INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng. **44**, 1429–1456 (1999)

THE MATERIAL POINT METHOD FOR SIMULATION OF THIN MEMBRANES

ALLEN R. YORK II[1,*], DEBORAH SULSKY[2] AND HOWARD L. SCHREYER[3]

[1]*Applied Research Associates, 811 Spring Forest Road, Raleigh, NL 27502, U.S.A.*
[2]*Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131, U.S.A.*
[3]*Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM 87131, U.S.A.*

SUMMARY

The material-point method (MPM) is extended to handle membranes, which are discretized by a collection of unconnected material points placed along each membrane surface. These points provide a Lagrangian description of the membrane. To solve for the membrane motion, data carried by the material points are transferred to a background mesh where equations of motion are discretized and solved. Then the solution on the background mesh is used to update the membrane material points. This process of combining Lagrangian and Eulerian features is standard in MPM; the modification for membranes involves merely an implementation of the constitutive equation in a local, normal-tangential coordinate system. It is shown that this procedure does, in fact, provide adequate resolution of membranes with thicknesses that can vary substantially from that of the background mesh spacing. A general formulation is given, but the implementation is in a two-dimensional code that provides a proof-of-principle.

Numerical examples including a spring, pendulum and a string with initial slack are used to illustrate the method. The string with slack uses an additional modification of the membrane constitutive equation that allows wrinkles to be modeled at low computational cost. Presented also are examples of two disks impacting, pinching a membrane and rebounding, a difficult problem for standard finite element codes. These simulations require a relaxation of the automatic no-slip contact algorithm in MPM. The addition of the capability to model membranes and the new contact algorithm provide a significant improvement over existing methods for handling an important class of problems. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: material point method; particle method; membrane; meshless; explicit; dynamics

1. INTRODUCTION

The material point method (MPM) has evolved from a particle-in-cell (PIC) method called FLIP originally developed at Los Alamos National Laboratory for fluid dynamics problems.[1,2] The basic formulation of the MPM with applications to materials having strength and stiffness has been described previously.[3–5] In the method, a mesh of Lagrangian material points is used to discretize one or more solid bodies. The material points have mass and velocity and carry stress

*Correspondence to: Allen R. York, Applied Research Associates, 811 Spring Forest Road, Raleigh, NL 27502, U.S.A.
E-mail: ayork@ara.com

Contract/grant sponsor: United States Department of Energy; Contract/grant number: DE-ACO4-94AL85000

CCC 0029–5981/99/101429–28$17.50
Copyright © 1999 John Wiley & Sons, Ltd.
Received 29 October 1997
Revised 18 May 1998

and strain, which may be history dependent. The interaction of the material points is calculated on a background Eulerian or spatial finite element mesh on which the momentum equation is solved. Material point quantities are updated from the solution on the Eulerian mesh with mapping functions.

The advantages of both Lagrangian and Eulerian methods are realized in the MPM. Mesh distortion and entanglement are not a problem as it might be with Lagrangian schemes since the background mesh is under user control and can be redefined each time step if desired. Advection of history-dependent variables is straightforward since the variables are defined on material points which themselves advect through the Eulerian mesh. Thus, numerical dissipation normally associated with an Eulerian method is avoided.

This paper describes an extension of the method to model membranes, which are structures, usually thin, with no bending stiffness. The traction is constant through the thickness of a membrane and is directed tangent to the membrane reference surface. Examples of physical structures that may be approximated by a membrane idealization are strings and thin wires (one-dimensional membranes), audio and video tape, fabrics (such as those used in airbags or parachutes), pressurized shelters, and the walls of living cells.

Other existing methods such as the finite element method certainly have the capability of simulating membranes.[6–8] The advancement here is the simplified framework in which the membrane formulation is implemented. A membrane surface is discretized by marking a set of material points on it. Thus, complicated mesh generation is not needed. The surface of material points is overlaid on a background computational mesh used to solve the governing equations of motion. The enforcement of tangential traction and the associated stress state in the membrane is done on a point-by-point basis, with the internal forces due to stresses determined on the background mesh. The mesh in all cases considered here is a square Cartesian grid. It is found that the governing equations for a membrane oriented arbitrarily to the mesh can be solved on a square grid with reasonable mesh sizes. The inherent feature of the MPM allowing the appropriate constitutive equation to be applied on a point-by-point basis and the new membrane formulation lays the foundation for studying fluid-structure interactions that will be taken up in future work. Fluid-membrane problems are an important subset of fluid-structure problems as evidenced by recent papers in the literature.[9–11]

The basic ideas presented here are somewhat similar to the method used by Peskin in simulating blood flow through the human heart.[13,14] Peskin immerses massless Lagrangian bars, chained together to form a surface, in an Eulerian mesh. As with our method, the Lagrangian bars do not have to coincide with the grid-node points of the mesh. Forces in the bars are interpolated to the mesh to provide constraints on the fluid motion in fluid-structure interactions. The bars then move in the computed velocity field. With the Peskin method, the connections between the bars are needed in order to compute the force which complicates the discretization of the surface by bars. The material points in the MPM carry mass and represent the membrane surface through continuum constitutive equations that describe properties of the material being modeled. Implementation of the constitutive model does not require connecting the material points.

There are several advantages inherent to the MPM simulation of membranes. The time step constraint for the explicit method is a function of the mesh size and not of the spacing between material points. No-slip contact is automatic and is provided at no additional computational cost. Since no-slip is the default contact condition between impacting bodies, if no alteration is made to the basic MPM algorithm, it is observed that under certain circumstances;

Copyright © 1999 John Wiley & Sons, Ltd.

bodies may numerically 'stick' to one another and not move away as they should. An explanation for this behaviour is given in Section 4, and a solution is proposed in the form of a contact-release algorithm. A simulation of impacting bars shows the effect of the improved algorithm.

Contact algorithms in standard updated Lagrangian finite element methods have difficulty in simulations where a membrane is pinched between two bodies. The difficulty arises because the thin membrane is continuously pushed out of one body and into another by the contact algorithm. The combined membrane and contact-release algorithm implemented in MPM is able to predict the deformation of two disks impacting and pinching a membrane. Results of this simulation are also given in Section 4.

The governing and discrete equations have been discussed in previous publications[3–5] and are briefly reviewed in Section 2. The modifications to the method for simulating a membrane are presented in Section 3 which also contains numerical examples including a spring, pendulum, and a string with initial slack. The computed, elastic vibrations of the spring are compared with an analytic solution, and the motion of the pendulum is compared to a finite element solution of the same problem. The string-with-slack example uses a modification to the membrane constitutive equation that allows wrinkles to be modeled at low computational cost. The idea is to allow a membrane material point to accumulate compressive strain without causing compressive stress. To date, the algorithm has only been implemented in a two-dimensional code, so calculations are restricted to one-dimensional membranes. Section 4 provides a description of the contact-release algorithm with example calculations consisting of impacting bars, and two disks impacting and pinching a membrane. Finally, concluding comments are presented.

2. DEVELOPMENT OF THE DISCRETIZED EQUATIONS

In the material point method the momentum equation is solved on the background Eulerian mesh. This background mesh provides a convenient means to define discrete derivatives. The source data for the solution of the momentum equation come from the material points. Mass and stresses at material points are mapped to the Eulerian mesh where conservation of momentum is applied (Figure 1).

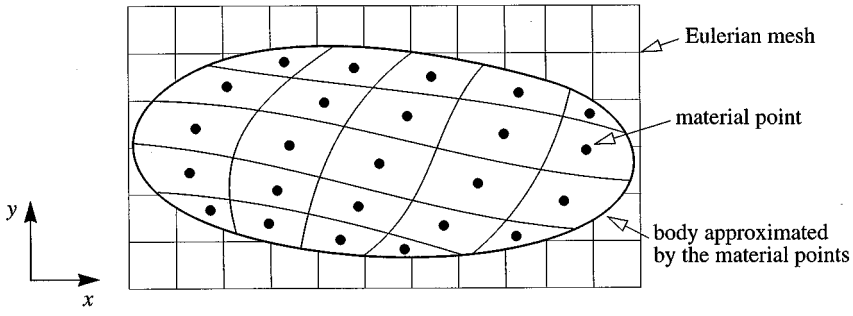


Figure 1. Mesh of Lagrangian material points representing a solid body overlaid on the computational mesh

Copyright © 1999 John Wiley & Sons, Ltd.

2.1. Spatial discretization of the governing equations

For the current position $\mathbf{x}$ and time t, let $\mathbf{v}(\mathbf{x}, t)$ be the mass density, $\rho(\mathbf{x}, t)$ be the velocity, $\boldsymbol{\sigma}(\mathbf{x}, t)$ be the Cauchy stress tensor, and $\mathbf{b}(\mathbf{x}, t)$ be the specific body force in the current configuration. Conservation of linear momentum is given by

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} = \rho \mathbf{a} \tag{1}$$

in which $\mathbf{a} = \dot{\mathbf{v}}$ is the acceleration, $\nabla(\ )$ and is the gradient with respect to the current configuration. The constitutive equation can be given by

$$\overset{\nabla}{\boldsymbol{\sigma}} = \mathbf{T} : \dot{\boldsymbol{\varepsilon}} \tag{2}$$

where the strain rate, $\dot{\boldsymbol{\varepsilon}}(\mathbf{x}, \mathbf{t})$, is the symmetric part of the velocity gradient

$$\dot{\boldsymbol{\varepsilon}} = \tfrac{1}{2}[(\nabla \mathbf{v}) + (\nabla \mathbf{v})^{\mathrm{T}}] \tag{3}$$

and $\mathbf{T}$ is the tangent modulus tensor. In all the simulations, $\mathbf{T}$ is the conventional fourth-order elasticity tensor since only materials with linear elastic properties are considered. The left-hand side of equation (2) denotes an objective stress rate; the Jaumann rate is used in this work since it is easy to implement and the strains are not that large.

