IMPLEMENTATION OF IRREGULAR MESH IN MATERIAL POINT METHOD (MPM) FOR SIMULATION OF MIXED MODE CRACK OPENING IN TENSION

By

VENKATESH KARUPPIAH

Bachelor of Engineering

Annamalai University

Tamilnadu, India 2000

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2004

IMPLEMENTATION OF IRREGULAR MESH IN

MATERIAL POINT METHOD (MPM) FOR

SIMULATION OF MIXED MODE

CRACK OPENING IN TENSION

Thesis Approved:

Dr. Ranga Komanduri

Thesis Advisor

Dr. Samit Roy

Dr. Hongbing Lu

Dr. Al Carlozzi

Dean of the Graduate College

ACKNOWLEDGEMENT

I wish to express my sincere thanks to my parents for their confidence in me. I would like to thank them for their encouragement at times of difficulty, love and understanding. Special thanks are due to my brother for his inspiration, motivation, love and guidance.

I would like to express my sincere appreciation to my advisor, Dr. Ranga Komanduri for his intelligent supervision, constructive guidance, financial support, inspiration, motivation and friendship. I would also like to express my sincere appreciation to Dr. Samit Roy and Dr. Hongbing Lu for providing encouragement and technical guidance throughout this study. I would like to express my gratitude to Dr. Bo Wang for his invaluable guidance, support and direct contribution toward the completion of this dissertation. Without their great friendship, understanding, and encouragement, it would be hard to believe that I could come this far. My master studies under them have been a wonderful and unforgettable period of time in my life.

This project has been funded by the grant (F49620-03-1-0281) from the Air Force Office of Scientific Research (AFOSR). The author thanks Dr. Craig S. Hartley for his interest and support of this work.

I wish to express my sincere gratitude to Nitin Daphlapurkar for his assistance and friendship throughout this project. I wish to extend my gratitude to Harish

Viswanathan, Rohit Raghav, and Yang Liu for their encouragement and friendship. Special thanks are due to Jin Ma for helping me with TECPLOT.

Special appreciation is due to Dr. Gary Young for supporting me in this study. Thanks are also due to Syed Kareem and other research members of our group for their support and helpful conversations who have made these years a pleasant experience.

Finally, I would like to thank the Department of Mechanical and Aerospace Engineering for providing me with the opportunity to pursue M.S at Oklahoma State University.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The General Problem

Recently, one area in which researchers have devoted much effort is the development of numerical codes designed to model manufacturing processes. It is a challenging problem because manufacturing processes involve large deformations. However, in many of these problems, the flow rate of the material through the process is much lower than the wave speed for either elastic or plastic wave propagation. There are numerous problems, which display material failure (fracture, separation) along both planar and curved surfaces. For both cases, it has been a cherished goal of many analysts to develop a numerical procedure that can predict such a phenomena. With such a capability, it would be possible to design materials with oriented properties (e.g., embedded fibers) to withstand failure in an optimal sense, to design machining process with more efficient cutting conditions that cause material failure, or provide a better indication of when structural failure might occur (e.g., seismic loading).

Researchers are often interested in determining the motion of a medium such as a solid, liquid or gas when it is acted upon by given forces and subject to given conditions. One approach to achieving this goal is to treat the material as if it is a continuum. It is assumed that at each point of this continuous matter there is a unique value for each of

the variable properties of the material, such as velocity, pressure and density. The continuous material then obeys a set of conservation laws, which govern the values of the variables of the material.

Eulerian or Lagrangian frames of reference are often used to describe such a continuous material. In the Eulerian frame, a control volume is fixed in space and the material is studied as it passes through this region. The solution in an Eulerian frame describes what is happening with the material at every point in the region as time passes. In a Lagrangian frame, a particular material point is selected and its motion is studied as time progresses. The solution in a Lagrangian frame describes the motion of each particle as a function of time. In either frame the governing equations for the material can be formulated as differential equations. Numerical techniques, such as FEM is one common technique used to determine the solution of the differential equations.

1.2 Computational Methods

There are large number of numerical methods described in the literature that attempt to provide approximate solutions of the differential equations governing the motion of a solid. One reasonable expectation of a numerical method is that it would compute sufficiently accurate solutions at a moderate cost. Ideally, such a method should also be easy to implement and resolve the material interfaces in a computationally efficient manner.

As earlier pointed out, numerical methods are typically classified as Eulerian or Lagrangian, depending on the frame of reference used to arrive at the solution. Thus, in an Eulerian code, a set of spatial points is chosen and solutions are obtained at these

points as time progresses. In a Lagrangian code, the solution is calculated at points in the body of interest, following the motion of the point as deformation occur. Some codes combine these two frames of reference, performing a Lagrangian time step followed by a remapping step that maps the solution from the distorted Lagrangian frame to spatially fixed Eulerian frame. Depending on the choice of a new frame, these codes are usually referred to as Eulerian codes, arbitrary Lagrangian Eulerian (ALE) codes, or combined Eulerian Lagrangian codes [6].These methods have some advantages as well as some disadvantages. In the following, the special features of some of these methods are discussed starting with the Lagrangian methods.

1.2.1 Lagrangian Methods

In the Lagrangian method a "mesh" is inserted in the domain of the material. The mesh is a discrete description of the continuous domain. Each body of the material has its own separate mesh. The differential equations are solved for each position of the mesh at discrete points in time. Since the mesh moves with the material, the motion of the mesh implies motion of the material [6]. This method is known as updated Lagrangian. There is another Lagrangian method used for solids that involve discretizing the reference configuration with a fixed grid and solving in those coordinates. Examples of Lagrangian codes used in the literature include HEMP, DYNA2D, DYNA3D, PRONTO, TENSOR AND EPIC [6].

There are several advantages of using the Lagrangian method. The first one is a description of the equations in a Lagrangian framework removes the nonlinear convective term in the conservation of momentum equation. A linear derivative is much easier to

handle in terms of finite difference or a finite element discretization. A second advantage of the Lagrangian method is that material interfaces are resolved naturally since almost all Lagrangian formulations assume that each element is restricted to a single material [6]. This also allows for constitutive equations to be readily applied. The boundary of the material is easily followed since any node of the mesh that originates on the boundary of the domain will remain on the boundary as the mesh deforms [7].

One consideration with the Lagrangian method is the determination of the relative motion of two or more meshes. Nothing in the basic method precludes separate bodies from overlapping. Contact algorithms have been developed to determine the relative motion of two or more meshes. Penalty methods and Lagrangian multiplier methods are two such contact algorithms. Special contact algorithms are necessary if a surface is allowed to come into contact with itself. The contact search is the dominant cost for many contact algorithms. Simplifying the problem geometry and the contact force calculations can reduce the computational costs of contact problems [6].

Because the mesh is not allowed to tear, penetration calculations are difficult to perform with the Lagrangian code. To solve this problem algorithms have been developed to determine where and how to separate the nodes.

As the mesh distorts with the motion of the material, the smallest dimension of the mesh becomes the restricting factor in the time step size. Mesh distortion is also a concern in the Lagrangian method as too much distortion results in loss of accuracy of the solution. If the mesh becomes too entangled, the calculation must be halted. One way to deal with this problem is to allow the Lagrangian calculation to run until the mesh becomes unacceptably distorted, create a new undistorted mesh, map the solution from

the distorted mesh to the new mesh, and then continue the Lagrangian calculation. This process is referred to as rezoning [6]. Since such algorithms need to be quite general in nature to handle an arbitrary mesh, their implementation is somewhat complex. The process of rezoning is usually computationally intensive and of low order of accuracy [6]. Ensuring conservation of the mapped quantities requires more complex algorithm. To reduce the computational cost, remeshing is done only when necessary during an otherwise Lagrangian calculation. In the next section the Eulerian methods are discussed.

1.2.2 Eulerian Methods

In the Eulerian method the mesh overlays the material being modeled. Calculations are made at the stationary nodes of this mesh as time progresses. The material flows through the mesh. The solution provides a "snapshot" of the material motion at a given set of spatial points as time progresses. The strengths and weakness as of an Eulerian method are almost exactly the opposite of those for the Lagrangian method.

The ability of an Eulerian mesh to allow arbitrarily large deformations is the major advantage of this method. Since the mesh does not deform in an Eulerian calculation, mesh entanglement is not an issue. Eulerian calculations do not have the accuracy losses associated with highly distorted elements that are often found in the Lagrangian calculations [6]. Another advantage of using Eulerian codes is that they do not need contact algorithms since only one mesh is used for the entire computational area.

Eulerian methods also have their disadvantages. As with their Lagrangian counterparts, algorithms have been developed to address these issues at the expense of an increase in the computational costs and algorithmic complexity. One of the disadvantages of Eulerian methods is that use of Eulerian coordinates requires discretization of the nonlinear convective term in the equation for conservation of momentum. This discretization is the primary source of numerical diffusion in an Eulerian code. A nonlinear equation requires more complicated methods to solve numerically, especially when using an implicit solver.

The nonlinearity can also be treated using an "operator split" technique. The equation is first solved using the Lagrangian motion of the grid and the resulting solution is then mapped back to the Eulerian grid. This is not a true operator split. There is no time step associated with the Eulerian step. Rather, it is simply a projection of the solution from one mesh onto another, with the Eulerian step acting as a continuous rezoning of the mesh [6]. Other than increase in the cost of remapping, the main disadvantage of this technique is the limitation on accuracy. Many material models are integrated in time with a first order method, with the result that the Lagrangian step is rarely fully second order accurate in time [6]. Also, some quantities such as kinetic energy may not be conserved by the remapping scheme. Care must also be taken when treating materials that require history variables to determine the current state. The calculation costs can be reduced considerably by performing the Eulerian remap only after several Lagrangian steps instead of after every step.

Another major disadvantage of the Eulerian method is the lack of resolution of material interfaces due to use of only one mesh. Eulerian codes allow several materials to

lie within a single element. The amount of each material in an element must be known, and rules must be established to determine how each material is moving. These rules track, capture or reconstruct the interface between materials. Handling elements with several materials adds a significant computational cost. Eulerian methods were at one time regarded as a last resort for solving a problem because of their poor resolution of material interfaces. This has changed with the introduction of high–resolution interface tracking algorithms [6]. Second order accurate algorithms have been developed to calculate the material transport between elements. Thus, handling elements with several materials are no longer the limiting factor in the accuracy of Eulerian calculations [6] although robustness of these algorithms may still be an issue.

Many Eulerian codes are restricted to rectangular zones, although some codes allow the use of meshes based on orthogonal curvilinear coordinate systems. This is especially true of methods that use edge-centered velocities rather than nodal velocities [6].

As with their Lagrangian counterparts, the use of Eulerian methods has its own advantages and disadvantages. According to Ref.6, the accuracy of an Eulerian calculation is equal to a Lagrangian calculation for problems that have a changing topology. Because purely Eulerian and purely Lagrangian methods both have their short comings, attempts have been made to combine the positive features of both methods without too much compromising. Some of the negative effects of these methods are discussed in the next two sections.

1.2.3 Arbitrary Lagrangian - Eulerian Methods

Often, the problem of interest in a computation involves deformations that are too severe to be handled by the same Lagrangian mesh during the entire calculation. At some point in the calculation, a new mesh must be generated and the old solution must be mapped from the old mesh onto the new mesh. The frequency of remapping and the choice of new mesh define the differences between Eulerian, ALE and rezoned meshes [6, 18]. The boundaries between the three types are somewhat blurry, but general distinctions have been formed. Rezoned meshes are used in otherwise strictly Lagrangian methods, as described in section 1.2.1. A new mesh is generated as infrequently as possible, with the current mesh being used when doing so does not compromise the accuracy of the solution. In Eulerian methods as described in section 1.2.2, the solution of a Lagrangian step is calculated and then remapped usually every step or every few steps to a spatially fixed Eulerian mesh. If the solution is mapped onto a different mesh that moves in a manner that may be independent of the material motion, the method is referred to as an ALE method. These methods are briefly discussed in this section. Examples of ALE methods adopted in the literature include CAVEAT, DYNA2D, HEMP, SALE, CALE, HELP and SHALE [6].

ALE methods have the same advantages as Eulerian codes, allowing arbitrarily large deformations and avoiding mesh entanglement, element distortion and contact algorithms. Using an arbitrary mesh allows the added advantage of better resolution of features such as shock waves. The ALE and Eulerian methods have similar disadvantages.

The algorithms for the remapping step are identical for both Eulerian and ALE codes. Eulerian codes are often less computationally costly than the ALE codes because of their spatially fixed mesh. Another disadvantage of ALE methods is that, like their Eulerian counterparts, these methods may have elements containing more than one material. The algorithms used to address this problem for Eulerian methods are also applicable to ALE methods. The complexity of handling elements with several materials can add a significant computational cost [6]. Algorithms to compute history variables on the remapped ALE mesh are also more complex than those for Eulerian methods, due to the arbitrary nature of an ALE mesh. Accurate remapping of the element-centered variables is the largest obstacle to attaining second accurate finite element ALE methods [6].

ALE methods have properties that are desirable for certain types of problems. However, they tend to be more costly than strictly Eulerian or Lagrangian methods. Another type of method combining Eulerian and Lagrangian methods is described in section 1.2.4.

1.2.4 Coupled Eulerian Lagrangian Methods

In some problems of interest the ability to represent one part of a problem with a Lagrangian mesh and another part with an Eulerian mesh is useful. Examples of this type of calculation include underwater explosions, where the fluid is Eulerian and the hull is Lagrangian, and a low velocity penetration calculation, where the penetrator is Lagrangian and the target is Eulerian [6]. This type of code is called an Eulerian

Lagrangian code. One code with this capability that is mentioned in the literature is PISCES [6].

The Eulerian mesh is used for the target where large deformations occur. The response is like a fluid here. The Lagrangian mesh is used for the region that is more rigid and has relatively small deformations. This allows each of the meshes to model the part of the problem for which they are best suited. By combining the two meshes, the computational efficiency of the method is improved.

The major disadvantage of this type of method is that the Lagrangian mesh can move through the Eulerian mesh. If each method uses a separate mesh, some type of algorithm is needed to apply boundary conditions on the fluid mesh from the solid, and vice versa. Usually, continuity of the normal component of velocity and continuity of traction are imposed. This issue is similar to contact algorithms.

Coupled Eulerian Lagrangian methods allow a reduction in computational cost by using each mesh on the area of problem for which each is most suited. The computational cost is raised, however, due to the necessity of a contact algorithm.

Not all computational methods are based on a mesh. In the next section, "meshless" or particle methods are discussed.

1.2.5 Particle Methods

Four variations of mesh-oriented numerical methods are discussed in the previous sections. Looking at the disadvantage of these methods, it appears that elimination of the mesh might result in methods that avoid some of these problems. Particle methods are a result of such attempts to remove the mesh.

Particle methods attempt to construct the approximation to the solution strictly in terms of nodes [8]. In these methods, the domain of interest is discretized by a set of nodes or particles. A shape function with compact support is defined for each node. The region in the function's support, usually a disc or rectangle, is called the domain of influence of the node [8]. The shape function typically has two parameters, providing the ability to translate and dilate the domain of influence of a shape function. The translation parameter allows the function to move around the domain, replacing the elements in a meshed method. The dilation parameter changes the size of the domain of influence of the shape function, controlling the number of calculations necessary to find a solution. As the dilation parameter becomes larger, larger time steps can be taken [34]. A set of basis functions also needs to be defined for a given problem. Examples of particle methods discussed in the literature include SPH, DEM, EFG, RKPM [6], PUFM [35], hp-clouds [36], finite point method [37] and FEAM [38].

The major advantage of a particle method is that the particles are not treated as a mesh. Therefore, mesh entanglement is not a problem and large deformations can be treated with these methods. Creating new meshes and mapping between meshes is eliminated. Refinement can be obtained by simply adding points in the region of interest [34, 39]. It should be noted that most of these methods use a mesh to do integration. FEAM and some versions of SPH use a mesh or particle binning to find nearest neighbors. However, the mesh may be simpler than would be needed for standard element based solution.

Another advantage of particle method is that there is no need to track the material interfaces, since each particle has its own constitutive properties [6].

There are also difficulties in using particle methods. One of the major disadvantages of these methods is their relatively high computational cost, particularly in the formation of the stiffness matrix. The supports of the shape functions usually cover more surrounding points than finite element shape functions do. In fact, the support of the kernel function must cover enough particles for the method to be stable [39]. The bandwidth of the resulting matrix is increased and more irregularity of the sparsity structure results, since the number of neighbors of a given point can vary from point to point. Thus, the number of numerical operations in the formation and application of these matrices is increases. Additionally, higher- order shape and basis functions are usually used, with the result that higher-order integrations are required. Construction of these shape functions is also costly [39].

Another problem is that the shape functions are not interpolatory in many cases [8]. This makes essential boundary conditions more difficult to apply. Some techniques that have been developed to address this problem are Lagrange multipliers, modified variational principles, penalty methods and coupling to finite element methods. However, there can be difficulties with using techniques also. For example, the Lagrange multiplier method requires solution of an even larger system of equations. In addition, Lagrangian multipliers tend to destroy any structure, such as being banded or positive definite, that the system might exhibit [8]. The modified variational approach applies boundary conditions of a lower order of accuracy. Coupling to finite element methods by using particle methods only in regions with large deformations and finite element methods elsewhere in the problem can reduce the cost of the solution. However, the shape functions at the interface become quite complicated and require a higher order of

quadrature [8]. Another method by Chen [40] uses the map from nodal values to the function space to get nodal values of the function space to get nodal values of the function, applies the boundary conditions, then transforms back to nodal values.

Many of these codes have been restricted to static problems. Chen has developed a dynamic code, but all of the basis functions are constructed in the original configuration [40].

Like their meshed counterparts, particle methods are useful for certain types of problems. However, as a result of their higher computational cost, they are not the method of choice for other types of problems. Particle-in-cell methods combine some aspects of both are meshed and meshless methods. These methods and the comparison between Explicit and Implicit methods are discussed in the next section.

1.3 Explicit and Implicit Methods

Solving a system of partial differential equations numerically requires both spatial and temporal discretization of the equations. Numerical methods involving a spatially discretized mesh have typically been classified by the choice of frame of reference as either Lagrangian or Eulerian. Some methods have combined aspects of both of these types, while particle methods have eliminated the mesh in favor of particle based spatial shape functions. Some of these differences are discussed in previous sections.

In the terms of their temporal discretization, numerical methods are classified as either explicit or implicit methods. In an explicit method, each unknown can be evaluated directly in terms of known quantities. In an implicit method, two or more quantities in the discretized equation are unknown at the same time. More complex methods are needed to

solve an implicit system of equations and iterative methods for linear or nonlinear systems.

There are several advantages to using an explicit method over an implicit method. According to [7], typically an explicit method requires fewer computations per time step. An explicit method usually has simpler logic than an implicit method, making it easier to deal with complex nonlinearities. An explicit method generally requires less storage than either the direct methods or the iterative methods of solving an implicit system. An explicit method requires less coding, making it useful for testing purposes. An explicit method is usually very reliable as to accuracy and completion of the computation when stability requirements are met.

The major disadvantage of using an explicit method it is usually only conditionally stable [7]. The result is that a very large number of very small time steps may be required.

The advantages and disadvantages of using an implicit are essentially the opposite of the explicit method [7]. The major advantage of using an implicit method is that a much larger time step may be used. The disadvantages of using an implicit method are numerous. Implicit methods use more complicated logic, resulting in more complex, larger codes. More storage is required, especially if direct elimination methods are used. More computations are required per time step. When using an implicit method, care must be taken to ensure that time steps are not so large that accuracy is lost, as time steps used may be large enough that time integration errors become dominant. In contrast, spatial discretization errors tend to be dominant in explicit methods.

