh distortion and element entanglement are alleviated. The solution accuracy of the meshfree methods is still dependent on the node regularities, though it is not as significant as that in the FE method.

The material point method (MPM) [10, 11] is an extension of the FLIP particle-in-cell method [12, 13] to solid mechanics. Being a fully Lagrangian particle method, it discretizes a material domain by using a collection of material points, called particles, which carry all state variables such as displacement, stress, strain, etc. The same computational grid is often used for all the time steps so that mesh distortion and element entanglement are avoided. Equations of motion are solved on the grid using the standard Lagrangian FE formulation. The numerical dissipation normally associated with a Eulerian method is eliminated. However, there is an unavoidable smoothing in this method that arises from the coarseness of the computational grid relative to the material point density. Moreover, the two mappings between the particles and grid points required in each time step increase the computational effort compared with the traditional FE method.

In this paper, an explicit material point finite element (MPFE) method is proposed and a computer code EMPFE-3D is developed for the analysis of hyper-velocity impact problems. The material domain is discretized by a mesh of finite elements. The momentum equations are solved on a predefined computational grid under the Lagrangian framework (like the MPM) in the large deformation zone and solved on the FE mesh (like the traditional Lagrangian FE method) elsewhere. The grid may be fixed in space or moved in a prescribed way. The nodes covered by the grid are treated as material particles, and the remaining nodes are treated as FE nodes.

The MPFE method yields the same results as the traditional FE method if there is no computational grid is defined. On the other hand, it yields the same results as the MPM if the grid covers the entire material domain at all time steps. The proposed method combines the advantages of the Eulerian and Lagrangian descriptions of the motion, while overcoming their drawbacks. It eliminates the difficulty of element entanglement and numerical dissipation. The

method is computationally efficient and can be easily implemented in an existing explicit FE code like DYNA3D.

2. GOVERNING EQUATIONS

Considering the updated Lagrangian description, the material motion is governed by the momentum equations [14]

$$\sigma_{ij,j} + \rho f_i = \rho \ddot{x}_i \quad \forall x_i \in V \quad (1)$$

subject to the traction boundary conditions

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

and the displacement boundary conditions

$$x_i(X_\alpha, t) = d_i(t) \quad \forall x_i \in \Gamma_d \quad (3)$$

where V is the material domain, Γ_t and Γ_d are, respectively, the boundary portions of V prescribed with traction and displacement, X_α is the Lagrangian co-ordinate, σ_{ij} is the Cauchy stress, ρ is the density, f_i is the body force density, \ddot{x}_i is the acceleration, the comma denotes covariant differentiation and n_j is the unit outward normal to the boundary. The repeated indices imply a summation over 1, 2 and 3.

The weak form of the momentum equations and the traction boundary condition is given by

$$\delta \Pi = \int_V \rho \ddot{x}_i \delta x_i dV + \int_V \rho \sigma_{ij}^s \delta x_{i,j} dV - \int_V \rho f_i \delta x_i dV - \int_{\Gamma_t} t_i \delta x_i d\Gamma = 0 \quad (4)$$

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

Mass conservation is trivially stated

$$\rho dV = \rho_0 dV_0 \quad (5)$$

where ρ_0 is the reference density. The energy equation

$$\dot{E} = J \sigma_{ij} \dot{\epsilon}_{ij} \quad (6)$$

is integrated in time and used for the equation of state evaluations which will be further discussed in Section 4.2. In (6), 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, $\dot{\epsilon}_{ij}$ is the strain rate.

3. SOLUTION SCHEME

The formulation presented in this paper is for 3D problems. For the sake of clarity, however, let us consider the impact problem of a 2D bar as shown in Figure 1. The bar is discretized by a mesh of finite elements. The lower part of the bar undergoes large deformation, so that severe element distortion occurs as seen in Figure 1(b). To overcome the difficulty arising from

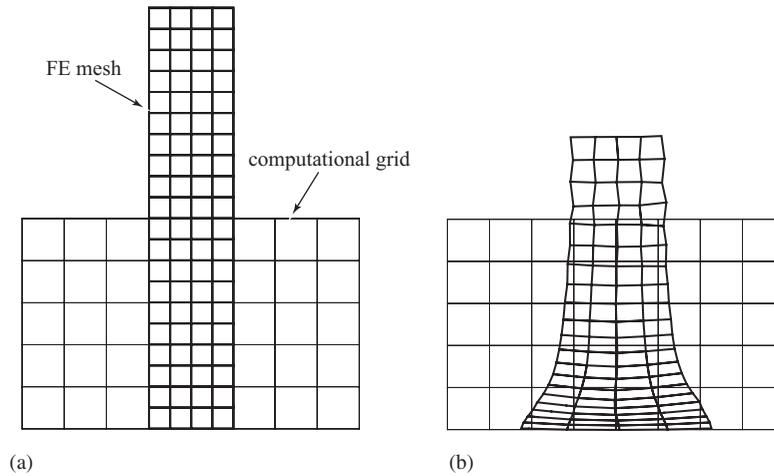


Figure 1. FE mesh and computational grid: (a) initial configuration; and (b) current configuration.

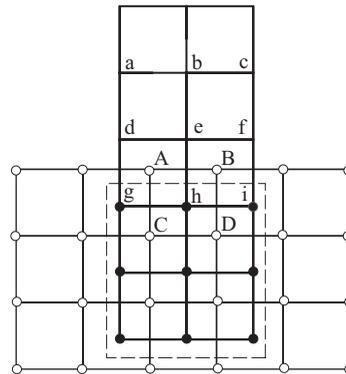


Figure 2. Solution scheme, (a)–(f) are FE nodes, (g)–(i) are particles, A–D are grid points.

element distortion and entanglement, a computational grid fixed in space is incorporated to serve as an updated Lagrangian frame.