In the following development all variables with a subscript i or j reference grid values, and variables with subscript p represent material point values. Equation (1) can be discretized as described in References 3–5 which results in a governing equation at each grid node of the Eulerian mesh. This equation takes the form

$$\sum_{j=1}^{N_n} m_{ij} \mathbf{a}_j = \mathbf{f}_i^{\mathrm{int}} + \mathbf{f}_i^{\mathrm{ext}} \quad i = 1, \ldots, N_n \tag{4}$$

where m_{ij} is the consistent mass matrix, $\mathbf{a}_j$ is the acceleration at node j, $\mathbf{f}_i^{\mathrm{int}}$ and $\mathbf{f}_i^{\mathrm{ext}}$ are the internal and external forces at node i, and N_n is the number of grid nodes. The internal force $\mathbf{f}_i^{\mathrm{int}}$ will be defined in the next section. To define the mass matrix, introduce nodal basis functions, $N_i(\mathbf{x})$ then the mass matrix is

$$m_{ij} = \sum_{p=1}^{N_p} m_p N_i(\mathbf{x}_p)\, N_j(\mathbf{x}_p) \tag{5}$$

where m_p is the mass of material point p and N_p is the total number of material points. In this work, the consistent mass matrix is replaced by the diagonal form simplifying equation (4) at the expense of introducing a slight amount of numerical dissipation[12] to give

$$m_i \mathbf{a}_i = \mathbf{f}_i^{\mathrm{int}} + \mathbf{f}_i^{\mathrm{ext}} \quad i = 1, \ldots, N_n \tag{6}$$

where the diagonal mass matrix is

$$m_i = \sum_p m_p N_i(\mathbf{x}_p) \tag{7}$$

Equation (6) can be recast into momentum form as

$$\frac{\mathrm{d}\mathbf{p}_i}{\mathrm{d}t} = \mathbf{f}_i^{\mathrm{int}} + \mathbf{f}_i^{\mathrm{ext}} \tag{8}$$

where $\mathbf{p}_i = m_i \mathbf{v}_i$ is the momentum, and i ranges from one to the number of grid nodes.

Copyright © 1999 John Wiley & Sons, Ltd.

2.2. Time integration algorithm

The discrete equations are to be solved at a discrete set of time steps, t^k, $k = 1, \ldots, K$. The superscript k indicates an approximation at time level k so that, for example, m_i^k is the approximation to $m_i(t^k)$. If we use an explicit integration of equation (8) to find the change in momentum, $\Delta\mathbf{p}_i$, the momentum for node i at the end of the Lagrangian step is

$$\mathbf{p}_i^{\mathrm{L}} = \mathbf{p}_i^k + \Delta\mathbf{p}_i = \mathbf{p}_i^k + \Delta t(\mathbf{f}_i^{\mathrm{int},\,k} + \mathbf{f}_i^{\mathrm{ext},\,k}) \tag{9}$$

where $\Delta t = t^{k+1} - t^k$. Note that the grid mass, m_i^k, depends on time due to the movement of material points through the grid; however, through the Lagrangian step defined in equation (8) the grid mass is constant. The momentum, $\mathbf{p}_i^k$, in equation (9) is determined from a mass-weighted mapping of material point velocities, $\mathbf{v}_p^k$, to the grid nodes

$$m_i^k \mathbf{v}_i^k = \sum_{p=1}^{N_p} m_p^k \mathbf{v}_p^k N_i(\mathbf{x}_p^k) \tag{10}$$

The material point velocity and position are updated using mapping functions, $N_i(\mathbf{x}_p^k)$, that map the grid node values of momentum and change in momentum to the material points

$$\begin{aligned} \mathbf{v}_p^{k+1} &= \mathbf{v}_p^k + \sum_{i=1}^{N_n} \Delta\mathbf{p}_i N_i(\mathbf{x}_p^k)/m_i^k \\ \mathbf{x}_p^{k+1} &= \mathbf{x}_p^k + \sum_{i=1}^{N_n} \Delta t\ \mathbf{p}_i^L N_i(\mathbf{x}_p^k)/m_i^k \end{aligned} \tag{11}$$

The strain increment $\Delta\boldsymbol{\varepsilon}_p$ at a material point is computed from the grid node velocities

$$\Delta\boldsymbol{\varepsilon}_p = \frac{\Delta t}{2} \sum_{i=1}^{N_n} (G_{ip}^k \mathbf{v}_i^{k+1} + (G_{ip}^k \mathbf{v}_i^{k+1})^{\mathrm{T}}) \tag{12}$$

where

$$G_{ip}^k \mathbf{v}_i^{k+1} = \nabla(\mathbf{v}_i^{k+1} N_i(\mathbf{x}))|_{x = x_p^x} \tag{13}$$

is the gradient of the grid velocity evaluated at $\mathbf{x}_p$, and the grid velocity $\mathbf{v}_i^{k+1}$ is determined using equation (10). The stress at a material point is updated using equation (2) in a frame indifferent formulation. For the Jaumann rate

$$\boldsymbol{\sigma}_p^{k+1} = \boldsymbol{\sigma}_p^k + \Delta t(\boldsymbol{\sigma}_p^k \cdot \mathbf{W}_p^k - \mathbf{W}_p^k \cdot \boldsymbol{\sigma}_p^k) + \mathbf{T}{:}\Delta\boldsymbol{\varepsilon}_p \tag{14}$$

where the vorticity is

$$\mathbf{W}_p^k = \frac{1}{2} \sum_{i=1}^{N_n} (G_{ip}^k \mathbf{v}_i^{k+1} - (G_{ip}^k \mathbf{v}_i^{k+1})^{\mathrm{T}}) \tag{15}$$

Then the internal forces are computed from the stress

$$\mathbf{f}_i^{\mathrm{int},k} = -\sum_p \frac{m_p}{\rho_p^k} G_{ip}^k \boldsymbol{\sigma}_p^k \tag{16}$$

where the quotient of material point mass and density, m_p/ρ_p^k, is the volume of material point p, and $\boldsymbol{\sigma}_p$ is the stress at material point p. The value of ρ_p^k is updated from the continuity equation

$$\rho_p^{k+1} = \rho_p^k/(1 + \mathrm{tr}(\Delta\boldsymbol{\varepsilon}_p)) \tag{17}$$

Copyright © 1999 John Wiley & Sons, Ltd.

where tr is the trace. In this work, the mapping $N_i(\mathbf{x})$ is given by bilinear nodal basis functions used routinely in the finite element method.

3. REPRESENTATION OF A MEMBRANE

The membrane is to be represented by individual material points which collectively define a membrane surface. Figure 2 illustrates the concept in two dimensions. Figure 2(a) shows a planar membrane contour and Figure 2(b) shows a material point discretization of the membrane. The material points are overlaid on the Eulerian mesh. A single layer of material points is used through the thickness of the membrane so the enforcement of constant traction through the thickness of the membrane is automatic.

Recall that the momentum equation is being solved at the nodes of the Eulerian mesh. Thus, the motion of the material points resulting from the solution of the momentum equation on the square Cartesian grid must be consistent with the forces in the membrane and the geometry of the membrane. This consistency is obtained by projecting the strains calculated for each material point [equation (12)] onto the local co-ordinate system of the membrane. This local system is defined in two dimensions by the normal and tangent vectors, $\mathbf{n}_p$ and $\mathbf{t}_p$ (Figure 3). The normals $\mathbf{n}_p$ are approximated to be perpendicular to a line connecting the two neighbouring material points.

With the tangent strain updated, the determination of stress for a material point with linear elastic material properties is straightforward. For example, if a uniaxial membrane or string is assumed, this projection results in a stress component tangent to the surface of the membrane, with all other components zero. The out of plane strains can be adjusted to give either a uniaxial

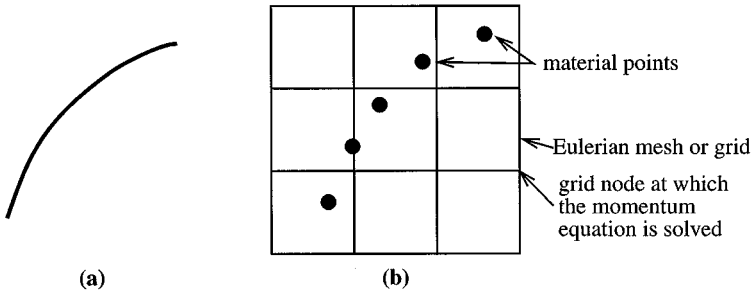


Figure 2. (a) Physical membrane contour; and (b) material point representation

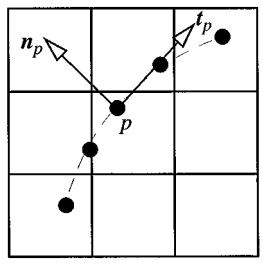


Figure 3. Local 2D co-ordinate system for a material point on the membrane

Copyright © 1999 John Wiley & Sons, Ltd.

stress or plane stress formulation. The stresses are projected back to the global co-ordinate system for the evaluation of internal forces which are defined at grid nodes.

Material points designated as membrane points must have assigned mass that takes into account the membrane thickness. The standard method for setting material point mass is element-based and has been used to initialize masses for material points representing solids without thin features.[3–5] The material points are assigned a fraction of the volume associated with each element. For example Figure 4(a) and (b) show a physical solid material and a MPM discretization. The shaded areas represent the volume, Ω_p, associated with each material point which, in this case, is one-quarter of the element volume because there are initially four material points per element. For two-dimensional calculations, element volume is computed as the cross-sectional area times unit width in the ignorable direction. In general, the membrane material points have no ordered relationship with the grid as shown in Figure 4(c) and (d). Thus, the masses of these material points should be initialized in a different manner.