Certain types of problems require small time steps in order to achieve appropriate accuracy in the solution, while others do not. Explicit methods are a more suitable approach to problems where high frequency components of the solution are a significant part of the response [41]. Examples of this type of problem include wave propagation problems such as shocks, blasts or any type of loading with a broad frequency range. Implicit methods, on the other hand, are more suited to problems with frequency components in the lower range. In particular, inertial structural dynamics problems fall into this category.

According to [41], for linear systems of equations, explicit and implicit methods are about equally difficult to apply and to implement. The only additional cost of implicit integration is the linear solution needed at each step. Explicit integration is usually straight forward.

For nonlinear problems, more elaborate methods of solution are needed. Implicit integration of nonlinear systems is much more expensive. Each nonlinear iteration entails the solution of a linear system containing a tangent matrix. The evaluation of this matrix and the solution of the associated linear system account for most of the cost of a nonlinear iteration. The use of methods that remains stable for large time steps, that minimize the number of evaluations of the tangent matrix, and that minimize the number of iterations per time step can help to reduce these costs. Matrix-free implementations that do not require actual formation of the tangent stiffness matrix and preconditioners can be used for this purpose.

The choice between implicit and explicit methods generally depends on the type of problem to be solved. For problems requiring high-frequency components, an explicit

method is a more appropriate choice. For problems with only low-frequency components, either type of method could be used. In this case, the implicit method becomes the solution of choice when the gain in size of the time step outweighs the higher cost of the iterative solution.

1.4 Outline of the Thesis

This thesis is divided into 8 chapters. Chapter 1 presents a description of several methods currently available to numerically model plastic deformation processes. In Chapter 2, a review of literature on topics of interest to the present investigation, namely particle-in cell method and other methods which lead to the developments of Material Point Method (MPM) are presented. Chapter 3 gives an introduction to the MPM and description of the governing equations used in MPM. Numerical implementation and discussion of various algorithms are used in the conventional MPM algorithm are also covered in this chapter. Chapter 4 gives the problem statement of the investigation. In Chapter 5, simulation of mixed mode crack by Material Point Method is discussed and methods to determine the fracture parameters are presented.

Chapter 6 describes a new MPM algorithm. MPM is particularly suited the problems undergoing large material distortions [12, 14, 36-38]. Chapter 7 presents results and discussion of MPM simulation of a tensile model with and without a crack for different times and loads. These results are compared with FEM simulation using ABAQUS/Explicit. Chapter 8 gives conclusions arising out of the present investigation and offers some suggestions for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Particle-In-Cell Methods (PIC)

According to Harlow [1] the Particle-in-Cell (PIC) method was developed in 1955 at Los Alamos National Laboratory for the solution of complex fluid dynamics problems. It is a combination of Lagrangian and Eulerian methods that naturally can handle no slip interfaces between materials and large slippage and distortions. Details of the code are discussed by Amsden [3].

In general idea behind the PIC method is to solve the governing equation on an Eulerian grid where derivatives can be conveniently defined. Information is then transferred from the grid to Lagrangian material particles via mapping functions. The material particles move or convect and carry with them certain properties. Variations on the method can occur by changing the mapping method. That is, the mapping functions themselves may be changed. In Harlow's classical version of PIC, velocities were mapped from the grid to the particle. In a less dissipative version called FLIP (FLuid-Implicit-Particle) [42, 43] material particle velocities are only updated from the grid solution.

An outline of a FLIP-type algorithm is as follows:

1. Solve the governing equation to obtain magnitudes of acceleration at the grid nodes.
2. Integrate accelerations to obtain velocities on the grid.
3. Map the acceleration to the particles to update the velocity.
4. Move the particles based on the velocity determined in Step 2.
5. Map particle quantities to the grid in preparation for the solution at the next time step,
6. Determine velocity gradient, strains, and stresses at nodes (or vertices),
7. Determine grid forces from stresses.

Sulsky and Brackbill [12] used a method similar to Peskin's [44] but based on the PIC method, to simulate suspended bodies moving in a fluid.

A force density term, F(x, t), is added to equations for Stokes' flow for an incompressible fluid

$$\begin{aligned} -\nabla + \mu\Delta u + F &= 0 \\ \nabla \cdot u &= 0 \end{aligned} \qquad (2.1)$$

where the gradient, ∇ and Laplacian, Δ are taken with respect to the current position, x, velocity u, and pressure p, and the force, F, is determined from the sum of the internal and external forces. The external forces may be those due to gravity or magnetic fields. The internal forces only exist in the suspended body and are due to the strains within the body.

The basic ideas behind the PIC or FLIP methods have been adapted recently to solid mechanics by changing step (6) of the FLIP-type algorithm. These field variables are evaluated at material points, and the resulting approach is applied to impact problems with elastic and elastic-plastic constitutive equations [2]

2.2 Material Point Method (MPM)

Sulsky and Schreyer [5] presented the general description of the material point method (MPM), along with special considerations relevant to axisymmetric problems. The method utilizes a material or Lagrangian mesh defined on the body under investigation, and a spatial or Eulerian mesh defined over the computational domain. The set of material points making up the material mesh is tracked throughout the deformation history of the body and these points carry with them a representation of the solution in a Lagrangian frame. Interactions among these material points are computed by projecting information they carry onto a background finite element mesh where equations of motion are solved. They reported that the material point method does not exhibit locking or an overly stiff response in simulations of upsetting.

The material point method (MPM) has recently been developed as a numerical method for solving problems in dynamic solid mechanics [2, 4, 5, and 45]. In MPM, a solid body is discretized into a collection of points much like a computer image is represented by pixels. As the dynamic analysis proceeds, the solution is tracked on the material points by updating all required properties, such as position, velocity, acceleration, stress state, etc. At each time step, the particle information is extrapolated to a background grid which serves as a calculational tool to solve the equations of motions. Once the equations are solved, the grid-based solution is used to update all particle properties. This combination of Lagrangian and Eulerian methods has proven useful for solving solid mechanics problems including those with large deformations or rotations and involving materials with history dependent properties such as plasticity or viscoelasticity effects [2, 4, 5, and 45]. MPM is amendable to parallel computation [46],

implicit integration methods [24] and alternative interpolation schemes that improve accuracy [47].

Although MPM uses a background grid and is frequently compared to finite element methods, a new derivation of MPM [24] presents it as a Petrov-Galerkin method that has similarities with meshless methods, such as Element-Free Galerkin (EFG) methods [48] and Meshless-Local Petrov-Galerkin (MLPG) methods [50, 51, 52]. The "meshless" aspect of MPM, despite the use of a grid, derives from the fact that the body and the solution are described on the particles while the grid is used solely for calculations. The body can translate through the grid. Furthermore, the grid can be discarded at each time step and redrawn which makes MPM suitable for adaptive mesh methods. It is essential for any extension to MPM, such as presented here, to preserve the separation between the grid and the particles. MPM, EFG, and MLPG differ in their approach to derive shape functions and in their selection of test functions during numerical implementation [47, 50]. One potential application of MPM is its use as a tool in dynamic fracture modeling. It was recently shown that MPM can accurately calculate fracture parameters, such as energy release rate [10] but those results were for a crack at a symmetrical plane and thus the crack could be described by symmetry conditions alone. Conventional MPM is not capable of handling explicit, internal cracks.

MPM has found application in the solution of a wide variety of problems, including silo discharge [15], membrane stretching [22], landfill settlement [19], elastic vibrations [17], collisions [4, 14–16], and the response of granular materials [9, 14,16].

CHAPTER 3

MATERIAL POINT METHOD

3.1 Introduction

The Material Point Method (MPM) is a numerical scheme for dynamically modeling problems in solid mechanics. MPM is an extension of the Fluid Implicit Particle method (FLIP) [42], a particle-in-cell (PIC) method. The PIC method proved useful for modeling highly distorted flows. On the other hand, the method exhibited a large dissipation of kinetic energy. When other more accurate methods were developed, PIC methods were thought to be obsolete. However, in the 1980s, Brackbill developed the FLIP particle-in-cell method [12, 42]. By mapping only the changes in the quantities of interest and by tracking more of these quantities on the particles, the accuracy of the computation was increased. Sulsky and her colleagues developed the MPM method in the 1990s [2, 4, 5]. In this method, a mesh of Lagrangian material points is used to discretize one or more solid bodies.

The material point method (MPM) is a particle method for simulations in computational fluid and solid mechanics. The method uses a regular structured grid as a computational scratchpad for computing spatial gradients of field variables. The grid is convected with the particles during deformations that occur over a time step, eliminating the diffusion problems associated with advection on an Eulerian grid. The grid is restored

to its original location at the end of a time step. In addition to avoiding the Eulerian diffusion problem, this approach also circumvents problems with mesh entanglement that can plague fully Lagrangian-based techniques when large deformations are encountered. MPM has also been successful in solving problems involving contact, having an advantage over traditional finite element (FE) methods in that the use of the regular grid eliminates the need for doing costly searches for contact surfaces.

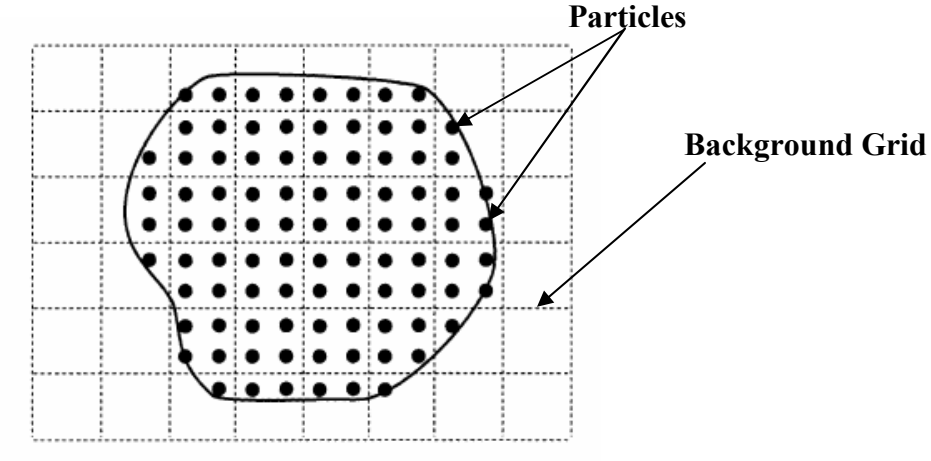


Figure 3.1 MPM Grid showing Particles and Background Grid.

It is only necessary to avoid taking a time step too large to tangle the mesh in one-step. It is not necessary to generate complex grids for this purpose. The method easily tracks contact discontinuities and material interfaces, since each material point maintains its material properties throughout the calculation. For the same reason, it is easy to apply constitutive equations on the material points.

The main disadvantage of MPM methods is the computational expense. Mapping quantities between the material points and the nodes of the grid and calculating the location of the material points on the new grid are the major additional costs. There can be fluctuations in solutions due to the transport of discrete material points across element

boundaries. Thin layers of material can be hard to resolve. Using more material points per element can reduce these last two problems.

3.2 Material Point Method

In this section, the conventional material point method (MPM) developed by Sulsky et al. [2, 4] will be summarized. In MPM, the material continuum is discretized into a finite collection of material points. Fig. 3.1 is a schematic MPM for a two-dimensional (2D) calculation. The solid line is the outline of the body to be analyzed. The black dots are the material points. Each material point is given an initial mass consistent with the material density and volume of the point. Material parameters, such as mass, displacement, velocity, stress, strain, internal energy, and temperature are assigned to each material point according to the material it represents. As the numerical solution proceeds, the material points are tracked and their states updated so that they carry the complete solution. Information from the material points is transferred to background computational grid nodes. The continuum equations are discretized at grid nodes using standard finite difference or finite element methods. The solutions at grid nodes are then used to update the position and velocity of the material points. Strain increments computed on the grid are interpolated to the material points and used in conjunction with constitutive equations to update stress states for each material point.

A material continuum is divided into a finite collection of discrete infinitesimal regions Ω_p ($p = 1,..., N_p$) called material points. Each material point is assigned a mass

m_p in Ω_p, where $m_p = \int_{\Omega_p} \rho(x) d\Omega$ and $\Omega = \bigcup \Omega_p$. Mass density can then be approximated as a sum of point masses using a Dirac function

$$\rho(x) = \sum_{p=1}^{N_p} m_p \delta(x - x_p) \tag{3.1}$$

All variables $\phi(x)$ such as coordinate, displacement, velocity, and acceleration need to be transferred between grid nodes to material points using the shape functions *N(x)*,

$$\phi(x) = \sum_{n=1}^{N} \phi^{(n)} N^{(n)}(x). \tag{3.2}$$

where *N* is the number of nodes in the grid and superscript (*i*) refers to the nodal values of $\phi(x)$. The details are given in Ref. [4]. The grid point accelerations are then used to update the position, velocity, stress, strain and temperature of the material points.

To begin the next time step, the velocity at grid points in the new grid can be calculated by extrapolation from material points as

$$\sum_{n'=1}^{n} m^{k(nn')} v_i^{k(n')} = \sum_{p=1}^{N_p} m_p v_i^{k(p)} N^{(p)(n)}. \tag{3.3}$$

$\Delta\sigma_{ij}^{s,(p)}$ at $X_i^{L(p)}$ are obtained from the strain increment with standard constitutive equation. The following extrapolation scheme will be used to transfer displacement, stress and strain information from particles to nodes at every time step

$$u_i^{(n)}(x) = \sum_{n=1}^{N} u_i^{(p)} N^{(p)(n)}(x) \tag{3.4}$$

$$\sum_{n'=1}^{n} m^{k(nn')} \sigma_{ij}^{k(n')} = \sum_{p=1}^{N_p} m_p \sigma_{ij}^{k(p)} N^{(p)(n)} \tag{3.5}$$

$$\sum_{n'=1}^{n} m^{k(nn')} \varepsilon_{ij}^{k(n')} = \sum_{p=1}^{N_p} m_p \varepsilon_{ij}^{k(p)} N^{(p)(n)}. \tag{3.6}$$

The mesh in the conventional MPM in all cases is considered as a square grid. This approach has limitations, the limitations are that the size of the grid cell mesh cannot be refined near the crack tip to account for the stress gradient and the crack can be created only in x or y directions. Details are given in next chapter. To overcome this limitations a new MPM algorithm has been developed (Chap. 6) which deals with the shortcomings of the conventional MPM. To prove the robustness of the new MPM algorithm a tensile problem with an inclined crack is solved (Chap. 7).

In MPM, three different algorithms can be used to update stress. The first one is update material point stresses average (USAVG) which is commonly used. The second one is the update material point stresses last (USL) algorithm and the third one is update material point stresses first (USF) algorithm [13].These algorithms are discussed in the following section. All the three algorithms are implemented in the conventional MPM code.

3.3 Update Material Point Stresses First (USF) Algorithm

As outlined in the previous section on the MPM algorithm, the stress on the material points may be updated based on the strain increment, calculated from the initial material point velocities interpolated to the grid. This option is referred to subsequently as the update-stress-first (USF) algorithm [13]. For this case, stress is calculated from the equation specified in the previous section and the grid velocity increment is determined from the updated stress.

3.4 Update Material Point Stresses Last (USL) Algorithm

The other option, equally valid from an algorithmic standpoint, is to update the stress on the material points at the end of the time step, using the strain increment, calculated from the updated material point velocities interpolated to the grid. This option is referred to subsequently as the update-stress-last (USL) algorithm [13]. For this case stress is calculated as specified above, but the grid velocity increment is determined from the initial stress.

3.5 Update Material Point Stresses Average (USAVG) Algorithm

The common option is used to update the stress on the material points before the current time step and after the time step and the calculated average value is taken into computation. This option is referred as the update-stress-average algorithm [11]. For this case, the grid velocity increment is determined from the average of initial and the final stress.

CHAPTER 4

PROBLEM STATEMENT

The material point method (MPM) proposed by Sulsky et al. [2, 4, 5, 9] has received increased applications to simulate dynamic problems in solid mechanics. MPM has demonstrated capabilities in the simulation of impact/contact, penetration, and interfacial crack growth problems. In MPM, a material continuum is discretized into a finite collection of material points. Two descriptions are used in MPM - one based on a collection of material points (Lagrangian) and the other based on a computational background grid (Eulerian). The background rectangular grid is used solely for calculations. The material points are followed throughout the deformation of a solid and provide a Lagrangian description that is not subjected to mesh entangling. As a result, MPM takes advantage of both the Eulerian and Lagrangian descriptions to possess capability to handle large deformations in a more natural manner so that mesh lock-up is avoided. Parallel computation is also straightforward because of the use of a grid structure that is consistent with parallel computing grids. Additionally, for problems involving contact [10], MPM is able to provide a naturally non-slip contact algorithm to avoid the penetration between two bodies based on a common background mesh.

Recently, MPM has been used for modeling fracture in materials [10-11]. Tan and Nairn [10] utilized MPM to model a crack in a plane problem, and computed dynamic energy release rates. Their work has considered the condition that that there should be no

interaction of particles between two free crack surfaces. The conventional MPM approach used a regular computation grid in which all MPM cells are of square shape and same size [10, 11, 13, 14, 20-22]. This results in two major limitations:

1) The crack orientations are confined to be only along *x*- or *y*- directions

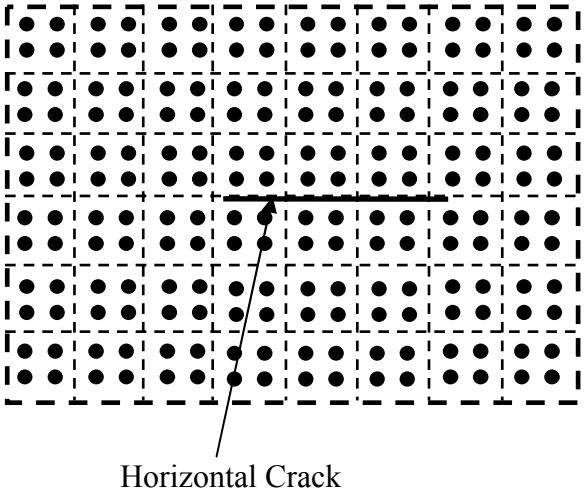


Figure 4.1 Horizontal Crack on Regular Mesh

2) The size of the grid cell mesh cannot be refined near the crack tip to account for the stress gradient.

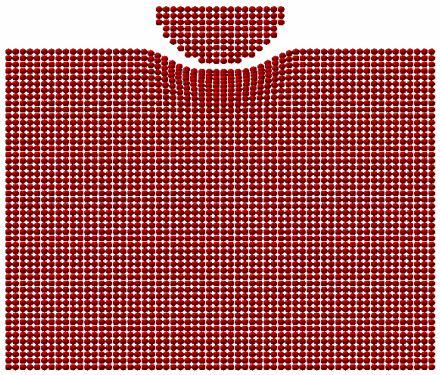


Figure 4.2 Model showing Stress Concentration

In reality, cracks can exist in arbitrary orientations. While a single crack in a structure can always be brought to be aligned with coordinate axes, un-aligned multiple cracks and crack kinking cannot be all brought to align with abscissa or ordinate. To model the

inclined cracks, and to use fine mesh close to the crack and coarse mesh in the far field for minimization of computational time while maintaining accuracy, a new approach is needed. This approach is the implementation of irregular mesh in MPM which is the subject of present investigation.

Validation was first made with a tension problem using two MPM models (regular and irregular mesh) and the results are compared with the results obtained from ABAQUS/explicit code. An inclined crack problem is solved as an example to demonstrate the capability of the arbitrary quadrilateral cells in the new algorithm. The stress and deformation fields are determined as a function of time. For this mixed mode crack fracture, energy release rates are calculated using the virtual crack closure method based on the nodal force and relative displacements at the crack tip. Stress intensity factors were also calculated using the displacement extrapolation method in terms of relative displacement and locations along free crack surfaces. The same problem is also solved using FEM.

