SCI INSTITUTE
TECHNICAL REPORT

Solving Time-Dependent PDEs using the Material Point Method, A Case Study from Gas Dynamics

L.T. Tran, J. Kim, M. Berzins

UUSCI-2007-010

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

July 1, 2007

Abstract:

The Material Point Method (MPM) developed by Sulsky and colleagues is currently being used to solve many challenging problems involving large deformations and/or fragmentations with some success. In order to understand the properties of this method, an analysis of the considerable computational properties of MPM is undertaken in the context of model problems from gas dynamics. The MPM method in the form used here is shown both theoretically and computationally to have first order accuracy for a standard gas dynamics test problem. Copyright ©2008 John Wiley & Sons, Ltd.

THE UNIVERSITY OF UTAH

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **00**:1–23 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

Solving Time-Dependent PDEs using the Material Point Method, A Case Study from Gas Dynamics.

† L.T.Tran [1,*], J.Kim [1] M. Berzins[1,2]

[1] *School of Computing, University of Utah, Salt Lake City, Utah* [2] *SCI Institute, University of Utah, USA.*

SUMMARY

The Material Point Method (MPM) developed by Sulsky and colleagues is currently being used to solve many challenging problems involving large deformations and/or fragementations with some success. In order to understand the properties of this method, an analysis of the considerable computational properties of MPM is undertaken in the context of model problems from gas dynamics. The MPM method in the form used here is shown both theoretically and computationally to have first order accuracy for a standard gas dynamics test problem. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: MPM Particle method, Error Estimates

1. INTRODUCTION

The need to solve problems involving large deformations in materials has led to the development of a number of new computational methods. Examples of such methods are meshfree and particle methods, for example as surveyed by [16], one of which is the relatively new Material Point Method of Sulsky et al., [26, 27], which may, perhaps, be described as a quasi-meshless method. This method (MPM) has evolved from the particle-in-cell(PIC) and FLIP methods [5] originally developed by Brackbill et al. see [3] and the references within. An interesting discussion of some of these methods and important theoretical results are given by Grigoryev et al. [10]. Two important features of MPM are the use of a grid as a scratchpad for calculations, hence the quasi-meshless characterization and the capability to model solid materials undergoing large deformation. An important aspect of the MPM method is that it has not yet been subjected to as much analysis as many of the methods surveyed by [16].

There has been considerable analysis of PIC type methods. One of fundamental aspects of PIC methods is a discretization of a material into particles, and the interpolation of information from particles to grids and vice-versa. In MPM, Lagrangian particles (or points) are used to discretize the volume of the fluid or solid. These material points carry with them properties such as mass, velocity, stress, strain and so on. The grid may be viewed as a temporary

*Correspondence to: mb@cs.utah.edu

Copyright © 2008 John Wiley & Sons, Ltd.

computational scratch pad, which can be reconnected at any time when a mesh distortion makes further calculation more difficult. Material response is governed by continuum mechanics constitutive models, which generates stress based on both the history and current mechanical states, [26]. The Generalized Interpolation Material Point method(GIMP), [2] provides a general formulation covering MPM methods. MPM has been used for applications such as the biomechanics of micro-vessels, the effects of wounding on heart tissue and the properties of foam under large deformation [7]. The method has also been used extensively in large-scale complex fluid-structure interactions, [13, 21], arising from the modeling of safety studies involving explosions. Given the use of the method on such important and challenging problems it is important to understand how accurate the method is.

In this paper, an analysis of the MPM procedure is considered in the context of a shock propagation problem, using a modified form of the method developed by [14]. This problem has also been studied by Brackbill [8], Sulsky [26], York et al.[30] and very recently in the context of SPH methods by Brown et al. [6]. A comparison betweem MPM and the SPH method has been undertaken by [18]. Although MPM is originally designed for solid mechanics problems, this test problem has the advantage of being sufficiently simple and well-understood to allow analysis of the method. Furthermore the problem's analytical solution can be used to evaluate the various sources of error in the MPM method. This paper describes the accuracy and stability properties of the MPM method in a way that also allows these properties to be extended to other more general situations. A particular focus of the paper is an analysis of different methods used to project information from particles onto the grid. The errors introduced when particles cross grid cells are also studied in some depth. The paper is complementary to other recent studies of the method [24, 25, 28, 29].

2. Problem Description

The model problem used here is that of Sod [23] who used a simple gas dynamics problem to investigate finite difference schemes for shock propagation type problems. This problem has an analytical solution and may be used to compare the result of MPM to the analytical solution. The same problem has often been used as a test problem for PIC and MPM methods, [30].

Sod's gas dynamics problem consists of a shock tube, where a diaphragm is located in the middle of the tube. Two sides of the diaphragm have different pressure and density, which make the fluid flows when the diaphragm is broken. The left side of density is 1 and pressure is also 1. The right side of density is 0.125 and pressure is 0.1, and the initial velocities of both regions are zero. At time t=0, the diaphragm is removed the motion of the compressible and inviscid fluid is governed by Euler's equations, which are,

$$\begin{aligned} \tfrac{\partial \rho}{\partial t} + \tfrac{\partial \rho v}{\partial x} &= 0 \\ \tfrac{\partial \rho v}{\partial t} + \tfrac{\partial (\rho v^2 + p)}{\partial x} &= 0 \\ \tfrac{\partial \rho e}{\partial t} + \tfrac{\partial v(\rho e + p)}{\partial x} &= 0 \end{aligned} \tag{1}$$

where e: total energy per unit volume, p: pressure, v: unit velocity, ρ: fluid density and $(x,t)\epsilon(0,1)\times(0,0.1)$ The state equation for pressure is

$$p = (\gamma - 1)\left(\rho e - \frac{\rho v^2}{2}\right) \tag{2}$$

Copyright © 2008 John Wiley & Sons, Ltd.

where $\gamma = 1.4$ denotes an ratio of specific heat for dry air as a perfect gas. These equations may be written e.g. using equations 14.45-14.47 of [15] in the form given by Sulsky et al [26] as:

$$\frac{\partial \epsilon}{\partial t} + v\frac{\partial \epsilon}{\partial x} + \frac{p}{\rho}\frac{\partial v}{\partial x} = 0 \tag{3}$$

$$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial x} + \rho\frac{\partial v}{\partial x} = 0 \tag{4}$$

where ϵ denotes internal energy. The state equation for pressure is then given by:

$$p = (\gamma - 1)\rho\epsilon. \tag{5}$$

The boundary conditions are those commonly used, [23].

3. MPM Spatial Discretization

3.1. Particle Basis Functions

The original form of the MPM method uses Delta functions for the basis functions associated with the particles:

$$\chi_p(x) = \delta(x - x_p)V_p, \quad p = 1, ..., np \tag{6}$$

where $x_p(t)$ are particle positions which are functions of time t and V_p is a particle volume that is discussed below. Bardenhagen and Kober [2] use the piecewise constant form instead:

$$\chi_p(x) = \begin{cases} 1 & if\ x\ \epsilon\Omega_p, \\ 0 & otherwise \end{cases} \tag{7}$$

where Ω_p is the interval $[x_p - h_p/2, x_p + h_p/2]$ with h_p is the particle width. This has the advantage that the functions form a partition of unity on the interval $[a, b]$:

$$\sum_{p=1}^{np} \chi_p(x) = 1 \ \ \forall\ x \in [a, b]. \tag{8}$$

For both choices of basis functions, the approximation to the function $f(x)$ in terms of particle values is then written as

$$f(x) \approx \sum_p f_p\chi_p(x) \ \ \forall\ x \in [a, b]. \tag{9}$$

The particle volumes are then defined by

$$V_p = \int_{\Omega^i} \chi_p(x)dx \tag{10}$$

where Ω^i is the domain of cell i that contains the particle p. In the case when $\chi_p(x)$ is defined as by equation (6) the "volume" of a particle will be defined in Section 5 below.

Copyright © 2008 John Wiley & Sons, Ltd.

3.2. Grid basis Functions.

The continuous representation of a function $g(x)$ using grid data g_i on a grid

$$a = x_0 < x_1 < ... < x_N = b \tag{11}$$

and where $I_i = [x_{i-1}, x_i]$ and $I_{i+1} = [x_i, x_{i+1}]$ and $h_i = x_i - x_{i-1}$; is given by

$$g(x) = \sum_{i=1}^{n_v} g_i S_i(x) \tag{12}$$

where $S_i(x)$ is the piecewise linear basis function with value one at node x_i in the mesh and value zero at all other nodes; often these points are equidistant with an uniform mesh spacing of h.