If s is the total length of the membrane, ρ^0 is the mass per unit area of the membrane material, and N_{m} is the total number of membrane material points, the mass per unit width of a membrane material point can be set as $m_p = s\rho^0/N_{\mathrm{m}}$ for uniform spacing of material points. The quantity Δs, where $\Delta s = s/N_{\mathrm{m}}$, is used to associate a length of the membrane with each material point. Typically, Δs is a fraction of the smallest element dimension. Fractions of one-third to one-tenth have been used in simulations. It is essential that the membrane material points not become separated by more than one element or the membrane will effectively be 'broken.' The effective thickness of the membrane is accounted for in ρ^0. It is shown that membranes can be simulated using a grid size substantially different than the membrane thickness.

3.1. *One dimensional membranes*

3.1.1. Spring-mass simulation. The spring-mass problem [Figure 5(a)] is a simple problem with an analytical solution and is used as a test for the MPM. The problem consists of a rigid mass

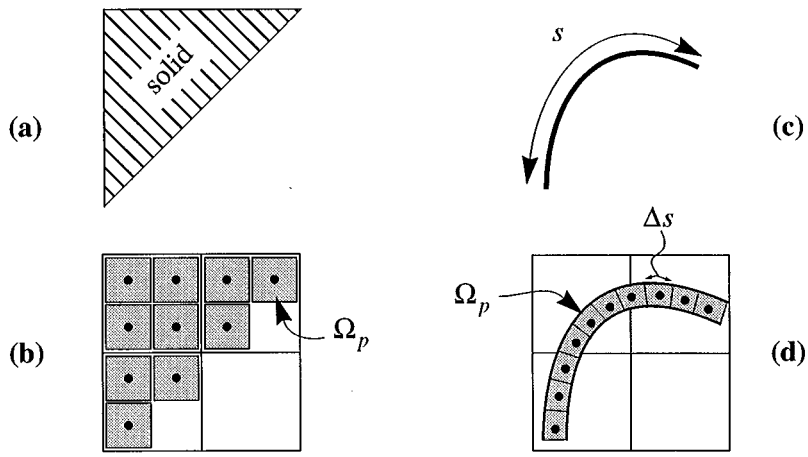


Figure 4. Physical representation of a solid (a) with its MPM representation; (b) and physical representation of a membrane; (c) with its MPM representation (d)

Copyright © 1999 John Wiley & Sons, Ltd.

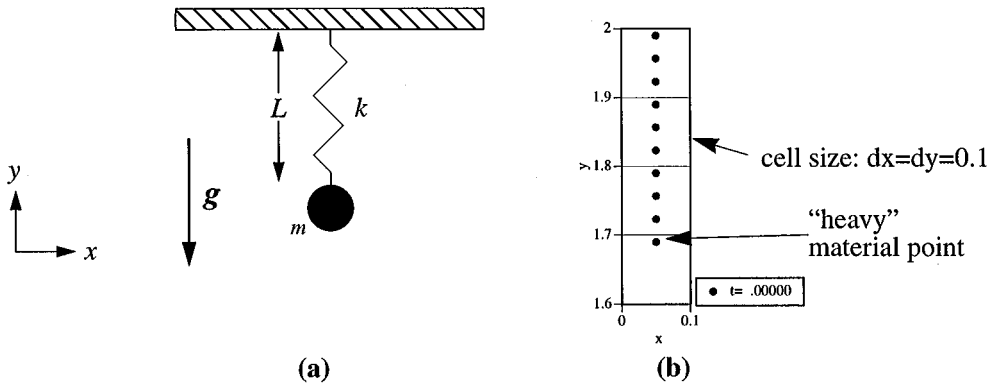


Figure 5. (a) Idealized spring-mass system; and (b) MPM discretization

attached to a massless elastic spring of unstretched length L and spring constant k which is connected to a stationary wall. All points lie in the x–y plane. A force due to gravity, g, acts on the system.

The spring-mass problem is discretized using ten membrane material points. The numerical model simulates the massless spring with material points having a small mass as compared to mass m. The initial positions of the material points and the computational grid for the simulation are shown in Figure 5(b).

The heavy material point in the simulation has a mass of 3·33 which is 10 000 times the mass of the other material points. The equivalent spring constant, k, of the system is AE/L where A is the cross-sectional area of the spring, E is Young's modulus, and L is the length of the string. For this simulation, $A = 0{\cdot}1$, $E = 1{\cdot}0\text{e}6$, $g = -250$, and $L = 0{\cdot}3$ which is the distance between the top and bottom material points.

To demonstrate the explicit time step limitation several time steps were used in this calculation. The wave speed, c, of the spring is

$$c = \sqrt{\frac{E}{\rho}} = \sqrt{\frac{1 \times 10^6}{0{\cdot}1}} = 3162 \tag{18}$$

and the critical time step, Δt_c, as defined by the Courant condition based on the element size is

$$\Delta t_c = \frac{\Delta x}{c} = \frac{0{\cdot}1}{3162} = 3{\cdot}16 \times 10^{-5} \tag{19}$$

where Δx is the smallest distance across a grid cell. If the time steps were governed by the material point spacing the Δx in equation (19) would be smaller, and thus, the critical time step would be smaller. It is interesting to note that this one-dimensional problem can be successfully run at $1{\cdot}6\Delta t_c$. At $1{\cdot}7\ \Delta t_c$ the calculation is unstable. A stability analysis is particularly complicated for this method given that one full time step involves both a Lagrangian step, where the material points move, and a regrid step. Also, material points may have irregular positions and they may change cells during a time step, which adds further complications. However, a one-dimensional analysis does show that for an extremely simple problem the method is stable at time steps up to

Copyright © 1999 John Wiley & Sons, Ltd.

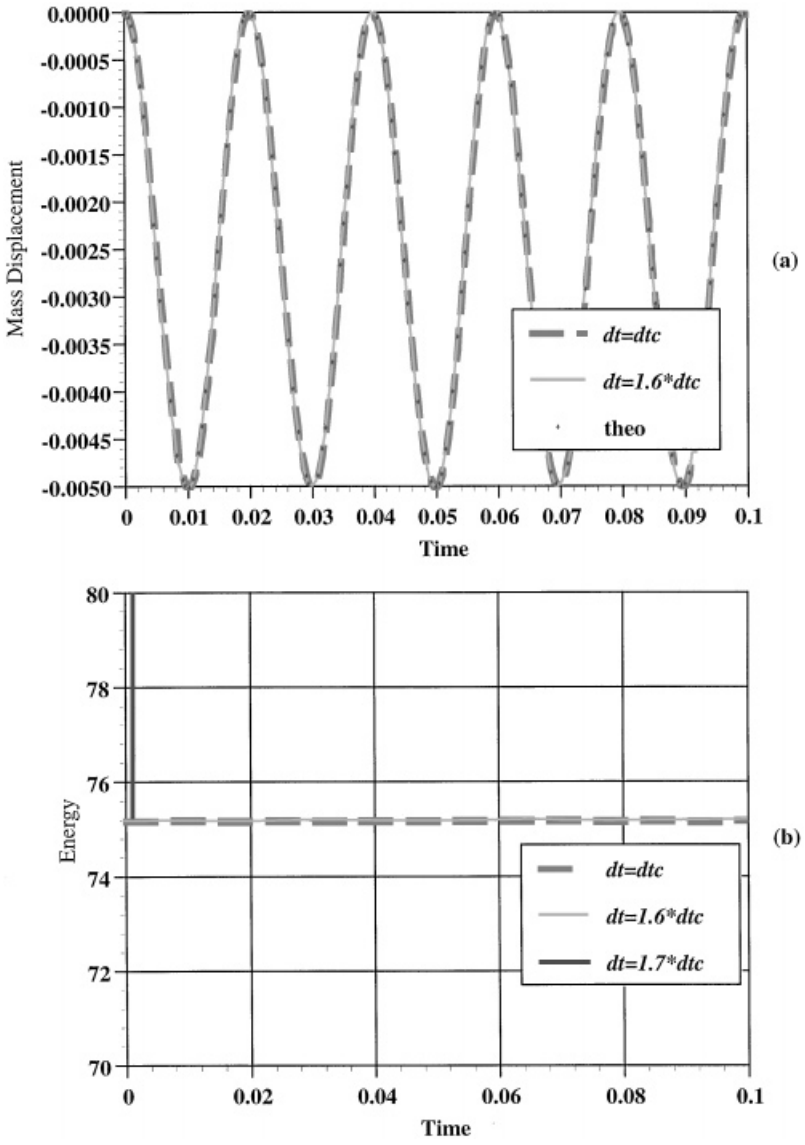


Figure 6. (a) Time history of mass displacement; and (b) energy history

$2\,\Delta t_c$ if there is one material point per element at the center of the element and is stable at $\Delta t = \Delta t_c$ if the material point is at the edge of the elements.[16]

Results of the MPM simulation and the theoretical results for mass deflection and energy are shown in Figure 6. Figure 6(a) provides displacement as a function of time while Figure 6(b) shows the corresponding total energy. The mass displacement compares favourably with the theoretical result using a time step of Δt_c and $1{\cdot}6\Delta t_c$. The energy history results are shown from

Copyright © 1999 John Wiley & Sons, Ltd.

simulations using time steps of Δt_c, $1{\cdot}6\Delta t_c$, and $1{\cdot}7\Delta t_c$. It is evident from the increasing energy shown in the plot that using a time step of $1{\cdot}7\Delta t_c$ causes an instability.

3.1.2. Pendulum simulation. The pendulum simulation is a test problem that demonstrates the ability of the method to resolve a membrane moving laterally through a Cartesian mesh. The MPM solutions are compared to a finite element solution, and numerical convergence and dissipation characteristics are illustrated.