For simplicity, the nodes covered by the computational grid are termed as *particles*, and the remaining nodes are termed as *FE nodes*. The number of particles varies from time to time because the nodes not covered by the grid currently may move into the grid at the consequent time steps. In Figure 2(a)–(f) are FE nodes, 2(g)–(i) are particles and A–D are grid points.

3.1. Solution of momentum equations on the grid

For the material subdomain covered by the computational grid, the momentum equations are solved on the computational grid. During this phase of solution, the particles are attached

to the grid and they deform with the grid. There is no particle convection through the grid, so the momentum equations are the same as those in Lagrangian formulation. There are no advective terms in the momentum equations, so it eliminates the numerical dissipation normally associated with a Eulerian method. After obtaining the kinematic solution on the grid, the state variables carried by the grid points are mapped back to the particles to update that of the particles. The deformed grid is discarded in the subsequent time step and a new regular grid, which can be the same as the previous undeformed grid, is used. In this light, the difficulty due to mesh distortion and element entanglement can be eliminated.

Because the particles are rigidly attached to the grid, the kinematic information can be mapped between the particles and grid points through the shape functions of the grid. For example, the particle acceleration \ddot{x}_{pi} can be obtained by mapping the grid point acceleration \ddot{x}_{gi} to the particle, namely,

$$\ddot{x}_{pi} = \sum_{g=1}^{n_g} N_{pg} \ddot{x}_{gi} \quad (7)$$

where n_g is the total number of grid points, $N_{pg} = N_g(x_{pi})$ is the value of the shape function associated with the g th grid point at the p th particle. In this paper, the 8-point hexahedron grid is used. If the p th particle is inside the hexahedron, the shape function is given by

$$N_g = \frac{1}{8}(1 + \xi \zeta_g)(1 + \eta \eta_g)(1 + \zeta \zeta_g) \quad (8)$$

where ξ_g, η_g and ζ_g take on their nodal values of $(\pm 1, \pm 1, \pm 1)$ at the grid points. If the p th particle is outside the hexahedron, $N_g = 0$.

Similar to the MPM, the material mass is lumped at particles. Therefore, the density ρ at any point x_i in the material domain can be approximated as

$$\rho(x_i) = \sum_{p=1}^{n_p} M_p \delta(x_i - x_{pi}) \quad (9)$$

where δ is the Dirac delta function, n_p is the total number of particles and x_{pi} are their co-ordinates. The mass M_p carried by node p can be approximated by

$$M_p = \frac{1}{8} \sum_e \rho_e V_e \quad (10)$$

where ρ_e and V_e are the density and volume of the elements connected to node p . The factor '1/8' is for 3/D and it should be changed to '1/4' for 2/D.

Substituting (7) and (9) into (4), and invoking the arbitrariness of δx_{hi} lead to

$$\dot{p}_{hi} = f_{hi}^{\text{int}} + f_{hi}^{\text{ext}} \quad (11)$$

where

$$p_{hi} = \sum_{g=1}^{n_g} m_{gh} \dot{x}_{gi} \quad (12)$$

is the momentum of the h th grid point in the i th direction. Moreover,

$$f_{hi}^{\text{int}} = - \sum_{p=1}^{n_p} V_p N_{ph,j} \sigma_{pij} \quad (13)$$

and

$$f_{hi}^{\text{ext}} = \sum_{p=1}^{n_p} M_p N_{ph} f_{pi} + \int_{\Gamma_i} N_h t_i \, d\Gamma \quad (14)$$

are the internal and external forces at h th grid point in the i th direction. $V_p = M_p/\rho_p$, ρ_p and σ_{pij} are the current density and stress tensor at the p th node, respectively, f_{pi} is the body force density at the p th node in the i th direction. In (12), grid mass matrix m_{gh} is given by

$$m_{gh} = \sum_{p=1}^{n_p} M_p N_{pg} N_{ph} \quad (15)$$

If a lumped grid mass matrix is used, Equation (12) can be rewritten as

$$p_{hi} = m_h \dot{x}_{hi} \quad (16)$$

where m_h is the lumped grid mass matrix given by

$$m_h = \sum_{p=1}^{n_p} M_p N_{ph} \quad (17)$$

Although M_p remains constant at all time, the grid point mass m_h varies with time and must be recalculated at the beginning of each time step.

3.2. Solution of momentum equations on the FE mesh

For the part of the material domain not covered by the computational grid, the momentum equations are solved on the Lagrangian FE mesh like the traditional FE method. The position of a material point X_α is given by the interpolation of FE nodal position as

$$x_i(X_\alpha, t) = \sum_{p=1}^{n_f} \phi_p x_{pi}(t) \quad (18)$$

where n_f is the total number of FE nodes, ϕ_p is the value of the standard element shape function at the point x_i associated with the p th FE node. x_{pi} is the nodal co-ordinate of the p th FE node in the i th direction. The integration over V in (4) can be approximated by the summation of the integration over all elements. Substituting (18) into (4) and using the lumped mass matrix result in

$$M_p \ddot{x}_{pi} = f_{pi}^{\text{int}} + f_{pi}^{\text{ext}} \quad (19)$$

where

$$f_{pi}^{\text{int}} = - \sum_e \int_{V_e} \phi_{p,j} \sigma_{ji} \, dV \quad (20)$$

$$f_{pi}^{\text{ext}} = \sum_e \left(\int_{V_e} \rho \phi_p f_i \, dV + \int_{\Gamma_{te}} \phi_p t_i \, d\Gamma \right) \quad (21)$$