3.3. Mapping from Particles to Grid.

The mapping from particle values to values at grid points is defined by the convolution of the linear basis functions or their gradients with the particle basis functions as follows. Let

$$\bar{S}_{ip} = \frac{1}{V_p} \int_{\Omega^i} S_i(x) \chi_p(x) dx \tag{13}$$

and

$$\bar{G}_{ip} = \frac{1}{V_p} \int_{\Omega^i} \frac{dS_i}{dx}(x) \chi_p(x) dx. \tag{14}$$

In the case of the standard MPM case when delta functions are used for the particles and linear basis functions are used for the grid, then, [2],

$$\bar{S}_{ip} = S_i(x_p) \tag{15}$$

and

$$\bar{G}_{ip} = \frac{dS_i}{dx}(x_p). \tag{16}$$

The mapping from particle values to values at grid points is then defined by

$$f(x_i) = \sum_{p=1}^{np} f(x_p)\ \bar{S}_{ip} \tag{17}$$

4. Computational Method

Given an initial distribution of particles on the domain, the point masses, m_p, are defined in terms of density by:

$$m_p = \int_{\Omega^i} \rho(x) \chi_p(x) dx. \tag{18}$$

A point density average, ρ_p, may also be defined by

$$\rho_p = m_p / V_p. \tag{19}$$

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using fldauth.cls

Particle momentum values, P_p,are given by

$$P_p = \int_{\Omega^i} \rho(x)v(x)\chi_p(x)dx, \tag{20}$$

where $\rho(x)$ is the continuum body's mass density and $v(x)$ is the velocity.
The Cauchy stresses are:

$$\sigma_p = \int_{\Omega^i} \sigma(x)\frac{\chi_p(x)}{V_p(x)}dx \tag{21}$$

where $\sigma(x)$ is continuum bodies initial Cauchy stress. In the most general case, the stress tensor is given by $\sigma = -pI + T$, where p is the pressure, T denotes the viscous stress tensor and I is an identity tensor whose size is same as the modeling dimension. In a perfect fluid model such as the gas dynamics problem considered here, the stress at a particle is equal to the pressure:

$$\sigma_p = -p_p. \tag{22}$$

4.1. Mesh and Particle Movement per Timestep

This subsection is an abbreviated form of the description of the MPM method in [25]. At the start of a timestep, the mass at each grid point, m_i, is calculated from the masses of the particles, by using the lumped mass matrix form of MPM, [26].

$$m_i = \sum_{p=1}^{np} \bar{S}_{ip} m_p, \quad i = 1, ..., nv. \tag{23}$$

Momentum at a grid node, P_i, is given by

$$P_i = \sum_{p=1}^{n_p} \bar{S}_{ip} m_p v_p, \quad i = 1, ..., nv. \tag{24}$$

The nodal velocity, v_i, is calculated from the mass and the momentum of the node:

$$v_i = \frac{P_i}{m_i}. \tag{25}$$

The force at each node, F_i^{int},is given by:

$$F_i^{int} = \sum_{p=1}^{np} p_p \bar{G}_{ip} V_p \tag{26}$$

where p_p is the particle pressure. The acceleration at a node, a_i, is calculated from the force and the mass at the node:

$$a_i = \frac{F_i^{int}}{m_i} \ . \tag{27}$$

The nodal velocity at the end of Lagrangian step is calculated using Euler's method:

$$v_i^{n+1} = v_i^n + a_i^n dt \tag{28}$$

Copyright © 2008 John Wiley & Sons, Ltd.

where a_i^n is the acceleration at time t_n. The particle velocity and location are updated using these new values:

$$v_p^{n+1} = v_p^n + \sum_{i=1}^{n_v} \bar{S}_{ip} a_i^n dt, \tag{29}$$

$$x_p^{n+1} = x_p^n + \sum_{i=1}^{n_v} \bar{S}_{ip} v_i^{n+1} dt. \tag{30}$$

Remark If v_p^{n+1} was used to replace the sum in the right side of equation (30), the time integration method could be viewed as a first order Runge-Kutta-Nystrom method, [9].

5. Application to Gas Dynamics

At the start of a timestep, the approximate particle volume for particle p can be calculated from the width of the cell it lies in, h_j, and the number of particles in that cell, N_p^j, by:

$$V_p = \frac{h_j}{N_p^j}. \tag{31}$$

while this is a reasonable approximation for compressible flow, and was first used by [14], it represents a departure from the standard MPM approach for solid mechanics, in which the volumes associated with particles are tracked, see [24] for an analysis of this case. The particle's mass is calculated from the density and the volume of the particle as

$$m_p = \rho_p V_p. \tag{32}$$

The mass at each grid point is calculated from the projection of the particle properties as in (23) and momentum at a grid node is given by equation (24). The nodal velocity is calculated from the mass and the momentum of the node as given in (25). The force at each node may be written as the jump on the averaged particle pressures:

$$F_i^{int} = p_{p,i}^- - p_{p,i}^+ \tag{33}$$

where

$$p_{p,i}^- = \sum_{p:x_p \in I_i} p_p \frac{1}{N_p^i} \tag{34}$$

$$p_{p,i}^+ = \sum_{p:x_p \in I_{i+1}} p_p \frac{1}{N_p^{i+1}}. \tag{35}$$

The internal force at a node is thus equal to the averaged pressure drop around that node. The acceleration at a node is calculated from the force and the mass at the node.

$$a_i = \frac{p_{p,i}^- - p_{p,i}^+}{m_i}. \tag{36}$$

The particle velocity and location are updated using these values by equations (28-30). This method of force calculation has been developed here as being more appropriate for compressible gas dynamics as it assumes that the particles within a cell have the same volume. Analysis and investigation of alternative methods for solid mechanics and different approaches to gas dynamics are provided by [10, 24, 30].

Copyright © 2008 John Wiley & Sons, Ltd.

5.1. Particle Energy, Density and Pressure Update

Once the nodal velocities are known as in equation (29), it is possible to update the velocity gradient and hence calculate the energy and a density of the particles at the next time step, as denoted by $\epsilon_p^{n+1}, \rho_p^{n+1}$ by:

$$\epsilon_p^{n+1} = \epsilon_p^n - \frac{p_p^n}{\rho_p^n}\frac{\partial v_p^{n+1}}{\partial x}dt, \tag{37}$$

and

$$\rho_p^{n+1} = \rho_p^n(1 - \frac{\partial v_p^{n+1}}{\partial x}dt), \tag{38}$$

where the velocity gradient of each particle is calculated using nodal velocities and the gradients of the nodal basis functions by using:

$$\frac{\partial v_p^{n+1}}{\partial x} = \sum_{i=1}^{nv} \bar{G}_{ip} v_i^{n+1}$$

where $\bar{G}_{ip}$ is defined by equation (14). The pressure update is given by:

$$p_p^{n+1} = (\gamma - 1)\rho_p^{n+1}\epsilon_p^{n+1} + a_v. \tag{39}$$

The term a_v is a standard artificial viscosity term which is defined by

$$a_v^n = \begin{cases} C^2 dx^2 \rho(\frac{\partial v_p}{\partial x})^2 & \text{if } \frac{\partial v_p}{\partial x} \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

where $C = 2.5$. This form of artificial viscosity was used by Monaghan and Gingold [19, 20] to reduce oscillations in the numerical solution SPH methods. This formula exploits the property of shock front that the gradient of velocity is less than zero, there. Using the pressure equation (5) to substitute for the pressure/density ratio in the energy equation gives:

$$\epsilon_p^{n+1} = \epsilon_p^n(1 - (\gamma - 1)\frac{\partial v_p^{n+1}}{\partial x}dt) - \frac{a_v}{\rho_p^n}\frac{\partial v_p^{n+1}}{\partial x}dt. \tag{40}$$

In the same way the pressure equation (5) may itself be rewritten as:

$$p_p^{n+1} = \left[(p_p^n - a_v^{n-1})(1 - \frac{\partial v_p}{\partial x}dt)) + a_v^n\right](1 - (\gamma - 1)\frac{\partial v_p^{n+1}}{\partial x}dt). \tag{41}$$

5.2. Positivity, Overshoots and Stability

As density, energy and pressure values are positive, their numerical approximations should also be positive. From equations (37)-(38) it may be seen that this occurs for the discrete density and energy equations under a Courant-like condition:

$$0 \leq \frac{\partial v_p^{n+1}}{\partial x}dt \leq 1. \tag{42}$$

Although this ensures that values of density and energy remain positive; local extrema may be caused by use of the velocity gradient from "old"cell when cell crossing occurs. Suppose