The physical pendulum problem is illustrated in Figure 7. A mass, m, is attached to a string of length L and thickness t which is connected to a stationary wall with a frictionless ball and socket joint. All bodies lie in the x–y plane. The string is initially inclined at an angle θ_0 from vertical. A field force due to gravity, g, acts on the system. Table I summarizes the problem parameters where ρ is density, E is Young's modulus, and ν is Poisson's ratio. Since the string is not rigid, an analytical solution describing the motion of the mass m does not exist. Thus, the MPM results are compared to a numerically converged finite element solution.

Four simulations with different degrees of mesh refinement are presented. A simulation performed with an explicit finite element method (FEM) using bar elements is used for comparison purposes.

The initial position of the material points and the computational domain for each simulation are shown in Figure 8. The edge length of a square computational element, Δ, is given in the caption. In each simulation, the material point at the end of the string is given a mass equal to 3·3. Figures 9 and 10 illustrate the position of the material points at various times in the first ($\Delta = 0{\cdot}2$) and fourth ($\Delta = 0{\cdot}025$) simulations which are carried out for about one and one-half oscillations.

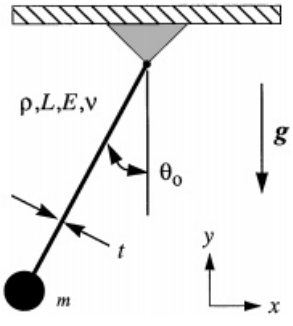


Figure 7. Pendulum problem set-up

Table I. Pendulum simulation parameters

Parameters	Value	Parameter	Value
L	0·73	E	1×10^6
m	3·3	ρ	1·0
g	20·0	ν	0·0
θ_0	19·8′	t	0·01

Copyright © 1999 John Wiley & Sons, Ltd.

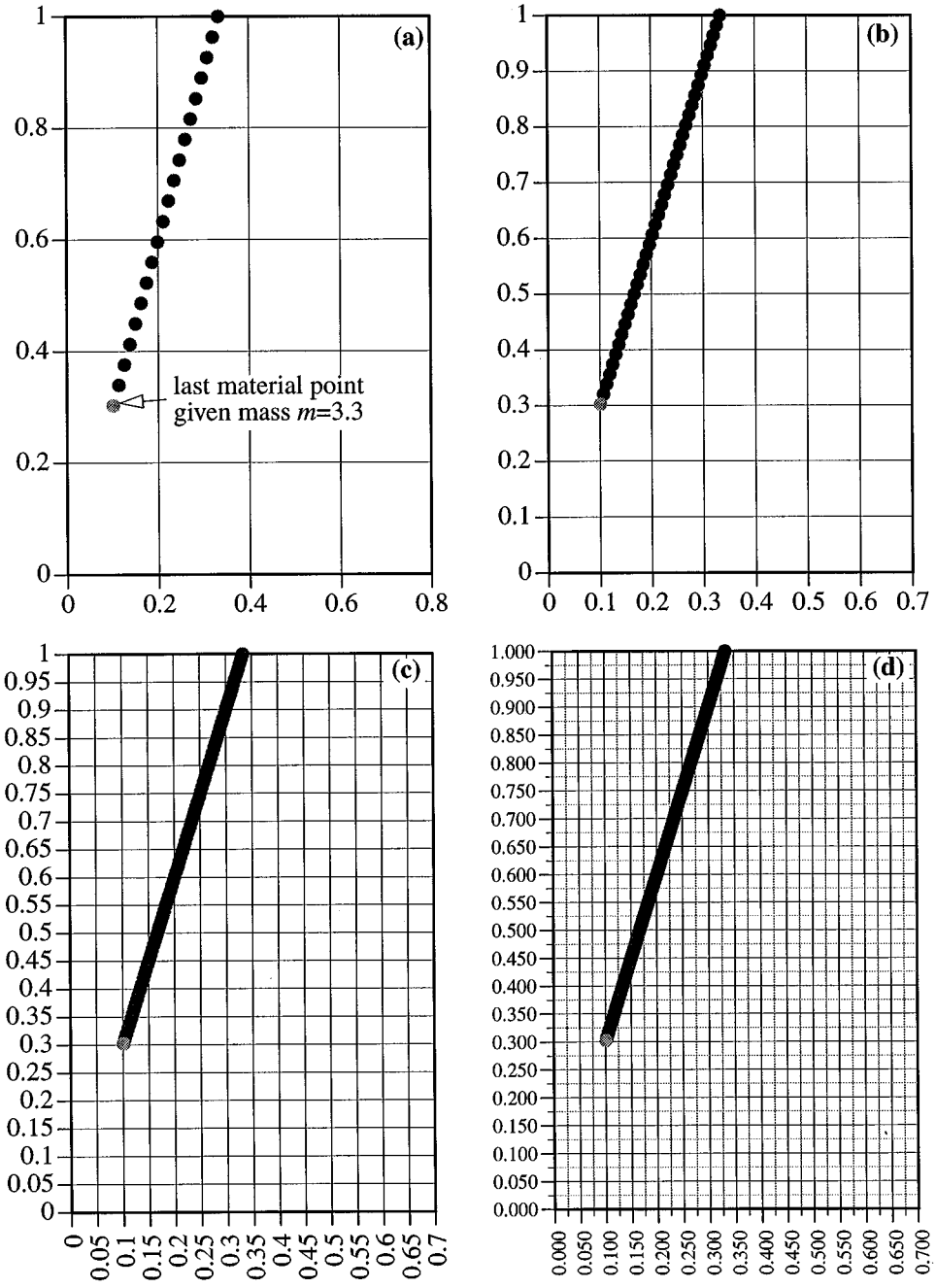


Figure 8. Pendulum problem simulations—initial conditions for $\Delta =$: (a) 0·2; (b) 0·1; (c) 0·05; and (d) 0·025

Copyright © 1999 John Wiley & Sons, Ltd.

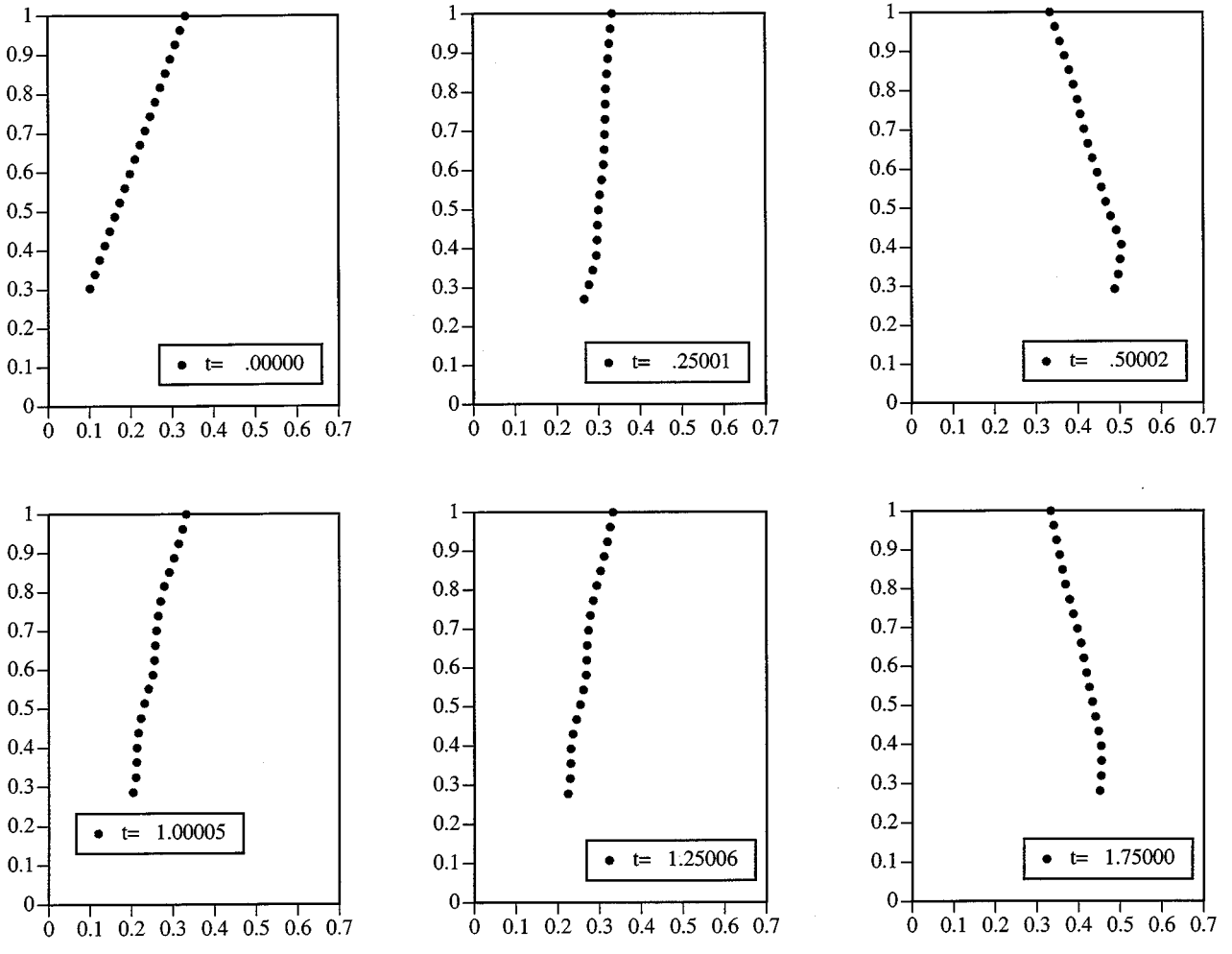


Figure 9. Pendulum position at various times—MPM simulation, $\Delta = 0{\cdot}2$

Copyright © 1999 John Wiley & Sons, Ltd.