It should be noted that the elements partially covered by the grid will contribute to the nodal forces of the particles. For example, element d-e-h-g in Figure 2 contributes to the nodal force

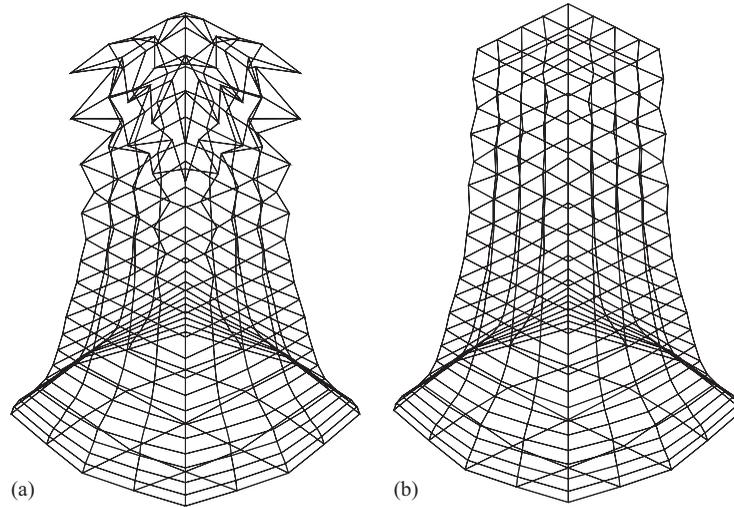


Figure 3. Hourglass modes: (a) without hourglass control; and (b) with hourglass control.

of particles *g* and *h*. This contribution must be mapped to the grid points when establishing the momentum equations. Consequently, Equation (14) should be rewritten as

$$f_{hi}^{ext} = \sum_{p=1}^{n_p} N_{ph}(M_p f_{pi} + f_{pi}^{ele}) + \int_{\Gamma_t} N_{ph} t_i \, d\Gamma \tag{22}$$

where f_{pi}^{ele} is the contribution to the nodal force of *p*th node in the *i*th direction from those elements partially covered by the grid, which can be evaluated by (20).

Although the proposed MPFE method can be implemented as an implicit method, this paper focuses on the explicit implementation, which is often used for simulating problems like hyper-velocity impact. The 1-point quadrature is used to evaluate the volume integration in (20) and (21), namely

$$f_{pi}^{int} = -\sum_e \phi_{ep,j} \sigma_{eji} V_e \tag{23}$$

$$f_{pi}^{ext} = \sum_e M_e \phi_{ep} f_{ei} + \sum_e \int_{\Gamma_{te}} \phi_{ep} t_i \, d\Gamma \tag{24}$$

where $M_e = \rho_e V_e$. The subscript *e* represents the value at the centre of element *e*.

Numerical studies show that the hourglass modes can easily be excited in the proposed EMPFE method as shown in Figure 3(a). Hence, a hourglass control scheme [14, 15] must be employed. In this paper, the following hourglass-resisting forces are added to the right side of (19):

$$f_{pi}^\Gamma = -\sum_{\alpha=1}^4 \beta_h h_{i\alpha} \Gamma_{\alpha p} \tag{25}$$

where

$$\beta_h = \frac{1}{4} Q_{hg} \rho V_e^{2/3} c \quad (26)$$

$$h_{i\alpha} = \sum_{k=1}^8 v_{ki} \Gamma_{\alpha k} \quad (27)$$

in which c is the material sound speed, and Q_{hg} is a user-defined non-dimensional constant usually set to a value between 0.05 and 0.15. Lastly, $\Gamma_{\alpha k}$ ($\alpha = 1, 2, 3, 4$; $k = 1, 2, \dots, 8$) are the hourglass base vectors.

With the hourglass control scheme, Figure 3(b) shows that the hourglass modes have been suppressed in a typical simulation.

4. NUMERICAL IMPLEMENTATION

It can be concluded by comparing (11)–(16) and (19)–(24) that the solution equations for the grid points and FE nodes are almost the same, except that (11)–(16) are solved at grid points while (19)–(24) are solved at FE nodes. In (11)–(16), the internal forces are mapped from the particles to the grid points to establish the momentum equations at the grid points. After solving the momentum equations, kinematic solutions are mapped back to the particles to update their material properties. In (19)–(24), the material data are always carried by the quadrature point of the elements. In other words, the particles play the roles of both the FE nodes and the quadrature points which carry the kinematic and material data, respectively.

4.1. Time integration

The central difference method with variable time step size is employed to integrate (11) and (19), see Figure 4. It can be noted in the figure that $t^{k+1} = t^k + \Delta t^{k+1/2}$, $t^{k+1/2} = t^k + \frac{1}{2} \Delta t^{k+1/2} = t^{k-1/2} + \Delta t^k$ and $\Delta t^k = \frac{1}{2} (\Delta t^{k-1/2} + \Delta t^{k+1/2})$.

Let the superscripts k and $k - 1/2$ denote the value at time t^k and $t^{k-1/2}$, respectively. Suppose x_{pi}^k and $\dot{x}_{pi}^{k-1/2}$ are known, and we are seeking for the solution at time t^{k+1} . The energy equation (6) can be integrated as

$$E^{k+1} = E^k + \frac{1}{2} J^k (\sigma_{ij}^k + \sigma_{ij}^{k+1}) \dot{\epsilon}_{ij}^{k+1/2} \Delta t^{k+1/2} \quad (28)$$