CHAPTER 5

NEW MPM ALGORITHM

5.1 MPM Algorithm

A new algorithm was developed involving an irregular mesh in MPM that can handle mixed mode crack propagation [58]. The algorithm adopted in the MPM code (which is looped till user-defined end time, with *user-defined time step*[*]) is:

In the preprocessing stage the data is read from the input file and the background grid is build. After building the background then the material points are placed in the cell based upon the shape functions derived from the natural co-ordinates.

1. "PrepMassmatrix": In this particle a Lumped Mass Matrix is prepared:

$$M_p = \text{particle mass} = \frac{(\text{Cell}_{\text{volume}}) \times (\text{density})}{(\text{total number of particles in each cell})}$$

the total number of particles in cell is 4 in case of 2D and 8 in 3D.This is done for the conservation of mass. After determining the mass, the time step is computed. The time step is the ratio of cell size to the wave speed. The smallest time step is taken for computation.

2. Find dimensionless particle locations and find grid momentum:
   a) ξ, η, and shape-functions are calculated for each particle based on 4-node isoparametric representation.
   b) Lumped Mass Mapping (particles to nodes)

$$m_n = mass_{node} = \sum_{neighbouring\_cell=1}^{max=8(4\ for\ 2D)} \sum_{p=1}^{no\_of\_particle\_in\_cell=4} \left(M_p \times N_p\right)$$

c) Momentum is mapped (particles to nodes) (for i =1 and 2 axes, i.e. x and y axes)

$$pk_n = momentum_{node} = \left(\sum_{neighbouring\_cell=1}^{max=8(4\ for\ 2D)} \sum_{p=1}^{no\_of\_particle\_in\_cell=4} \left(\left(M_p \times V_p\right) \times N_p\right) \right)$$

where, V_p is the particle velocity

d) Strain update:

- Straintime = (timestep)/2
- Impose displacement boundary conditions: This is done by making respective momentum equal to zero.
- Get grid velocities:

$$pk_n = momentum_{node} = \frac{\left(\sum_{neighbouring\_cell=1}^{max=8(4\ for\ 2D)} \sum_{p=1}^{no\_of\_particle\_in\_cell=4} \left(\left(M_p \times V_p\right) \times N_p\right) \right)}{mass_{node}}$$

- Update strain:

$$\varepsilon_p^{xx} = \left[\sum_{neighbouring\_node=1}^{(4\ for\ 2D)} \left(v_n^x \times \frac{\partial N_p}{\partial \xi}_p \right) \right] \text{ x Straintime}$$

$$\varepsilon_p^{yy} = \left[\sum_{neighbouring\_node=1}^{(4\ for\ 2D)} \left(v_n^y \times \frac{\partial N_p}{\partial \eta}_p \right) \right] \text{ x Straintime}$$

$$\varepsilon_p^{xy} = \left[\sum_{neighbouring\_node=1}^{(4\ for\ 2D)} \left(v_n^x \times \frac{\partial N_p}{\partial \eta}_p \right) \right] \text{ x Straintime}$$

$$\varepsilon_p^{yx} = \left[\sum_{neighbouring\_node=1}^{(4\ for\ 2D)} \left(v_n^y \times \frac{\partial N_p}{\partial \xi}_p \right) \right] \text{ x Straintime}$$

- Specific stiffness matrix used to find stress from strain (constitutive matrix) is given by

$E/(1-\nu^2)*\rho$	$nu* E/(1-\nu^2)* \rho$	0
$Nu* E/(1-\nu^2) * \rho$	$E/(1-\nu^2) * \rho$	0
0	0	G/ ρ

- Particle stress and strain are calculated using, hypoelastic and adjusting for rotations using midpoint or endpoint derivatives

e) Get total grid point forces:

$$F_n^{TOTAL} = \left(F_n^{internal} + F_n^{external} + F_n^{damping}\right)$$

where, $F_n^{external}$ is given by

- External Force (for i=1 and 2 axes)

$$F_n^{external} = \sum_{neighbouring\_cell=1}^{max=8(4\ for\ 2D)} \sum_{p=1}^{no\_of\_particle\_in\_cell=4} \left(F_p^{external} \times N_p\right)$$

where, N_p is the Shape Function

$F_n^{external}$ = Mapped external force on node

$F_p^{external}$ = User applied external force on particle

- Internal Force (for i=1 and 2 axes)

$$F_n^{internal} = \sum_{neighbouring\_cell=1}^{max=8(4\ for\ 2D)} \sum_{p=1}^{no\_of\_particle\_in\_cell=4} \left(F_p^{internal} \times N_p\right)$$

where, internal force on particle is

$$F_p^{internal} = \left[\left(\sigma^s(i,1) \times N_{,1}\right) + \left(\sigma^s(i,2) \times N_{,2}\right)\right] \times M_p \ldots \text{For 2D-MPM}$$

where,

$N_{,1}$ is the derivative of shape function w.r.t. local-x (ξ)

$N_{,2}$ is the derivative of shape function w.r.t. local-y (η)

$\sigma^s(i, j)$ is the specific stress, with force in I-direction

- Currently no damping; Hence, $F_n^{damping} = F_p^{damping} = 0$

f) Impose Zero F^{total} on Displacement BC nodes.

3. Update Information

- Update grid momentum (for all the nodes):

$$pk_n^{t+\Delta t} = pk_n^t + \left(F_n^{total} \times timestep\right)$$

- Find new velocity at particles:

$$\Delta V_p = \sum_{neighbouring\_node=1}^{(4\ for\ 2D)} \frac{\left(pk_n^{t+\Delta t} \times N_p\right)}{m_n}$$

- Find new particle acceleration:

$$\Delta A_p = \sum_{neighbouring\_node=1}^{(4\ for\ 2D)} \frac{\left(F_n^{total} \times N_p\right)}{m_n}$$

- Update particle position:

Pos = ΔV_p x (timestep)

- Update particle velocity:

Velocity particle = ΔA_p x (timestep)

4. Rezero the momentum on grid and map the momentum from particle to grid again and again update particle strain (step number 2 (c) and 2 (d))

5. A loop is run to check assign the particles to the respective cells in which they fall. This is done using Ray Crossing Algorithm (Sec. 5.3)

In this investigation, the conventional MPM algorithm is modified to accommodate irregular mesh. The important thing is the modification involving determination of the local coordinates of the particle positions and the development of ray crossing algorithm to investigate which particle belongs to which cell after each time step.

5.2 Local (Natural) Coordinates for Updated Positions of Particles

These equations are derived with the help of Dr. Bo Wang. After the particle locations in the grid cells are determined, local (natural) coordinates for updated positions of particles are calculated for the next iteration in the MPM computation. In 2D situation, the shape function is of the same form as that used in finite element analysis and can be expressed as

$$N_i = \frac{1}{4}(1+\xi_0)(1+\eta_0) \qquad \xi_0 = \xi\xi_i,\ \eta_0 = \eta\eta_i \quad (i = 1,2,3,4) \tag{5.1}$$

and $i(\xi_i, \eta_i)$: 1 (-1, -1); 2 (1,-1); 3 (1, 1); 4(-1,1). The coordinates of the material points could be determined in terms of the coordinates of cell nodes which material points are associated with.

$$x_p = \sum_{n=1}^{N} x_i N_i(\xi_p, \eta_p) \quad y_p = \sum_{n=1}^{N} y_i N_i(\xi_p, \eta_p). \tag{5.2}$$

That is,

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta) \tag{5.2a}$$

$$N_2 = \frac{1}{4}(1+\xi)(1-\eta) \tag{5.2.b}$$

$$N_3 = \frac{1}{4}(1+\xi)(1+\eta) \tag{5.2.c}$$

$$N_4 = \frac{1}{4}(1-\xi)(1+\eta) \tag{5.2d}$$

$$Xp = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4 \tag{5.3}$$

$$Yp = N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4 \tag{5.4}$$

Substitute the values of N_1, N_2, N_3, N_4 in to Eqs. (5.3) and (5.4)

$$Xp = \frac{1}{4}(1-\xi)(1-\eta)x_1 + \frac{1}{4}(1+\xi)(1-\eta)x_2 + \frac{1}{4}(1+\xi)(1+\eta)x_3 + \frac{1}{4}(1-\xi)(1+\eta)x_4 \tag{5.5}$$

Multiplication by 4 on both sides would give

$$4Xp = (1-\xi)(1-\eta)x_1 + (1+\xi)(1-\eta)x_2 + (1+\xi)(1+\eta)x_3 + (1-\xi)(1+\eta)x_4 \tag{5.6}$$

Taking the common terms

$$4Xp = (1-\xi)((1-\eta)x_1 + (1+\eta)x_4) + (1+\xi)((1-\eta)x_2 + (1+\eta)x_3) \tag{5.7}$$

Similarly, substituting the values in Eq (5.3) and taking the common terms will give

$$4Yp = (1-\xi)((1-\eta)y_1 + (1+\eta)y_4) + (1+\xi)((1-\eta)y_2 + (1+\eta)y_3) \tag{5.8}$$

5.2.1 Parallel in X-direction

Applying the condition $y_1 = y_2, y_3 = y_4$ in Eq. (5.8) yields

$$4Yp = (-2y_1 + 2y_4)\eta + 2y_1 + 2y_4 \tag{5.9}$$

Eq (5.7) becomes

$$4\mathrm{Xp} = (-\mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3 - \mathrm{x}_4)\xi + (-\mathrm{x}_1 - \mathrm{x}_2 + \mathrm{x}_3 + \mathrm{x}_4)\eta + (\mathrm{x}_1 - \mathrm{x}_2 + \mathrm{x}_3 - \mathrm{x}_4)\xi\eta + \mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3 + \mathrm{x}_4 \tag{5.10}$$

Solving Eq. (5.9) and Eq. (5.10)

We can find η from Eq. (5.9) and substitute the value of η in Eq. (5.10) to find ξ

5.2.2 Parallel in Y-Direction

Applying this condition in Eq. (5.7) $x_1 = x_4, x_2 = x_3$

Eq. (5.7) comes

$$4Xp = (-2x_1 + 2x_2)\xi + 2x_1 + 2x_2 \tag{5.11}$$

Eq (5.8) becomes

$$4\mathrm{Yp} = (-\mathrm{y}_1 + \mathrm{y}_2 + \mathrm{y}_3 - \mathrm{y}_4)\xi + (-\mathrm{y}_1 - \mathrm{y}_2 + \mathrm{y}_3 + \mathrm{y}_4)\eta + (\mathrm{y}_1 - \mathrm{y}_2 + \mathrm{y}_3 - \mathrm{y}_4)\xi\eta + \mathrm{y}_1 + \mathrm{y}_2 + \mathrm{y}_3 + \mathrm{y}_4 \tag{5.12}$$

solving Eq. (5.11) and Eq. (5.12)

we can find ξ from Eq (5.11) and plug in the value of ξ in Eq. (5.12) to find η

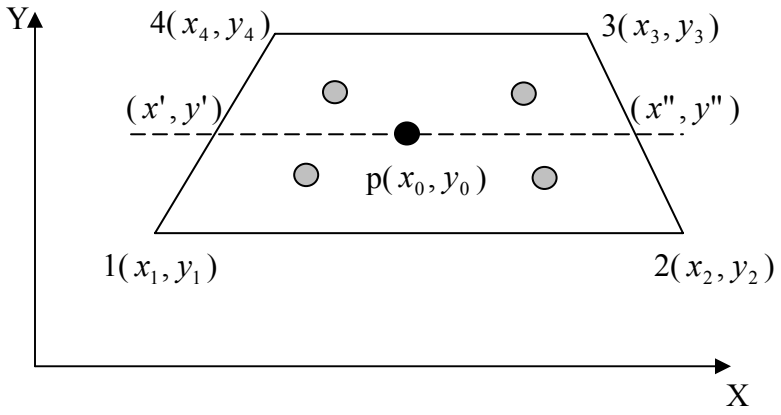


Figure 5.1 Quadrilateral Cell

5.2.3 Not Parallel in Either Direction

$$4Xp = (-x_1 + x_2 + x_3 - x_4)\xi + (-x_1 - x_2 + x_3 + x_4)\eta + (x_1 - x_2 + x_3 - x_4)\xi\eta + x_1 + x_2 + x_3 + x_4 \tag{5.13}$$

$$4Yp = (-y_1 + y_2 + y_3 - y_4)\xi + (-y_1 - y_2 + y_3 + y_4)\eta + (y_1 - y_2 + y_3 - y_4)\xi\eta + y_1 + y_2 + y_3 + y_4 \quad (5.14)$$

Taking the co-efficients of $\xi, \eta, \xi\eta$ from eq (5.13) and eq(5.14)

$\alpha_1 = (-x_1 + x_2 + x_3 - x_4)$ $\beta_1 = (-y_1 + y_2 + y_3 - y_4)$

$\alpha_2 = (-x_1 - x_2 + x_3 + x_4)$ $\beta_2 = (-y_1 - y_2 + y_3 + y_4)$

$\alpha_3 = (x_1 - x_2 + x_3 - x_4)$ $\beta_3 = (y_1 - y_2 + y_3 - y_4)$

$\alpha_4 = 4Xp - (x_1 + x_2 + x_3 + x_4)$ $\beta_4 = 4Yp - (y_1 + y_2 + y_3 + y_4)$

$$\alpha_4 = \alpha_1\xi + \alpha_2\eta + \alpha_3\xi\eta \quad (5.15)$$

$$\beta_4 = \beta_1\xi + \beta_2\eta + \beta_3\xi\eta \quad (5.16)$$

Solving Equations (5.15) and (5.16)

(5.15) x β_3 gives $\alpha_4 . \beta_3 = \beta_3 . \alpha_1\xi + \beta_3 . \alpha_2\eta + \beta_3 . \alpha_3\xi\eta$ (5.17)

(5.16) x α_3 gives $\beta_4 . \alpha_3 = \alpha_3 . \beta_1\xi + \alpha_3 . \beta_2\eta + \alpha_3 . \beta_3\xi\eta$ (5.18)

(5.17) - (5.18) yields

$$\alpha_4 . \beta_3 - \beta_4 . \alpha_3 = (\beta_3 . \alpha_1 - \alpha_3 . \beta_1)\xi + (\beta_3 . \alpha_2 - \alpha_3 . \beta_2)\eta \quad (5.19)$$

from Eq. (5.19)

$$\xi = (\alpha_4 . \beta_3 - \beta_4 . \alpha_3 - (\beta_3 . \alpha_2 - \alpha_3 . \beta_2)\eta)/(\beta_3 . \alpha_1 - \alpha_3 . \beta_1) \quad (5.20)$$

Plug Eq. (5.20) in Eq. (5.15) which gives

$$\alpha_4 = \alpha_1 . ((\alpha_4 . \beta_3 - \beta_4 . \alpha_3 - (\beta_3 . \alpha_2 - \alpha_3 . \beta_2)\eta)/(\beta_3 . \alpha_1 - \alpha_3 . \beta_1)) + \alpha_2\eta + \alpha_3 . ((\alpha_4 . \beta_3 - \beta_4 . \alpha_3 - (\beta_3 . \alpha_2 - \alpha_3 . \beta_2)\eta)/(\beta_3 . \alpha_1 - \alpha_3 . \beta_1))\eta \quad (5.21)$$

Solving Eq. (5.21)

$$(\alpha_3 . (\alpha_2 . \beta_3 - \beta_2 . \alpha_3)/\beta_3 . \alpha_1 - \alpha_3 . \beta_1)\eta^2 + (\alpha_1 . (\alpha_2 . \beta_3 - \beta_2 . \alpha_3)/(\beta_3 . \alpha_1 - \alpha_3 . \beta_1) - \alpha_2 - \alpha_3 . (\alpha_4 . \beta_3 - \beta_4 . \alpha_3)/(\beta_3 . \alpha_1 - \alpha_3 . \beta_1))\eta + \alpha_4 - (\alpha 1 (\alpha_4 . \beta_1 - \beta_4 . \alpha_3)/(\beta_3 . \alpha_1 - \alpha_3 . \beta_1)) = 0 \quad (5.22)$$

Taking the co-efficents of η^2 , η and the constants from Eq.(5.22)

A = α_3. (α_2. F- β_2. α_3)/ β_3. α_1- α_3. β_1; co-efficents of η^2

B = α1* (α_2. β_3- β_2. α_3)/ (β3 α_1- α_3. β_1) - α_2 - α_3. (α_4. β_3- β_4 . α_3)/ (β3. α_1- α_3. β_1); co-efficents of η

C= α_4 - (α_1. (α_4. β_3- β_4. α_3)/ (β3. α_1- α_3. β_1)); constant

Solving the above quadratic equation Eq. 5.22 by using the formula

η_1 = (-B + √ (B*B-4*A*C))/ (2*A); (5.23)

η_2 = (-B - √ (B*B-4*A*C))/ (2*A); (5.24)

The absolute values of η_1 and η_2 are compared and which ever values lie within '1' is taken as η

And substituting the value of η in the Eq. (5.20) will give ξ

5.3 Ray Crossing Algorithm

In MPM computation, the ray crossing algorithm [Fig. 5.2] is employed to determine which cell in the background grid mesh a particle belongs after the deformation [55]. In this section, the ray crossing algorithm is summarized as follows: A polygon made up of *N* vertices (x_i,y_i) where *i* ranges from *0* to *N-1* is considered herein. The last vertex (x_N,y_N) is assumed to be the same as the first vertex (x_0,y_0), i.e., the polygon is closed in Fig. 5.2. To determine the status of a point (x_p,y_p) consider a horizontal ray emanating from (x_p,y_p) and to the right. If the number of times this ray intersects with the line segments making up the polygon is even, the point is outside the

polygon. Whereas if the number of intersections is odd, the point (x_p,y_p) lies inside the polygon. Fig. 5.2 is a schematic illustrating the principle of ray crossing technique.

The number of intersections made by A is odd; so, the point is inside. The same case can be made for point C. Similarly, we can see in the Fig 5.2 points B and D crosses the polygon in even number. Therefore, these points are outside the polygon.

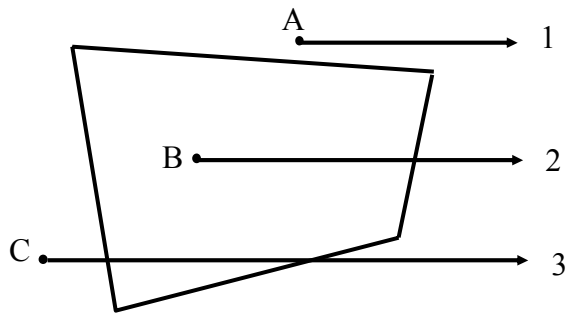


Figure 5.2 Ray Crossing Algorithm [55]

5.4 MPM Computational Algorithm

The MPM computational algorithm is summarized as follows:

1. Initialize material point locations, velocities, strains and stresses,
2. Add external load to the material points,
3. Determine the local co-ordinates. The difference between the conventional MPM and the new MPM algorithm is in this step, where a new approach has been derived for computing the local co-ordinates (Sec. 5.2).
4. Determine the shape function (based upon the derived local co-ordinates) for that particle with respect to the four surrounding nodes,

5. Map particle momentum, mass to the grid, using shape function based upon conservation of momentum and mass,
6. Update strain. The strain is updated here if the method is USAVG (Sec. 3.5), if the method is USL (Sec. 3.4) then the strain is not updated in this step,
7. Compute the internal force on node,
8. Map the external load from the particles to the nodes using shape function,
9. Calculate the total force on the nodes,
10. Impose the displacement boundary condition on the nodes.
11. Update momentum on the grid,
12. If the method is USL then determine the grid velocities,
13. Update material point locations based on grid velocity and update their velocities,
14. If the method is USL then update strain and stresses at the material points,
15. If the method is USAVG then extrapolate new particle velocities to the grid and determine new velocity gradients and strains and stresses at the material points,
16. Check whether the material point has crossed the element boundary using ray-crossing algorithm (Sec. 5.3). In case of conventional MPM, there is no need of special algorithm, as the grid cells are always square or rectangle in shape.
17. Regrid
18. Go to step 2.