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using fldauth.cls

that there are two adjacent particles in different mesh intervals with densities, ρ_p and ρ_{p+1}. Suppose further that

$$\rho_p(t_n) < \rho_{p+1}(t_n), \tag{43}$$

and that the velocity gradients differ in adjacent intervals so that:

$$(1 - dt\frac{\partial v_p^{n+1}(x_p)}{\partial x}) >> (1 - dt\frac{\partial v_p^{n+1}(x_{p+1})}{\partial x}), \tag{44}$$

then it is possible that one particle will over take the other in magnitude:

$$\rho_p(t_{n+1}) > \rho_{p+1}(t_{n+1}), \tag{45}$$

this may result in a new extremal value. A similar argument may be developed for the creation of new extrema in energy. In order to prevent this further artificial diffusion is applied in the case when extrema occur in velocity $(v_{i-1} - v_i)(v_{i+1} - v_i) > 0$. The new value of velocity is then calculated by the addition of an artificial viscosity like that approximates to $\frac{\partial x^2}{3}\frac{\partial^2 v}{\partial x^2}$ gives:

$$v_i = v_i + \frac{v_{i-1} - 2v_i + v_{i+1}}{3} \tag{46}$$

and the same approach is applied if extrema are detected in density.

5.3. Particle Redistribution

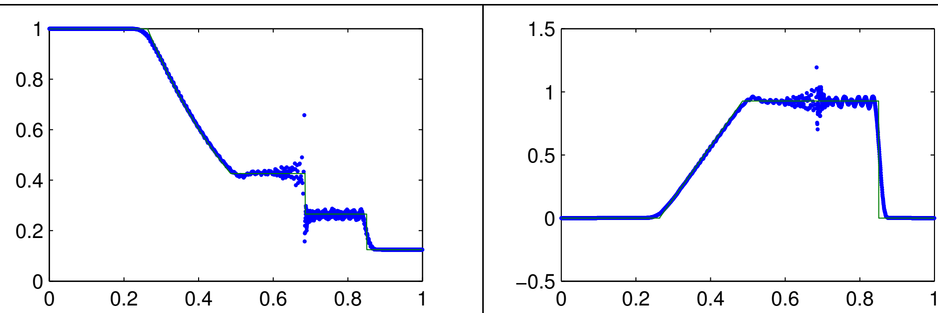
Once particles move to an adjacent cell, the changed number of particles in a cell is used to calculate new particle volume and mass after the density calculation is completed. If there were too few particles per cell and some of these particles move from one cell to another cell, it is possible for a cell not to have any particles. This may cause stability problems. To prevent this situation, care must be taken in the initial assignment of particles, see Section (6.1). The main idea is to ensure that there are always sufficient number of particles per cell. This may be obtained by redistribute of particles or by ensuring that particles are placed where they will move into cells with less particles. It may also be necessary to create new particles in the empty cells with the particles' properties obtained by interpolating the particles' properties in the adjacent cells.

6. Gas Dynamics Computational Experiments

6.1. Initial Uniform Particle Distribution

In these experiments the spatial mesh is fixed, and as particles can move from one cell to another cell, the number of particles in a cell varies, and so does their volume according to equation (31). Since we assume each material point is part of a perfect compressible gas, changing the particle's volume is a reasonable modeling assumption. Initially, same number of particles per a cell is used. Figures 1a and 1b below are the results after 0.2 seconds. The initial number of particles in a cell is 8, the cell size is 0.005, and time step is 0.00025. Each dot represents a material point and a solid line is the analytical solution.

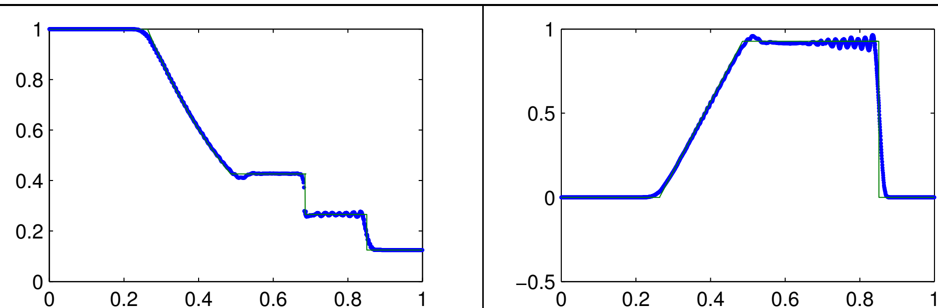
Copyright © 2008 John Wiley & Sons, Ltd.



[Fig. 1a] Density Error
L1 Norm: 0.00640
L2 Norm: 0.01524

[Fig. 1b] Velocity Error
L1 Norm: 0.01847
L2 Norm: 0.05800

Figures 1a and 1b show large errors behind the shock front. The smoothing process described in Section (5.2) was applied to density and velocity as a remedy for this. Figures 2a and 2b show the solution after the smoothing process was applied. The error norm after the smoothing process is about 67 to 90% of that when smoothing is not applied.



[Fig. 2a] Density Error
L1 Norm: 0.00434
L2 Norm: 0.01046

[Fig. 2b] Velocity Error
L1 Norm: 0.01471
L2 Norm: 0.05070

In investigating the relationship between the error and cell size, number of particles, and time interval, a smaller cell size generated more accurate results. When number of particles is too small (1, 2 or 3), the computation was inaccurate or unstable no matter how small the cell size was. When the number of particles in a cell was between 4 to 8 and a sufficiently small cell size was used the best results were obtained. Also it is interesting to see that the smaller cell size doesn't reduce the need for a certain number of particles in a cell in order for a stable and accurate result to be obtained. Smaller time-steps generated slightly better results at the cost of increased calculation times. The conclusion thus is that between 4 and 8 particles should be used in this ... with this method. Section (6.2) below will show that this may be modified based on the difference of density in various regions.

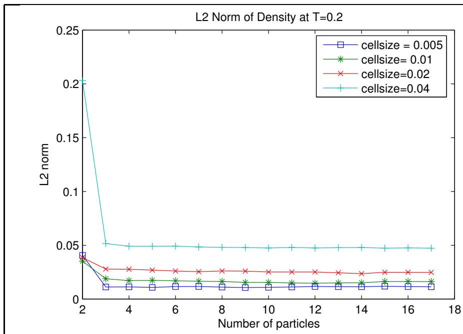
Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using *fldauth.cls*

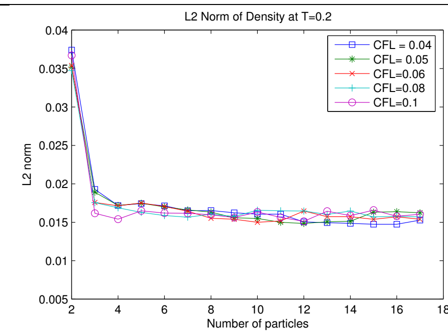
Table I. Values of Stable Time Step-size

Cell Size (dX)	0.005		0.01		0.015	
	N.S.*	S*	N.S.	S	N.S.	S
Max stable time step (dt)	0.00057	0.0006	0.00114	0.00124	0.00171	0.00185
Max stable CFL(dT/dX)	0.114	0.124	0.114	0.124	0.114	0.123

*(N.S.: Non Smoothing Process, S: Smoothing Process Applied)



[Fig. 3a] Error vs Cell Size



[Fig. 3b] Error vs CFL Number

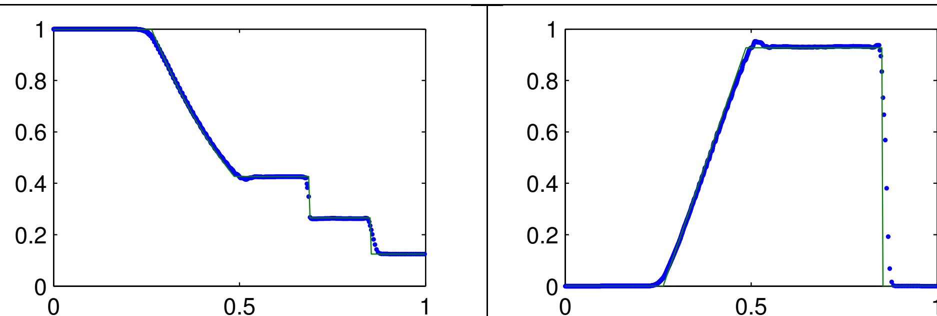
Fig 3a shows the final L2 error norms with a fixed time interval but different cell sizes and shows that the L2 norm is decreasing as the cell size decreases. The figure also shows that the number of particles is not the main factor for a smaller error norm if the number of particles is not too small. Fig 3b shows that for the same cell size but different time steps the error did not change much for CFL numbers below 0.1. Fig 3b shows slightly increasing errors as the time interval decreases, perhaps due to error buildup over the larger number of steps, but overall the spatial error dominates the temporal error. In order to investigate stability and choice of CFL number, two test cases were used. First of all, cell size and number of particles per cell were fixed, and time interval is changed. When the cell size was 0.005, the modeling system was unstable if the time interval was bigger than 0.00057. The meaning of "unstable" is that the particle's velocity was so large that the particle left the spatial domain. Table I show that the method generates stable results only if the CFL number is smaller than about $0.11 \sim 0.12$, and the smoothing process allows a slightly larger CFL number than the non-smoothing process does.

