Structured Mesh Refinement in Generalized Interpolation Material Point (GIMP) Method for Simulation of Dynamic Problems

Jin Ma, Hongbing Lu, and Ranga Komanduri[1]

Abstract: The generalized interpolation material point (GIMP) method, recently developed using a C^1 continuous weighting function, has solved the numerical noise problem associated with material points just crossing the cell borders, so that it is suitable for simulation of relatively large deformation problems. However, this method typically uses a uniform mesh in computation when one level of material points is used, thus limiting its effectiveness in dealing with structures involving areas of high stress gradients. In this paper, a spatial refinement scheme of the ~~structural~~ Structured grid for GIMP is presented for simulations with highly localized stress gradients. A uniform ~~structural~~ Structured background grid is used in each refinement zone for interpolation in GIMP for ease of generating and duplicating ~~structural~~ structured grid in parallel processing. The concept of influence zone for the background node and transitional node is introduced for the mesh size transition. The grid shape function for the transitional node is modified accordingly, whereas the computation of the weighting function in GIMP remains the same. Two other issues are also addressed to improve the GIMP method. The displacement boundary conditions are introduced into the discretization of the momentum conservation equation in GIMP, and a method is implemented to track the deformation of the material particles by tracking the position of the particle corners to resolve the problem of artificial separation of material particles in GIMP simulations. Numerical simulations of several problems, such as tension, indentation, stress concentration and stress distribution near a crack (mode I crack problem) are presented to ~~validate~~ verify this refinement scheme.

keyword: GIMP, Material Point Method, Mesh refinement R, FEM

1 Introduction

The material point method (MPM) uses a collection of material points, mathematically represented by Dirac delta functions to represent a material continuum (Sulsky, Zhou, and Schreyer (1995); Hu and Chen (2003); Guilkey and Weiss (2003)). A spatially fixed background grid, and interpolation between grid nodes and material points are introduced to track physical variables carried by the material points in the Lagrangian description. Field equations are solved on the background grid in the Eulerian description. Physical variables are interpolated from the solutions on the background grids to material points back and forth for solution and convection of physical variables. In general, the isoparametric shape functions, same as those used in the finite element method (FEM), are used. As the MPM simulation is independent of the background grid, a ~~structural~~ Structured grid is usually employed for purposes of simplicity. The movement of the material points represents the deformation of the continuum. MPM has been demonstrated to be capable of handling large deformations in a natural way (Sulsky, Zhou, and Schreyer (1995)). However, primarily due to the discontinuity of the gradient of the interpolation function at the borders of the neighboring cells, artificial noise can be introduced when the material points move just across the grid cell boundaries, leading to simulation instability for MPM. The generalized interpolation material point (GIMP) method, introduced by Bardenhagen and Kober (2004) can resolve this problem. In GIMP a C^1 continuous interpolation function is used and each material point/particle occupies a region. GIMP has been demonstrated to be stable and capable of handling relatively large deformations (Ma, Lu, Wang, Roy, Hornung, Wissink, and Komanduri (2005)).

The current MPM typically uses a uniform background mesh for solving the field equations. However, this is not efficient when stress gradients are high such as stress concentrations in a plate with a hole, or the stress field of a workpiece under indentation. In contrast, transitional

[1] Correspondence author, Tel: 405-744-5900; Fax: 405-744-7873; e-mail: ranga@ceat.okstate.edu All authors are with the School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078

mesh is effective in solving problems involving rapidly varying stress in an area. Wang, Karuppiah, Lu, Roy, and Komanduri (2005) have presented a method using an irregular background mesh to deal with problems involving rapidly varying stress, such as stress field near a crack. However, this approach does not use regular structured background mesh so that mesh generation encounters the same difficulty as FEM, and leads to the loss of some advantage of MPM on the ease of generating mesh for a complex problem.

The use of structured grid in GIMP has facilitated the implementation of GIMP in parallel processing. A refinement scheme based on splitting and merging material particles was proposed by Tan and Nairn (2002). Recently, a multilevel refinement algorithm has been developed for parallel processing using the structured adaptive mesh refinement application infrastructure (SAMRAI) (Hornung and Kohn (2002); Ma, Lu, Wang, Roy, Hornung, Wissink and Komanduri (2005)). The computational domain was divided into multiple nested levels of refinement. Each grid level is uniform but has a different cell size. Smaller material particles and smaller cell sizes are used in each finer level. Two neighboring levels are connected by overlapped material particles of the same size and data communication between levels is performed at predefined intervals. However, the refinement through material particles requires extra communication and simulation time. In this paper, a refinement for GIMP based on the transitional grid nodes is developed. This refinement is natural and does not involve extra simulation time. Moreover, the refined grid remains uniformly structured in each refinement region.

While the problem associated with artificial noise has been resolved with the use of GIMP method, it has been observed recently that material separation could occur if the deformation of the material particles was not tracked, Guilkey (2005). Tracking the deformation of material particles properly in GIMP is necessary especially when the material particles are stretched. In this paper, an approach is developed for tracking the particle deformation to resolve material point separation problem. This paper focuses on the refinement scheme for ~~structural~~ Structured grid. Several numerical problems, such as tension, indentation, stress concentration and stress distribution near a crack (mode I crack problem) were simulated to ~~validate~~ verify this refinement algorithm, as well as to demonstrate the effectiveness of tracking particle deformations.

2 GIMP

For the purpose of completeness, the basic equations in GIMP (Bardenhagen and Kober (2004)) are summarized here. In dynamic simulations, the mass and momentum conservation equations are given by

$$\frac{d\rho}{dt} + \rho\nabla\cdot\mathbf{v} = 0, \text{ and} \tag{1}$$

$$\rho\mathbf{a} = \overset{\nabla\cdot\sigma}{\underline{\nabla\boldsymbol{\sigma}}} + \mathbf{b} \quad \text{in } \Omega, \tag{2}$$

where ρ is the material density, $\mathbf{a}$ is the acceleration, σ and $\mathbf{b}$ are the Cauchy stress and body force density, respectively. The displacement and traction boundary conditions are given as

$$\mathbf{u} = \overline{\mathbf{u}} \text{ on } \partial\Omega_u, \tag{3}$$

$$\boldsymbol{\tau} = \overline{\boldsymbol{\tau}} \text{ on } \partial\Omega_\tau, \tag{4}$$

space

where $\partial\Omega_u \subset \partial\Omega, \partial\Omega_\tau \subset \partial\Omega\text{and}\partial\Omega_u \cap \partial\Omega_\tau = 0$.In variational form, the momentum conservation equation can be written as

$$\int_\Omega \rho\mathbf{a}\cdot\delta\mathbf{v}d\mathbf{x} = \int_\Omega \nabla\boldsymbol{\sigma}\cdot\delta\mathbf{v}d\mathbf{x} + \int_\Omega \mathbf{b}\cdot\delta\mathbf{v}d\mathbf{x} - \alpha\int_{\partial\Omega_u}(\mathbf{u}-\overline{\mathbf{u}})\cdot\delta\mathbf{v}d\mathbf{x}, \tag{5}$$

space

where$\delta\mathbf{v}$ is an admissible velocity field, α is a penalty parameter we introduce herein to impose the essential boundary conditions and $\alpha >> 1$, Atluri and Zhu (1998), Atluri and Zhu (2000). Applying the chain rule, $\underline{\nabla\boldsymbol{\sigma}}\cdot\delta\mathbf{v} = \nabla(\boldsymbol{\sigma}\cdot\delta\mathbf{v}) - \boldsymbol{\sigma} : \nabla\delta\mathbf{v}$, and the divergence theorem, $\nabla\cdot\sigma$ Eq. (5) can be written as

$$\int_\Omega \rho\mathbf{a}\cdot\delta\mathbf{v}d\mathbf{x} + \int_\Omega \boldsymbol{\sigma} : \nabla\delta\mathbf{v}d\mathbf{x} = \int_\Omega \mathbf{b}\cdot\delta\mathbf{v}d\mathbf{x} + \int_{\partial\Omega_\tau} \overline{\boldsymbol{\tau}}\cdot\delta\mathbf{v}dS + \int_{\partial\Omega_u} \boldsymbol{\tau}_u\cdot\delta\mathbf{v}dS - \alpha\int_{\partial\Omega_u}(\mathbf{u}-\overline{\mathbf{u}})\cdot\delta\mathbf{v}dS \tag{6}$$