These steps are repeated until the user defined time is reached. The program stops computing if the material point crosses the element boundary.

Grid mass is determined by mapping material point masses to the grid with the shape functions. Thus the small grid mass is due to the small values of shape functions.

When momentum is used, the numerator and denominator of the material point equations are balanced by the shape function value. That is, the numerator and denominator both contain a multiplication by the shape function, and numerical problems are avoided.

5.5 3D Material Point Method (MPM)

The analysis of complex three-dimensional components has become a common task in recent years in several fields. The implementation of 3D MPM should address more complex shapes. The 2D new MPM algorithm is extended to 3D, which can handle both the regular and irregular 3D elements. The modifications are made in the 2D MPM algorithm (Sec. 5.1) to extend to 3D.

The changes that are made in 2D to make it work for 3D are as follows, The update is made by extending the global co-ordinates from x, y to x, y and z and the local coordinates from ξ, η to ξ, η and τ . The local coordinates of the particles are determined by using Newton-Raphson method [56]. The shape functions are extended to 8-node isoparametric representation. The stiffness matrix is extended to 3D [57]. All the other variables such as position, velocity, acceleration, strain, stress are extended to "Z". The ray crossing algorithm which is used in 2D MPM is extended to 3D which is explained in the following section.

5.5.1 3D Ray Crossing Algorithm

The first step is to test the point along the XY axis same as 2D, but an extra loop is added to loop through all the faces of the element. We trace the point along the XY

plane in all the faces by generating a ray from the test point and the crossing number is tracked through all the faces. The next step is to trace the point along the Z axis and then store the crossing number in both the cases and check whether the crossing is odd or even, if the crossing is odd then the point is inside or if it is even then the point is outside. If the point is exactly on any vertex then it is checked by computing the equation of plane and checking its value to be exactly equal to zero, if yes then we consider the point to be inside.

5.5.2 Newton-Raphson Method

An iteration method for solving a system of n non-linear equations

$$f_1(x) = f_2(x) = ... = f_n(x) = 0$$

for the n variables $x = (x_1, x_2, ..., x_n)$. An approximate solution *x* must be known. Then a better approximation $x' = x + \Delta x$ is found from the approximate equations

$$f_j(x + \Delta x) = f_j(x) + \sum_{k=1}^{n} J_{jk}(x)\Delta x_k = 0, \qquad j = 1,...,n,$$

which are linear equations in the unknown Δx. The matrix J is the Jacobi matrix,

$$J_{jk} = \frac{\partial f_j}{\partial x_k}$$

The process is iterated until it converges, usually until Δx is smaller than the accuracy wanted in the solution, or until all the $f_j(x)$ are "sufficiently close to 0". Convergence may, of course, not be obtained if the first approximation was poor. In the two-dimensional case the Newton-Raphson formula

$$x' = x + \Delta x = x - f(x)/ f'(x)$$

has a very simple geometrical interpretation: it is the extrapolation to 0 along the tangent to the graph of $f(x)$ (also called *Newton's rule*). Only approximate solutions for Δx are required. A small error in Δx will not destroy the convergence completely, but may make it linear instead of quadratic. Hence also the Jacobian matrix J needs to be calculated only approximately, in particular it need often not be recalculated for each iteration. Double computer precision for x and $f(x)$ but single precision for J and Δx may give double precision for the final solution.

In fact, the Newton-Raphson method may be applied even to linear equations in order to give double precision solutions using single precision subroutines.

CHAPTER 6

MIXED MODE CRACK IN MPM

6 .1 Introduction

As a demonstration of the capability of the new MPM with irregular mesh, one can consider a inclined crack problem in mixed mode fracture. The theoretical framework on the numerical analysis of mixed mode fracture is explained in this chapter.

In engineering applications, cracks subjected to mixed mode loading can be attributed primarily due to three factors:

1. Mixed remote loading, i.e., remote normal and shearing forces acting on a component having a crack perpendicular to the normal loading direction,
2. Deflected or inclined crack under normal/uniaxial remote loading, and
3. Mechanical and/or thermal loads combined with arbitrary restraint conditions, producing a multiaxial loading conditions on a crack.

As mixed mode fracture problems are common in structural analysis [25], the understanding and analysis of mixed mode fracture problem is important in fracture mechanics. In numerical analysis, there are at least three methods available for the computation of the stress intensity factor (SIF) under mixed mode I/mode II loading conditions [26]:

1. The displacement extrapolation method,

2. The potential energy release rate method computed by means of a modified crack-closure integral technique, and
3. The direct J-integral computation method using the equivalent domain integral together with a mode decomposition scheme.

Bittencourt et al. [28] showed that for sufficiently refined finite element meshes all three methods give essentially the same results. Among the three methods, the displacement extrapolation method is the most convenient method to calculate Mode I and Mode II stress intensity factors in terms of relative displacements and locations along the free crack surfaces. This method is discussed in detail in the following sections and included in the MPM.

6.2 Fracture Modes

One of the most common failure modes for composite structures is delamination [19]. Figure 6.1 shows common fracture modes [49]. The remote loadings applied to composite components are typically resolved into interlaminar tension and shear stresses at discontinuities that create mixed-mode I, II, and III delaminations. To characterize the onset and growth of these delaminations the use of fracture mechanics has become a common practice over the past two decades [9-11]. The total strain energy release rate, G_T, the mode I component due to interlaminar tension, G_I, the mode II component due to interlaminar sliding shear, G_{II}, and the mode III component, G_{III}, due to interlaminar scissoring shear need to be calculated. In order to predict delamination onset or growth for two-dimensional problems, these calculated G components are compared to interlaminar fracture toughness properties measured over a range from pure mode I

loading to pure mode II loading [10-11, 17]. The virtual crack closure technique (VCCT) [24-26] is widely used for computing energy release rates.

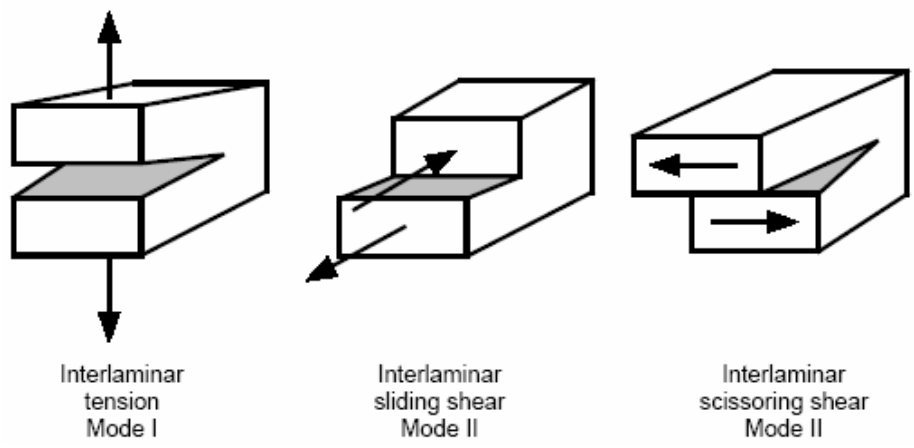


Figure 6.1 Fracture Modes [49]

6.3 Fracture Parameters for a Mixed Mode Crack

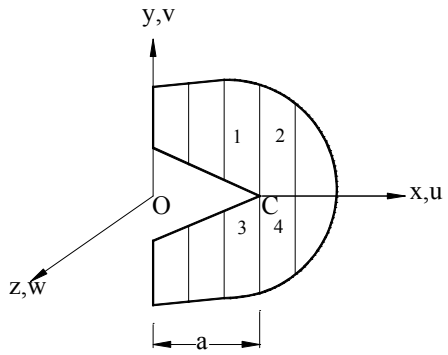
In computational fracture mechanics, the energy release rate for the crack propagation can be calculated using the Irwin's virtual crack closure method [29-30]. In this the crack is assumed to propagate an infinitesimal increment and the energy released during crack growth equals that required to close the propagated crack to its initial crack size. Based on this assumption, the total energy release rate for a mixed crack growth increment Δa is given by

$$G = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \sigma \cdot \Delta \bar{u} da = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} \{ \int_0^{\Delta a} \sigma_{yy} \Delta v da + \int_0^{\Delta a} \sigma_{yx} \Delta u da + \int_0^{\Delta a} \sigma_{yz} \Delta w da \} \quad (6.1)$$

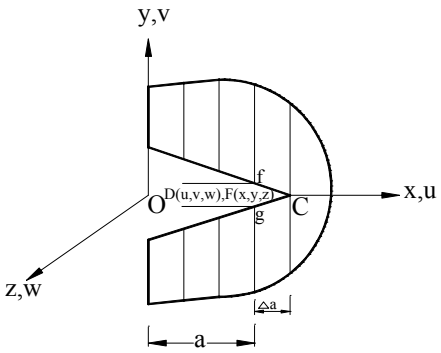
and for the 2D case, the Mode I and Mode II energy release rates for a mixed crack fracture will be

$$G_I == \lim_{\Delta a \to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \sigma_{yy} \Delta v da \ , \quad (6.2)$$

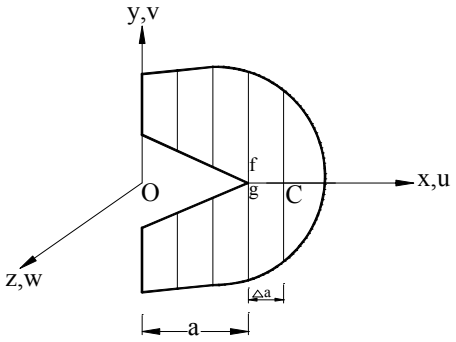
$$G_{II} = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} \int_0^{\Delta a} \sigma_{yx} \Delta u \, da \quad . \tag{6.3}$$



(a) Initial Stage



(b) Intermediate stage



(c) Final stage

Figures 6.2 (a) – (c) Schematic of Irwin Crack Closure Technique [29-30]

In numerical methods, the integral can be approximated using the nodal forces and nodal displacements of the crack tip. In Figs. 6.2 (b)-(c), the work required to close the propagated crack to its original crack size is given by

$$\Delta W = \frac{1}{2}[F_x(u_f - u_g) + F_y(v_f - v_g) + F_z(w_f - w_g)] \tag{6.4}$$

Based on the definition of G in Eq. (6.3), the energy release rate can be expressed as

$$G = \frac{\Delta W}{\Delta A} \tag{6.5}$$

where ΔA is the crack area increment due to the crack extension increment Δa. Thus, the Irwin's virtual crack closure technique can be implemented in FE and MPM methods. Usually, in elastic regime the stress intensity factor is directly related to the energy release rate. However, in the mixed mode situation this relation cannot be used easily.

6.3.1 Displacement Extrapolation Method

The displacement extrapolation method is used to determine the Mode I and Mode II stress intensity factors. Elastic solutions for the displacements at and near the crack tip are used in this method. Paris and Sih [31] gave the displacements for linear elastic materials as

$$u = \frac{K_I}{4G}\sqrt{\frac{r}{2\pi}}[(2\chi - 1)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}] - \frac{K_{II}}{4G}\sqrt{\frac{r}{2\pi}}[(2\chi + 3)\sin\frac{\theta}{2} - \sin\frac{3\theta}{2}] + O(r) \tag{6.6a}$$

$$v = \frac{K_I}{4G}\sqrt{\frac{r}{2\pi}}[(2\chi - 1)\sin\frac{\theta}{2} - \sin\frac{3\theta}{2}] - \frac{K_{II}}{4G}\sqrt{\frac{r}{2\pi}}[(2\chi + 3)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}] + O(r) \tag{6.6b}$$

$$w = \frac{2K_{III}}{G}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2} + O(r) \tag{6.6c}$$

where *u, v, w* are the local Cartesian displacements, (*r,* θ) are the local polar coordinates, G is the shear modulus, $\chi = 3 - 4\nu$ for plain strain or axisymmetric conditions, and $\chi = (3-\nu)/(1+\nu)$ for plane stress conditions, ν is the Poisson's ratio, and $O(r)$ represents terms of order *r* or higher.

Neglecting the higher order terms, and evaluating Eqs. [6.6(a to c)] at $\theta = \pm 180^{\circ}$ gives displacements along the free crack faces as

$$u = \frac{K_{II}}{2G}\sqrt{\frac{r}{2\pi}}(1+\chi) \tag{6.7a}$$

$$v = \frac{K_{I}}{2G}\sqrt{\frac{r}{2\pi}}(1+\chi) \tag{6.7b}$$

$$w = \frac{2K_{III}}{G}\sqrt{\frac{r}{2\pi}} \tag{6.7c}$$

Eqs. [6.7(a) - (c)] yield to

$$K_{I} = \sqrt{2\pi}\,\frac{G}{1+\chi}\,\frac{|\Delta v|}{\sqrt{r}} \tag{6.8a}$$

$$K_{II} = \sqrt{2\pi}\,\frac{G}{1+\chi}\,\frac{|\Delta u|}{\sqrt{r}} \tag{6.8b}$$

$$K_{III} = \sqrt{2\pi}\,\frac{G}{1+\chi}\,\frac{|\Delta w|}{\sqrt{r}} \tag{6.8c}$$

where Δu , Δv , and Δw are the relative displacements of one crack face with respect to the other. $\frac{|\Delta v|}{\sqrt{r}}, \frac{|\Delta u|}{\sqrt{r}}$, and $\frac{|\Delta w|}{\sqrt{r}}$ can be calculated based on the nodal displacements and locations along the free crack faces determined from the MPM computation.

CHAPTER 7

RESULTS AND DISCUSSION

The main objective of this investigation is to develop a new algorithm in MPM computation to allow the use of arbitrary quadrilateral cells in the background grid. Simple tensile specimens with/without an inclined crack are simulated using the refined MPM. The same models are analyzed using the commercially available ABAQUS/Explicit code. All the models have the same material properties. The linear elastic properties of the material are given as:

Mass density, $\rho = 1 gm/cm^3$

Young's modulus, E = 10000 MPa and

Poisson's ratio, $\nu = 0.001$. [20]

7.1 MPM and FEM Modeling

To validate the new MPM algorithm with an irregular mesh, a simple tensile test is simulated using two MPM models (regular and irregular mesh), and one FEM model using the commercially available ABAQUS/Explicit code. The geometry and boundary conditions of the tensile model without a crack is shown in Fig. 7.1. For the model without a crack the tensile load is applied at one end and the other end is fixed. Both ends

are subjected to tensile loads for the specimen with a crack. The geometry and boundary conditions of the tensile model with an inclined crack is shown in Fig. 7.2.

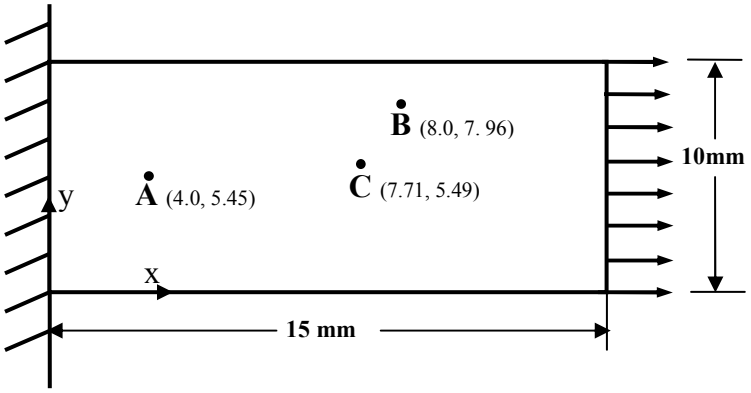


Figure 7.1 Tensile Model without crack

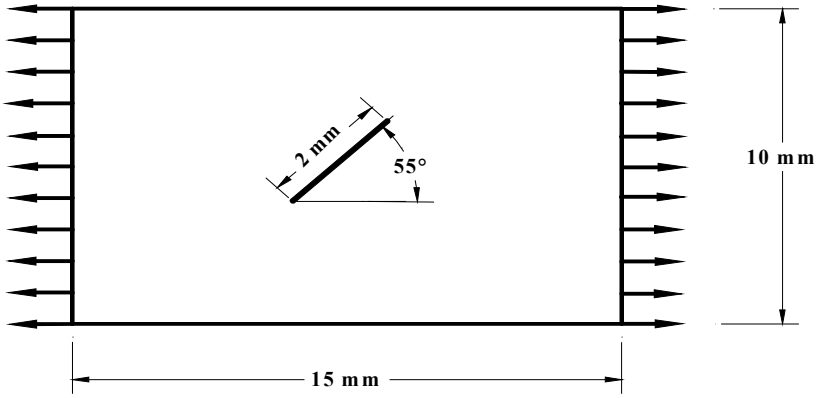


Figure 7.2 Tensile Model with an Inclined Crack

In both MPM and FEM models free crack surfaces are introduced using an approach that allows the placement of two sets of nodes at the same locations along the crack line to avoid the interaction between neighboring cells/elements on the two sides of the crack line, as shown in Fig. 7.3.

To allow direct comparison between MPM and FE results, meshes used in both models are the same. Details of the FEM tensile model with an inclined crack are shown

in Fig 7.4. Preprocessing code I-Deas is used to generate the MPM and FEM meshes and ABAQUS/Explicit is used for FE analysis.

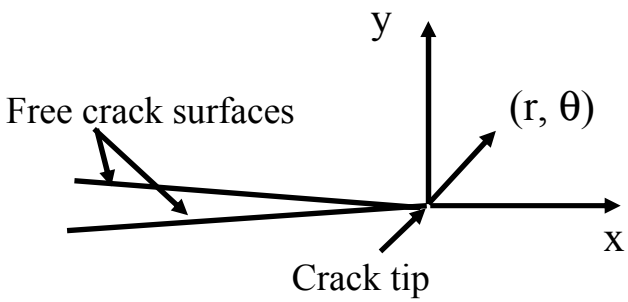


Figure 7.3 Model showing Free Crack Surfaces

To characterize the stress singularity near the crack tip, very fine mesh is used around the crack and a coarse mesh is used in the far field to minimize the computation time and to raise the computation capacity as shown in Fig 7.4. This adaptive mesh will also improve the accuracy of fracture parameter calculations. During the entire computation in MPM, the background grid mesh never changes. Thus, material points can move in or out of the cells after deformation.

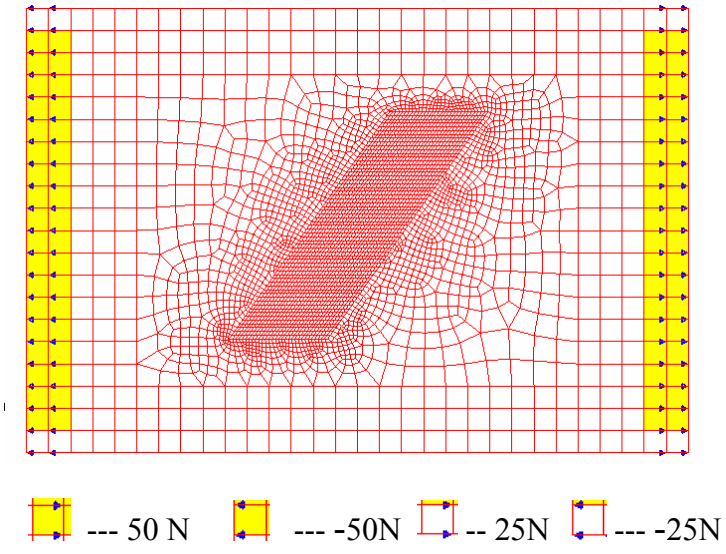


Figure 7.4 FE Crack Model Showing Boundary Conditions

Figs. 7.5 (a) –(c) show the MPM and FEM mesh for the tensile models without crack and Figs. 7.6 (a) – (b) with an inclined crack, respectively. MPM can create regular square or rectangular elements, if we specify the required dimensions. It does not require any preprocessor to create the mesh. There are 600 particles in the MPM model with regular mesh as shown in Fig. 7.5 (a).