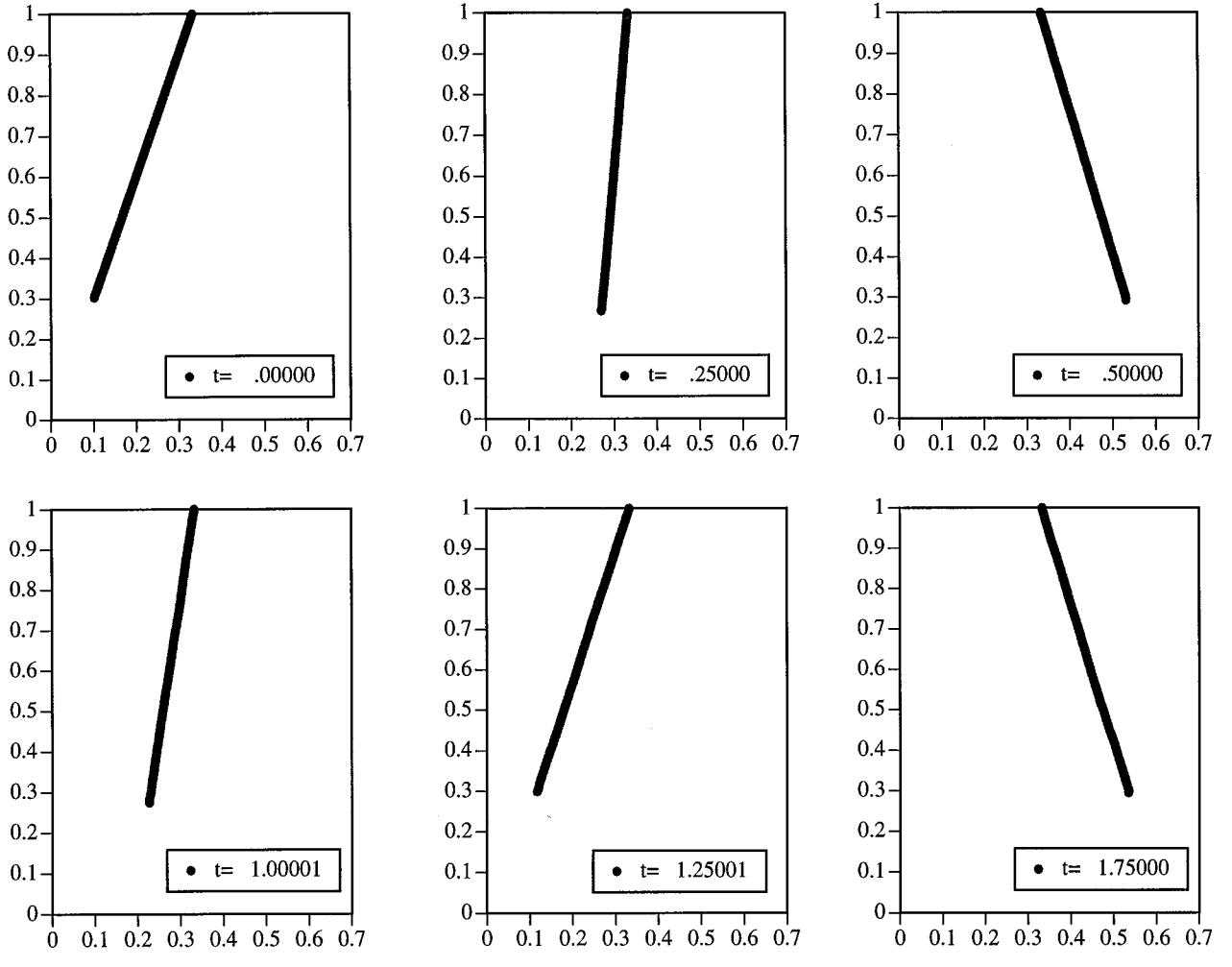


Figure 10. Pendulum position at various times—MPM simulation, $\Delta = 0{\cdot}025$

Copyright © 1999 John Wiley & Sons, Ltd.

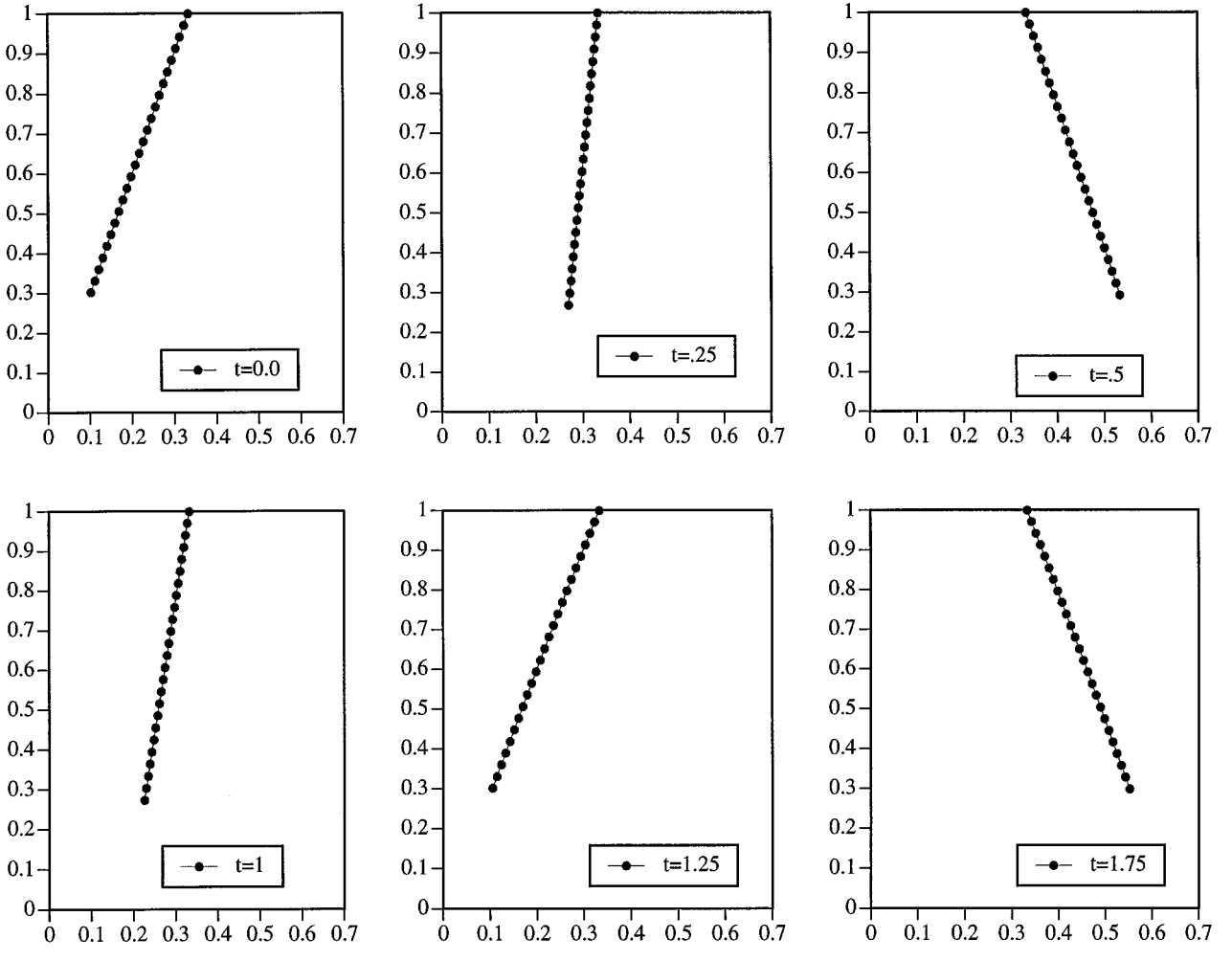


Figure 11. Pendulum position at various times—FEM simulation

Copyright © 1999 John Wiley & Sons, Ltd.

Figure 11 shows plots of the FEM simulation, where the element nodes are connected to emphasize this calculation is done with bar elements. The finite element solution does not change if a larger number of elements is used.

Figure 12 compares the time history of the angle theta obtained from the simulations. Theta is defined by the angle made by a straight line joining the first and last material points. The MPM simulations agree reasonably well with the explicit FEM simulation, and convergence toward the FEM simulation is observed. As the simulation is refined, the length of the string and direction of forces in the string are more accurately resolved on the computational grid. The material points stay aligned better as the simulation is refined. The reduced amplitude of oscillations (seen in Figure 12) in the MPM results when compared with the FEM results indicates numerical dissipation. However, the dissipation is decreasing with mesh refinement. Also, note that the membrane thickness, 0·01, is substantially different than the grid dimension which ranges from 0·2 to 0·025.