where τ_u is the resultant traction due to the displacement boundary condition on $\partial\Omega_u$. In GIMP, the domain Ω is discretized into a collection of material particles, with Ω_p as the domain of particle p. The physical quantities, such as the mass, stress and momentum can be defined for each particle. For example, the momentum for particle p can be expressed as $\mathbf{p}_p = \int\limits_{\Omega_p} \rho(\mathbf{x})\mathbf{v}(\mathbf{x})\chi_p(\mathbf{x})d\mathbf{x}$, where $\mathbf{v}(\mathbf{x})$ is the velocity and $\chi_p(\mathbf{x})$ is the particle characteristic function. The momentum conservation equation can be discretized as

$$\begin{aligned}\sum_p \int\limits_{\Omega\cap\Omega_p} \frac{\dot{\mathbf{p}}_p\chi_p}{V_p}\cdot\delta\mathbf{v}d\mathbf{x} + \sum_p \int\limits_{\Omega\cap\Omega_p} \sigma_p\chi_p : \delta\mathbf{v}d\mathbf{x} \\ = \sum_p \int\limits_{\Omega\cap\Omega_p} \frac{m_p\chi_p}{V_p}\mathbf{b}\cdot\delta\mathbf{v}\mathbf{dx} + \sum_p \int\limits_{\partial\Omega_\tau\cap\Omega_p} \overline{\boldsymbol{\tau}}\cdot\delta\mathbf{v}d\mathbf{x} \\ + \sum_p \int\limits_{\partial\Omega_u\cap\Omega_p} \boldsymbol{\tau}_u\cdot\delta\mathbf{v}dS - \alpha\sum_p \int\limits_{\partial\Omega_u\cap\Omega_p} (\mathbf{u}-\overline{\mathbf{u}})\cdot\delta\mathbf{v}dS \quad (7)\end{aligned}$$

where $V_p = \int\limits_{\Omega\cap\Omega_p} \chi_p(\mathbf{x})d\mathbf{x}$ is the particle volume. Introducing a background grid and the grid shape function $S_i(\mathbf{x})$ that satisfies partition of unity $\sum\limits_i^{S_i(\mathbf{x})} = 1$, the admissible velocity field can be represented by the grid nodal data as $\delta\mathbf{v} = \sum\limits_i \delta\mathbf{v}_i S_i(\mathbf{x})$. Without the loss of generality, we take $\mathbf{u}$ in Eq. (7) to be the displacement of the boundary particles at the current time step and $\tau_u = \boldsymbol{\sigma}_p\mathbf{n}_u$ where $\mathbf{n}_u$ is the unit outward normal to $\partial\Omega_u$. The momentum conservation, Eq. (7), can eventually be written for each node i as

$$\dot{\mathbf{p}}_i = \mathbf{f}_i^{int} + \mathbf{f}_i^b + \mathbf{f}_i^\tau + \mathbf{f}_i^u, \quad (8)$$

where the time rate ~~of~~ change of nodal momentum $\dot{\mathbf{p}}_i = \sum\limits_p \overline{S}_{ip}\mathbf{p}_p/\Delta t$, the nodal internal force vector $\mathbf{f}_i^{int} = -\sum\limits_p \boldsymbol{\sigma}_p\cdot\overline{\nabla S}_{ip}V_p$, the nodal body force vector $\mathbf{f}_i^b = \sum\limits_p m_p\mathbf{b}\overline{S}_{ip}$ and the nodal traction force vector $\mathbf{f}_i^\tau = \sum\limits_p \int\limits_{\partial\Omega_\tau\cap\Omega_p} \overline{\boldsymbol{\tau}}S_i(\mathbf{x})dS$. $\mathbf{f}_i^u$ is the force vector induced by the essential boundary condition given by

$$\mathbf{f}_i^u = \sum_p \int\limits_{\partial\Omega_u\cap\Omega_p} \boldsymbol{\sigma}_p\mathbf{n}_u S_i(\mathbf{x})dS - \alpha\sum_p \int\limits_{\partial\Omega_u\cap\Omega_p} (\mathbf{u}_p - \overline{\mathbf{u}})S_i(\mathbf{x})dS. \quad (9)$$

$\overline{S}_{ip}$is weighting function between particle p and node i given as

space

$$\overline{S}_{ip} = \frac{1}{V_p}\int\limits_{\Omega\cap\Omega_p} \chi_p(\mathbf{x})S_i(\mathbf{x})d\mathbf{x}. \quad (10)$$

The weighting function in GIMP is C^1 continuous and satisfies partition of unity. The momentum conservation, Eq. (8), can be solved at each node to update the nodal momentum, acceleration, and velocity. These updated nodal quantities can be interpolated to the material particles to update the particles, as given by Bardenhagen and Kober (2004). It may be noted that the mass of each material particle does not change, so that the mass conservation equation is satisfied automatically.

In the discretization of the weak form of the momentum conservation equation, a background grid is used. However, the computation is independent of the grid from one increment to another. Hence a spatially fixed structured grid can be used for convenience. In the background grid, no nodal connectivity is required and the integration is never performed on the element domain. Similar characteristics have been reported for other meshless methods, such as the meshless local Petrov-Galerkin (MLPG) method, Atluri and Shen (2002).

For a uniform ~~structural~~ grid, the grid shape function in 3D is defined as the product of three nodal tent functions (Bardenhagen and Kober (2004)) structured

$$S_i(\mathbf{x}) = S_i^x(x)S_i^y(y)S_i^z(z), \quad (11)$$

in which the nodal tent functions are in the same form, e.g.,

$$S_i^x(x) = \begin{cases} 0 & x - x_i \le -L_x \\ 1 + (x - x_i)/L_x & -L_x \le x - x_i \le 0 \\ 1 - (x - x_i)/L_x & 0 \le x - x_i \le L_x \\ 0 & L_x \le x - x_i \end{cases} \quad (12)$$

Fig. 1 shows one 2D grid cell with four nodes. In this paper, the particle characteristic function of the material particle located at $(\mathrm{x}_p, \mathrm{y}_p)$ is taken as

$$\chi_p(\mathbf{x}) = \chi_p^x(x)\chi_p^y(y), \quad (13)$$

where $\chi_p^x(x) = H[x - (x_p - l_x)] - H[x - (x_p + l_x)]$ and H denotes the step function.

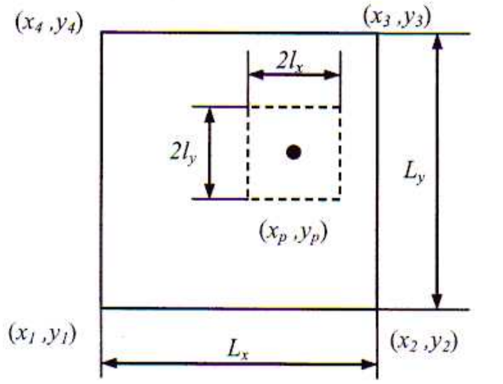


Figure 1 : 2D representation of a particle and a grid cell

Lower case

3 Structured Mesh Refinement

In this section, a refinement scheme for a ~~structural~~ structured mesh in GIMP is described. Since GIMP shares some characteristics with meshless methods, we expect the GIMP method to have h convergence in simulation (Atluri and Shen (2002)). The momentum conservation equation is essentially solved at each node (see Eq. (8)). Therefore, the number of equations to be solved is the same as the number of nodes. Finer grid and smaller material particles will lead to more accurate results. In some simulations, high stress gradients exist in small regions. For instance, in indentation with a sharp tip, the stress gradient is high in the workpiece beneath the indenter tip. In simulation of fracture problems, the stress gradient at the crack tip is high and of particular interest. Consequently, finer grid is needed for these regions; but away from these regions, a coarser grid can be used to reduce the computational cost. In conclusion, a uniform grid can be either too computationally expensive if it is too fine, or inaccurate, if it is coarse. A non-uniform grid with refinement can provide accurate results while minimizing the overall computational time.