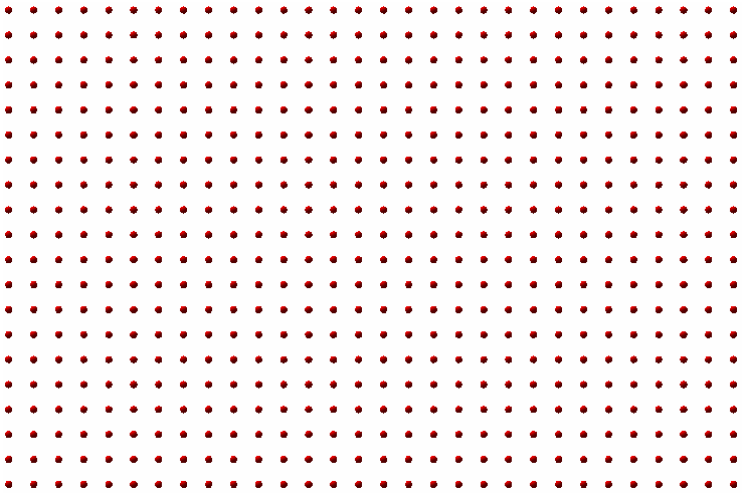


Figure 7.5 (a) Regular MPM Mesh

To create the irregular mesh, first the mesh is modeled using I-Deas. The I-Deas software will generate an input file, which will have the nodal co-ordinates and the element connectivity. The generated input file is then converted into MPM format by using a macro developed in EXCEL and then the corresponding load and the material properties are updated for the current input file and then given as an input to the MPM program. Thus, the MPM model with irregular mesh is created. The model shown in Fig 7.5 (b) contains 1,616 particles.

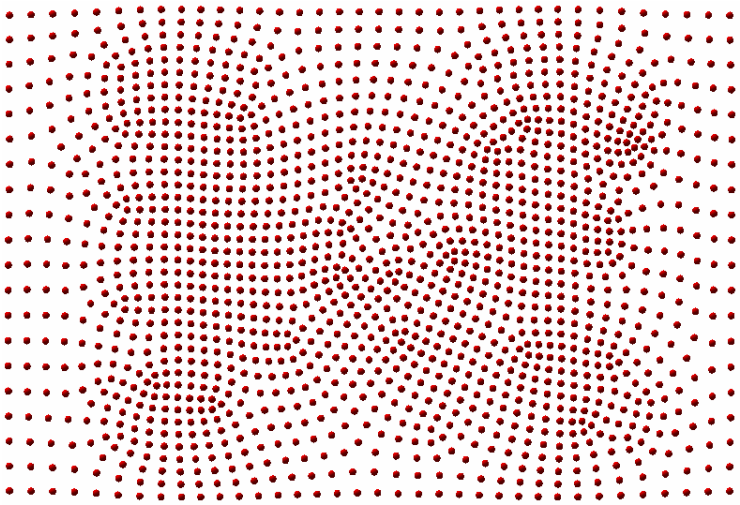


Figure 7.5 (b) Irregular MPM Mesh

For FEM model, the input file that is generated by I-Deas is given as an input to ABAQUS/Explicit code, which is used for the computation of FEM models. There are 433 nodes and 404 elements in the FEM model as shown in Fig. 7.5 (c).

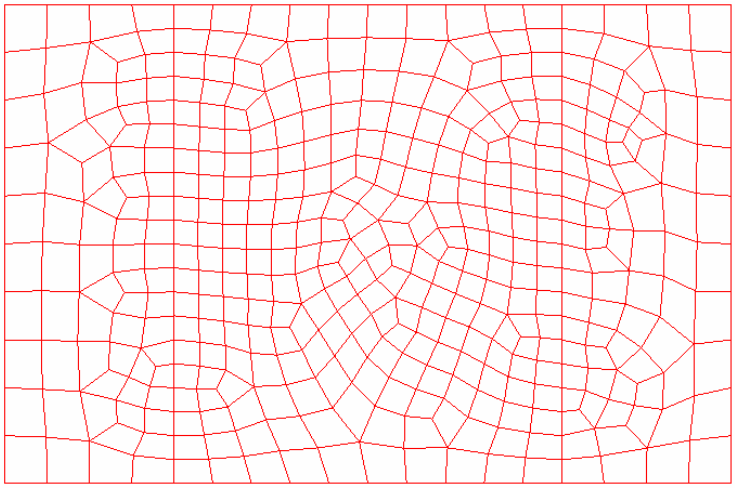


Figure 7.5 (c) Irregular FEM Mesh

For the tensile model with an inclined crack, the MPM model is created by the same procedure as we have used to create the MPM irregular mesh. The MPM model with an inclined crack has 14,628 particles as shown in Fig. 7.6 (a)

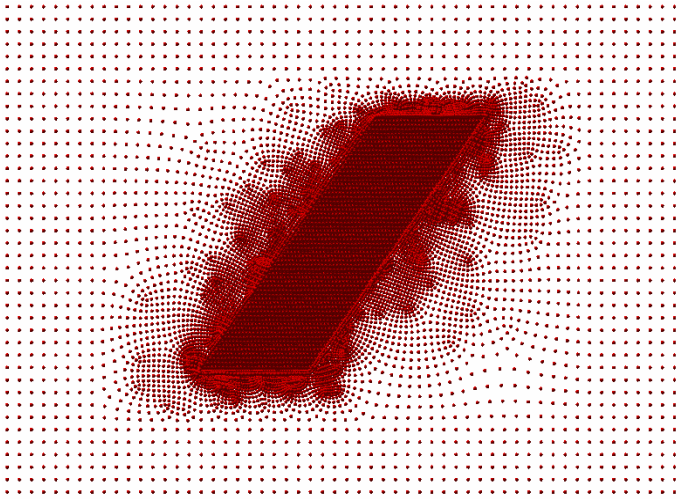


Figure 7.6 (a) MPM mesh with crack

The same input file, used to create the MPM model with a crack, is used in creating the FEM model. It has 3,767 nodes and 3,697 elements as shown in Fig. 7.6 (b).

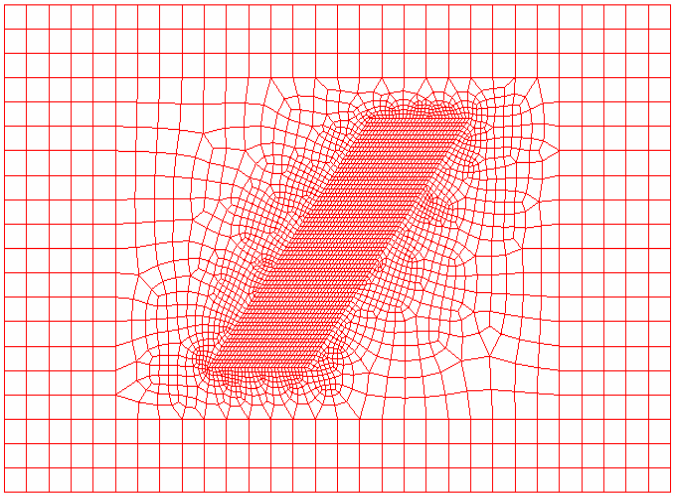


Figure 7.6 (b) FE Mesh with Crack

For explicit dynamic MPM/FEM analyses, the time step increment depends on the length of the cell/element and the stress wave speed. The minimum time step increment is taken to implement the numerical simulations. At the end of each time step, the deformed grid is effectively reset to its undeformed position. The details about the total time and the time step increment for each model are given in Table 7.1.

Table 7.1 Total time and time step increment for each model

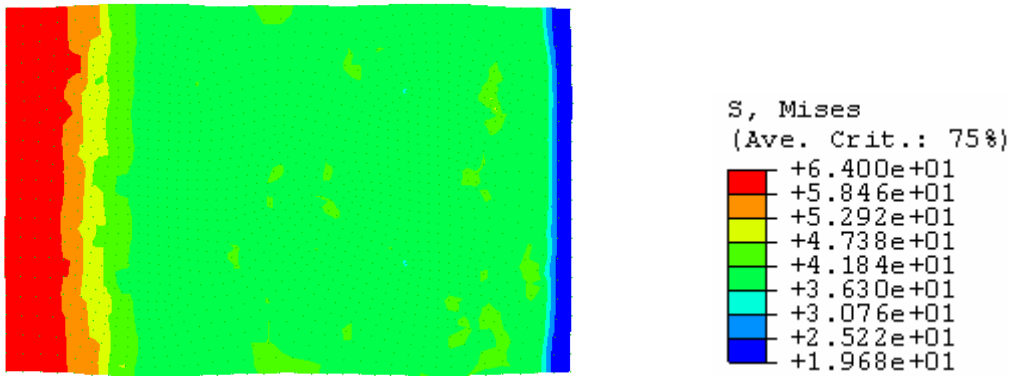
Model	Total time, μs	Time step increment, ns
MPM with a regular mesh	5.1	14
MPM with an irregular mesh	5.1	73
FEM	5.1	73
MPM with a crack	5.1	14
FEM with a crack	5.1	14

7.2 Tensile Model without a Crack

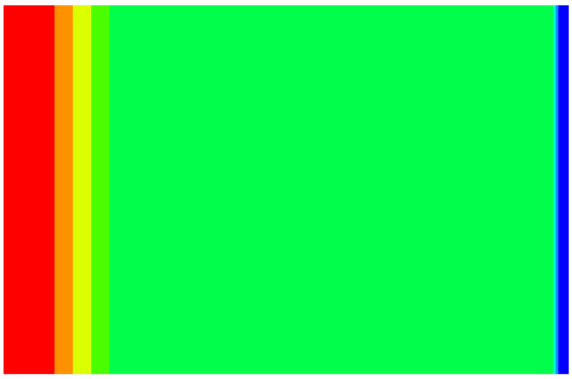
Explicit dynamic simulations were conducted on the tensile specimen without a crack using two MPM models (regular and irregular mesh) and one FEM model. A tensile load of 400 N is applied at one end and the other end is fixed. After running MPM and FEM simulations, displacement and stress fields were obtained for these three tensile models. To validate the new algorithm the following validations are performed.

The first validation is done by comparing the overall stress profiles of the three models. The stresses are calculated using the USL method (Sec 3.4). To ensure direct comparison, the von Mises stress contour plots are compared at the final time step for all three models. The FEM von Mises stress contour plot is directly obtained from ABAQUS. As MPM cannot directly generate contour plots, the output data is passed on to TECPLOT to generate the von Mises stress contours. Figs. 7.7 (a) – (c) shows von Mises stress distribution contours for t = 5μs for the MPM models with irregular/regular mesh and FE model without crack. In Fig. 7.7 (c), the FE results are smooth, because the

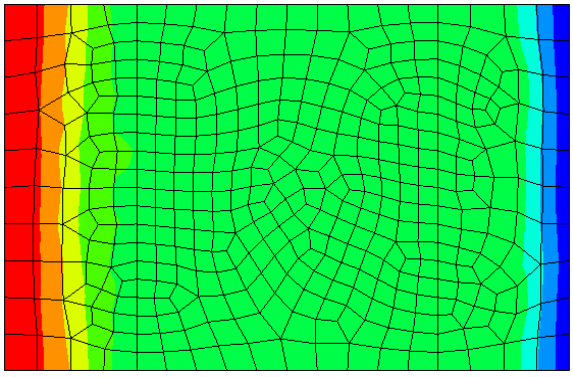
ABAQUS does post processing to smoothen the data. The overall stress profiles of all the models are same which exhibits the robustness of the code.



(a) Irregular MPM Mesh



(b) Regular MPM Mesh



(c) FE Mesh

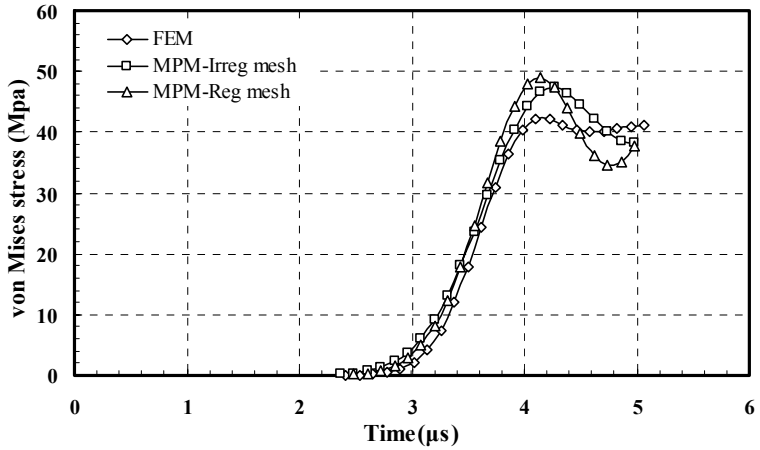
Figure 7.7 (a)-(c) Comparison of the von Mises Stress Contours for Irregular MPM/Regular MPM and FE Mesh at t = 5μs

The second validation is performed by comparing the displacements along x-direction and von Mises stress as a function of time at some specific locations A, B, C (Fig. 7.1) as shown in Fig. 7.9 (a) –(c). As the load is applied on one end, the stress wave travels from right to left side with time at the stress wave speed of C = 3162.3 m/sec.

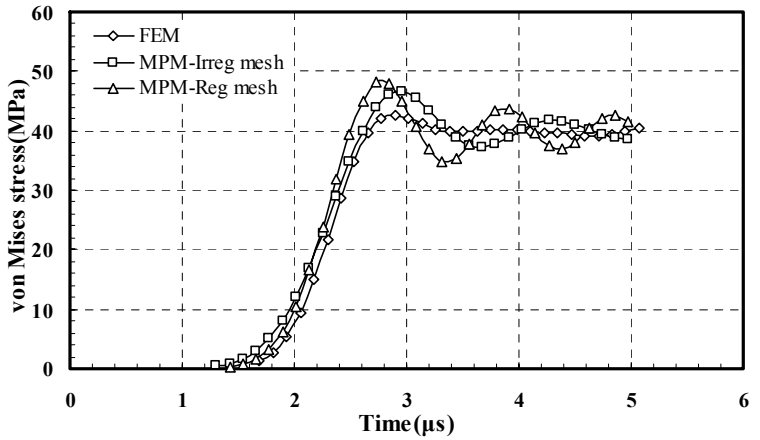
In Fig. 7.8 (a) – (c) the von Mises stress is plotted for MPM (regular/irregular) models and compared with FEM. In this figures the stresses are calculated using the USAVG method (Sec 3.5), where as in Fig. 7.9 (a) – (c) the values are calculated using the USL method (Sec 3.4) which gives the smooth result than the USAVG case , that's the reason why USL method is chosen for the computation.

The von Mises stress and the displacement values are taken from the nodes in the case of FEM model. In the case of MPM models (irregular/regular) one cannot take the values from the nodes as the background grid in the MPM does not deform, so one has to take the values from the surrounding particles closer to the node and take the average of those values. Thus, slight differences are observed in the values between the FEM and the MPM models.

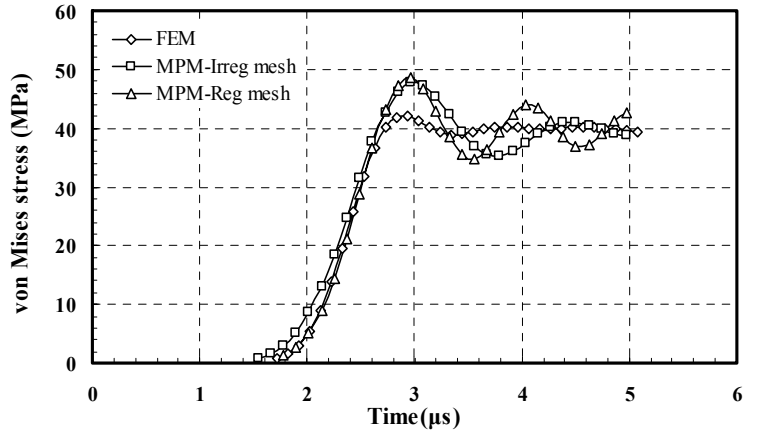
At t = 0.5 L/C where L is the length of the model, the stress wave reaches exactly at the center of the model at t = 2.3μs. The deformation of the particle closer to location A starts deforming at t = 3μs, as location A is closer to the left side of the model. The stress wave takes time to reach the location where as at location B and C the deformation starts at t = 2μs as they are closer to the right end of the model. This can be observed in cases of both the displacements and the von Mises stress.



(a) von Mises stress vs. time at location A (4.0 mm, 5.45 mm)



(b) von Mises stress vs. time at location B (8.0 mm, 7.96 mm)



(c) von Mises stress vs. time at location C (7.71 mm, 5.49 mm)

Figure 7.8 (a) – (c) von Mises stress (USAVG) plot vs. time

The deformation of particles at locations B and C are the same as they are in the same section of the model. The von Mises stresses in the Figs 7.9 (a) – (c) are close to the theoretical value of 40 MPa.

It can be seen from Figs 7.7 (a) – (c) and 7.9 (a) – (c) that FEM and MPM (irregular/regular) are in good agreement, indicating the validity of the new MPM algorithm using irregular grid mesh.