6.2. Alternative Particle Distribution

Although the smoothing process reduced much of instability of the particles, there are still remaining spurious oscillations in the solution. Brackbill [4] showed that the ringing instability in the PIC method was reduced with smaller number of particles, see Section (8) below. This result suggests using a smaller number of particles. However using ($2 \sim 3$ particles) increases error and one particle in a cell may generate unstable results. This problem is overcome by noting that, based on the given initial condition, the gas to the right of the diaphragm has a lower density. Hence particles are assigned in proportion to the relative density of the

Copyright © 2008 John Wiley & Sons, Ltd.

gasses. Since the gas on the left side's has normalized density 1 and that on the the right has density 0.125, eight particles per a cell are assigned to the left and one particle to the right. Interestingly, this particle distribution gives a stable result although the number of particles on the right side is one per cell. During the time integration process, the left particles move rightwards. As there are enough number of particles on the left side and these particles move to the right where there is only one particle per cell, the solution process remains stable as we are constantly introducing particles into the cells on the right. Figures 4a and 4b show the results from using fewer particles on the righthand side of the diaphram. The Smoothing process of Section (5.2) was also applied. Comparing these images to Fig. 2a and Fig 2b, this approach results in fewer oscillations, but has a similar error norm to the previous cases. A generalization of this approach is to equally distribute particles with respect to density.



[Fig. 4a] Density
L1 Norm: 0.005389
L2 Norm: 0.013781

[Fig. 4b] Velocity
L1 Norm: 0.014942
L2 Norm: 0.056484

7. Time Integration Error and Grid Crossing by Particles

7.1. Time Integration Discontinuities Arising from Grid Crossing

The comparative lack of smoothness of the spatial basis grid functions used in the MPM translates into a lack of smoothness in time when particles cross grid points and then have properties that are redefined in terms of the basis functions in the next interval. The definition of particle velocity updates in terms of nodal velocity values means that the higher time derivatives of the particle velocity are discontinuous when a particle crosses a grid point. This may be illustrated by considering equation (28) which is a forward Euler discretization of

$$\dot{v}_p = \sum_i S_i(x_p)a_i. \tag{47}$$

If the point $x_p(t)$ is in the interval $[x_{i-1}, x_i]$ then this equation may be written as

$$\dot{v}_p = \alpha_i a_{i-1} + (1-\alpha_i)a_i, \ \ \alpha_i = \frac{x_p(t) - x_i}{x_{i-1} - x_i}, \tag{48}$$

whereas if the point $x_p(t)$ is in the interval $[x_i, x_{i+1}]$ then this equation may be written as

$$\dot{v}_p = \alpha_{i+1} a_i + (1-\alpha_{i+1})a_{i+1}, \ \ \alpha_{i+1} = \frac{x_p(t) - x_{i+1}}{x_i - x_{i+1}}. \tag{49}$$

Copyright © 2008 John Wiley & Sons, Ltd.

The second derivative of v_p when the point $x_p(t)$ is in the interval $[x_{i-1}, x_i]$ is given by

$$\ddot{v}_p = \alpha_i \dot{a}_{i-1} + (1-\alpha_i)\dot{a}_i + \dot{x}_p \frac{a_{i-1} - a_i}{x_{i-1} - x_i} \tag{50}$$

or if the point $x_p(t)$ is in the interval $[x_i, x_{i+1}]$ then

$$\ddot{v}_p = \alpha_{i+1} \dot{a}_i + (1-\alpha_{i+1})\dot{a}_{i+1} + \dot{x}_p \frac{a_i - a_{i+1}}{x_i - x_{i+1}}. \tag{51}$$

The jump in the second derivative of particle velocity as the particle crosses the point x_i is given by

$$[\ddot{v}_p^+ - \ddot{v}_p^-]_{x_i} = \dot{x}_p \left[\frac{a_i - a_{i+1}}{x_i - x_{i+1}} - \frac{a_{i-1} - a_i}{x_{i-1} - x_i} \right]. \tag{52}$$

The local error associated with one step of the forward Euler method applied to equation (29) is given by

$$le = \frac{dt_1^2}{2} \ddot{v}_p. \tag{53}$$

This formula does not apply if $\ddot{v}_p$ is discontinuous with "left"and "right" values denoted by $\ddot{v}_p^-$ and $\ddot{v}_p^+$ respectively. One standard ODE method for crossing a discontinuity is to march up to it with one step of size dt_1 and one step from it of size dt_2. The local error for an Euler time-step in region one may be estimated by

$$le_1 \approx \frac{dt_1^2}{2} \ddot{v}_p^-, \tag{54}$$

and the local error for an Euler time-step in region two is estimated by

$$le_2 \approx \frac{dt_2^2}{2} \ddot{v}_p^+ \tag{55}$$

by assuming that the second derivatives may be regarded as constant on a step. It may be shown by using techniques such as those used by Shampine [22], that the error introduced over one time-step, denoted here by e_p^{p+1}, that crosses the discontinuity is then the sum of these local errors and the difference between the one and two step solutions, i.e.

$$e_p^{p+1} = le_1 + le_2 + (\bar{v}_p^{n+1} - v_p^{n+1}) \tag{56}$$

where v_p^{n+1} is the solution computed using one Euler step of size dt and where $\bar{v}_p^{n+1}$ is the solution computed using two Euler steps of size dt_1 and dt_2. The next two sub-sections will show that the gap between the two Euler solutions $(\bar{v}_p^{n+1} - v_p^{n+1})$ is one power of dt less than the local errors for both velocity and position errors.

7.2. Time Integration Errors in Velocity

Having determined the nature of the discontinuity it now remains to determine the error introduced by stepping over it. In both these cases the discontinuity in the first time derivative of the right hand side of equation (43)(after noting that x_p is time dependent) means that the time integration method accuracy is restricted to first order unless special action is taken, [1], p.64. It is worth noting that with a standard p.d.e method discontinuities in time derivatives

Copyright © 2008 John Wiley & Sons, Ltd.

do not occur in the same way as when material point method particles cross cells. In the case when a particle x_p lies in I_i and passes over a mesh cell then

$$v_p^{n+1} = v_p^n + \left[a_{i-1} + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(a_i - a_{i-1})\right] dt. \tag{57}$$

Alternatively, the forward Euler method may be applied to march up to the edge of the cell in one step and then take another step to arrive at the same point. For the first sub step of length dt_1 the velocity is

$$\bar{v}_i = v_p^n + \left[a_{i-1} + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(a_i - a_{i-1})\right] dt_1. \tag{58}$$

For the second sub step

$$\bar{v}_p^{n+1} = \bar{v}_i + [a_i + dt_1 \dot{a}_i]\, dt_2 \tag{59}$$

where $dt = dt_1 + dt_2$. Hence the difference in the velocities calculated using the two approaches is given by:

$$\bar{v}_p^{n+1} - v_p^{n+1} = (a_i - a_{i-1}) \left[\frac{x_i - x_p^n}{x_i - x_{i-1}}\right] dt_2 + dt_1 dt_2 \dot{a}_i, \tag{60}$$

and so may be written as

$$\bar{v}_p^{n+1} - v_p^{n+1} \approx C dt_2 (a_i - a_{i-1}) + h.o.t \tag{61}$$

where $C = \left[\frac{x_i - x_p^n}{x_i - x_{i-1}}\right]$ and where $0 \leq C \leq 1$. For the Euler equations considered here the values of $(a_i - a_{i-1})$ may be as large as 10^3. This dictates the use of a time-step of the order of that used in Section (6). †

7.3. *Time Integration Errors in Spatial Position*

Having determined the nature of the discontinuity it now remains to determine the error introduced by stepping over it. In both these cases the discontinuity in the higher derivative means that the time integration method accuracy is again restricted to first order. In the case when a particle x_p lies in I_i and passes over a mesh cell then

$$x_p^{n+1} = x_p^n + \left[v_{i-1}^{n+1} + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(v_i^{n+1} - v_{i-1}^{n+1})\right] dt \tag{62}$$

which may be written as

$$x_p^{n+1} = x_p^n + \left[v_{i-1}^n + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(v_i^n - v_{i-1}^n)\right] dt + \left[a_{i-1}^n + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(a_i^n - a_{i-1}^n)\right] dt^2. \tag{63}$$

As stated above, consider using the forward Euler method to march up to the edge of the cell in one step and then in another step to step to the same time point. For the first step

$$x_i = x_p^n + \left[v_{i-1}^n + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(v_i^n - v_{i-1}^n)\right] dt_1 + \left[a_{i-1}^n + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}}(a_i^n - a_{i-1}^n)\right] dt_1^2. \tag{64}$$

†The reader should note that throughout C will be used as a generic constant whose value may be different each time it is used.