Grid refinement should maintain the same characteristics of the structured grid as much as possible, in order to replicate the grid generation in parallel processing. The proposed refinement scheme is illustrated in Fig. 2 with one particle per cell assigned. The material particles that fill each cell are square in nature but denoted as circles for clarity. To understand this grid, one can consider that there are two overlapped structured grids. The coarse grid covers a rectangular region from (0, 0) to (8, 6) and the fine grid covers a region from (2, 2) to (6, 6). For each grid, the shape function can be evaluated from Eq. (12). To be consistent with any other general refinement, it is required that the region of the fine grid to be smaller than the coarse grid.

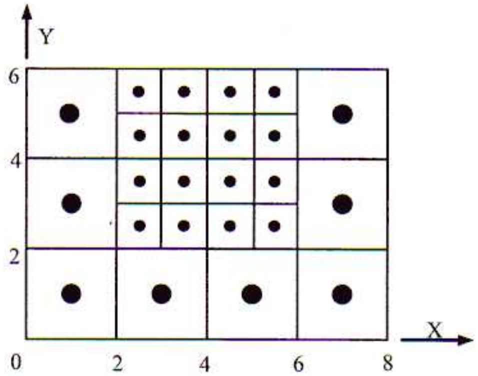
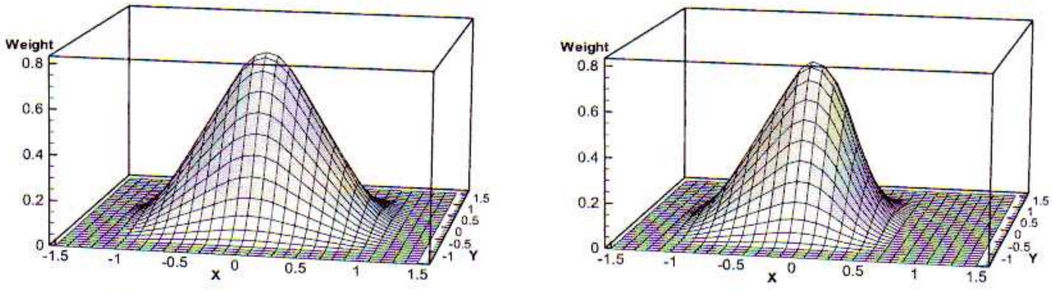


Figure 2 : Refinement of structured grid with a refinement ratio of two

When these two grids are merged into one, the shape function and the weighting function for the nodes at the boundary of the finer grid, for example, the nodes at (2, 2) and (2, 3) should be changed. These nodes are called transition nodes. To facilitate the computation of the interpolation function, we define an influence zone for each node, denoted as $[L_x^-, L_x^+, L_y^-, L_y^+]$ in 2D or $[L_x^-, L_x^+, L_y^-, L_y^+, L_z^-, L_z^+]$ in 3D. The symbols in the square bracket define the size of the influence zone, whereas the subscript denotes the coordinate axis and the superscript denotes the direction of the axis. For example, L_x^- and L_x^+ represent the sizes in the negative and positive X direction, respectively. The influence zone for each node in 2D is rectangular and it extends to the next immediate grid line to the left, right, bottom and top of the node. If no more grid lines exist in any direction, such as the boundary nodes, the size is zero in that direction. For example, in the refined grid in Fig. 2, the influence zones for the nodes at (2, 3) and (2, 4) are both [2, 1, 1, 1]. The influence zone for the node at (2, 2) is [2, 1, 2, 1]. Based on this definition, the influence zone for ~~the~~ node this ~~at the same location~~ in the coarse grid is [2, 2, 2, 2], and in the uniformly fine grid is [0, 1, 0, 1].

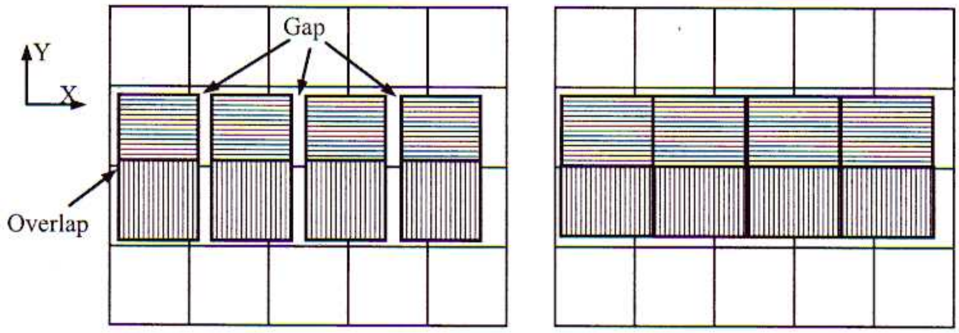
Based on the influence zone, the nodal tent function in each direction can be modified as, for example, in the X



(a) Influence zone: [1, 1, 1, 1]

(b) Influence zone: [1, 0.5, 1, 1]

Figure 3 : Effect of influence zone on the weighting functions



(a) No particle deformation

(b) Actual particle deformation

Figure 4 : Schematics showing overlaps and gaps that may occur when particle deformation is not tracked

direction,

$$S_i^x(x) = \begin{cases} 0 & x - x_i \leq -L_x^- \\ 1 + (x - x_i)/L_x^- & -L_x^- \leq x - x_i \leq 0 \\ 1 - (x - x_i)/L_x^+ & 0 \leq x - x_i \leq L_x^+ \\ 0 & L_x^+ \leq x - x_i \end{cases} \quad . \tag{14}$$

Eq. (14) can be substituted into the grid shape function (Eq. (11)), and the weighting function between the particle p and the node i can be evaluated as

$$\overline{S}_{ip}^x = \begin{cases} 0 & B \leq -L_x^- \; or \, A \geq L_x^+ \\ \frac{b-a+(b^2-a^2)/(2L_x^-)}{2l_p} & b \leq 0 \\ \frac{b-a-(b^2-a^2)/(2L_x^+)}{2l_p} & a \geq 0 \\ \frac{b-a-a^2/(2L_x^-)-b^2/(2L_x^+)}{2l_p} & otherwise \end{cases} \tag{15}$$

where $A = x - x_i - l_p$, $B = x - x_i + l_p$, $a = \max(A, \ -L_x^-)$ and $b = \min(B, \ L_x^+)$. When $L_x^- = L_x^+$, Eqs. (14) and (15) are degraded to the cases for uniform grid. Without detailed proof, the grid shape function and the weighting function still satisfy partition of unity. Similarly, the gradient of the modified weighting function can be computed.

It may be noted that in Fig. 2 the refinement ratio is two, i.e., the length of a side of a coarse cell is twice that of the fine cell. To maintain the convenience of the structured grid, only integer refinement ratio should be used. All nodal positions can be computed from the domain of each grid and the cell sizes. The proposed refinement scheme can be applied to any integer refinement ratio and for multiple times for successive refinements. As an example, the weighting function between a particle of size 0.5×0.5 and a node at (0, 0) with an influence zone of [1, 1,1, 1] is shown in 4 (a). The particle is on the X-Y plane and the weighting function is computed at each particle position. For comparison, the influence zone is changed to [1, 0.5, 1, 1], representing a transitional node,

Fig.