4.1.1. *Time integration on the FE mesh.* From (19), the acceleration of FE nodes at time t^k can be obtained as

$$\ddot{x}_{pi}^k = (f_{pi}^{k,int} + f_{pi}^{k,ext}) / M_p \quad (29)$$

The nodal velocity at time $t^{k+1/2}$ and the nodal position at time t^{k+1} are given by

$$\dot{x}_{pi}^{k+1/2} = \dot{x}_{pi}^{k-1/2} + \ddot{x}_{pi}^k \Delta t^k \quad (30)$$

$$x_{pi}^{k+1} = x_{pi}^k + \dot{x}_{pi}^{k+1/2} \Delta t^{k+1/2} \quad (31)$$

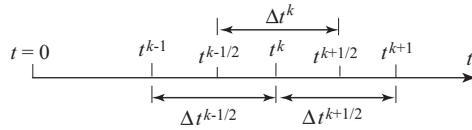


Figure 4. Time integration.

where the time step size, $\Delta t^{k+1/2}$, is determined by

$$\Delta t^{k+1/2} = \min(L_e/c) \tag{32}$$

where $L_e = V_e/A_{e \max}$ is the characteristic length of element e , $A_{e \max}$ is the area of the largest side of element e and c is the material sound speed.

4.1.2. *Time integration on the grid.* Similar to (30), the grid momentum at time $t^{k+1/2}$ can be obtained from (11) as

$$p_{hi}^{k+1/2} = p_{hi}^{k-1/2} + \Delta p_{hi}^{k+1/2} \tag{33}$$

where

$$\Delta p_{hi}^{k+1/2} = (f_{hi}^{k,int} + f_{hi}^{k,ext})\Delta t^k \tag{34}$$

The particle velocity at time $t^{k+1/2}$ and the particle position at time t^{k+1} can be obtained by mapping the momentum from the grid points to the particles as

$$\dot{x}_{pi}^{k+1/2} = \dot{x}_{pi}^{k-1/2} + \sum_{h=1}^{n_g} \Delta p_{hi}^{k+1/2} N_{ph}^k / m_h^k \tag{35}$$

$$x_{pi}^{k+1} = x_{pi}^k + \Delta t^{k+1/2} \sum_{h=1}^{n_g} p_{hi}^{k+1/2} N_{ph}^{k+1/2} / m_h^{k+1/2} \tag{36}$$

In (36), $N_{ph}^{k+1/2}$ and $m_h^{k+1/2}$ are the value of the shape function N_{ph} and grid point mass m_h at time step $t^{k+1/2}$, which are unknown. However, $N_{ph}^{k+1/2} / m_h^{k+1/2}$ can be approximated by N_{ph}^k / m_h^k .

4.2. *Stress update*

After obtaining the positions and velocities of all FE nodes and particles at time t^{k+1} from time integration, the stress can be updated by using a constitution model. Stress update is carried out at the particles in the material subdomain covered by the grid, and carried out at the element centres elsewhere. The stresses at time t^{k+1} can be obtained by

$$\sigma_{ij}^{k+1} = \sigma_{ij}^k + \dot{\sigma}_{ij}^{k+1/2} \Delta t^{k+1/2} \tag{37}$$

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

$$\dot{\sigma}_{ij} = \sigma_{ij}^\nabla + \sigma_{il}\omega_{lj} + \sigma_{jl}\omega_{li} \tag{38}$$

in which $\omega_{ij} = \frac{1}{2}(\dot{x}_{i,j} - \dot{x}_{j,i})$ is the spin tensor, and σ_{ij}^∇ is the Jaumann (co-rotational) stress rate and determined from the strain rate $\dot{\epsilon}_{ij} = \frac{1}{2}(\dot{x}_{i,j} + \dot{x}_{j,i})$ by a constitution model. Substituting (38)

into (37) leads to

$$\sigma_{ij}^{k+1} = \sigma_{ij}^k + (\sigma_{il}^k \omega_{lj}^{k+1/2} + \sigma_{jl}^k \omega_{li}^{k+1/2}) \Delta t^{k+1/2} + \sigma_{ij}^{\nabla k+1/2} \Delta t^{k+1/2} \quad (39)$$

Four kinds of constitution models have been implemented in our code EMPFE-3D. They are the elastic model, the elastic-perfectly plastic model, the elasto-plastic hydrodynamic model with isotropic hardening (LS-DYNA material mode 10) and the Johnson–Cook plasticity model [14, 16] (LS-DYNA material mode 15). In the Johnson–Cook model, the equivalent tensile flow stress is expressed as [17]

$$\sigma_y = (A + B\varepsilon^n)(A + C \ln \dot{\varepsilon}^*)(1 - T^{*m}) \quad (40)$$

where ε is the 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. The effects of temperature are ignored in this paper by omitting the last set of parentheses in (40).

The Mie–Gruneisen equation of state [18] is implemented for the last two material models to update the pressure for the hyper-velocity impact problems

$$P = (C\mu + D\mu^2 + S\mu^3) \left(1 - \frac{\gamma\mu}{2}\right) + \gamma E \quad (41)$$

where $\mu = \rho/\rho_0 - 1 = V_0/V - 1$, $\gamma = \gamma_0 \rho_0/\rho$, $C = \rho_0 c_0^2$, $D = C(2\lambda - 1)$, $S = C(\lambda - 1)(3\lambda - 1)$, c_0 is the sound wave speed, λ and γ_0 are the Gruneisen coefficients, and E is the internal energy per initial volume. C, D and S are the material constants.

In the implementation of the constitution models, the strain rate $\dot{\varepsilon}_{ij}$ is required to be calculated at the particles and at the element centres not covered by the grid. The strain rate at the element centre can be readily evaluated by using the FE mesh as that in the traditional FE method. Because the particles are attached to the grid and deform with the grid, the strain rate at the particles can be evaluated based on the velocities of the grid points as

$$\dot{\varepsilon}_{pij} = \frac{1}{2} \sum_{g=1}^{n_g} (N_{pg,j} \dot{x}_{gi} + N_{pg,i} \dot{x}_{gj}) \quad (42)$$

4.3. Description of the algorithm

The implementation of the proposed method is very similar to that of the traditional FE method, and can be easily implemented in an existing FE code like DYNA3D. The method can be implemented as follows.