Copyright © 2008 John Wiley & Sons, Ltd.

For the second step

$$\bar{x}_p^{n+1} = x_i + v_i^{n+1} dt_2 \tag{65}$$

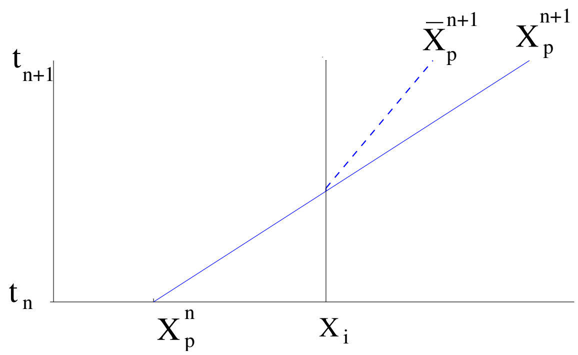
and so

$$\bar{x}_p^{n+1} = x_i + v_i^n dt_2 + a_i^n dt\ dt_2 \tag{66}$$

where $dt = dt_1 + dt_2$. Hence the difference between the positions calculated by the two approaches is:

$$\bar{x}_p^{n+1} - x_p^{n+1} = (v_i^{n+1} - v_{i-1}^{n+1}) \left[\frac{x_i - x_p^n}{x_i - x_{i-1}} \right] dt_2 - \left[a_{i-1}^n + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}} (a_i^n - a_{i-1}^n) \right] dt_1 dt_2 \tag{67}$$

Figure 6 illustrates the different values of spatial position that may result when the



[Fig. 5] Mesh Crossing Diagram.

discontinuity is and is not considered.

Dividing both sides of equation (67) by $(x_i - x_{i-1})$ gives:

$$\frac{\bar{x}_p^{n+1} - x_p^{n+1}}{x_i - x_{i-1}} \approx dt_2 \frac{(v_i^{n+1} - v_{i-1}^{n+1})}{(x_i - x_{i-1})} C + \frac{dt_1 dt_2}{x_i - x_{i-1}} \left[a_{i-1}^n + \frac{x_p^n - x_{i-1}}{x_i - x_{i-1}} (a_i^n - a_{i-1}^n) \right] \tag{68}$$

where $C = \left[\frac{x_i - x_p^n}{x_i - x_{i-1}} \right]$ and where $0 \leq C \leq 1$.

The term $\frac{\bar{x}_p^{n+1} - x_p^{n+1}}{x_i - x_{i-1}}$ is the relative error in the position. And as the values of $(v_i^{n+1} - v_{i-1}^{n+1})$ are $O(1)$ for the case considered here it follows that limiting $dt \frac{\partial v}{\partial x} < 0.1$ will control the relative position error on the step, as also suggested in Section 6.2.

8. Spatial Error Estimation.

In evaluating the spatial error there are three main sources of errors: the mass mapping error introduced by equation (23), the momentum mapping error introduced by equation (24), and the force mapping error introduced by equation (26). Before considering these equations it is helpful to establish some notation relating to an important result Theorem 2.3 of Hickernell

Copyright © 2008 John Wiley & Sons, Ltd.

[12], who proves that for any function $f(x) \in X^p \equiv \left[f : \frac{df}{dx} \in L^p([0,1])\right]$:

$$\left| \int_0^1 f(y)dy - \frac{1}{N_p} \sum_{i=1}^{N_p} f(z_i) \right| \leq D_2(P, N_p) \ \left\| \frac{df}{dx} \right\|_2 \tag{69}$$

where

$$D_2(P, N_p) \ = \sqrt{\frac{1}{12N_p^2} \ + \frac{1}{N_p} \sum_{i=1}^{N_p} \left(z_i - \frac{2i-1}{2N_p} \right)^2}, \tag{70}$$

z_i is an ordered set of the points $x_p \in [0,1]$. Although Hickernell proves the result for more general norms, the above result is sufficient for this analysis. It is important to translate Hickernell's result to the sub-intervals used in the MPM method. The constant $D_2(P, N_p)$ is unchanged except that the points $\frac{2i-1}{2N_p}$ need to be translated to the interval I_{i+1}. In considering an integral over a domain of width h, Theorem (2.3) then becomes

$$\left| \frac{1}{h_{i+1}} \int_{x_i}^{x_{i+1}} f(y)dy - \frac{1}{N_p^{i+1}} \sum_{i=1}^{N_p^{i+1}} f(x_i) \right| \leq D_2(P, N_p^{i+1}) \ (h_{i+1})^{1/2} \left\| \frac{df}{dx} \right\|_{2,h_{i+1}} \tag{71}$$

where

$$\left\| \frac{df}{dx} \right\|_{2,h_{i+1}} = \left[\int_{x_i}^{x_{i+1}} \left(\frac{df}{dx} \right)^2 dx \right]^{1/2} \tag{72}$$

and where

$$D_2(P, N_p^{i+1}) \ = \sqrt{\frac{1}{12(N_p^{i+1})^2} \ + \frac{1}{N_p^{i+1} h^2} \sum_{i=1}^{N_p^{i+1}} \left((hz_i + x_i) - \left(x_i + \frac{(2i-1)h}{2N_p^{i+1}} \right) \right)^2} \tag{73}$$

It should also be noted that from the mean value theorem for integration

$$(h_{i+1})^{1/2} \left[\int_{x_i}^{x_{i+1}} \left(\frac{df}{dx} \right)^2 dx \right]^{1/2} = (h_{i+1}) \left| \frac{df}{dx}(\xi) \right| \tag{74}$$

for some $\xi \in I_{i+1}$. Hence

$$\left| \int_{x_i}^{x_{i+1}} f(y)dy - \frac{h_{i+1}}{N_p^{i+1}} \sum_{i=1}^{N_p^{i+1}} f(x_i) \right| \leq D_2(P, N_p^{i+1}) \ h_{i+1}^2 \left| \frac{df}{dx}(\xi) \right| \tag{75}$$

The values of $D_2(P, N_p^{i+1})$ clearly depend on the point distribution and thus in turn on the problem being solved. Considering the worst case of particles negligible distances apart at the end of an interval it is straightforward to show that

$$\frac{1}{2\sqrt{3}N_p^{i+1}} \leq D_2(P, N_p^{i+1}) \leq \frac{1}{\sqrt{3}} \tag{76}$$

Copyright © 2008 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

This result has a similar form to the results of Vshivkov[10] (as quoted by Brackbill, [3]) except that the key difference here lies in the choice of quadrature rule. Vshivkov calculates the error, δ_k, in the charge density at node k as computed with the PIC. His result states that

$$\delta_k \leq \left(\frac{3\rho_{\text{av}}^2}{2\rho_{\text{min}}} + h \frac{\rho_{\text{av}}^2 \rho_{\text{max}}}{6\rho_{\text{min}}^3} \left| \frac{\partial \rho}{\partial x} \right|_{\text{max}} \right) \frac{1}{N^2} + \frac{h^2}{12} \left| \frac{\partial^2 \rho}{\partial x^2} \right|_{\text{max}} \tag{77}$$

where N is the average number of particles in a cell. In the discussion that follows it is convenient to assume that the meshpoints are evenly spaced i.e.

$$h = h_{i+1} = h_i. \tag{78}$$

8.1. Ringing Instability

It is also important to remark that, as with any quadrature rule, there exist values of $f(x)$ such that $f(x_j) = 0$. For example if

$$f(x) = \prod_{j=1}^{N_p^i} (x - x_j) \tag{79}$$

then the integral approximation is zero and the error is the value of the integral. Furthermore there are functions which are non-zero at the particle points such as

$$f(x_i) = (-1)^i \tag{80}$$

which in the case of even numbers of mesh points will give a zero contribution to the integral. The problem is made worse by the fact that the quadrature rule is essentially using a piecewise constant approximation to function in forming the integral in the most general case. This loss of information due to quadrature is known as the "Ringing Instability" and is a well-known feature of particle methods that is attributed to the under-representation of particle data on the grid. Brackbill [4] and Macneice [17] explain this instability in terms of Fourier analysis.