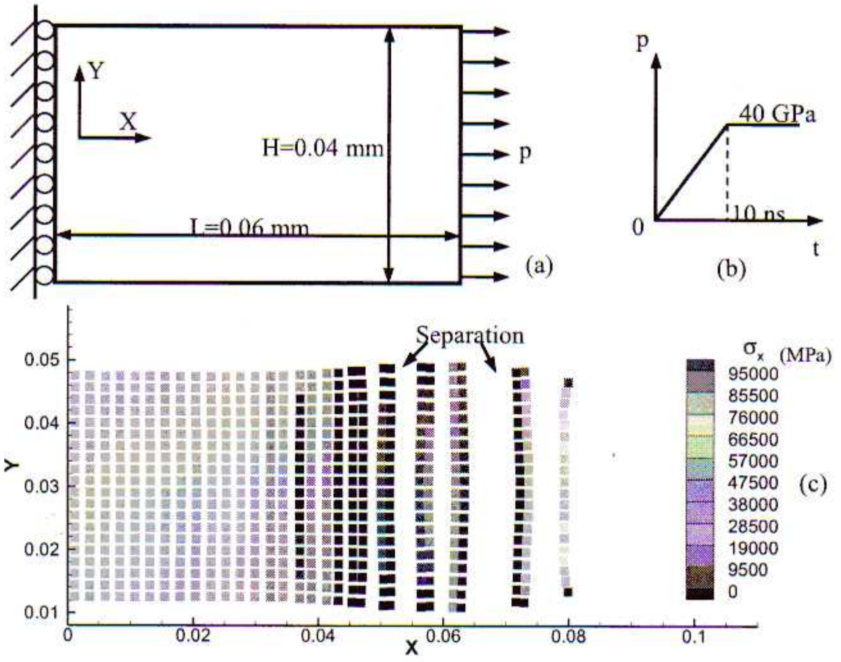


Figure 5 : Simulation showing separation when the particle deformation is tracked by strain

while other conditions are the same. The weighting function for this case is plotted in Fig. 4 (b). It can be seen that the weighting function for the transitional node is still C^1 continuous.

4 Numerical simulations

4.1 Tracking particle deformation

Prior to presenting the results of structured mesh refinement, tracking particle deformation is addressed first since this is necessary in later simulations to achieve accuracy. The material particles are initialized into regular shapes, normally square and cube for 2D and 3D simulations, respectively. All the physical quantities in a particle domain are considered to be uniform. The shape of the particle changes during deformation. So, it is important to track the deformation of each particle. Fig. 4 illustrates the deformation of the particles in 2D when the particles are stretched in the X-direction. If the particle deformation is not tracked, gaps will form between neighboring particles in the X-direction, as shown in Fig. 4 (a). Due to the Poisson's effect, there will be overlapping between particles in the Y-direction, if the particles do not follow the deformation of the materials properly. When the stretch and gaps are large enough, the particles would be separated. Fig. 4 (b) shows the correct deformation in which contiguous particles remain contiguous after deformation.

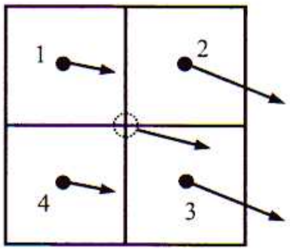
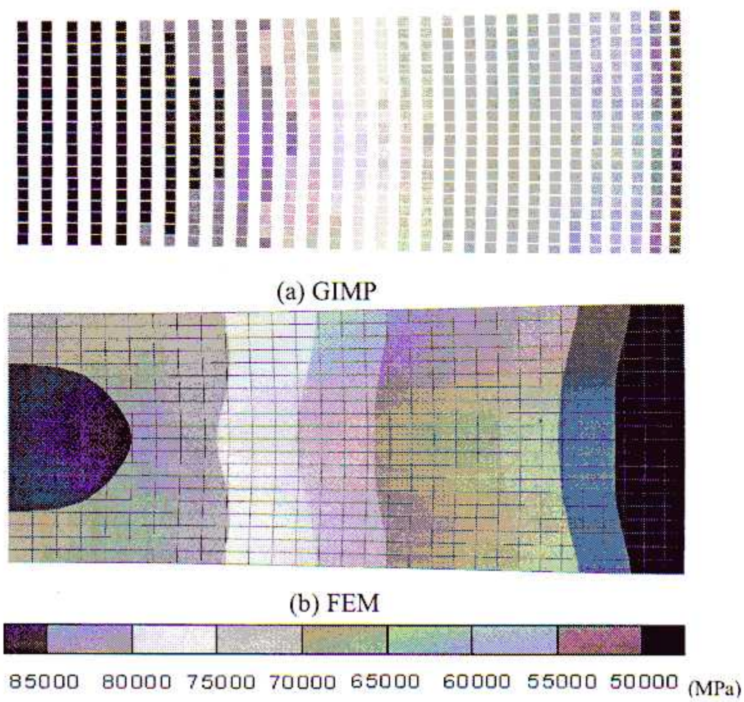


Figure 6 : Velocity field of a continuum region (the arrows represent both direction and magnitude)

To track the particle deformation, a convenient approach is to calculate the deformed particle shape based on strain history. Since the linear strain increment is computed in GIMP, the effectiveness and validity of tracking particle deformation based on strain would be limited to relatively small deformations. As an example, Fig. 5 (a) shows the simulation of a uniaxial tension problem under plane strain conditions. The material is silicon. Its mass density is 2.71 g/cm^3, Young's modulus 175.8 GPa, and the Poisson's ratio 0.28. The background grid size is 0.002 × 0.002 mm^2. One particle per cell is assigned initially and the time step is 0.02 ns (1 ns = 10^{-9} s). The applied



(a) GIMP

(b) FEM

Figure 7 : GIMP results with tracking deformation of corners and their comparison with FEM

pressure increases linearly with time from 0 to 10 ns and is then maintained constant, as shown in Fig. 5 (b). The elongation in the X-direction of the particle is computed as $(1+\varepsilon_x)l_x^0$, where $l_x^0$is the initial length of the particles. Separation of the particles occurred at $\sim$25% strain at ~~when~~ 6 ns before the full pressure was applied, as shown in Fig. 5 (c). Similar problems have been reported by Guilkey (2005).

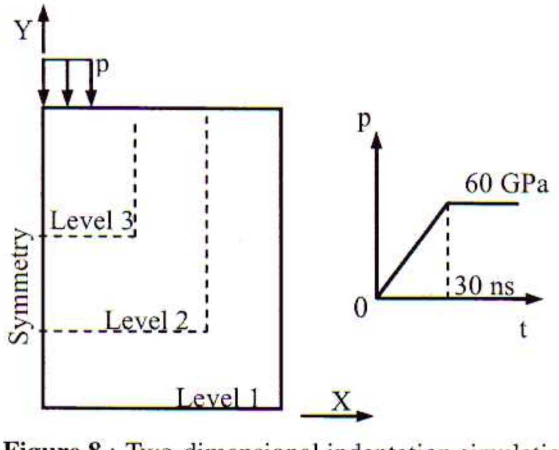
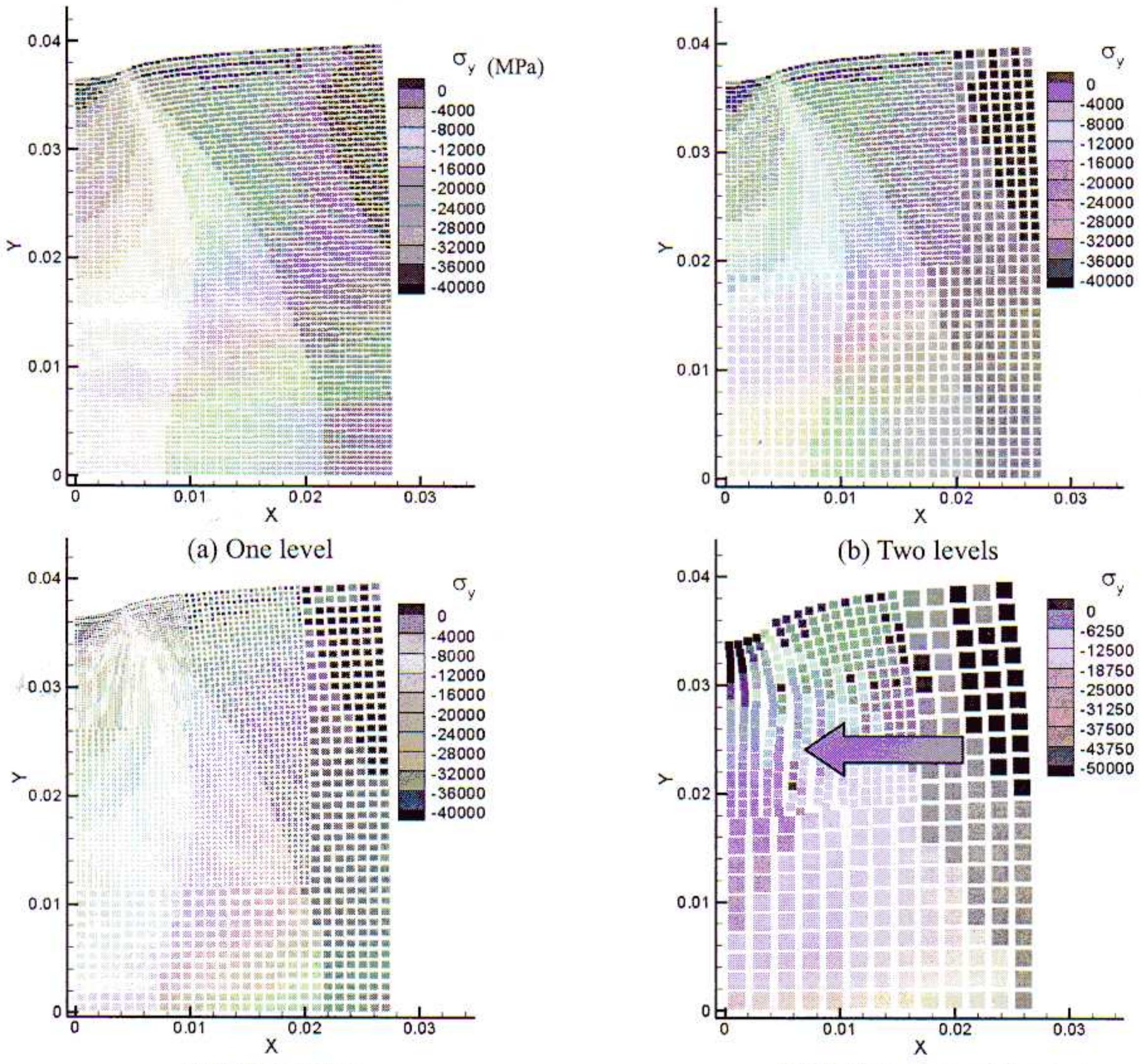