3.2. *String with slack*

An addition can be made to the formulation that enables a material point to accumulate compressive strain without causing a compressive stress. The idea is to simulate a wrinkle which can occur when a membrane or string is in compression without having to resolve the

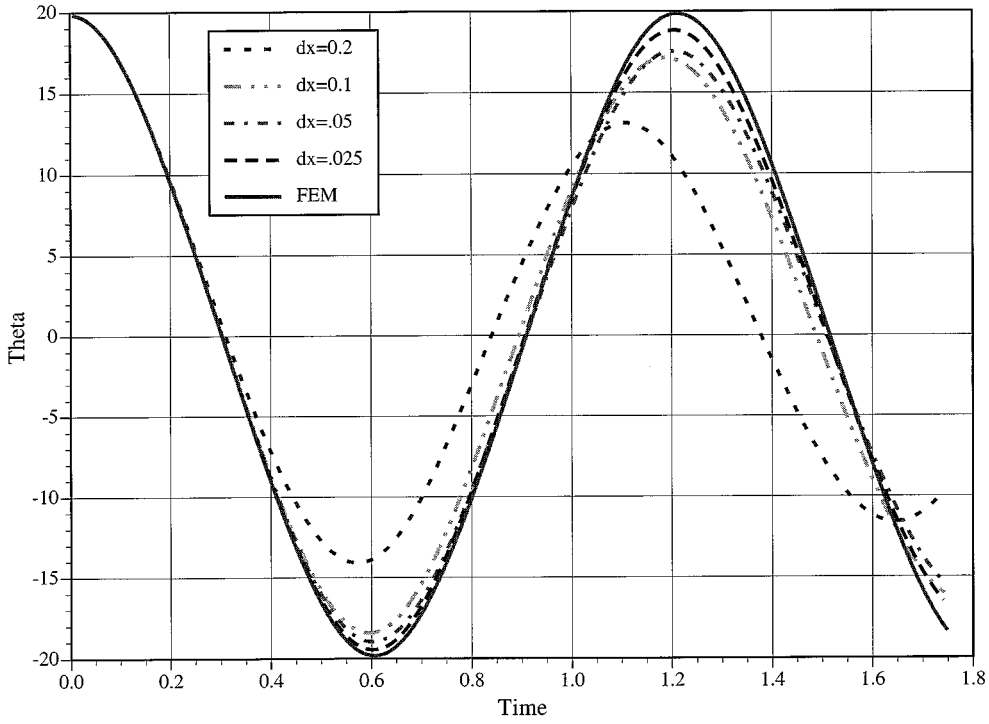


Figure 12. Comparison of angle theta for swinging pendulum simulations

Copyright © 1999 John Wiley & Sons, Ltd.

buckling/bending of the membrane, possibly with a fine mesh. This enables a more realistic membrane simulation at low cost. The addition to the method is to assign a Young's modulus of zero for compressive strains. In two dimensions a compressive strain is a negative tangent strain, ε_{t_p}. Figure 13 shows the effective Young's modulus $\mathbf{E}_{t_p}$ as a function of the tangent strain.

A simple example using a string-mass system is used to demonstrate the use of the one-way constitutive equation. The physical problem being simulated is illustrated in Figure 14(a) which shows a mass, $m = 3{\cdot}3$, attached to a string of stiffness $k = 3{\cdot}33 \times 10^5$ ($k = 0$ for compression). The MPM representation is shown in Figure 14(b) at time $t = 0$. Material points in the middle of the string are given an initial negative strain in the vertical direction. With the one-way constitutive

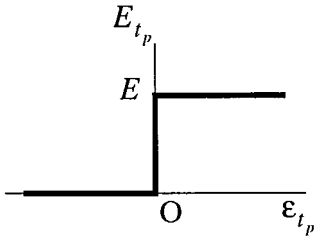


Figure 13. One-way stiffness to simulate wrinkles in a membrane

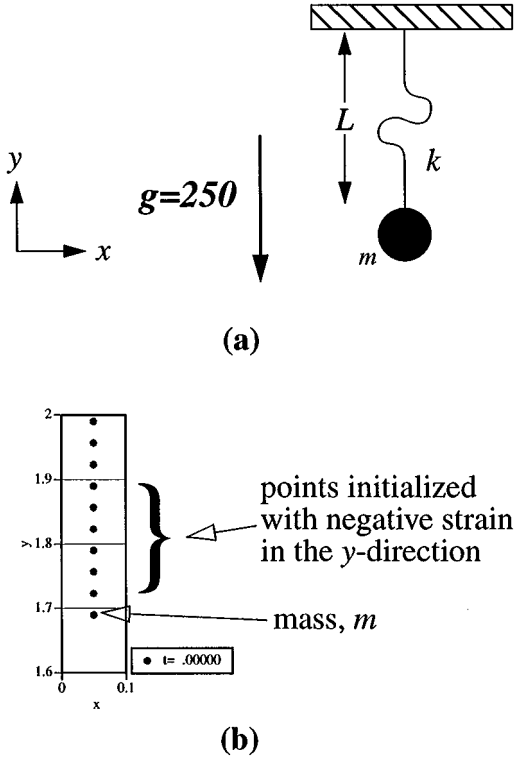


Figure 14. String-mass with initial slack: (a) physical; and (b) MPM representations

Copyright © 1999 John Wiley & Sons, Ltd.

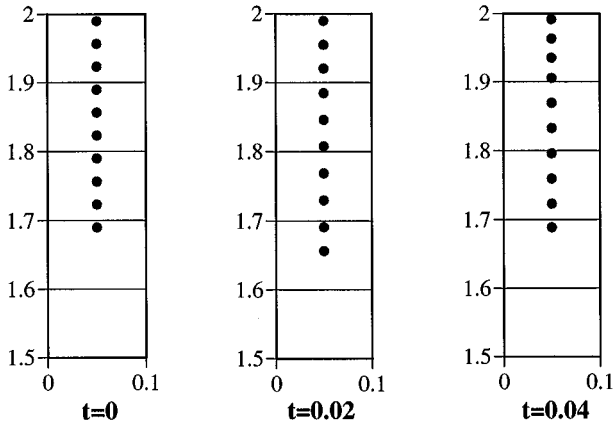


Figure 15. Material point positions at various times

model there is no force generated by the material points until the material point has a positive tangent strain (in tension).

The results are presented in Figures 15 and 16. Figure 15 shows the material point positions at various times. Figure 16(a) shows the displacement of the mass, and Figure 16(b) is the energy history. The plot of energy reveals the details of the simulation.

For $t < 0{\cdot}013$ the mass is in free fall, and the slack is being taken out of the material points initialized with negative strain (segment 1–2). At about $t = 0{\cdot}013$ the slack is taken up, the string pulls on the mass, and the mass begins to decelerate (segment 2–3). From here until about $t = 0{\cdot}023$ the string acts as a spring in tension (segment 2–4). At about $t = 0{\cdot}023$ the string is in compression and does not generate a resistance force. Thus, the mass begins free flight up and is decelerated by gravity (segment 4–5). At about $t = 0{\cdot}0375$ the mass reaches its initial position and has zero velocity. From here the cycle begins again. The slight slope in the total energy indicates a small amount of numerical dissipation. The sketches of Figure 16(c) qualitatively show the sequences of movement.

During development of the membrane formulation in the MPM framework, many different simulations were run as tests. In general it appeared that the results were slightly more physical when the one-way constitutive equation was used since qualitative comparisons showed that the material points seemed to collectively form a smoother contour.

4. IMPACT

4.1. Contact-release algorithm

Several sample problems involving contact show that two bodies sometimes stick to one another unphysically when they should separate. This occurs when material points from two bodies come close to each other. The basic MPM algorithm moves material points in a single-valued velocity field defined by element shape functions, equation (11); so, points that are close together move with similar velocity. The strain increment experienced by a material point arises

Copyright © 1999 John Wiley & Sons, Ltd.

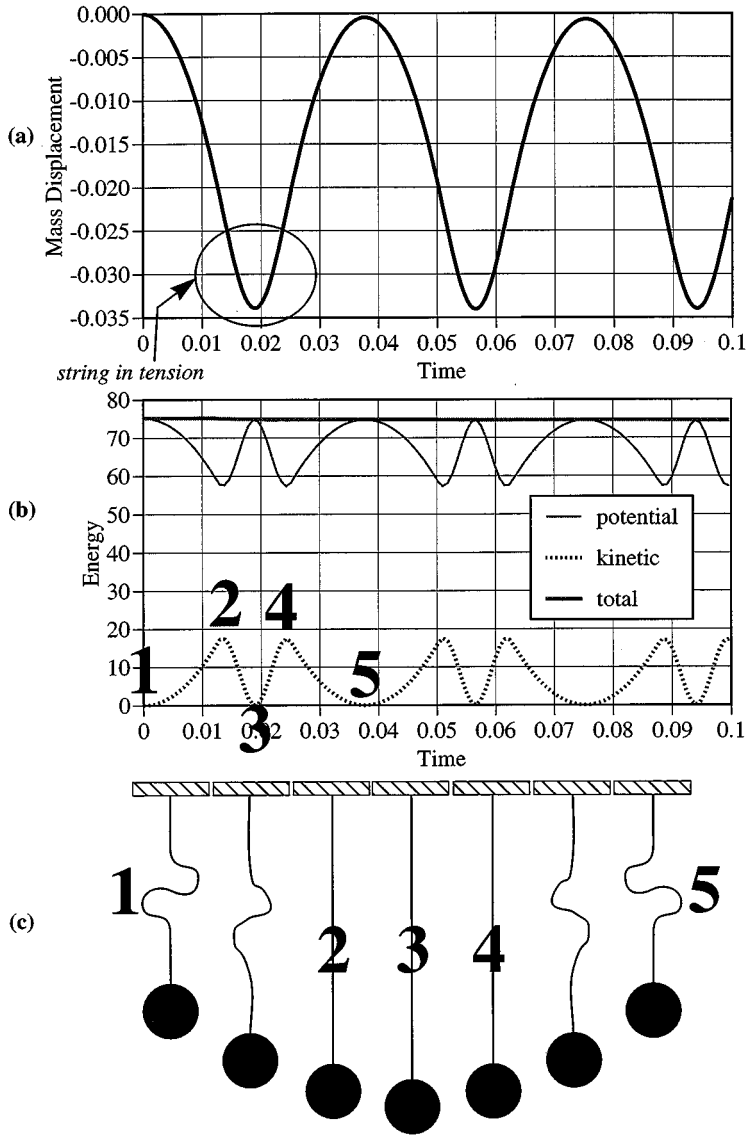


Figure 16. (a) Displacement as a function of time; (b) energy as a function of time; and (c) qualitative illustration of mass location and wrinkles at discrete times

from differencing the velocity field on the background mesh. If material points from two different bodies occupy the same element in the mesh, they will experience the same increment in strain. This means that even as bodies start to move apart, they will experience a strain if they are in the same element. Separation will not occur if the bodies do not have enough kinetic energy to overcome this strain energy. In reality, there should be no constraint on the bodies when they move apart.