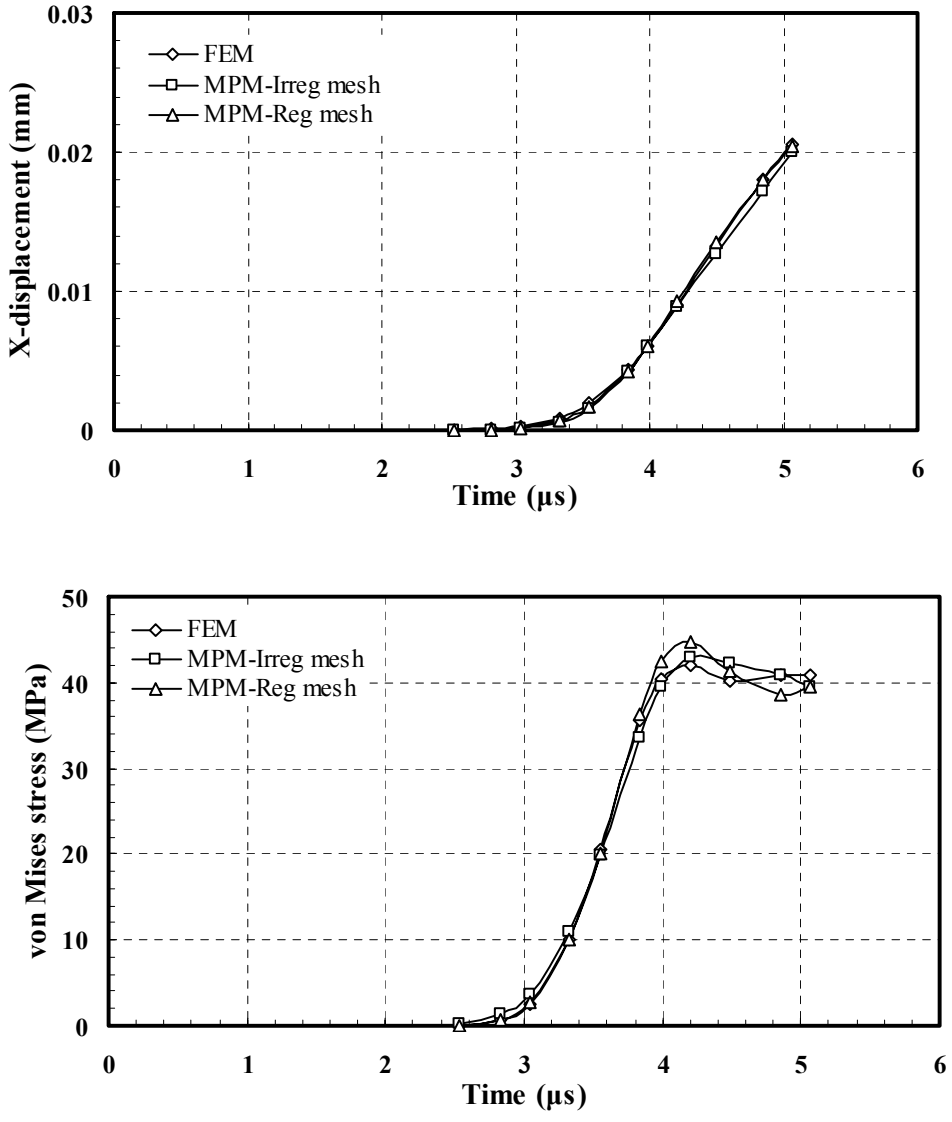


Figure 7.9 (a) Comparison of the X-displacement and the von Mises Stress with time at location A (4.0 mm, 5.45 mm)

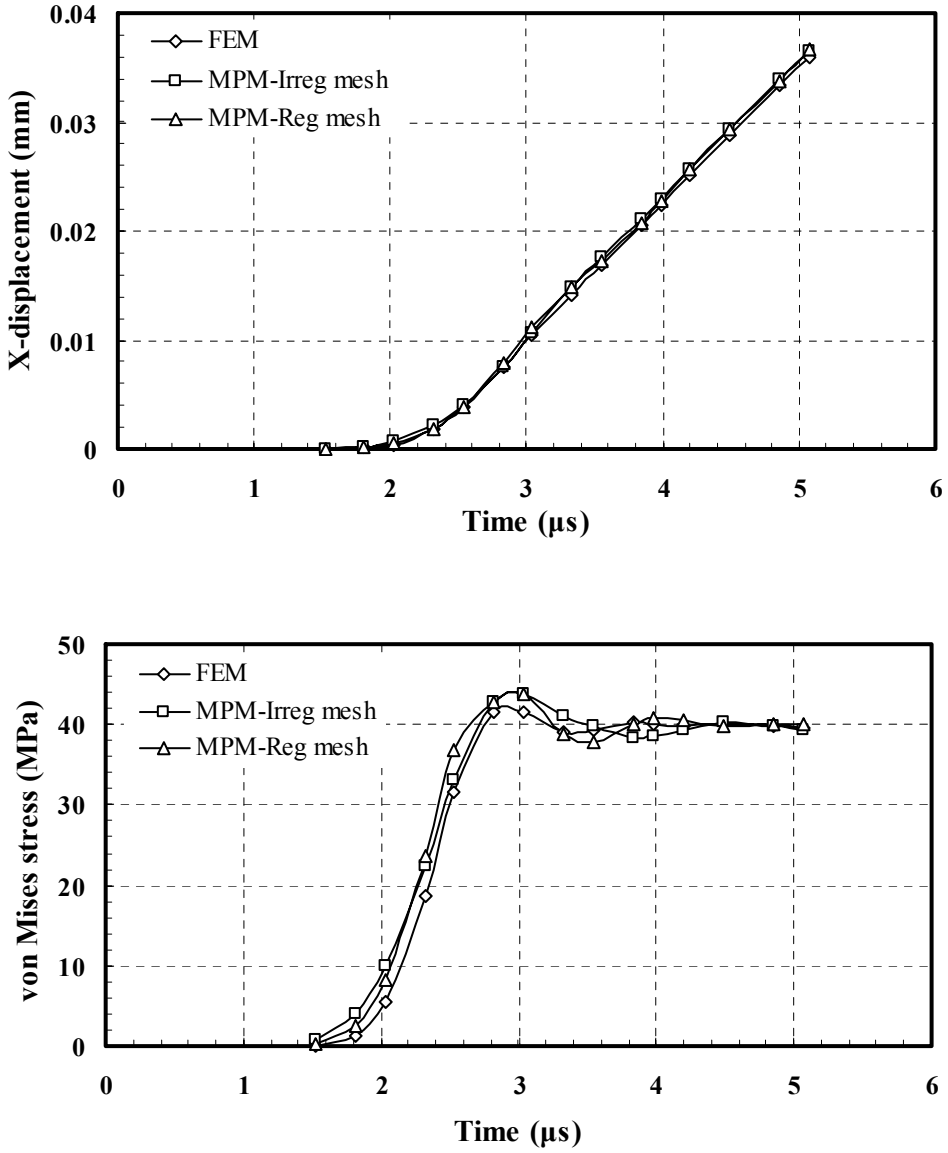
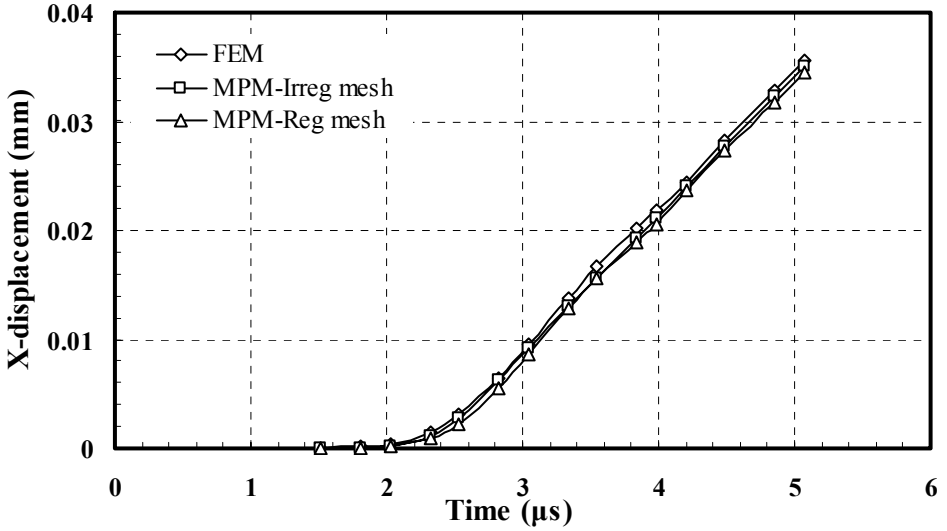


Figure 7.9 (b) Comparison of the X-displacement and the von Mises Stress with time at location B (8.0 mm, 7.96 mm)



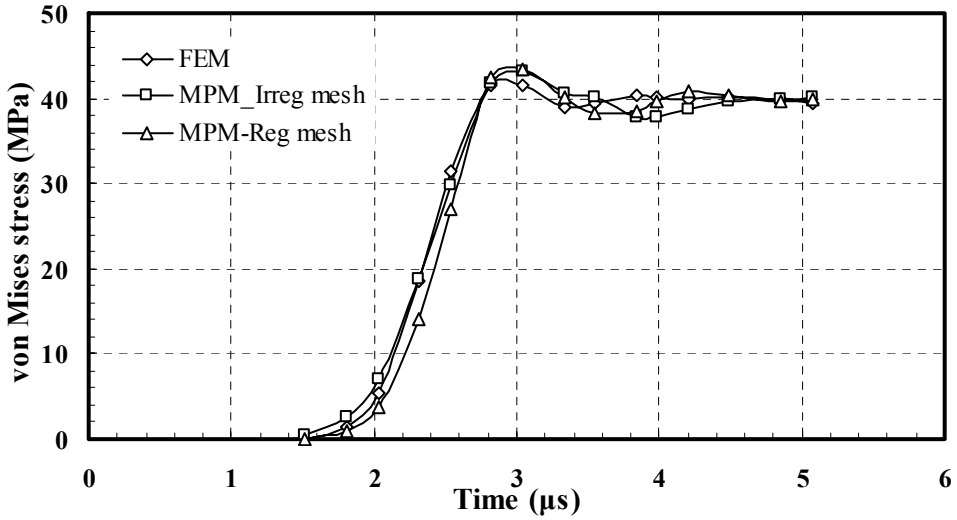


Figure 7.9 (c) Comparison of the X-displacement and the von Mises Stress with time at location C (7.71 mm, 5.49 mm)

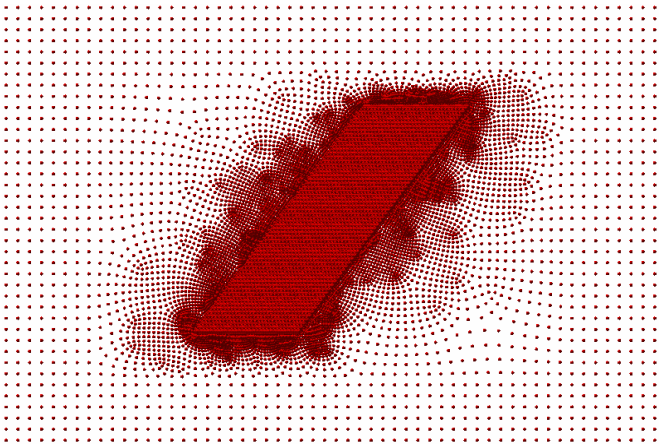
7.3 Tensile Model with an Inclined Crack

A specimen with an inclined crack is subjected to a tensile load to demonstrate the capability of arbitrary quadrilateral cells in the background mesh. As shown in Fig. 7.6 (a) – (b) the MPM and FEM mesh are modeled using I-Deas as discussed in the previous

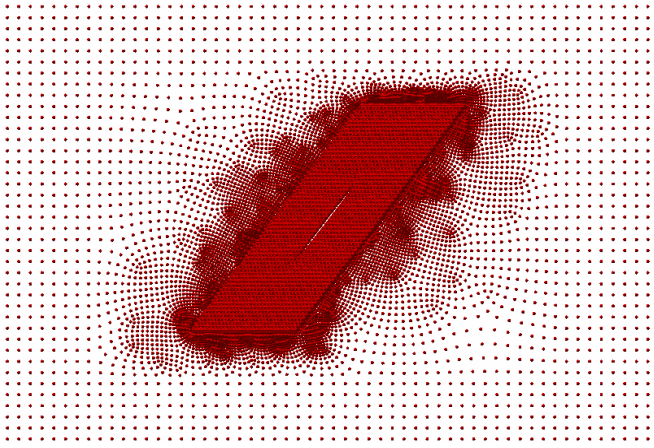
sections. On both ends of the model an equivalent tensile load of 2000N is applied. In the case of FEM model the same load is applied as shown in Fig. 7.4. As soon as the load is applied the stress wave starts moving from both ends as time increases from t = 0 to 5.1μs. Figs 7.10 (a) – (f) show the snapshots of the movement of the stress wave for the MPM model showing various stages of the crack opening at different time steps.

Fig 7.10 (a) shows the initial stage of the model. At this stage the load is not yet applied to the model so one cannot observe any deformation in the particles. In Fig 7.10 (b), the particle starts moving as the stress wave reaches the center of the model. The particles on the free crack surface starts deforming and one can observe the starting stage of the crack opening. As time step increases the stress wave travels further to the end of the model, Figs 7.10 (c) – (e) show the intermediate and the final stages of the crack opening. The stress wave after reaching the left end of the model, reflects back and becomes a compressive wave and the crack opening gets reduced which one can observe in the Fig 7.10 f.

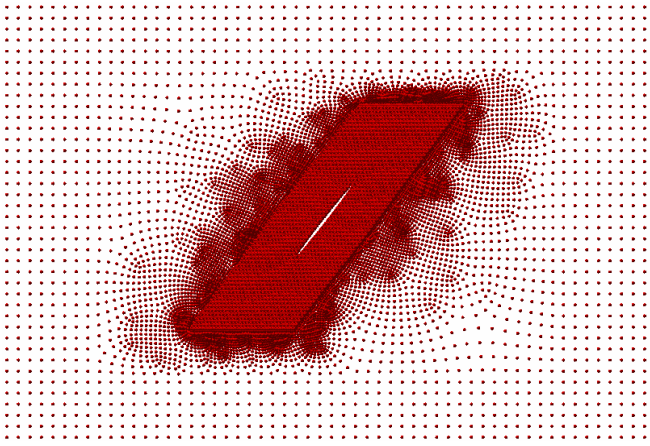
Figs 7.11 (a) – (f) shows different stages of the stress wave movement for the FEM model. Fig. 7.11 (a) show the initial stage of the model. At t = 2.5 μs the stress wave starts moving and reaches the center of the model. In Fig 7.11 (c) – (f) one can observe the stress profiles of the model gradually moving and as the crack starts opening there exist high stress near the crack which shows the stress singularity near the crack tip.



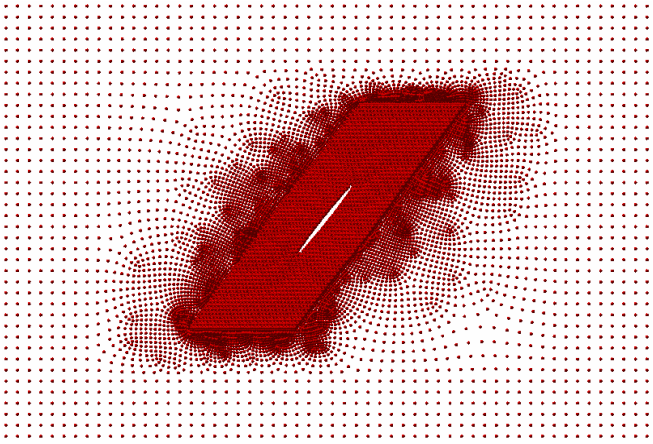
(a) At t = 0μs



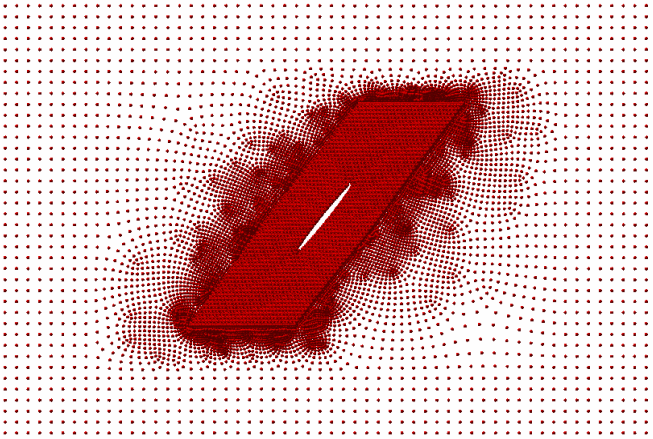
(b) At t = 2.5μs



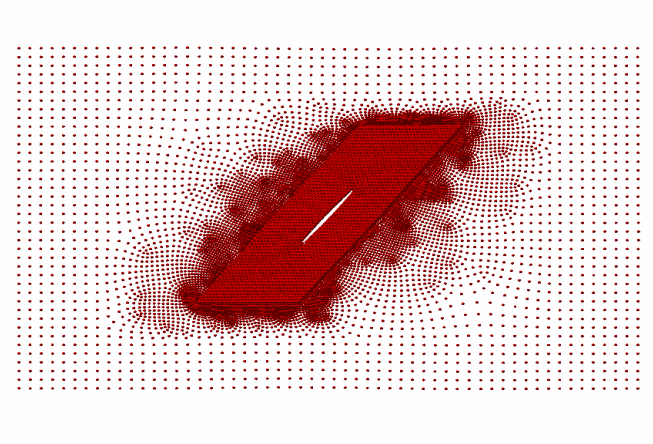
(c) At t = 3μs



(d) At t = 3.5μs

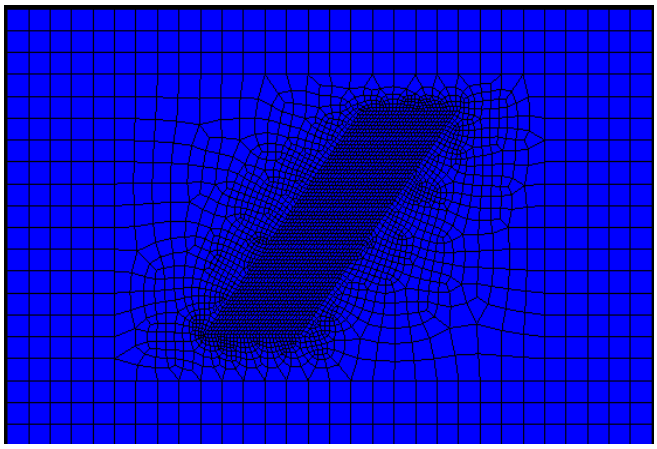


(e) At t = 4μs

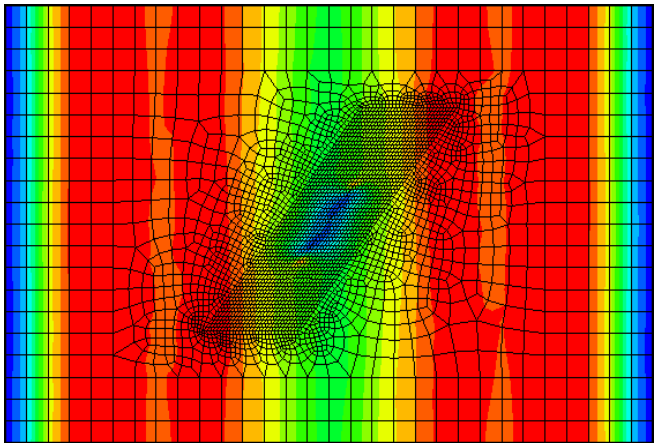


(f) At t = 5μs

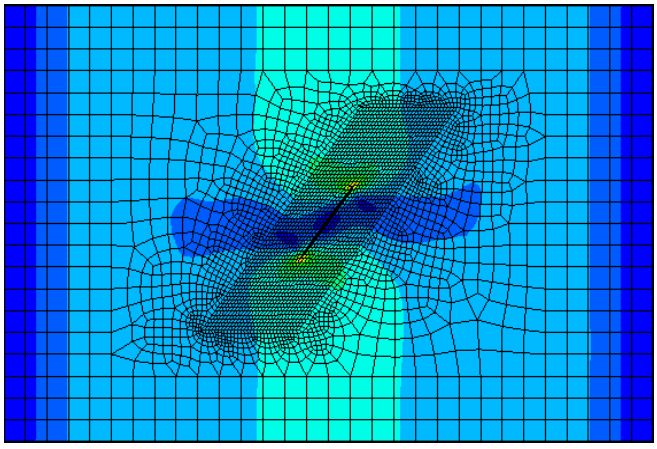
Figures 7.10 (a) – (f) Simulation of MPM crack model showing various stages of crack opening



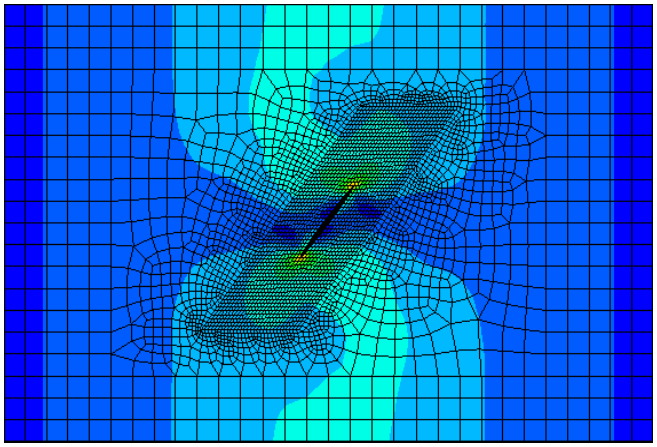
(a) At t = 0μs



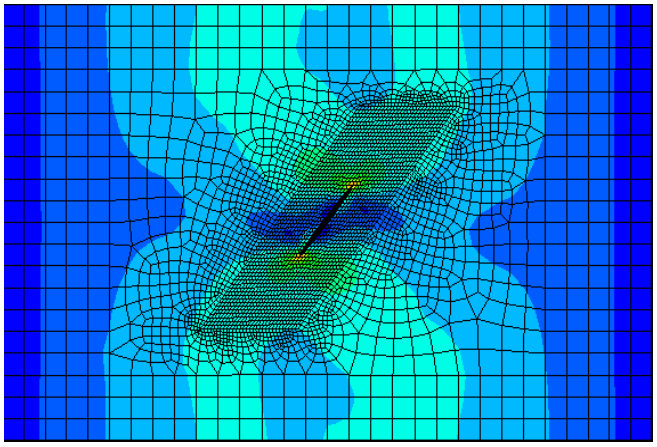
(b) At t = 2.5μs



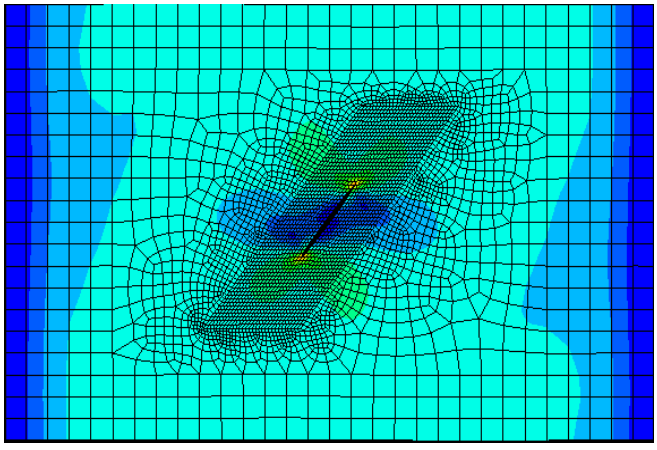
(c) At t = 3μs



(d) At t = 3.5μs



(e) At t = 4μs



(f) At t =5μs

Figures 7.11 (a) – (f) Simulation of FEM crack model showing various stages of crack opening and stress contours.