8.2. Mass Projection Error

The mass error associated with equation (23) is denoted by E_m^i and is defined by

$$E_m^i = \int_{x_{i-1}}^{x_{i+1}} \rho(x) S_i(x) dx - m_i \tag{81}$$

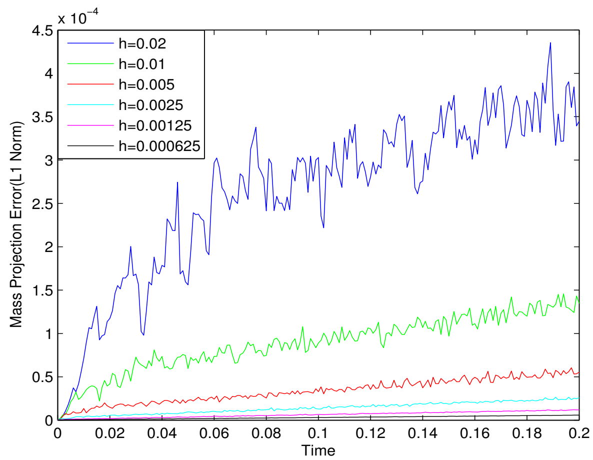
where there are N_p^j points in the interval I_j. This may be written more explicitly in terms of the points in each interval, by using equation (15) and (23), as

$$E_m^i = \int_{x_{i-1}}^{x_i} \rho(x) S_i(x) dx - \frac{h}{N_p^i} \sum_{p: x_p \in I_i} S_i(x_p) \rho_p + \int_{x_i}^{x_{i+1}} \rho(x) S_i(x) dx - \frac{h}{N_p^{i+1}} \sum_{p: x_p \in I_{i+1}} S_i(x_p) \rho_p \tag{82}$$

The error term is thus composed of two terms each of which is similar to the right side of equation (75):

$$\left| E_m^i \right| \leq D_2(P, N_p^i) h^2 \left| \frac{d(\rho(x) S_i(x))}{dx} (\xi_1) \right| + D_2(P, N_p^{i+1}) h^2 \left| \frac{d(\rho(x) S_i(x))}{dx} (\xi_2) \right| \tag{83}$$

for some $\xi_1 \in I_i$ and some $\xi_2 \in I_{i+1}$. However as the first derivative of $S_i(x)$ depends on $\frac{1}{h}$, this results in the mass error E^i_m being first order in h. An approximate L_1 norm of mass projection error is calculated by using the trapezoidal quadrature rule, based upon the true error in the mass at mesh points. The result in Figure 6 shows how the mass projection error grows for different mesh sizes and is first order of mesh size as expected. The errors grow in time in a way that is consistent with first time integration using the forward Euler method.



[Fig. 6] L1 norm of mass errors at nodes for different meshes

8.3. Momentum Projection Error

The momentum error associated with equation (36) is denoted by E^i_P and is given by a similar expression as the mass error except that terms of the from $\rho(x)S_i(x)$ are replaced with $v(x)\rho(x)S_i(x)$ i.e.

$$\left|E^i_P\right| \leq D_2(P, N^i_p)h^2 \left|\frac{d(\rho(x)v(x)S_i(x))}{dx}(\xi_1)\right| + D_2(P, N^{i+1}_p)h^2 \left|\frac{d(\rho(x)v(x)S_i(x))}{dx}(\xi_2)\right| \quad (84)$$

for some $\xi_1 \in I_i$ and some $\xi_2 \in I_{i+1}$. It follows that the momentum error is also first order in h. A graph of the momentum projection error is very similar to Fig. 6.

8.4. Velocity Projection Error

The nodal velocity, v_i, is calculated from the mass and the momentum of the node as in equation (25). The exact projected velocity is given by

$$v^{expro}_i = \frac{P^{expro}_i}{m^{expro}_i} = \frac{\int S_i(x)\rho(x)v(x)dx}{\int S_i(x)\rho(x)dx}. \quad (85)$$

While the division by an integral containing $\rho(x)$ may be problematic; the method described above has a number of steps to ensure that at least one particle with mass is in every cell interval.

The error in the velocity projection, $E^i_{vproj}(t)$, is defined by:

$$E^i_{vproj}(t) = v^{expro}_i - v_i. \quad (86)$$

Let $v(x_i,t)$ be the exact nodal velocity at t, and define the error from projection in the exact value as:

$$E^i_{v1}(t) = v(x_i,t) - v_i^{expro}. \tag{87}$$

Then overall error in velocity projection may be split into two parts:

$$E^i_v(t) = v(x_i,t) - v_i = E^i_{v1}(t) + E^i_{vproj}(t). \tag{88}$$

Let:

$$\frac{\delta^2 U}{\delta x^2}(x,t) = \rho(x,t)v(x,t) \tag{89}$$

and:

$$\frac{\delta^2 V}{\delta x^2}(x,t) = \rho(x,t). \tag{90}$$

Then the exact velocity is defined by:

$$v(x_i,t) = \frac{h\frac{\delta^2 U}{\delta x^2}(x,t)}{h\frac{\delta^2 V}{\delta x^2}(x,t)}. \tag{91}$$

Using integration by parts, the projection of the velocity satisfies:

$$\begin{aligned} v_i^{expro} &= \frac{\int S_i(x)\rho(x)v(x)dx}{\int S_i(x)\rho(x)dx} \\ &= \frac{\frac{1}{h}(U(x_i-H,t) - 2U(x_i,t) + U(x_i+H,t))}{\frac{1}{h}(V(x_i-H,t) - 2V(x_i,t) + V(x_i+H,t))}. \end{aligned}$$

Define two projection errors $E^i_U(t)$ and $E^i_V(t)$ by:

$$E^i_U(t) = h\frac{\delta^2 U}{\delta x^2}(x_i,t) - \int S_i(x)\rho(x)v(x)dx \tag{92}$$

where using standard finite differeence analysis $E^i_U = O(h^3) + H.O.T$, and

$$E^i_V(t) = h\frac{\delta^2 V}{\delta x^2}(x_i,t) - \int S_i(x)\rho(x)dx \tag{93}$$

and where $E^i_V = O(h^3) + H.O.T$ similarly. The partial projection error $E^i_{v1}(t)$ is then given by:

$$\begin{aligned} E^i_{v1}(t) &= \frac{h\frac{\delta^2 U}{\delta x^2}(x,t)}{h\frac{\delta^2 V}{\delta x^2}(x,t)} - \frac{\int S_i(x)\rho(x)v(x)dx}{\int S_i(x)\rho(x)dx} \\ &= \frac{1}{\int S_i(x)\rho(x)dx}(E^i_U - v(x_i,t)E^i_V). \end{aligned}$$

As E^i_U and E^i_V are third order in h and $\int S_i(x)\rho(x)dx$ is first order in h, it follows that $E^i_{v1}(t)$ is second order in h. The second part of the projection error is defined by:

$$\begin{aligned} E^i_{vproj}(t) &= \frac{\int S_i(x)\rho(x)v(x)dx}{\int S_i(x)\rho(x)dx} - \frac{P_i}{m_i} \\ &= \frac{1}{m_i}(E^i_p(t) - v(x_i,t)E^i_m(t)) \end{aligned}$$

where

$$E_p^i(t) = \int_{x_{i-1}}^{x_{i+1}} \rho(x) S_i(x) v(x) dx - \frac{h}{N_p^i} \sum_{p:x_p \in I_i} S_i(x_p) \rho_p v_p - \frac{h}{N_p^{i+1}} \sum_{p:x_p \in I_{i+1}} S_i(x_p) \rho_p v_p$$