Copyright © 1999 John Wiley & Sons, Ltd.

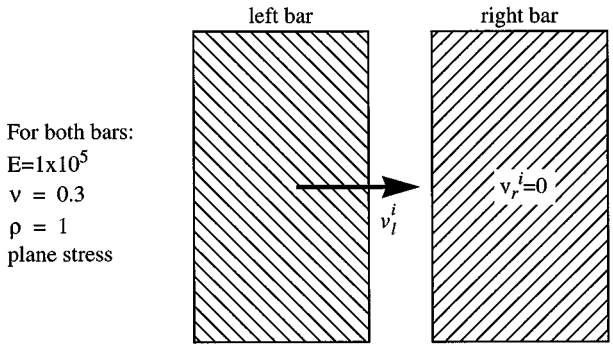


Figure 17. Initial conditions for two-bar-impact

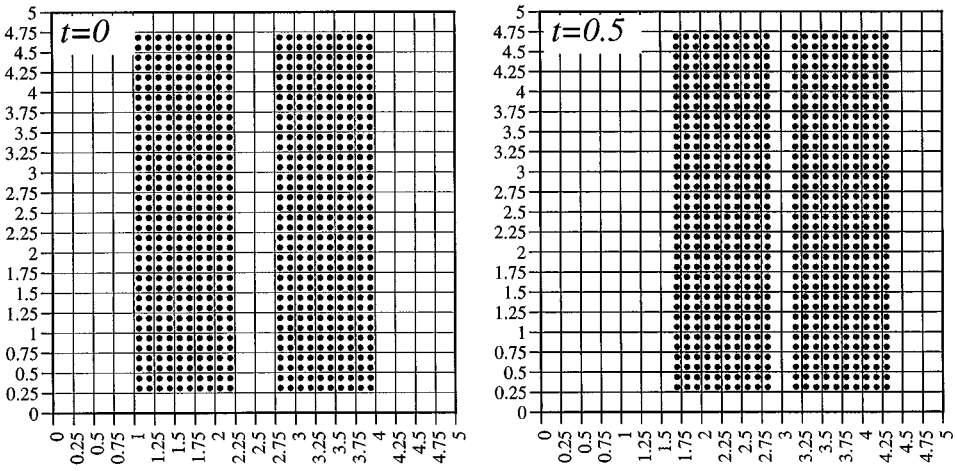
A simple example of two impacting bars demonstrates the sticking problem, and then an algorithm is described that allows the bars to release unimpeded from each other. The initial conditions for the two-bar impact problem are shown in Figure 17. The left bar has an initial nonzero velocity v_l^i, and the right bar is initially stationary $v_r^i, = 0$. For an ideal, elastic collision, after impact the velocity of the right and left bars should be $v_r^a = v_l^i$ and $v_l^a = 0$, respectively. The superscript a denotes the velocity after impact for the two bars. For deformable bars, the collision is not ideal and some kinetic energy is converted to strain energy associated with elastic vibrations so that v_r^a will be somewhat smaller in magnitude than v_l^i, and v_l^a will be nonzero.

Figure 18(a) shows the initial material point configuration and the configuration at $t = 0{\cdot}5$. It can be seen that the two bodies do not separate if the simulation is done with the basic MPM algorithm. This is more evident in Figure 18(b) which shows the center-of-mass velocity of each bar as a function of time. The velocities of the two bars oscillate as the bars move off to the right together. The vibrations of the bars sometimes cause loss of contact in the simulation. The portion of the velocity plots between $t = 0{\cdot}35$ and $t = 0{\cdot}425$ where the two bars have constant velocity occurs when the bars are separated by one element, and thus are not in contact. Contact is defined to be the case when two bodies share a grid node, i.e., when material points of two bodies both contribute to the solution at a grid node. This does not necessarily mean that material points from two bodies are in the same computational element.

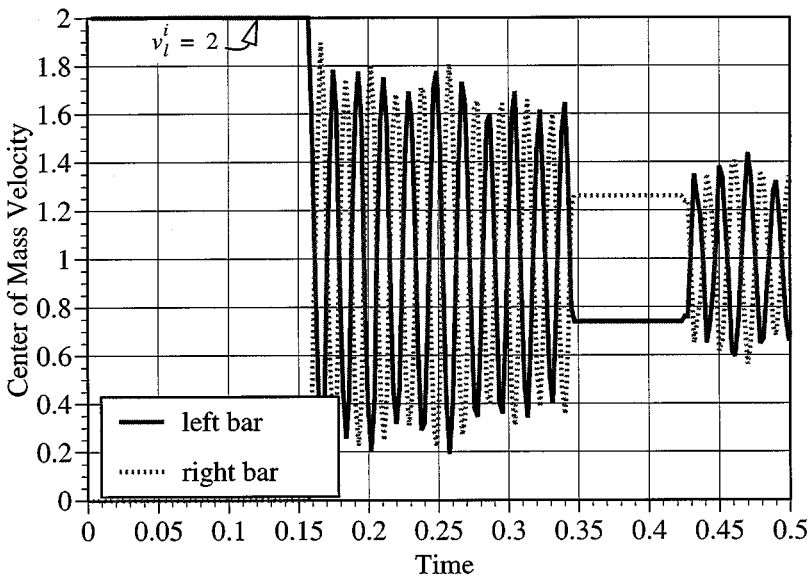
Another way to look at the problem is to consider the constitutive equation for material at the surface of a body. Free surfaces identify a state of plane stress. When two bodies come in contact, normal and shear components of the traction vector can be sustained and the constitutive equation must reflect this condition. When the bodies start to release, the normal component of traction becomes tensile. In fact the constitutive equation should be altered back to plane stress but in a general algorithm this step is never performed. The result is an 'apparent' sticking which is not a physical phenomenon. Possible solutions to relieve the artificial sticking consist of either altering the constitutive algorithm to plane stress or checking the nodal velocities to identify when release should occur and then altering the nodal forces. The latter approach is probably the most efficient and is described next.

A simple algorithm is proposed that allows bodies to release. The idea is that if the bodies are moving toward one another, as determined by some test criterion, the standard MPM method is

Copyright © 1999 John Wiley & Sons, Ltd.



(a)



(b)

Figure 18. Two-bar impact simulation with the MPM: (a) material point positions; and (b) center of mass velocities

used. That is, the material points are moved in the usual 'centre-of-mass' velocity field which automatically enforces the no-penetration condition. If the bodies are moving away from one another, they move in their own velocity field which allows the separation to occur unimpeded. This algorithm is efficient since it can be applied to all bodies with one sweep over the computational mesh without iteration. The details are given next.

Copyright © 1999 John Wiley & Sons, Ltd.

Recall from equation (6) that centre-of-mass velocities are obtained by solving and integrating the governing equation

$$\sum_{j=1}^{N_n} m_{ij}^k \mathbf{a}_j^k = \mathbf{f}_i^{\text{int},k} + \mathbf{f}_i^{\text{ext},k} \quad i = 1, \ldots, N_n \tag{20}$$

where $\mathbf{f}_i^{\text{int}}$ is the internal force at grid node i resulting from all material points even if they are from different bodies and the superscript k indicates the time level. New centre-of-mass velocities are obtained from integrating the centre-of-mass accelerations

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \Delta t \mathbf{a}_i^k \tag{21}$$

The grid node velocities at time level k are determined from material point velocities at time level k as

$$\sum_{j=1}^{N_n} m_{ij}^k \mathbf{v}_j^k = \sum_{p=1}^{p} m_p \mathbf{v}_p^k N_i(\mathbf{x}_p^k) \tag{22}$$

The mass weighting and the incorporation of all material points in this equation motivates the term 'centre-of-mass' velocity field. Equations (20)–(22) can also be written for a separate body I as

$$\sum_{j=1}^{N_n} m_{ij}^{(I)} \mathbf{a}_j^{(I)} = \mathbf{f}_i^{\text{int},(\text{I})} + \mathbf{f}_i^{\text{ext}} \quad i = 1, \ldots, N_n \tag{23}$$

where the contributions to the internal force are obtained from material points in body I. Equation (23) is integrated to give a velocity field on the grid due only to body I

$$\mathbf{v}_i^{k+1,(I)} = \mathbf{v}_i^{k,(I)} + \Delta t \mathbf{a}_i^{k,(I)} \tag{24}$$

where the velocity at the previous time step is determined from material points of body I

$$\sum_{j=1}^{N_n} m_{ij}^{k,(I)} \mathbf{v}_j^{k,(I)} = \sum_{p=1}^{P} m_p \mathbf{v}_p^{k,(I)} N_i(\mathbf{x}_p^{k,(I)}) \tag{25}$$

If the bodies are determined to be in contact, equations (20)–(22) are used, and if they are releasing from one another equations (23)–(25) are used.

The contact-release criterion is evaluated at each grid node that is shared by two bodies. The velocity of each body, as defined at grid nodes, is compared to the centre-of-mass velocity at corresponding grid nodes. Release is allowed only if the following criterion is satisfied for both bodies:

$$(\mathbf{v}_i^{k,(I)} - \mathbf{v}_i^{(k)}) \cdot \mathbf{n}_i^{(I)} > 0 \tag{26}$$

where $\mathbf{n}_i^{(I)}$ is the outward grid normal from body I, and i ranges from one to the number of shared grid nodes.