1. Input phase: read FE nodal co-ordinates, element connectivities, material properties, and initial conditions.
2. Initializing phase: calculate the mass M_p carried by all nodes using (10), create the computational grid, which is fixed in space in this paper.
3. Solution phase: Loop until the current time t^k is greater than the termination time t^f :
 - (a) Loop over all nodes to calculate the kinetic energy of the system.
 - (b) Loop over all nodes to set their flags for identifying the FE nodes and particles.

- (c) Loop over all nodes including FE nodes and particles to calculate their external forces.
 - (d) Loop over all particles to update their stresses.
 - (e) Loop over all FE elements not fully covered by the grid to update stresses at their centres; to calculate the internal and hourglass-resisting nodal forces according to (23) and (25) for FE nodes; to determine the new time step size Δt^{k+1} according to (32).
 - (f) Loop over all particles to calculate the internal and external grid point forces according to (13) and (22).
 - (g) Loop over all grid points to integrate the momentum equations to (33) and apply the displacement boundary conditions.
 - (h) Loop over all particles to update their velocities and positions according to (35) and (36).
 - (i) Loop over all FE nodes to calculate their accelerations according to (29) and apply the displacement boundary conditions.
 - (j) Loop over all FE nodes to update their velocities and positions according to (30) and (31).
 - (k) Update the time by $t^{k+1} = t^k + \Delta t^{k+1}$.
4. End loop.

5. NUMERICAL EXAMPLES

5.1. Elastic bar impact on a rigid wall

To illustrate the basic idea of the MPFE method, an elastic bar impacting on a rigid wall is investigated, as seen in Figure 5. The material properties of the bar are taken to be $\rho = 8.9 \times 10^3 \text{ kg/m}^3$, $E = 10^5 \text{ MPa}$, and $\nu = 0.3$. The initial velocity of the bar is 200 m/s.

In the computation, the momentum equations are solved on the computational grid fixed in space in the material subdomain covered by the grid, and solved on the FE mesh elsewhere, see Figure 5(a). The nodes covered by the grid are termed as particles and the remaining nodes are termed as FE nodes. The particles inside the grid and element centres not covered by the grid carry all state variables as discussed in Section 3.

After impacting, the bar is bounced back and moves at velocity of around 200 m/s and the kinematic energy of the bar is shown in Figure 6. The computation is aborted at the position shown in Figure 5(c), at which all particles have moved out of the computational grid. To resume the computation, a new computational grid should be exploited which can cover all particles.

5.2. Taylor's impact

The cylinder-impact test is a simple and inexpensive test introduced by Taylor in 1948. It provides data for the appropriate range of strains and strain rates, so several researchers have used this test to estimate the dynamics flow stresses for various materials [17]. This test has also been used to validate computer codes, such as Eulerian wavecode [11, 19], CSQ, HEMP and FLIP.

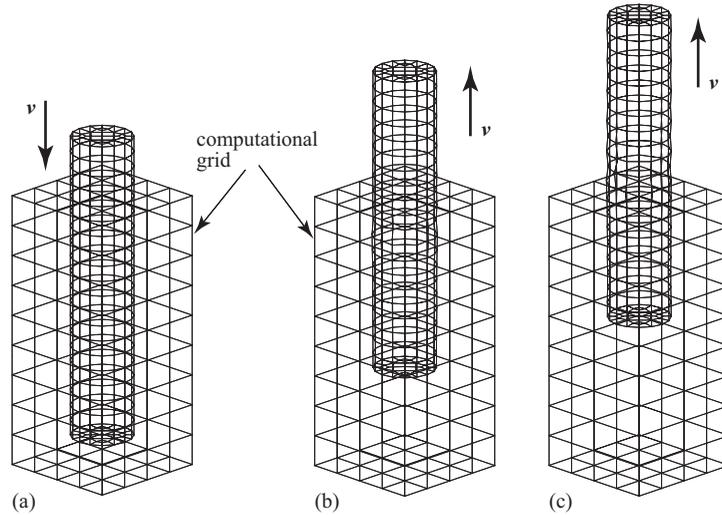


Figure 5. Elastic bar impact on a rigid wall.

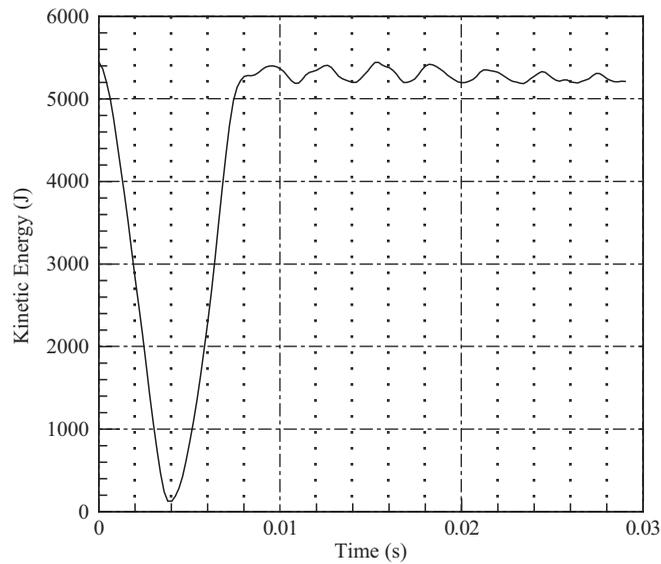


Figure 6. Kinematic energy of the bar.