Using a Taylors series expansion of velocity about x_i gives:

$$E_p^i(t) = v(x_i,t) E_m^i(t) + v_x(x_i,t) E_{vp1}^i(t) + \frac{v_{xx}(x_i,t)}{2} E_{vp2}^i(t) + ...+ \quad (94)$$

where:

$$E_{vpk}^i = \int_{x_{i-1}}^{x_{i+1}} S_i(x)\rho(x)(x-x_i)^k dx - \frac{h}{N_p^i} \sum_{p:x_p \in I_i} \bar{S}_{ip} \rho_p (x_p - x_i)^k - \frac{h}{N_p^{i+1}} \sum_{p:x_p \in I_{i+1}} \bar{S}_{ip} \rho_p (x_p - x_i)^k \quad (95)$$

Therefore:

$$E_v^i(t) = E_{v1}^i(t) + \frac{1}{m_i} \left(v_x(x_i,t) E_{vp1}^i(t) + \frac{v_{xx}(x_i,t)}{2} E_{vp2}^i(t) + ... \right) \quad (96)$$

Using Hickernell's result from equation (71), gives:

$$\left| E_{vpk}^i \right| \leq D_2(P, N_p^i) h^2 \left| \frac{d(\rho(x) S_i(x)(x-x_i)^k)}{dx}(\xi_1) \right| + D_2(P, N_p^{i+1}) h^2 \left| \frac{d(\rho(x) S_i(x)(x-x_i)^k)}{dx}(\xi_2) \right| \quad (97)$$

for some $\xi_1 \in I_i$ and some $\xi_2 \in I_{i+1}$. For the lowest order term $k = 1$ this is second order.

8.5. *Acceleration Projection Error*

We define the projection error in acceleration, E_a^i, is:

$$E_a^i = a(x_i,t) - a_i \quad (98)$$

where $a(x_i,t)$ is the exact acceleration at node x_i at time t. As for the velocity projection error, the acceleration projection error may be split into two parts:

$$E_a^i = (a(x_i,t) - a_i^{expro}) + (a_i^{expro} - a_i) = E_{a1}^i(t) + E_{aproj}^i(t) \quad (99)$$

where a_i^{expro} is exact nodal acceleration obtained by projecting the exact pressure and density onto the mesh points, and a_i is calculated nodal acceleration from (30). The error $E_{a1}^i(t)$ may be shown to be second order in h using the same approach as in equations (85) to (94). The second part of acceleration projection error is:

$$E_{aproj}^i(t) = \frac{\frac{1}{h}(\int_{x_{i-1}}^{x_i} p(x)dx - \int_{x_i}^{x_{i+1}} p(x)dx)}{\int_{x_{i-1}}^{x_{i+1}} \rho(x) S_i(x) dx} - \frac{1}{N_p^i} \sum_{p:x_p \in I_i} p_p - \frac{1}{N_p^{i+1}} \sum_{p:x_p \in I_{i+1}} p_p \quad (100)$$

Then:

$$E_{aproj}^i(t) = \frac{1}{m_i}(E_F^i(t) - a(x_i,t) E_m^i(t) \quad (101)$$

where:

$$E_F^i = \left(\frac{1}{h} \int_{x_{i-1}}^{x_i} p(x)dx - \frac{1}{N_p^i} \sum_{p:x_p \in I_i} p_p \right) + \left(-\frac{1}{h} \int_{x_i}^{x_{i+1}} p(x)dx + \frac{1}{N_p^{i+1}} \sum_{p:x_p \in I_{i+1}} p_p \right). \quad (102)$$

Copyright © 2008 John Wiley & Sons, Ltd.

Expanding the values of pressure about x_i gives:

$$\frac{1}{h}\int_{x_{i-1}}^{x_i} p(x)dx - \frac{1}{N_p^i}\sum_{p:x_p\in I_i} p_p = p_x(x_i)\left(\frac{x_i+x_{i-1}}{2} - \frac{1}{N_p^i}\sum_{p:x_p\in I_i} x_p\right) + \frac{p_{xx}(x_i)}{2}\left(\int_{x_{i-1}}^{x_i}\frac{(x-x_i)^2}{h}dx - \frac{h}{N_p^i}\sum_{p:x_p\in I_i}\frac{(x_p-x_i)^2}{h}\right)$$

and similarly for the interval $[x_i, x_{i+1}]$. The lowest order term in the error is then:

$$E_F^i = p_x(x_i)\left(h - \frac{1}{N_p^i}\sum_{p:x_p\in I_i} x_p + \frac{1}{N_p^{i+1}}\sum_{p:x_p\in I_{i+1}} x_p\right) + H.O.T. \tag{103}$$

In order to investigate the order of this term it is necessary to consider the evolution of the points that contribute to the calculation of acceleration at the point x_i at time t_n. Let means of particle positions and velocities be defined by

$$\bar{x}_{i+1}^n(t) = \frac{1}{N_p^{i+1}}\sum_{p:x_p(t_n)\in I_{i+1}} x_p(t) \tag{104}$$

$$\bar{v}_{i+1}^n(t) = \frac{1}{N_p^{i+1}}\sum_{p:x_p(t_n)\in I_{i+1}} v_p(t) \tag{105}$$

Furthermore define

$$\frac{d\bar{v}_i^n}{dx}(t) = \frac{\bar{v}_{i+1}^n(t) - \bar{v}_{i+1}^n(t)}{\bar{x}_{i+1}^n(t) - \bar{x}_i^n(t)} \tag{106}$$

From equations (104),(105) and (107) it follows that

$$\bar{x}_{i+1}^n(t_{n+1}) - \bar{x}_i^n(t_{n+1}) = [1 + \Delta t\frac{d\bar{v}_i^n}{dx}(t_n)](\bar{x}_{i+1}^n(t_n) - \bar{x}_i^n(t_n)) \tag{107}$$

and hence that the gap between the means may be related back to the initial mesh distribution.

$$\bar{x}_{i+1}^n(t_{n+1}) - \bar{x}_i^n(t_{n+1}) = \prod_j [1 + \Delta t\frac{d\bar{v}_i^n}{dx}(t_j)](\bar{x}_{i+1}^n(t_0) - \bar{x}_i^n(t_0)). \tag{108}$$

Suppose that initially all the points are evenly distributed at time t_0 with spacing h_p, then

$$(\bar{x}_{i+1}^n(t_0) - \bar{x}_i^n(t_0)) = h_p(N_p^{i+1} + N_p^i)/2 \tag{109}$$

where the interval spacing h is connected to the initial particle spacing h_p through

$$h = h_p(N_p^0 + 1) \tag{110}$$

where N_p^0 is the total number of points in every interval at t_0. Hence

$$\bar{x}_{i+1}^n(t_{n+1}) - \bar{x}_i^n(t_{n+1}) = h\prod_j [1 + \Delta t\frac{d\bar{v}_i^n}{dx}(t_j)]\left[\frac{N_p^{i+1} + N_p^i}{2(N_p^0+1)}\right]. \tag{111}$$

Copyright © 2008 John Wiley & Sons, Ltd.

Using the CFL condition as defined by equation (46) then gives

$$\bar{x}_{i+1}^n(t_{n+1}) - \bar{x}_i^n(t_{n+1}) = h[1 + h\ CFL\ K]\frac{N_p^{i+1} + N_p^i}{2(N_p^0 + 1)} + h.o.t \tag{112}$$

where

$$K = \sum_j [\frac{d\bar{v}_i^n}{dx}(t_j)] \tag{113}$$

This result shows that the acceleration order may be first order if local velocity gradients are "small" if particles are rezoned as to be closer to evenly spaced as in section (5.3).

8.6. Velocity gradient error.

The accuracy of the equations used to update energy and density in Section 5.1 depends on the accuracy of the velocity gradient and velocity gradient at any particle $x_p \in I_{i+1}$ is defined as:

$$\frac{\delta v}{\delta x}(x_p) = \frac{v_{i+1} - v_i}{x_{i+1} - x_i} - (\frac{x_{i+1} + x_i}{2} - x_p)\frac{\delta^2 v}{\delta x^2}(x_p) + H.O.T \tag{114}$$

The velocity gradient error at particles is rewritten as:

$$E_{VG}^p = \frac{E_v^{i+1} - E_v^i}{h} + \frac{\Delta t}{h}[E_a^{i+1} - E_a^i] - (\frac{x_{i+1} + x_i}{2} - x_p)\frac{\delta^2 v}{\delta x^2}(x_p) \tag{115}$$

Thus the velocity gradient error depends on teh first order interpolation error.