Figure 8 : Two-dimensional indentation simulation

Tracking particle deformation by strain could be more effective if nonlinear strain is used in the GIMP method. However, recovery of the deformed shape based on nonlinear strain involves additional complications. Based on the GIMP algorithm, there is another convenient approach to track the particle deformation. Numerically, the displacement and velocity of each particle in GIMP are computed at the center of the particle to represent the entire particle domain. However, in reality, the velocity and deformation at the corners of a particle can be different from the center. ~~4.2~~ Fig. 6 shows four 2D contiguous particles sharing one common corner point at the middle. This corner point should have unique displacement and velocity. As a result, it is helpful to track the displacement and velocity of each corner to track the particle deformation.

It is not difficult to compute the velocity of the particle corner given its location. It is computed from the interpolation from the background grid, similar to the center of the particle. For a 2D particle, in addition to updating the position of the center of the particle, the positions of the four corners are updated at each increment. To compute the weighting function for the corners, a fictitious size can be assigned to each corner. Numerical simu-



(a) One level

(b) Two levels

(c) Three levels

(d) Deformation not tracked

Figure 9 : Comparison of the stress distributions at different levels of refinements in force indentation

lation shows that the result is not sensitive to this size in the range of 10% to 80% of the initial particle size. The new particle shape can be obtained by connecting the four corners with straight lines. In order to avoid numerical integration of the interpolation function, it is assumed that the deformed material particle shape is rectangular with edges parallel to the coordinate axes. The size of the rectangle is, therefore, determined from the extent of the corners. As will be demonstrated later in this section, this assumption does not introduce any significant error while it can greatly improve the efficiency of the GIMP algorithm.

Using this approach to track the particle deformation, the problem in Fig. 5 (a) was simulated again with GIMP and the results at 20 ns are plotted in Fig. 7 (a). No separation of particles was seen during the entire simulation up to 50% strain. It is noted that each material particle is plotted as a square of the same size and the particle deformation is not shown due to software limitations on visualization. For comparison, the same problem is simulated using FEM (Abaqus/Explicit) and the FE result is shown in Fig. 7 (b). It can be seen that these two sets of results agree reasonably well with each other; the maximum difference in maximum tensile stress is ∼8%.

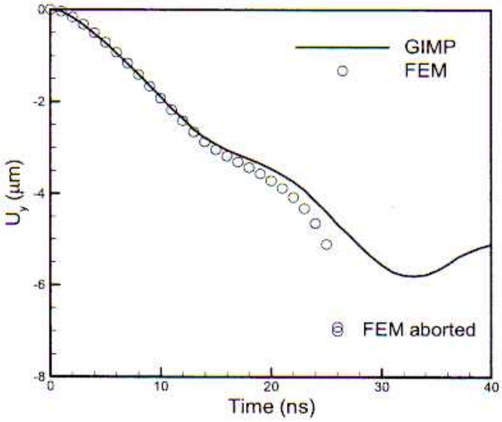


Figure 10 : Comparison of displacement history with FE

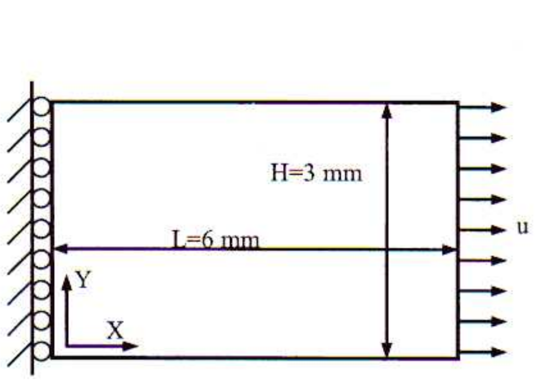
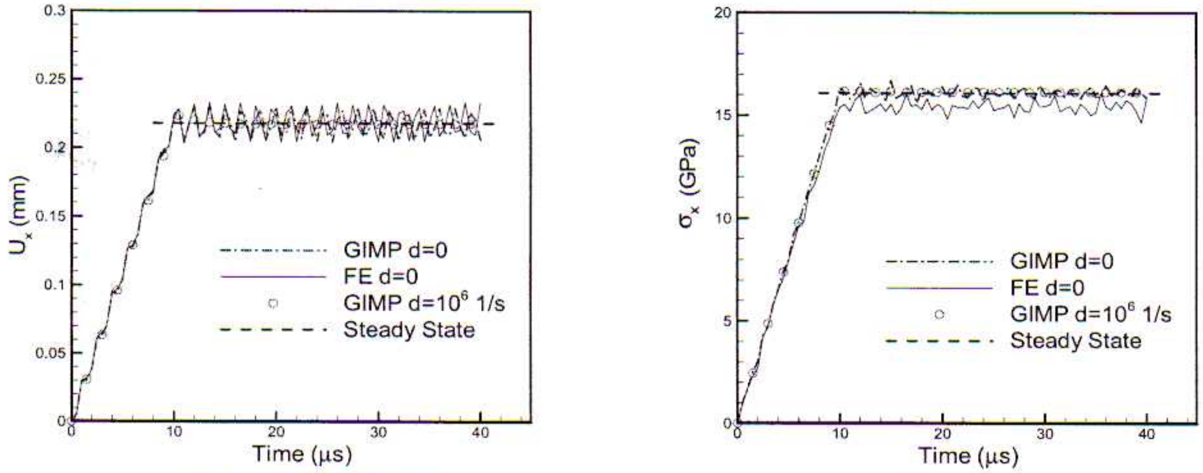


Figure 11 : A simple tension problem with displacement boundary conditions



(a) Displacement history

(b) Stress history

Figure 12 : Comparisons of the displacement and stress

4.2 *Indentation problem*

To ~~validate~~ verify the refinement algorithm, a 2D indentation problem was simulated. Pressure is applied at the top of the workpiece, as shown in ~~4.3~~ Fig. 8. The dashed lines indicate the borders of refinement levels. The work material is silicon with the properties the same as those given in the previous section. Due to symmetry with respect to the Y-axis, only half of the workpiece is modeled. The size of the model is 0.027 × 0.04 mm^2. The pressure increases linearly with time from t = 0 - 30 ns and is then kept constant at 60 GPa.