To quantify the degree of agreement between the computed cylinder shapes and the experimental data, Johnson and Holmquist [17] introduced the following average error:

$$\bar{\Delta} = \frac{1}{3} \left(\frac{|\Delta L|}{L} + \frac{|\Delta D|}{D} + \frac{|\Delta W|}{W} \right) \quad (43)$$

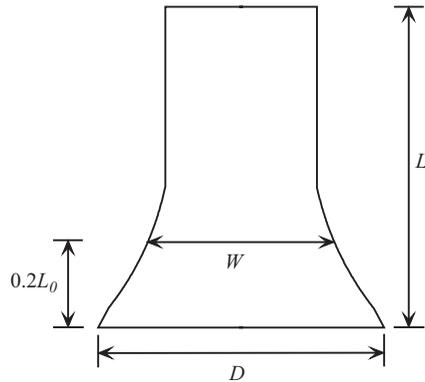


Figure 7. Geometry of the deformed cylinder in Taylor's test.

Table I. Material constants of the cylinder.

| ρ (kg/m ³) | E (MPa) | ν | A (MPa) | B (MPa) | n | C |
|-----------------------------|-------------------|-------|-----------|-----------|-----|-----|
| 8930 | 117×10^3 | 0.35 | 157 | 425 | 1.0 | 0.0 |

Table II. Comparison between numerical results and test data for the cylinder-impact problem.

| | L (mm) | D (mm) | W (mm) | $\bar{\Delta}$ |
|-----------|----------|----------|----------|----------------|
| Test data | 16.2 | 13.5 | 10.1 | — |
| FEM | 16.27 | 13.33 | 10.18 | 0.008 |
| MPM | 16.35 | 13.4 | 9.9 | 0.012 |
| MPFE | 16.38 | 13.4 | 9.9 | 0.013 |

where L , D and W are, respectively, the deformed length, diameter of the deformed end and diameter at $0.2L_0$ from the deformed end from the test results, see Figure 7. ΔL , ΔD , and ΔW are the differences between the computed and experimental results.

The cylinder is made of OFHC copper and its material constants with respect to the Johnson–Cook model are listed in Table I [17]. The initial length, diameter and velocity of the cylinder are $L_0 = 25.4$ mm, $D_0 = 7.6$ mm and $v_0 = 190$ m/s, respectively. The cylinder is discretized by 299×67 elements with 332×68 nodes, see Figure 8(a). Three different computational attempts are taken. In the first attempt (FEM), no computational grid is used, so that the solutions are identical to that of the traditional FE method. In the second attempt (MPM), a $21 \times 21 \times 26$ hexahedral grid with 1 mm grid point spacing is used to cover the whole cylinder. Thus, the solutions are identical to that of the MPM. In the third attempt (MPFE), a $21 \times 21 \times 13$ hexahedral grid with 1 mm grid point spacing is used to cover the lower half of the cylinder. Table II compares the numerical results with the experimental data. It can be seen that the numerical results agree well with the experimental data. The final configuration of the cylinder is shown in Figure 8(b).

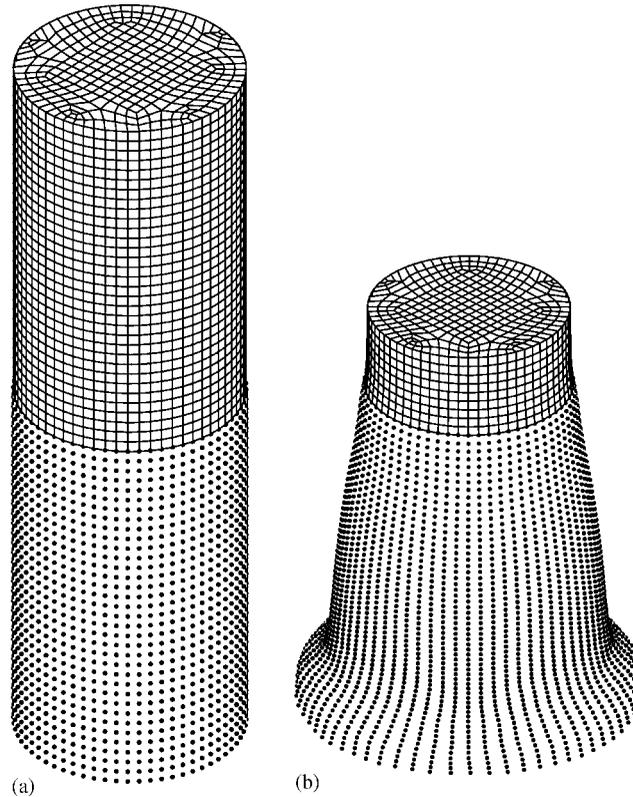


Figure 8. (a) Discretization; and (b) the final shape of the cylinder.

Table III. Mesh distortion and time step size.

| | χ_{\max} | χ_{\min} | Δt_{\max} (ms) | Δt_{\min} (ms) | Time steps | CPU time (s) |
|------|------------------------|------------------------|------------------------|------------------------|------------|--------------|
| FEM | 7.849×10^{-2} | 8.298×10^{-6} | 9.264×10^{-6} | 7.893×10^{-7} | 17987 | 5969 |
| MPFE | 7.849×10^{-2} | 7.573×10^{-2} | 9.264×10^{-6} | 9.242×10^{-6} | 5443 | 1744 |

To investigate the mesh distortion, the following measure is employed:

$$\chi = \min_e A_{e \min} / A_{e \max} \quad (44)$$

in which $A_{e \min}$ and $A_{e \max}$ are the area of the smallest and largest side of element e . Due to the material hardening effect, the mesh distortion is not severe.