To validate the tensile model with an inclined crack, the stress distributions along the crack line are compared for FEM and MPM at t = 4μs as shown in Fig 7.13. Stress distribution curve for MPM model at different times with respect to the distance from the crack tip is shown in Fig 7.12. In this figure, there exists a high stress gradient near the crack tip showing the stress singularity of the crack. It starts decreasing as one moves away from the crack tip. The trend is similar at different time steps showing the stress distribution well in agreement with the stress singularity. It can be seen from Fig 7.13 the results of both MPM and FE models are consistent. The small difference may be due to the computational errors in averaging the values of the particles to the corresponding nodes of FEM.

7.3.1 Energy Release Rate

An adaptive mesh is used in MPM to improve the accuracy of fracture parameter calculations. Using the crack closure technique described in Sec. 6.3, the dynamic energy release rates in Mode I and Mode II are calculated. As we are solving a mixed mode crack problem both Mode I and Mode II plays an important role. The total energy release rate is obtained by adding Mode I and Mode II energy release rates. The Mode I, Mode II and the total energy release rate are compared with FEM and it can be seen from Fig. 7.14 (a) – (c) that MPM results are in good agreement. From these figures it can be seen that as soon as the stress wave reaches the center of the model at approximately t = 2.3μs, the crack gradually starts opening as shown in Fig 7.14 b and the energy release rate starts increasing as shown in the Figs 7.14 (a) –(c).

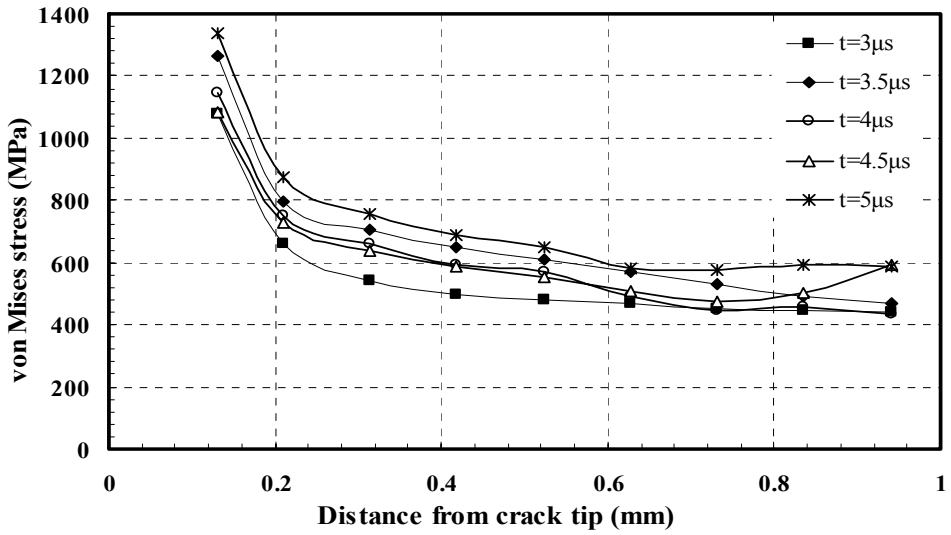


Figure 7.12 Stress distributions along the crack line near the crack tip at different times

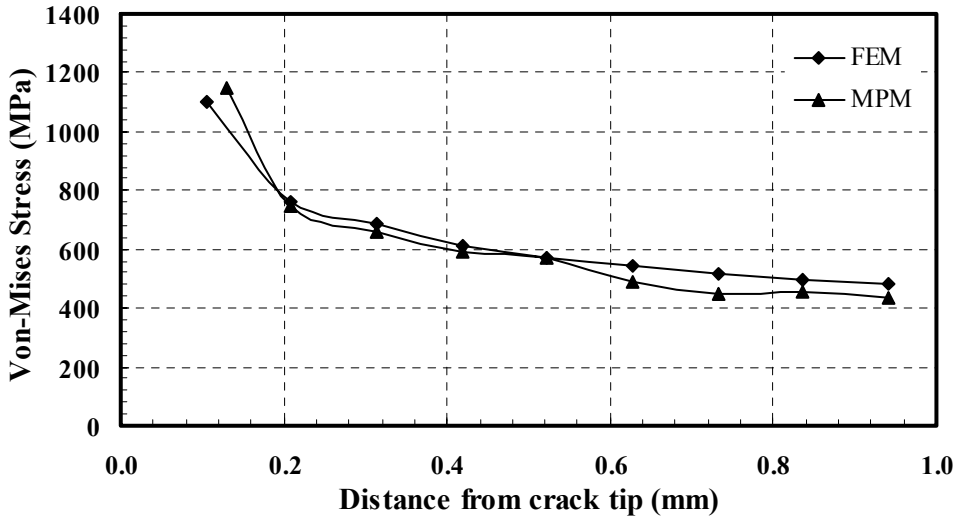
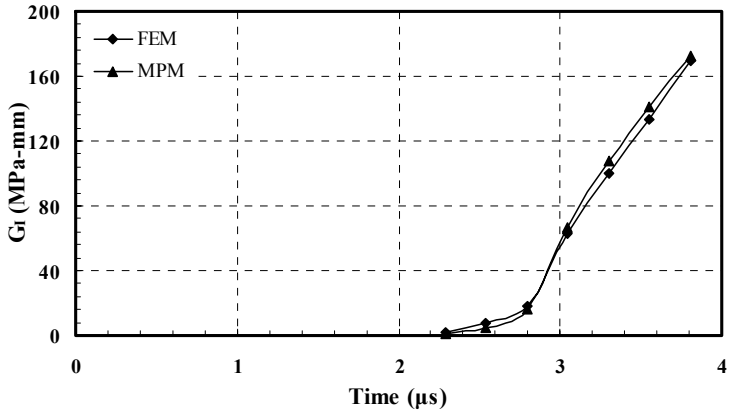
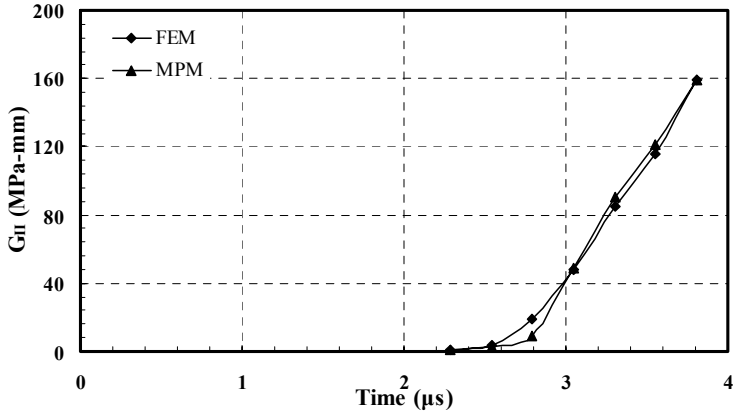


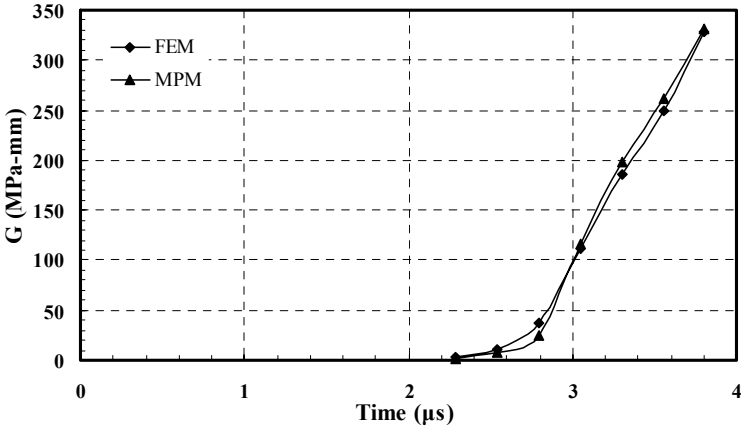
Figure 7.13 Stress distribution along the crack line near the crack tip at t = 4μs



(a) Mode I Energy release rate vs. time



(b) Mode II energy release rate vs. time

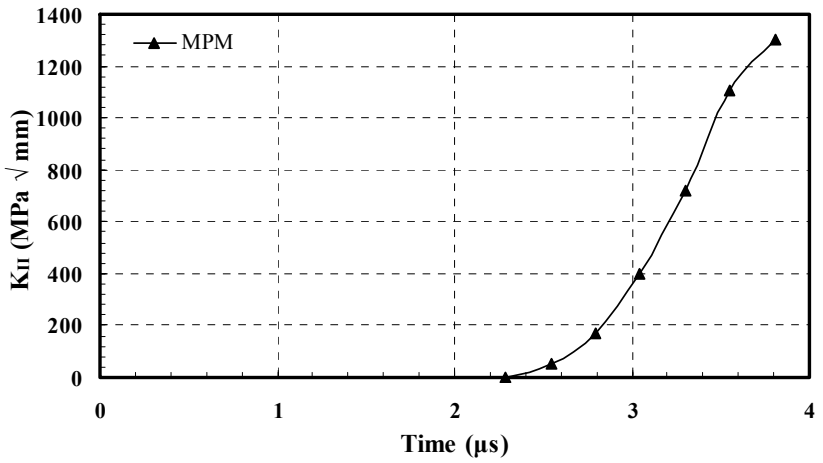


(c) Total energy release rate vs. time

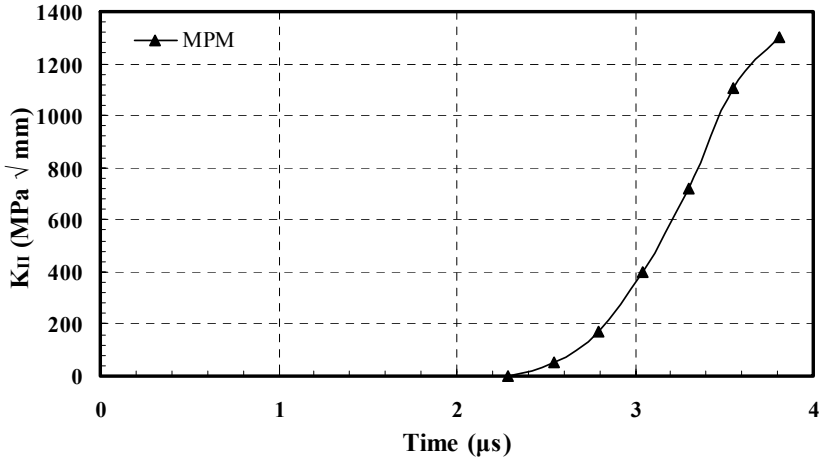
Figure 7.14 (a) – (c) Mode I, Mode II, and Total Energy Release Rate vs. Time

7.3.2 Stress Intensity Factor

The displacement extrapolation method (Sec. 6.3.1) is used to determine the Mode I and Mode II stress intensity factors for MPM and FE models. The Mode I and Mode II stress intensity factor vs. time curves are shown in Fig 7.15 (a) and (b). After the stress wave from both ends of the model arrives to the center of the model, the crack starts opening and K_I and K_{II} starts increasing with time.



(a) Mode I stress intensity factor vs. time



(b) Mode II stress intensity factor vs. time

Figure 7.15 (a) – (b) Mode I and Mode II Stress Intensity Factor vs. Time

From the above results from MPM and FEM models with an inclined crack, it can be seen that the proposed MPM algorithm with an irregular grid mesh can be used to simulate the mixed mode crack growth and explicit dynamic fracture mechanics computations can be implemented.

7.4 3D Irregular MPM

To validate the new 3D MPM algorithm with an irregular mesh, a simple tensile specimen was used to simulated two MPM models (regular and irregular mesh) and compared with the analytical solution [20]. The geometry and boundary conditions of the tensile model are shown in Fig. 7.16. A tensile load is applied at one end and the other end is fixed.

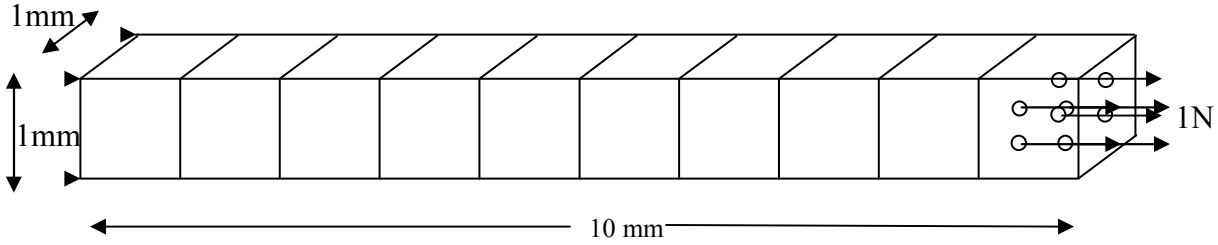


Figure 7.16 Schematic of the 3D Tensile Model showing geometry and boundary conditions

Explicit dynamic simulations are conducted on regular and irregular MPM models. A tensile load of 1N is applied on one end and fixed along x-direction in the other end. The load is equally distributed on all the 8 particles. The material properties used for the tensile model are the same as the ones used for 2D.

The 3D regular mesh shown in Fig 7.17 has 80 material points and it is simulated for t = 5μs. The boundary conditions are applied as shown in Fig 7.16.

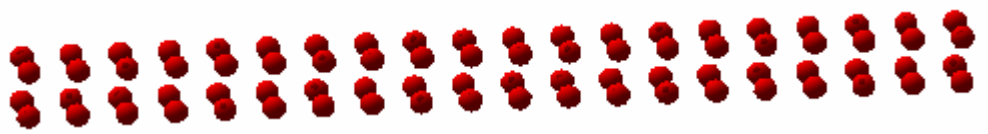


Figure 7.17 3D Regular MPM Mesh

The 3D irregular mesh shown in Fig 7.18 has 80 material points and to ensure direct comparison same boundary conditions are applied to the irregular model and simulated for the same time as the regular mesh model.

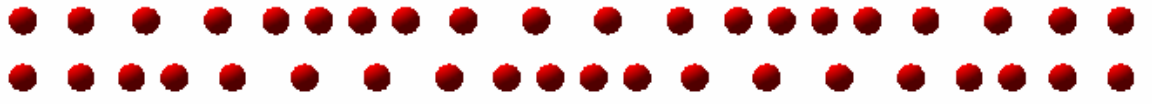


Figure 7.18 (a) Front View of 3D Irregular MPM Mesh

Figure 7.18 (b) Tilted View of 3D Irregular MPM Mesh

Chen and Brannon [20] has reported an analytical solution for a dynamic problem with same material properties, geometry, and boundary conditions. This analytical solution is used to validate the 3D MPM algorithm.

Fig 7.19 shows the stress S_{xx} along the length of the model at t = 0.5 L/C where L is the length of the tensile bar shown in Fig 7.16 and C is the wave speed which can be calculated based upon the material properties of the model. The plot shows that the regular and irregular mesh are good in agreement. As the mesh is too coarse it is not consistent with the theoretical model. It is possible to get good agreement by refining the mesh.

Fig. 7.20 shows the stress wave propagation at t = 1.5 L/C, i.e when the stress wave has reached one end of the model and reflected back to the center. The stress wave is plotted for regular mesh, irregular mesh and the analytical model as referred before the values would agree if we refine the mesh further.

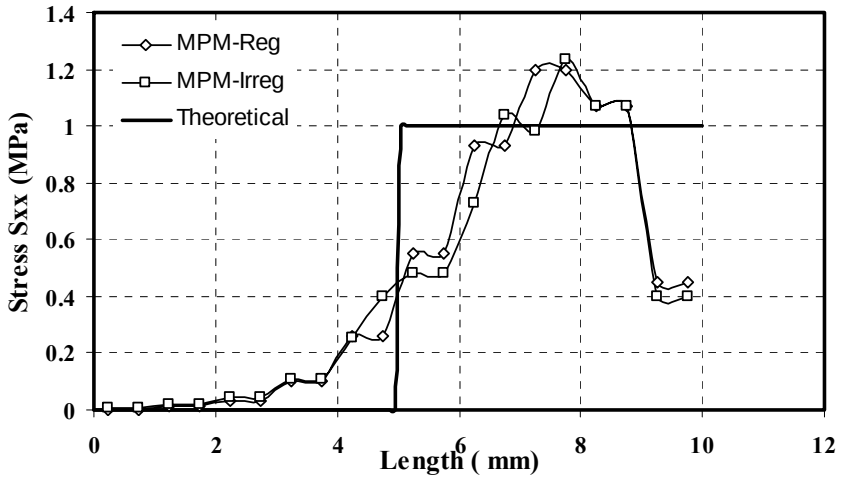


Figure 7.19 Stress along the length of the Model for Regular, Irregular MPM and Analytical Model at t = 0.5 L/C (1.6μs)

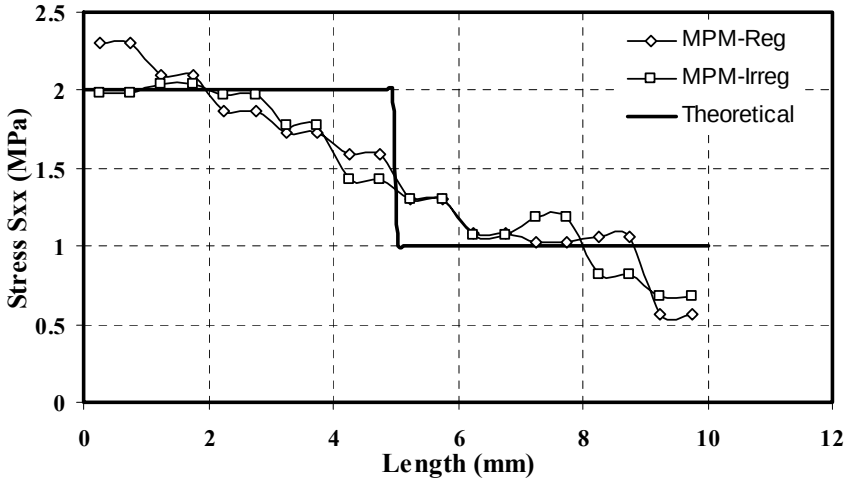


Figure 7.20 Stress along the length of the Model for Regular, Irregular MPM and Analytical Model at t = 1.5 L/C (4.8μs)

To resolve the large difference in the stress values the coarse mesh is refined as shown in fig. 7.21 and fig. 7.22. The dimensions and boundary conditions of the models are as shown in fig. 7.16. An equivalent load of 1N is applied on all the material points to the left end of the model. The irregular mesh shown in fig. 7.21 is created using ABAQUS and there are 6880 material points in the model. The regular mesh is created using the MPM code.