Several simulations were performed under different settings for the purpose of comparison. In the first simulation, a uniform grid with a cell size of 0.001 × 0.001 mm^2 is used and the time step is 20 ps (1 ps = 10^{-12} s). The stress distribution in the workpiece at t = 20 ns is shown in ~~4.4~~ Fig. 9 (a). In this figure, the units of length and stress are mm and MPa, respectively. In the second simulation, as indicated in ~~4.4~~ Fig. 9 (b), two levels of refinements are used with the refinement ratio to be 2. The cell lengths are 0.002 mm and 0.001 mm for the first and second levels, respectively. The fine level covers a rectangular area of the workpiece from (0, 0.02) to (0.02, 0.04). The grid is fixed in space; the material particles initially in the fine region move to the coarse region during deformation. As shown in ~~4.4~~ Fig. 9 (b), some fine material particles have moved below the line Y = 0.02 mm. It may be noted that in ~~4.4~~ Fig. 9 the material particles are plotted as squares corresponding to their initial sizes. Gaps between particles are intentionally shown to depict the particle sizes.

Three levels of refinements are used in the third simulation of the same problem. In this simulation, the time step is 10 ps and the results are shown in ~~4.4~~ Fig. 9 (c). The stresses agree very well with the previous two simulations. ~~4.4~~ Fig. 9 (d) shows the result when the particle deformation is not tracked. A severe material separation was observed at t = 36 ns, as indicated by the arrow. Additionally, the displacement history of the particle in the middle of the top surface for the third simulations is shown in Fig. 10. The same result of the integration point of the element in the middle of the top surface in FE simulation using ABAQUS/Explicit is also shown in Fig. 10 for comparison. It can be seen that the displacement compares well with FE up to t = ∼20 ns. After this time, FE simulation aborted due to mesh distortion. This demonstrates the capability of GIMP using a grid with ~~structural~~ Structured refinement in handling large deformations.

It may be noted that if the exterior corners of the surface particles are tracked from the nodal interpolations in the same way as the interior corners, simulations tend to become unstable due to erroneous surface corner displacements. This problem was observed to be strictly and consistently associated with the particles with external tractions applied. It is caused by insufficient nodal interpolation. To eliminate this problem, the exterior corners of the surface particles were tracked by strain only, as used in these simulations.

4.3 ~~Validation~~ Verification of the displacement boundary condition

We next describe the results on the ~~validation~~ verification of the displacement boundary conditions. For dynamic simulations, an artificial damping may be introduced. With damping, the nodal momentum can be updated as

$$\Delta\mathbf{p}_i = (\mathbf{f}_i^{int} + \mathbf{f}_i^{b} + \mathbf{f}_i^{\tau} + \mathbf{f}_i^{u} - d\mathbf{p}_i)\Delta t, \tag{16}$$

where d is the artificial damping coefficient.

A rectangular slab is fixed on the left and a displacement boundary condition is applied on the right, as shown in ~~4.5~~ Fig. 11. The material is silicon and its properties are given in the previous section. The cell size in this simulation is 0.5 × 0.5 mm^2 and four particles are assigned to each cell initially. The time step is 5 ns. The prescribed displacement increases linearly with time to 0.5 mm at t = 10 μs, which corresponds to a velocity of 50 m/s, and remains constant thereafter. This problem is simulated in FE using Abaqus/Explicit for comparison. The displacement in the X-direction, U$_x$, for a particle initially centered at (2.625, 1.625) as a function of time is plotted in Fig. 12 (a). It can be seen that without damping, the vibrations of displacement from both FEM and GIMP simulations are in phase before t = 14 μs, but out of phase afterwards. The dashed line represents the steady state displacement at this point. It can be seen that when an artificial damping of 10^6 s^{-1} is used, the GIMP solution converges quickly. The error of the converged displacement is 0.46%. Fig. 12 (b) shows the comparison of the normal stress in the X-direction. Good comparison between GIMP and FE has been obtained. With damping, the GIMP solution converges to the analytical solution for static simulations.

4.4 Stress concentration problem

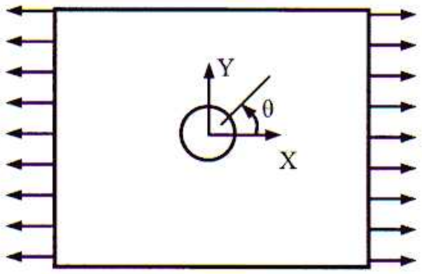
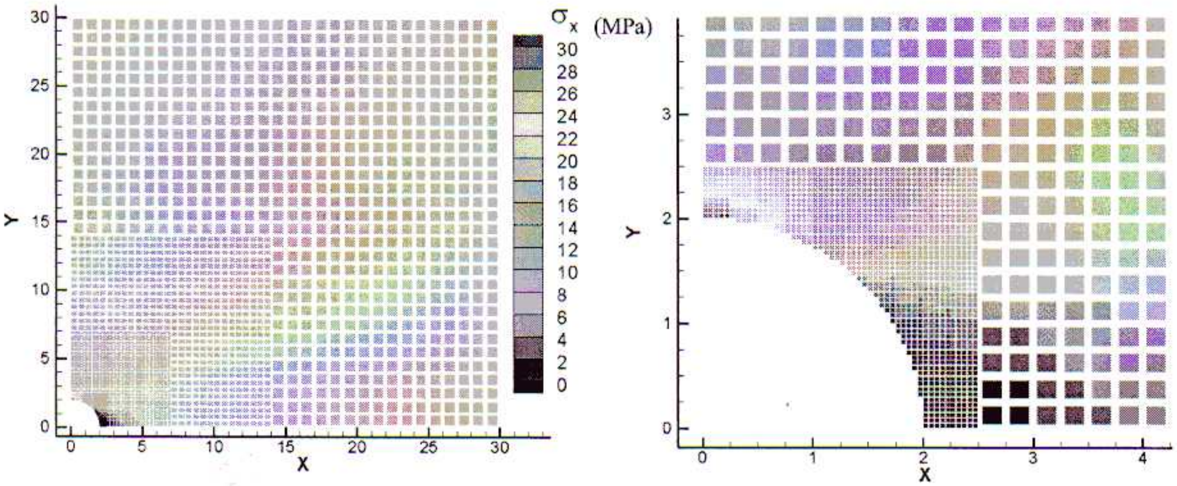


Figure 13 : A plate with a circular hole subjected to tension

Fig. 13 shows a copper plate (60 × 60 mm^2) with a central hole (4 mm diameter) subjected to a distributed load. This problem is simulated using GIMP as a dynamic problem with a damping factor of 1000 s^{-1} and three levels of refinement. The cell sizes at these three levels are 1.0 mm, 0.5 mm, and 0.25 mm, respectively. One particle is assigned to each cell not adjacent to the hole initially. The cells close to the circular hole are assigned 25 particles each to model the circular edge more accurately with the use of square particles. It may be noted that all the particles occupy square areas initially. The time step is 10 ns and the applied distributed load intensity p = 10 MPa. The stress distribution after 4000 steps is shown in Fig. 14 (a) and the area close to the hole (bottom left corner in Fig. 14 (a)) is magnified in



(a) Overall (b) Magnified

Figure 14 : Normal stress in the X-direction with p = 10 MPa

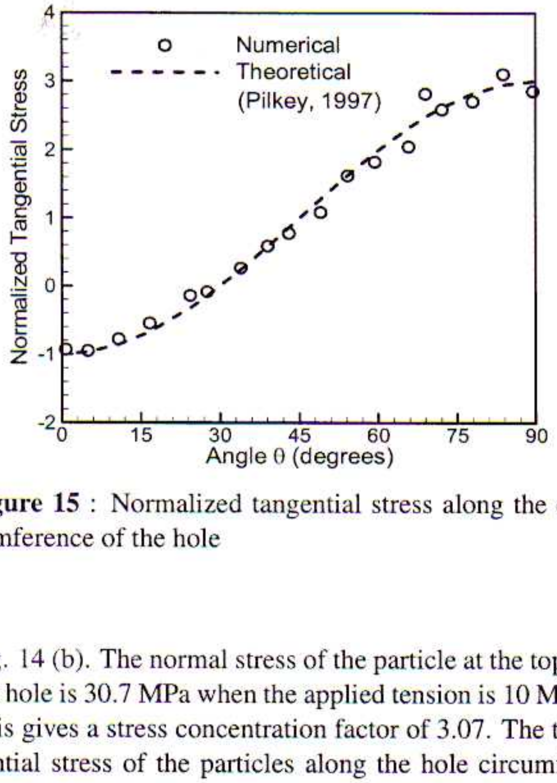


Figure 15 : Normalized tangential stress along the circumference of the hole

Fig. 14 (b). The normal stress of the particle at the top of the hole is 30.7 MPa when the applied tension is 10 MPa. This gives a stress concentration factor of 3.07. The tangential stress of the particles along the hole circumference, normalized by the applied pressure, is plotted in Fig. 15 in comparison with the theoretical value, Pilkey (1997). A good agreement between the numerical simulation and theoretical value is obtained. This demonstrates that the GIMP refinement algorithm is effective for problems involving significant stress gradients. Furthermore, with the use of small square particles, GIMP is capable of modeling curved surfaces.