To further investigate the effects of mesh distortion on the proposed method, the copper is taken to be elastic-perfectly plastic, i.e. B is set to zero in Table I. Table III compares the mesh distortion, time step size and CPU time consumed at $t = 50.4 \mu\text{s}$ for FEM and MPFE. Figure 9 shows the deformed configurations obtained by FEM and MPFE. Despite the of

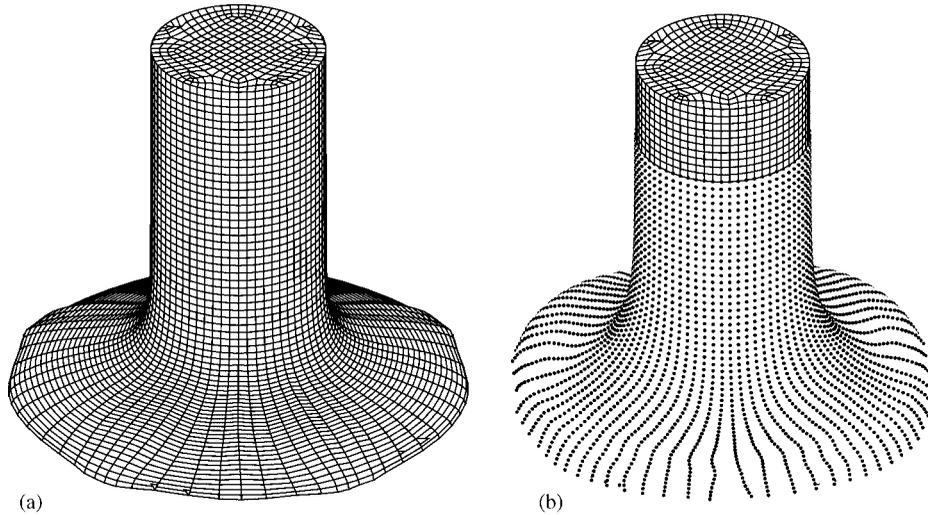


Figure 9. Comparison of the configuration at $t = 50 \mu\text{s}$: (a) FEM; and (b) MPFEM.

Table IV. Material constants of the projectile and plate.

| ρ (kg/m^3) | E (MPa) | ν | A (MPa) | B (MPa) | n | C |
|----------------------------|-------------------|-------|-----------|-----------|-----|-----|
| 8930 | 117×10^3 | 0.35 | 90 | 392 | 0.5 | 0.0 |

geometric similarity of the deformations, the time step size ($= \min(L_e/c)$) for FEM becomes very small due to the severe mesh distortion, and it is prohibitively expensive to continue the computation. On the other hand, the time step size for MPFE remain practically constant during the computation. The CPU time consumptions are in the ratio of 3:1.

5.3. Orbital debris shielding

Due to the extreme mesh distortion, conventional FE codes have been proven to be inefficient in the simulation of hyper-velocity impact problems like the design of orbital debris shielding.

To illustrate the capability of the EMPFE-3D code in the simulation of hyper-velocity impact problems, the impact of a spherical projectile with velocity of 6.6 km/s on a thin plate is examined. The diameter of the projectile is 8 mm and the plate thickness is 3 mm. The projectile and plate are made of copper, which is modelled by the Johnson–Cook plasticity model with the material constants given in Table IV. The Mie–Gruneisen equation of state is used with constants listed in Table V to update the pressure. A contact–impact algorithm similar to that proposed in Reference [20] is implemented in EPFE-3D to deal with the contact between the projectile and plate.

The projectile and the centre part of the plate, as shown in Figure 10, are expected to undergo severe deformation. To eliminate the difficulties arising from the pertinent mesh distortion, the momentum equations of them are solved in a computational grid with 1 mm grid-point spacing,

Table V. Constants in the Mie–Grüneisen equation of state.

| c_0 (m/s) | λ | γ_0 |
|-------------|-----------|------------|
| 3600 | 1.49 | 1.96 |

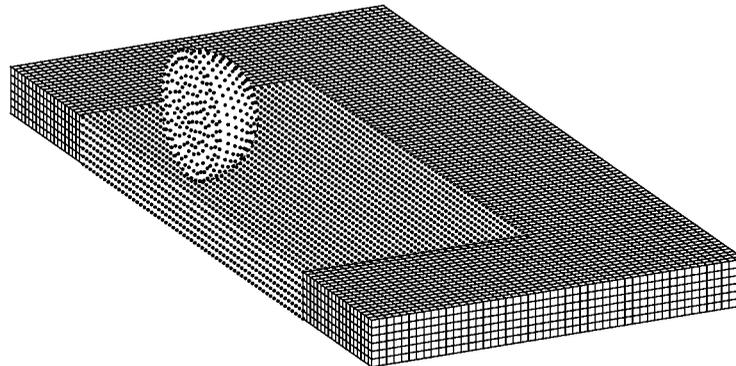
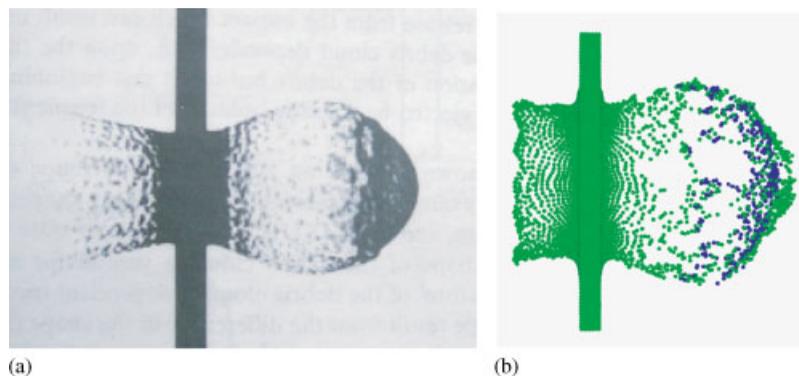


Figure 10. Impact of a projectile on a thin plate.