9. Combining the error estimate results.

The density errors at T=0.2 in the apprpoximate L1-Norm and L2-Norm for different mesh sizes are shown in Table II. We are using same CFL as in section 6 and the initial number of particles per cell is also 8 throughout. The numbers in this Table indicate that the density error is order of h in the approximate L1-Norm and order of $h^{\frac{1}{2}}$ in the approximate L2-Norm. To understand the orders of these norms a detailed inspection of the order of accuracy in each

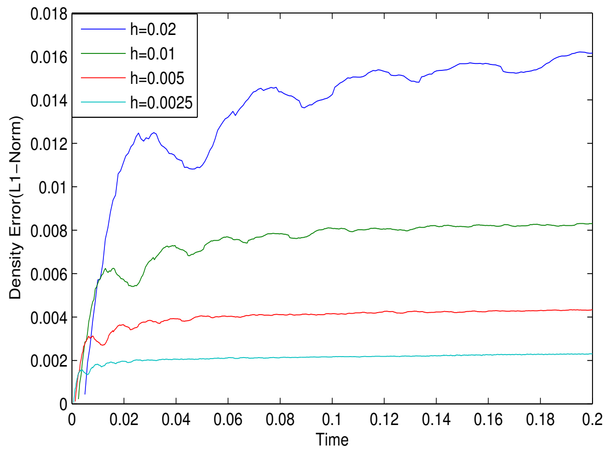
Table II. Density Error at T=0.2 in L1-Norm and L2-Norm and Pointwise maximum error at mesh points.

h	L1-Norm	L2-Norm	Max. Density Err. at Mesh Pts
0.02	0.00161	0.02484	0.1051
0.01	0.00831	0.01587	0.0812
0.005	0.00434	0.01046	0.1139
0.0025	0.00231	0.00759	0.1063
0.00125	0.00136	0.00626	0.1002
0.000625	0.00110	0.00619	0.0989

part of the spatial domain was made. In the regions around the contact discontinuity and the

Copyright © 2008 John Wiley & Sons, Ltd.

shock the maximum pointwise error does not decrease but the interval over which it occurs is reduced with the mesh spacing h. The approximate L1 norm is $h|E_{max}|$ while the approximate L2 norm is $\sqrt{h}|E_{max}|$, thus giving rise to the observed orders of convergence. Figure 7 shows the evolution in time of the L1 norm of the density error for different mesh sizes.



[Fig. 7] L1 norm of density errors over time for different meshes

10. Summary

In this paper the accuracy properties of a variant of the Material Point Method are investigated in depth on a well-known test problem in one space dimension. The analysis leads to the same conclusion that the method is order one half to first order in accuracy for a sufficently small CFL number. The analysis also shows that this accuracy depends on a sufficiently well-behaved point distribution. This point distribution can be verified computationally in a straightforward manner. Computational experiments have been used to show that the observed experiments match the computed experiments. The importance of this analysis is that it provides a way to make a more formal assessment of many of the errors in MPM type methods. This inturn makes it possible to start to consider an analysis of higher space dimensional Meterial Point Methods.

Acknowledgement: This work was supported by the University of Utah's Center for the Simulation of Accidental Fires and Explosions (C-SAFE) funded by the Department of Energy, under subcontract No. B524196. The authors would like to thank their colleagues Jim Guilkey, Mike Kirby, Mike Steffen, and Phillip Wallstedt at Utah for many helpful suggestions, and the referees for their constructive comments.

REFERENCES

1. Ascher UM Petzold LR Computer Methods for Ordinary Differential Equations and Differential Algebraic equations. SIAM, Philadelphia, 1998.

Copyright © 2008 John Wiley & Sons, Ltd.

2. Bardenhagen SG Kober EM The Generalized Interpolation Material Point Method. CMES vol 5, no 4 pp477-495.
3. Brackbill J.U. Particle Methods. International Jour. Numer. Meths. in Fluids, 2005:47:693-705
4. Brackbill J.U. The Ringing Instability in Particle in Cell Calculations of Low-Speed Flow. Journal of Computational Physics 75, 469-492 (1988).
5. Brackbill J.U. and Ruppel H.M. FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flow in two Dimensions. Journal of Computational Physics 65, 314-343 (1986).
6. Brownlee J, Levesley J, Houston P, and Rosswog S. Enhancing SPH using moving least-squares and radial basis functions. In Proc. A4A5 (Algorithms for Approximation) , Chester UK, Jul. 18-22 2005, Springer, 2007.
7. Brydon AD , Bardenhagen SG, Miller EA , Seidler GT. Simulation of the densification of real open-celled foam microstructures. Journal of the Mechanics and Physics of Solids, 53:2638–2660, 2005.
8. Burgess D, Sulsky D and Brackbill J.U. Mass Matrix formulation of the FLIP Particle in Cell Method Journal of Computational Physics 103, 1-15 (1992).
9. Chawla M.M. and Subramanian R. regions of absolute stability of explicit Runge Kutta Nystrom methods for $y'' = f(x, y, y')$. Journal of Computational and Applied Math. 11 (1984) 259-266.
10. Grigoryev Yu n., Vshivkov V.A. and Fedoruk M.P. Numerical Particle in Cell Methods Theory and Applications. VSP, Utrecht, Boston, 2002.
11. Guo Y Nairn JA. Calculation of j-integral and stress intensity factors using the material point method. Computer Modeling in Engineering and Sciences, 6:295–308, 2004.
12. Hickernel F.J. A Generalized Discrepancy and Quadrature Bound Mathematics of Computation, 67:299–322, 1998.
13. Henderson TC, McMurtry PA, Smith PJ Voth GA, Wight CA, Pershing DF Simulating Accidental Fires and Explosions Computing in Science and Engineering March/April 2000 (Vol.2, No.2) 64-76.
14. Kim J. MPM Masters Project report unpublished. 2004
15. Leveque R.J. Finite Volume Methods for Hyperbolic Problems Cambridge Texts in Applied mathematics, Cambridge Univesity Press, 2002.
16. Li S. and Liu W.K. Meshfree and particle methods and their applications. Appl. Mesh Rev. Vol. 55, no 1, January 2002, 1-33.
17. MAcNeice, P. Particle mesh techniques. NASA Contractor Report 4666, Hughes STX, Goddard Space Center, Greenbelt MD 20771.
18. Ma S, Zhang X, Qiui XM Comparison study of MPM and SPH in modeling hypervelocity impact problems. Impact Engineering (in press).
19. Monaghan JJ and Gingold RA Shock Simulation by the Particle Method SPH Journal of Computational Physics 52, 374-389 (1983).
20. Monaghan JJ and Pongracic H. Artificial Viscosity for Particle Methods Applied Numerical Mathematics 1. 1985 187-194.
21. Parker SG A Component-Based Architecture for Parallel Multi-physics PDE Simulation Computational Science - ICCS 2002: Int.Conf., Amsterdam, The Netherlands, April 21-24, 2002. Proceedings, Part III PMA. Sloot, CJ Kenneth Tan, JJ Dongarra, AG Hoekstra (Eds.): Lecture Notes in Computer Science, Vol 2331 Springer-Verlag GmbH, 2002, ISSN: 0302-9743.
22. Shampine LF Local Error Estimation by Doubling. Computing. Vol 34,2, 179-190.
23. Sod GA A survey of several difference methods for systems of nonlinear hyperbolic conservation laws. Journal of Computational Physics. 27. (1978) 1-31.
24. Steffen M; Kirby RM; Berzins M (2008) Analysis and reduction of quadrature errors in the material point method (MPM). International Journal for Numerical Methods in Engineering. DOI: 10.1002/nme.2360.
25. Steffen M; Wallstedt PC; Guilkey JE; Kirby RM; Berzins M Examination and Analysis of Implementation Choices within the Material Point Method (MPM). In Computer Modeling in Engineering and Sciences Vol. 32 No. 2(2008):107-127.
26. Sulsky D, Chen Z and Schreyer HL A particle method for history-dependent materials. Comput. Meths in Appl Mech. and Engng 118 (1994):179-196.
27. Sulsky D, Zhou S-J and Schreyer HL Application of a particle-in-cell method to solid mechanics. Computer Physics Communications 87 (1995):236-252 .
28. Wallstedt P; Guilkey J. (2007): Improved velocity projection for the material point method. Computer Modeling in Engineering and Sciences, vol. 19, pp. 223-232.
29. Wallstedt, PC; Guilkey JE (2008) An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. Journal of Computational Physics. (in press)
30. York AR, Sulsky D and Schreyer HL Fluid- membrane interaction based on the material point method. Int. Jour. for Numer. Meths. in Engng. 2000 48:901-924.

Copyright © 2008 John Wiley & Sons, Ltd.