4.5 *Static stress intensity factor*

We next use the GIMP refinement algorithm to determine the stress field and stress intensity factor for a mode-I no dash crack problem to determine the capability of the GIMP refinement algorithm in simulation of stress distribution near a crack. Guo and Nairn (2003) have successfully extended the MPM method to compute stress distribution in a plate with explicit cracks using multiple nodal fields along the crack surface. The physical quantities of material particles on each side of the crack were interpolated using variables in the field on that side of crack surface. In their simulation, a uniform mesh was used. Since the stress gradient at the crack tip is very high, a refined mesh near the crack tip and a coarse mesh in the far field should lead to savings in computational time while maintaining the same accuracy with the use of a uniform fine mesh. The fracture problem is thus an appropriate problem to evaluate the refined GIMP algorithm. For this purpose, we model the same fracture problem by Guo and Nairn (2003) using MPM, that is, a double cantilever beam (DCB) with a crack as shown in Fig. 16. In the area close to the crack tip, finer meshes are used, while in the area far away from the crack tip, coarse meshes are used. The thickness of the plate is 1 mm, thereby justifying a plane stress condition for this problem. The

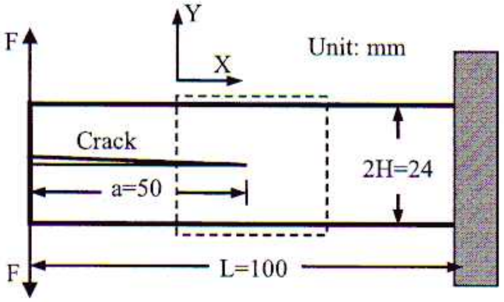


Figure 16 : Geometry of a double cantilever beam with a crack

material of the DCB is considered to be homogeneous, isotropic, and linearly elastic. Its density, Young's modulus, and Poisson's ratio are 1500 kg/m^3, 2300 GPa, and 0.33, respectively. The applied force is $F = 4 \times 10^{-4}$ N and this results in a mode I crack problem. The static stress intensity factor for the DCB can be calculated using the following equation (Kanninen 1973)

$$K_I = 2\sqrt{3}\frac{F(a+3H/2)}{H^{3/2}}. \tag{17}$$

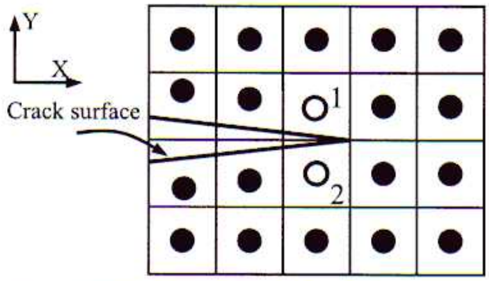
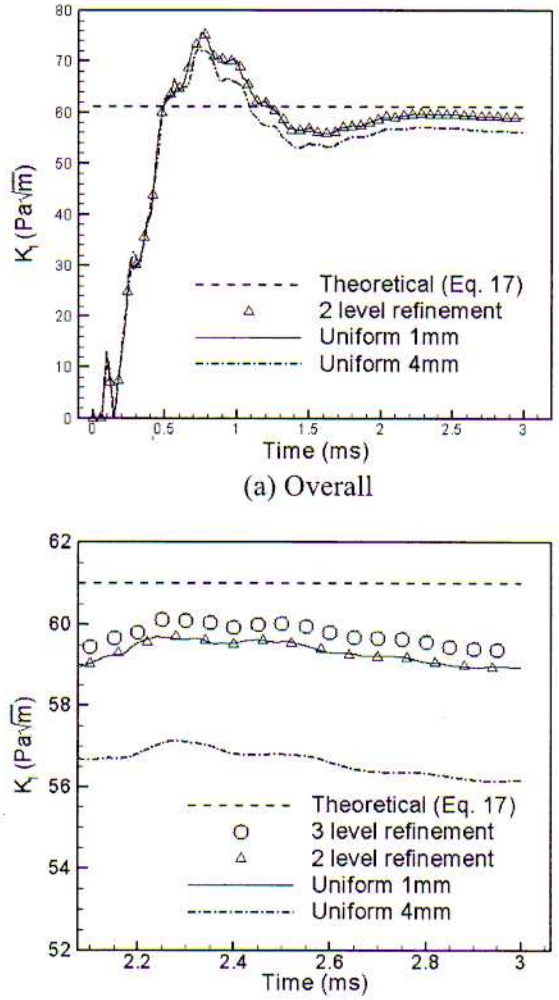


Figure 17 : Material points and background grid around the crack tip

This problem has been simulated using MPM with uniform grids of three sizes, 4 mm, 2 mm, and 1 mm (Guo, Nairn (2003)). They have computed the energy release rate and determined the stress intensity factor from J-integral. Their results indicate that the stress intensity factor determined from finer grid is more accurate and closer to the theoretical value.



(a) Overall

(b) Magnified

Figure 18 : Computed stress intensity factor as a function of time

In the mesh refinement GIMP algorithm used in this study, the energy release rate was computed using the virtual crack closure technique (Rybicki and Kanninen (1977); Wang, Karuppiah, Lu, Roy, and Komanduri (2005)). The energy released during an infinitesimal crack growth of Δa is assumed to be the energy required to close the crack to its initial size. Hence, the energy release rate G is determined by

$$G = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \boldsymbol{\sigma} \cdot \Delta \mathbf{u} da, \tag{18}$$

$(\bar{\sigma}\cdot n)$ (bold)
where n is the crack surface normal.

For the 2D mode I crack in the X-direction,

$$G_I = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \sigma_{yy} \cdot \Delta u_y da. \qquad (19)$$

In GIMP, the energy release rate can be computed as

$$G_I = \frac{1}{2t\Delta a} F_{tip}(u_y^1 - u_y^2), \qquad (20)$$

where the superscripts 1 and 2 denote the two material particles immediately to the left of the crack tip as shown in Fig. 17, Δa is the X distance between particle 1 and the crack tip, t is the thickness of the beam. F_{tip} is the nodal force to hold the crack tip together (Rybicki and Kanninen (1977)) and is computed as the crack tip nodal force from one side of the crack in this simulation.

The mode I stress intensity factor for the static crack is given by

$$K_I = \sqrt{GE}. \qquad (21)$$

GIMP simulations were carried out using four material points per cell. The time step is 0.1 μs, and a damping coefficient of 4000 s^{-1} is used to allow the results to converge to the static data. The computed stress intensity factors using uniform grids of 4 mm and 1 mm, respectively, are plotted in Fig. 18 (a) and the theoretical value calculated from Eq. (17) is also shown for comparison. For simulation with refinement, two levels of refinement were used, i.e., 2 mm grid size for the coarse level and 1 mm grid size for the fine level, and the extent of the fine level is indicated by the dashed square in Fig. 16. The computed stress intensity factor using the refinement algorithm is also plotted in Fig. 18 (a). It is seen that the results from two levels of refinement are identical to the results from one uniform fine level with 1 mm cell size. Moreover, the simulation time using the refinement algorithm is 38% shorter than that of one uniform fine level. The computed stress intensity factor became even closer to the theoretical value when a third level of refinement was added for the crack tip as shown in Fig. 18 (b). For the case of using uniform grid of 2 mm, the computed stress intensity factor using GIMP and MPM (Guo and Nairn 2003) are plotted in Fig. 19. The MPM results were computed using a damping factor of 1000 s^{-1}, and therefore, more oscillations can be seen as expected.