Figure 11. Comparison of the debris cloud at $9 \mu\text{s}$: (a) experimental result [21]; and (b) EMPFE-3D result.

and are solved in the FE mesh elsewhere. The projectile is modelled by 4984 tetrahedron elements with 1085 nodes, and the plate is modelled by 30 000 hexahedron elements with 36 057 nodes. The total number of degree of freedom is 111 426.

Figure 11(a) shows a typical debris cloud from hyper-velocity impact of a spherical projectile into a thin plate [21]. The computed debris cloud by EMPFE-3D is shown in Figure 11(b). The computational result compares favourably with the experiment in that the configuration of the debris clouds agree.

In MPM, there is no interaction between material particles separated by grid cells, such as particles A' and B' in Figure 12. Therefore, the interaction between particles A and B

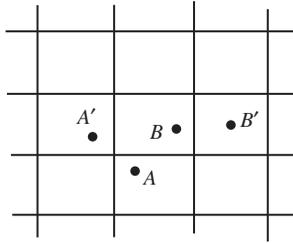


Figure 12. Fracture simulation in MPM.

will vanish when they move to the positions of A' and B' , which is similar to the fracture phenomena. It implies that the MPM possess a intrinsic fracture analysis capacity whose fracture criteria is the maximum space criteria, namely fracture occurs when the space between two particles is greater than the grid spacing. Hence, fracture can be simulated approximately in MPM without using any special material model.

We have investigated several other hyper-velocity impact problems using MPM, such as impact of projectile on thin plates and thick plates, and found that the numerical results agree well with those obtained by experiments and other numerical methods. A material model including fracture will be included in EMPFE-3D to refine the fracture analysis.

6. CONCLUSION

An explicit material point finite element method is proposed and a computer code EMPFE-3D is developed. Numerical examples have been presented to demonstrate the capability of the EMPFE-3D code for 3D impact problem with impact velocity from 190 to 6600 m/s.

The proposed method is very simple and straightforward to be implemented in an existing FE code, such as DYNA3D. It combines the advantages of the Lagrangian and Eulerian description, and is free from difficulties arising from severe mesh distortion. The method is promising in the simulation of hyper-velocity impact problems.

Although the computational grid is taken to be fixed in space in the computation, it might be updated adaptively to further improve the accuracy of the solution.

It should be noted that the MPFE method connects MPM and FE by using the FE nodes in the distorted region as material points. Therefore, the FE mesh is quite fine compared to the background MPM computational grid. Since the resolved length scale in MPM is the background grid spacing, this leads to a less refined calculation in the more highly deformed region and a fine mesh in the regions that are only deforming slightly. An adaptive scheme can be used to refine the computational grid in the highly deformed region.

REFERENCES

1. Zienkiewicz OC, Taylor RL. *The Finite Element Method* (5th edn). Butterworth-Heinemann: Stoneham, MA, 2000.
2. Liu WK, Belytschko T, Chang H. An arbitrary Lagrangian–Eulerian finite element method for path-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1986; **58**:227–245.

3. Gadala MS, Wang J. ALE formulation and its application in solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 1998; **167**:33–55.
4. Belytschko T, Krongauz Y, Organ D *et al.* Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.
5. Li S, Liu WK. Meshfree and particle methods and their applications. *Applied Mechanics Review* 2002; **55**:1–34.
6. Liu GR, Liu MB. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific: Singapore, 2003.
7. Atluri SN, Zhu TL. New concepts in meshless methods. *International Journal for Numerical Methods in Engineering* 2000; **47**(1–3):537–556.
8. Johnson GR, Stryk RA, Beissel SR. SPH for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1–4):347–373.
9. Zhang X, Liu XH, Song KZ *et al.* Least-squares collocation meshless method. *International Journal for Numerical Methods in Engineering* 2001; **51**:1089–1100.6.
10. Sulsky D, Chen Z, Schreyer HL. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:179–196.
11. Sulsky D, Zhou SJ, Schreyer HL. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**:136–252.
12. Brackbill JU, Ruppel HM. FLIP: a method for adaptively zoned, particle-in-cell calculations in two dimensions. *Journal of Computational Physics* 1986; **65**:314–343.
13. Brackbill JU, Kothe DB, Ruppel HM. FLIP: a low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications* 1988; **48**:25–38.
14. Hallquist JO. *LS-DYNA Theoretical Manual*. Livermore Software Technology Corporation, 1998.
15. Flangan DP, Belytschko T. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering* 1981; **17**:679–706.
16. Johnson GR, Cook WH. A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. *The Seventh International Symposium on Ballistics*, The Hague, The Netherlands, April 1983.
17. Johnson GR, Holmquist TJ. Evaluation of cylinder-impact test data for constitutive model constants. *Journal of Applied Physics* 1988; **64**:3901–3910.
18. Kohn BJ. Compilation of Hugoniot equation of state. *Technical Report No. AFWL-TR-69-38*, 1969.
19. Predebon WW, Anderson CE, Walker JD. Inclusion of evolutionary damage measures in Eulerian wavecodes. *Computational Mechanics* 1991; **7**:221–236.
20. Bardenhagen SG, Brackbill JU, Sulsky D. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering* 2000; **187**:529–541.
21. Anderson CE, Timothy J, Trucano G *et al.* Debris cloud dynamics. *International Journal of Impact Engineering* 1990; **9**(1):89–113.