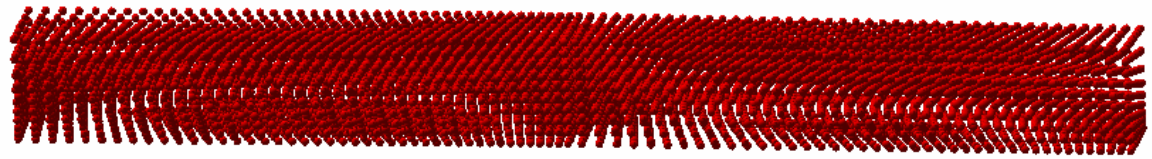


Fig 7.21 Irregular refined MPM mesh

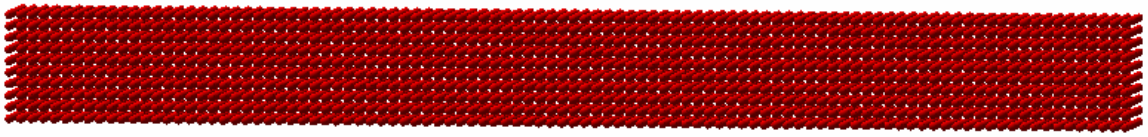


Fig 7.22 Regular refined MPM mesh

Fig. 7.23 shows the stress wave propagation at t = 0.5 L/C, i.e when the stress wave has reached the center of the model. The stress wave is plotted for regular mesh, irregular mesh and the analytical model. The plot shows that the regular and irregular mesh are good in agreement with the theoretical value when we refine the mesh.

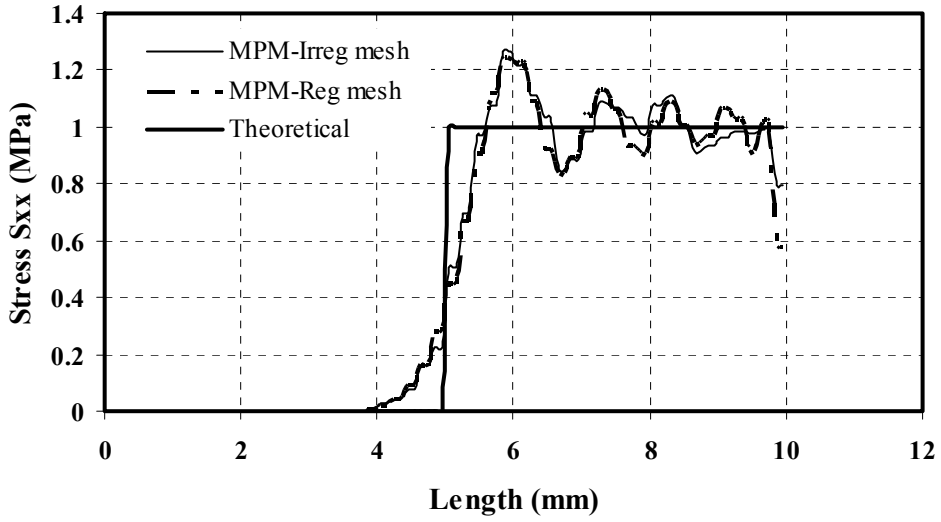


Fig 7.23 Stress along the length of the Refined Model for Regular, Irregular MPM and Analytical Model at t = 0.5 L/C (1.6μs)

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

1. A new 2D MPM algorithm with an irregular mesh has been developed to enable the use of arbitrary quadrilateral cells so that meshes can be generated to align an inclined crack with a set of nodes on MPM cells.
2. The introduction of arbitrary quadrilateral MPM cells enable cell refinement at high stress-gradient convenient to implement.
3. The new MPM algorithm can accommodate inclined crack issues and implement explicit fracture mechanics computations.
4. MPM simulations of tension model indicate that results from both regular and irregular mesh are consistent and agree with FEM using ABAQUS/Explicit.
5. For dynamic mixed mode crack fracture, a tensile model with an inclined crack has been simulated using this new MPM algorithm with adaptive mesh. This adaptive mesh can improve the accuracy of fracture parameter calculations.
6. From MPM simulations, the energy release rates were calculated and these results are in good agreement with those from FE analysis. Mode I and Mode II stress intensity factors were also calculated from MPM simulations.

8.2 Future Work

1. The new 2D MPM algorithm with irregular mesh should be extended to elastic-plastic and meso-plastic materials. After detailed verifications of the elastic-plastic algorithm in the conventional MPM code, it will be implemented in the new MPM algorithm and the validation would be done by simulating a tensile model with a regular mesh with the irregular mesh in the new MPM code and comparing with the FEM. The same method should be applied in meso-plastic case.
2. To simulate crack propagation in a elastic and elastic-plastic models. The crack – propagation can be implemented in the new MPM algorithm by introducing the critical shear stress, if it exceeds that value then the crack starts propagating.
3. The 3D MPM algorithm with irregular mesh should be extended to elastic-plastic and meso-plastic materials.
4. To simulate dislocations in the crystal, computational fracture mechanics should be used with irregular mesh MPM algorithm in elastic materials.
5. To investigate fracture behavior of cracks, computational fracture mechanics should be used with the new MPM algorithm for elasticity, elastic-plasticity and meso-plasticity.
6. To implement adaptive mesh in new MPM algorithm for simulating crack propagation problems.
7. To introduce contour integral method to compute energy release rate for an elastic-plastic models.

REFERENCES

1. Harlow, F.H., "The Particle-in-Cell Computing Method for Fluid Dynamics," Methods in Computational Physics, Editors Alder, B., Fernback,S., and Rotenberg., 3 (1964) 319-343

2. Sulsky, D., Chen, Z., and H. L. Schreyer, "A Particle Method for History-Dependent Materials," Comp. Meth. in Appl. Mech. and Eng.118 (1994) 179-196

3. Amsden,A.A., "The Particle-in-Cell Method for the Calculation of the Dynamics of Compressible Fluids," LA-3466, Los Alamos National Laboratory, Los Alamos, New Mexico. (1966)

4. Sulsky, D., Zhou, S-J., and H. L. Schreyer, "Application of a Particle-in-cell Method to Solid Mechanics," Computer Physics Communications 87 (1995) 236-252.

5. Sulsky, D., and H. L. Schreyer, "Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems," Comput. Meths. Appl. Mech. Eng., 139 (1996) 409-429.

6. Benson, D., "Computational Methods in Lagrangian and Eulerian hydrocodes," Comput. Meth. Appl. Mech.Eng., 99 (1992) 235-394

7. Belytschko, T., “An Overview of Semidiscretiztion and Time Integration Procedures,” Computational Methods for Transient Analysis, Vol I, T.Belytschko and T.J.R Hughes, ed.,Elsevire Science Publishers (1983)

8. Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P., “Meshless Methods:An Overview and Recent Developments,” Computer Methods in Apllied Mechanics and Engineering 139 (1996) 3-47

9. Shen, Y-L., Li, W., Sulsky, D. L., and H. L. Schreyer, “Localization of Plastic Deformation Along Grain Boundaries in a Hardening Material,” Int. J. of Mech. Sci. 42 (2000) 2167-2189

10. Tan H., and J. A. Nairn, “Hierarchical, Adaptive, Material Point Method for Dynamic Energy Release Rate Calculations,” Comput. Methods Appl. Mech. Eng. 191 (2002) 2095-2109

11. Nairn, J. A., “Material Point Method Calculations with Explicit Cracks,” Tech Science Press CMES, (2003).

12. Burgess, D., Sulsky, D., and Brackbill, J.U., “Mass Matrix Formulation of the FLIP Particle-in-cell Method,” Journal of Computational Physics 103 (1992) 1-15

13. Bardenhagen, S. G., “Energy Conservation error in the material point method for solid mechanics,” J. of Computational Physics 180 (2002) 383-403

14. Bardenhagen, S. G., Brackbill, J. U. and D. Sulsky, “The Material Point Method for Granular Materials,” Comput. Methods Appl. Mech. Eng. 187 (2000) 529-541

15. Sulsky, D., Ayton, G., Bardenhagen, G.S., McMurtry, P., Voth, G.A. “Interfacing Continuum and Molecular Dynamics: An Application to Lipid Bilayers,” Journal of Chemical Physics, 114 (15) (2001) 6913-6924

16. Cummins, S. J. and J.U. Brackbill, “An Implicit Particle-in-Cell Method for Granular Materials,” J. of Computational Physics 180 (2002) 506-548

17. Chen, Z., Hu, W., Shen, L., Xin, X., and R. Brannon, “An Evaluation of the MPM for Simulating Dynamic Failure with Damage Diffusion,” Engineering Fracture Mechanics 69 (2002) 1873-1890.

18. Herrmann, W., and Bertholf, L.D., “Explicit Lagrangian Finite-difference Methods,” Computational Methods for Transient Analysis, Vol. I, T.Belytschko and T.J.R Hughes, ed., Elsevier Science Publishers (1983)

19. Schreyer, H.L., Sulsky, D. L. and S.-J. Zhou, “Modeling delamination as a Strong Discontinuity with the Material Point Method,” Comput. Methods Appl. Mech. Eng. 191 (2002) 2483-2507.

20. Chen, Z. and R. Brannon, “An Evaluation of the Material Point method,” SAND Report, SAND2002-0482, (February 2002).

21. Hu, W. and Z. Chen, “A Multi-mesh MPM for Simulating the Meshing Process of Spur Gears,” Computers and Structures 81 (2003) 1991-2002.

22. York, A. R., Sulsky, D. L., and H. L. Schreyer, “The Material Point Method, for Simulation of Thin Membranes,” Int. J. Numer. Eng. 44 (1999) 1429-1456

23. Sulsky, D and A. Kaul, “Implicit Dynamics in the Material-Point Method,” Comp. Meths. Appl. Mechs. Engrg., 193 (2004) 1137-1170

24. Guilkey J. E., and J.A. Weiss, “Implicit Time Integration for the Material Point Method: Quantitative and Algorithmic Comparisons with the Finite Element Method,” Int. J. Numer. Meth. Eng.; 57 (9) (May 2003) 1323-1328.

25. Ma, C., and Chen, S. “Dynamic Stress Intensity Factor for Subsurface Inclined Cracks,” Journal of Engineering Mechanics - ASCE, 120 (3) (1994) 483–498

26. Buchholz, F., Chergui, A., and Richard, H., “Fracture analyses and experimental results of crack growth under general mixed mode loading conditions,” Engineering Fracture Mechanics, 71 (2004) 455-468.

27. Miranda, A., Meggiolaro, M., Castro, J., and Martha, L., “Fatigue life prediction of complex 2D components under mixed-mode variable amplitude loading,” International Journal of Fatigue, 25 (2003) 1157-1167.

28. Bittencourt, T., Wawrzynek, P., Ingraffea, A., and Sousa J., “Quasi-automatic simulation of crack propagation for 2D LEFM problems,” Engineering Fracture Mechanics 55 (1996) 321-334.

29. Liebowitz, H., and Moyer, E., “Finite Element Methods in Fracture Mechanics,” Comput. Struct., 31 (1) (1989) 1-9.

30. Rybicki, E., and Kanninen, M., “A Finite Element Calculation of Stress Intensity Factors by a Modified Crack Closure Integral,” Engineering Fracture Mechanics 9 (1977) 931-938.

31. Paris, P., and Sih, G., “Stress Analysis of Cracks, Fracture Toughness and Testing and its Applications,” STP381, Philadelphia: ASME (1965) 30-83.

32. Sulsky, D., and BrackBill, J.U., “A Numerical Method for Suspension Flow,” Journal of Computational Physics 96 (2) (1991) 339-368

33. Attaway, S.W., Heinstein, M.W., Mello, F.J., and Swegle, J.W. “Coupling of Smooth Particle Hydrodynamics with finite element method,” Nuclear Eng. Design, 150 (1994) 199--205

34. Liu, W.K., Jun, S., Li, S.F., Adee, J., Belytschko, T., “Reproducing Kernel Particle Methods for Structural Dynamics,” International Journal for Numerical Methods in Engineering 38 (10) (1995) 1655-1679

35. Babuska, I., Guo, B.Q., “Approximation Properties of the h-p Version of the Finite Element Method,” Comp. Meth. in Appl. Mechanics and Eng. 133 (3-4) (1996) 319-346

36. Oden, J.T., Duarte, C.A.M., Zienkiewicz, “A New Cloud-based h-p Finite Element Method,” Comp. Meth. in Appl. Mech. Eng. 153 (1-2) (1998) 117-126

37. Onate, E., Idelsohn, S., Zienkiewicz, O.C., Taylor, R.L., “A Finite Point Method in Computational Mechanics: Applications to Convective Transport and Fluid Flow,” International Journal for Numerical Methods in Engineering 39 (22) (1996) 3839-3866

38. Pipkins, D.S., Atluri, S.N., “Applications of the Three-dimensional Finite Element Alternating Method,” Finite Elements in Analysis and Design 23 (2-4) (1996) 133-153

39. Chen, J.S., Pan, C., Wu, C.T., Liu, W.K., “Reproducing Kernel Particle Methods for Large Deformation Analysis of Non-linear Structures”, Comp. Meth. in Appl. Mech. and Eng. 139 (1996) 195-228

40. Chen, J.S., Pan, C., Wu, Roque, C.M.O.L., Wang, H.P., “A Lagrangian Reproducing Kernel Particle Method for Metal Forming Analysis,” Comp. Mech. 22 (1998) 289-307

41. Geradin, M., Hogge, M., and Idelshon., “Implicit Finite Element Methods,” Computational Methods for Transient Analysis, Vol. I, T.Belytschko and T.J.R Hughes. ed., Elsevier Science Publishers (1983)

42. Brackbill, J.U., and Ruppel, H.M., “FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions,” Journal of Computational Physics, 65 (1986) 314-343

43. Brackbill, J.U., Kothe, D.B., Ruppel, H.M., “FLIP: A Low-Dissipation, Particle-in-Cell Method for Fluid Flow,” Computer Physics Communications, 48 (1988) 25-38

44. Peskin, C.S., “Numerical Analysis of Blood Flow in the Heart,” Journal of Computational Mechanics, (25) (1977) 220-252

45. Zhou, S.J., “The Numerical Prediction of Material Failure Based on the Material Point Method,” PhD thesis, Department of Mechanical Engineering, University of New Mexico, 1998.

46. Parker, S.G., “A Component based Architecture for Parallel Multi-Physics PDE Simulation,” (ICCS 2002) Workshop on PDE Software.

47. Bardenhagen, S.G., Kober, E.M., “The Generalized Interpolation Material Point Method,” in press (2003)

48. Belytschko, T., Lu, Y.Y., Gu, L. “Element Free Galerkin Methods,” Int. J. Numer. Methods in Eng., 39 (1996) 923-938

49. http://simscience.org/cracks/advanced/math1.html

50. Atluri, S.N., Shen, S. “The Meshless Local Petrov-Galerkin (MLPG) method: A Simple & Less Costly Alternative to the Finite Element and Boundary Element Methods,” Computer Modeling in Engineering & Sciences, 3 (2002) 11-52

51. Atluri, S.N., Shen, S.P. “The Meshless Local Petrov-Galerkin (MLPG) Method,” Tech.Science Press.(2002)

52. Atluri, S.N., Zhu, T. “A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics,” Computational Mechanics, 22 (1998) 117-127

53. Libersky, L.D., Petschek, A.G., Carney, A.G., Hipp, T.C., and Allahdadi, F.A. “ High Strain Lagrangian Hydrodynamics – A Three Dimensional SPH Code for Dynamic Material Response,” Journal of Computer Physics 109 (1993) 67 – 75

54. Brackbill, J.U. “The Ringing Instability in Particle-in-Cell Calculations of Low-Speed Flow,” Journal of Computational Physics 75 (2) (1998) 469-492

55. Haines, E. “Graphics Gems IV: Point in Polygon Strategies,” AP Professional, Boston, (1994) 24-46

56. http://mathworld.wolfram.com/NewtonsMethod.html

57. Reddy, J.N. “An Introduction to the Finite Element Method,” 2nd Edition, McGraw-Hill series in Mechanical Engineering (1993).

58. Wang, B., Karuppiah, V., Lu, H.B., Roy, S., Komanduri, R. “2D Mixed Mode Crack Opening Simulation Using Material Point Method (MPM),” Submitted for publication (2004).

VITA

Venkatesh Karuppiah

Candidate for the Degree of

Master of Science

Thesis: IMPLEMENTATION OF IRREGULAR MESH IN MATERIAL POINT METHOD (MPM) FOR SIMULATION OF MIXED MODE CRACK OPENING IN TENSION

Major Field: Mechanical Engineering

Biographical:

Education: Received Bachelor of Engineering degree in Mechanical Engineering from Annamalai University, Tamilnadu, India in May 2000. Completed the requirements for the Master of Science degree with a major in Mechanical and Aerospace Engineering at Oklahoma State University in May, 2004.

Experience: Graduate Research Assistant in Mechanical and Aerospace Engineering Department, Oklahoma State University, Stillwater, Oklahoma, January 2003 -present.

Graduate Teaching Assistant in Mechanical and Aerospace Engineering Department, Oklahoma State University, Stillwater, Oklahoma, January 2002 to August 2003.

Professional Membership: ASME.

Name: Venkatesh Karuppiah Date of Degree: May, 2004

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: IMPLEMENTATION OF IRREGULAR MESH IN MATERIAL POINT METHOD (MPM) FOR SIMULATION OF MIXED MODE CRACK OPENING IN TENSION

Pages in Study: 87 Candidate for the Degree of Master of Science

Major Field: Mechanical Engineering

Scope and Methodology of Study: 2D conventional MPM uses a regular grid with uniform square cells resulting in limitation in dealing with inclined cracks and dislocations. A new algorithm with irregular mesh has been developed for Material Point Method to allow the use of arbitrary quadrilateral cells in the background grid mesh. This new algorithm has been applied to solve mixed mode crack problems in computational fracture mechanics. Fracture parameters have also been investigated from both MPM and FEM simulations.

Findings and Conclusions: A new algorithm is developed to enable MPM to use arbitrary quadrilateral cells in the background grid mesh. Ray crossing algorithm is employed to determine which cell in the background grid mesh a particle belongs after the deformation. Validation was first made with a tension problem and compared with the results obtained from ABAQUS/explicit code. An inclined crack problem is solved as an example to demonstrate the capability of the arbitrary quadrilateral cells in the new algorithm. The same problem is also solved using ABAQUS to validate the new algorithm. For dynamic mixed mode crack fracture, the energy release rates were calculated from MPM. These results are in good agreement with FEM. Mode I and Mode II stress intensity factors were also calculated from MPM simulations.

ADVISOR'S APPROVAL: Dr. Ranga Komanduri