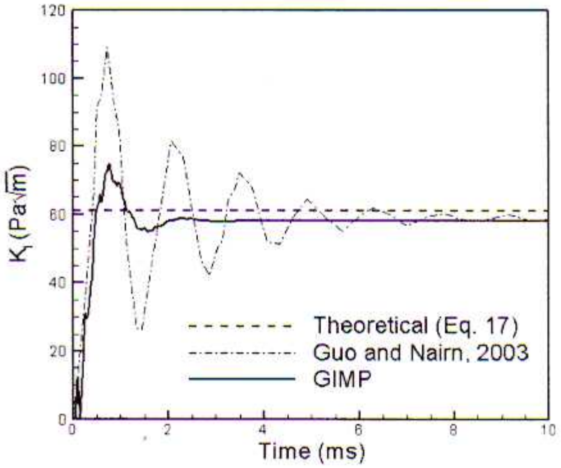


Figure 19 : Comparison of the stress intensity factor with MPM results

Fig. 20 ~~5~~ shows the distribution of σ_y at t = 2.5 ms when the force applied on the beam was changed to $F = 4$ N, 10000 times of the previous value, with other parameters the same. In this simulation, three levels of refinements with cell sizes of 1 mm, 0.5 mm and 0.25 mm, were used and the different density of material points in each level can be clearly seen in the figure. The computed stress intensity factor, scaled by 10000 times, still compares well with the theoretical value. It is noted that deformation near the crack surfaces has caused material points crossing cell boundaries, a situation where MPM would give numerical noise such as alternating stress signs. Despite the extent of the deformation, and material points crossing cell boundaries, GIMP method with the use of particle deformation tracking still gives correct results, further ~~validating~~ verifying the structured refinement methods developed herein.

5 Conclusions

1. A spatial refinement scheme for a ~~structural~~ structured grid was developed by adding transitional nodes and by changing the influence zone of the transition nodes in GIMP. The influence zone is square for uniform grid nodes and rectangular for transitional nodes. This influence zone affects the computation of the nodal shape function. The computation of the weighting function remains the same as for the uniform grid. The refinement scheme can be applied successively and the refined grid remains structural in each refinement level, i.e., every node can be determined by the extent of the grid level and cell size. The

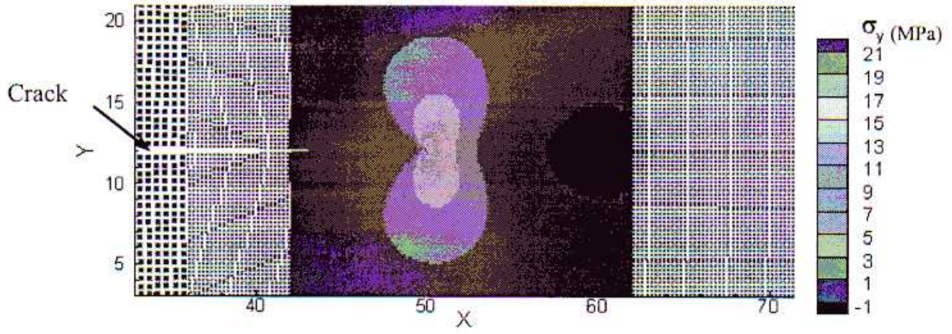


Figure 20 : Stress distribution in the beam using three levels of refinements at t = 2.5 ms

refinement scheme was implemented and several problems such as tension, indentation, stress concentration, and stress distribution near a crack (mode I crack problem) were modeled to demonstrate its effectiveness and accuracy. A good agreement has been obtained between numerical and theoretical results, indicating the validity of the structured mesh refinement for GIMP scheme.

2. The GIMP algorithm has also been extended to include the displacement boundary condition, based on the approach used in the meshless local Petrov-Galerkin (MLPG) method, Atluri and Zhu (2000). A penalty parameter is used to impose the displacement boundary condition and a nodal force vector because the displacement boundary condition is introduced to the nodal momentum governing equation. A uniaxial tension problem with constant pulling velocity was simulated to ~~validate~~ verify the displacement boundary condition.

3. A method to track the material particle deformation was developed and ~~validated~~ verified in one example. When the particle deformation is not tracked, artificial separation was observed when the particle strain increases to a certain level. In tensile simulations, when the normal strain is ∼25%, material particles tend to separate from the body. Our approach tracks the displacement of each corner of the material particles. Since neighboring particles share corners, no separation would occur during deformation using this approach.

Acknowledgement: The work was supported by a grant from the Air Force Office of Scientific Research (AFOSR) through a DEPSCoR grant (No. F49620-03-1-0281). The authors would like to thank Dr. Craig S. Hartley, and Dr. Jaimie Tiley, Program Managers for the Metallic Materials Program at AFOSR for the support of and interest in this work. Thanks are also due to Dr. Richard Hornung and Dr. Andy Wissink of the Lawrence Livermore National Laboratory for providing the code and assistance with the use of SAMRAI. One of the authors (RK) also acknowledges the A.H. Nelson, Jr., Endowed Chair in Engineering for additional support. Thanks are due to Dr. Samit Roy, formerly at OSU-Tulsa for participating in the project.

References

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

Atluri, S.N.: Zhu, T. (2000): The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics. *Computational Mechanics*, vol. 25, pp. 169-179.

Atluri, S.N.; Shen, S. (2002): The meshless local Petrov-Galerkin (MLPG) method: A simple and less-costly alternative to the finite element and boundary element methods. *Computer Modeling in Engineering & Sciences,* vol. 3, n. 1, pp. 11-51.

Bardenhagen, S.G.; Kober, E.M. (2004): The generalized interpolation material point method. *Computer Modeling in Engineering & Sciences,* vol. 5, n. 6, pp. 477-496.

Guilkey, J.E. (2005): Use of MPM in fully Lagrangian mode to eliminate cell crossing error, *First MPM Workshop*, March 13-14, 2005, Salt Lake City, Utah.

Guilkey, J.E.; Weiss, J.A. (2003): Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method, *Int. J. Numerical Methods in Engineering,* vol. 57, pp. 1323-1338.

Guo, Y.; Nairn, J. (2003): Calculation of J-integral and stress intensity factors using the material point method. *Computer Modeling in Engineering & Sciences,* vol.6, n. 3, pp. 295-308.

Hornung, R.D.; Kohn, S.R. (2002): Managing application complexity in the SAMRAI object-oriented framework. *Concurrency and Computation: Practice and Experience,* vol. 14, pp. 347-368.

Hu, W.; Chen, Z. (2003): A multi-mesh MPM for simulating the meshing process of spur gears. *Computers & Structures,* vol. 81, pp. 1991-2002.

Kannien, M.F. (1973): An augmented double cantilever beam model for studying crack propagation and arrest. *Int. J. Fracture*, vol 9, pp. 646-664.

Ma, J.; Lu, H.; Wang, B.; Roy, S.; Hornung, R.; Wissink, A.; Komanduri, R. (2005): Multiscale simulations using generalized interpolation material point (GIMP) method and SAMRAI parallel processing. *Computer Modeling in Engineering & Sciences,* vol 8, n2, pp.135-152.

Pilkey, W.D. (1997): Peterson's Stress Concentration Factors, 2nd Edition, John Wiley & Sons, New York.

Rybicki, E.F.; Kanninen, M.F. (1977): A finite element calculation of stress intensity factors by a modified crack closure integral. *Engineering Fracture Mechanics*, vol. 9, pp. 931-938.

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

Tan, H.; Nairn, J.A. (2002): Hierarchical, adaptive, material point method for dynamic energy release rate calculations. *Computer Methods in Applied Mechanics and Engineering,* vol. 191, pp. 2095-2109.

Wang, B.; Karuppiah, V.; Lu, H.; Roy, S.; Komanduri, R. (2005): Two-dimensional Mixed Mode Crack Simulation Using the Material Point, *Mechanics of Advanced Materials and Structures*~~(in press).~~, vol. 12, pp. 471-484.