The method for calculating the grid-based normals in equation (26) uses the colour function approach described next. Each material point of a particular body is assigned a colour (number) unique to that body. Interpolation of the material point colour to the background grid defines the colour function. The normals to bodies are determined by taking the gradient of this colour function.

Copyright © 1999 John Wiley & Sons, Ltd.

Let $\chi_p^{(I)}$ be the colour density of a material point belonging to body I. To smooth the function, interpolate the material point colour to element centres using quadratic B-spline interpolation[15]

$$\chi_c^{(I)} = \sum_p \chi_p^{(I)} S_c^{(2)}(\mathbf{x}_p) \tag{27}$$

where each material point will contribute to nine element centres with the use of the weighting function $S_c^{(2)}$, $(\mathbf{x}_p)$, and $\chi_c^{(I)}$ is the element-centred colour function for body I. The normal for grid node i is calculated by mapping the element-centred colour function to grid nodes using the same gradient operator as in equation (13)

$$n_i^{(I)} = \sum_c G_{ci} \chi_c^{(I)} \tag{28}$$

An example of the normals calculated using equation (28) is shown in Figure 19. Due to the nine point stencil of the quadratic interpolating function more normals are calculated than would be used in the contact-release algorithm and, thus, only the pertinent normals are shown in the figure.

4.2. Two-bar impact with contact-release algorithm

The application of the contact-release algorithm to the two-bar impact problem shows significantly better results. Figure 20(a) shows the material point positions at $t = 0$ and 0·5. At $t = 0{\cdot}5$ the right bar has moved away from the left bar. This is evident in Figure 20(b) where the

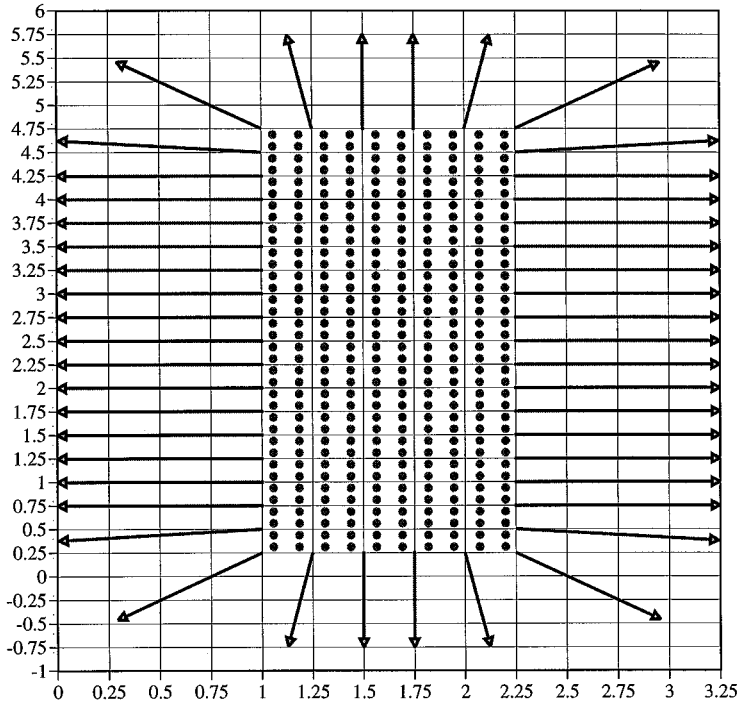


Figure 19. Grid normals for one bar of the two-bar impact simulation

Copyright © 1999 John Wiley & Sons, Ltd.

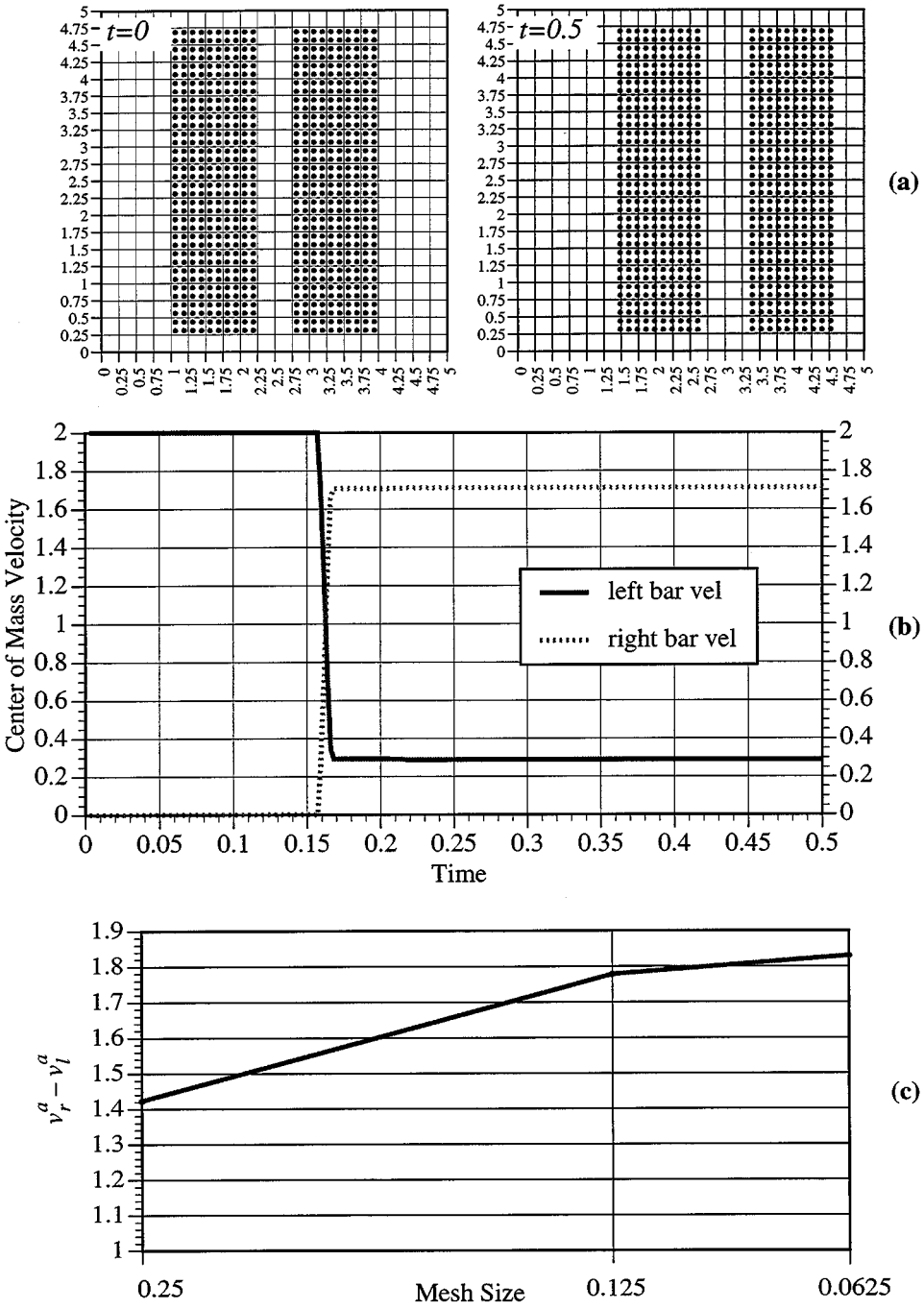


Figure 20. Two-bar impact simulation with the contact-release algorithm: (a) material point positions; (b) center of mass velocities; and (c) results with mesh refinement

Copyright © 1999 John Wiley & Sons, Ltd.

centre of mass velocity of the left bar decreases from 2 to about 0·3 and the right bar increases from 0 to about 1·7. Two more simulations were run with increasingly finer mesh sizes, and the velocity difference between the right and left bars increases and shows convergence. Figure 20(c) illustrates the velocity difference, $v_r^a - v_l^a$ after impact for the three cases considered.

4.3. Disks impacting a wire

A hypothetical problem of two disks impacting a stationary wire is used to demonstrate the membrane and the pinching of the membrane between the two disks. The contact release algorithm is used so that the disks separate from the wire. As mentioned earlier this problem is difficult for conventional updated Lagrangian FE methods. To obtain wire deformation a FE simulation must define contact between the disk and wire. If the disk/wire contact is used for the second disk as well, the solution will prove difficult when the wire is pinched due to excessive movement of the wire into and out of the disks as the contact algorithm is applied. The MPM avoids this problem altogether.

The problem set-up is shown in Figure 21. The initial disk velocities are $v_{Ix}^i = 1{\cdot}0$ and $v_{IIx}^i = -1{\cdot}1$. The initial velocity and position of disk II is set so that it impacts the wire at approximately the peak of the membrane deformation due to the impact of disk I. The constitutive model for the disks is plane stress, and that of the wire is uniaxial stress. The material properties used are listed in Table II.

Plots of the particle positions for various times (listed in the top left hand corner of each plot) are shown in Figure 22. A square mesh was used with dimension 0·25. Disk I moves into and

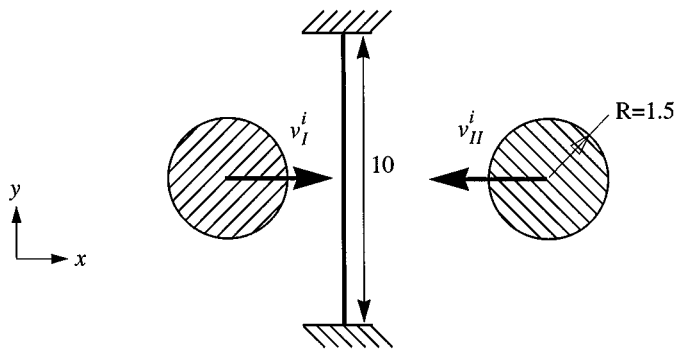


Figure 21. Two disks and wire

Table II. Parameters for the disk and wire simulation

Parameter	Disk value	Wire value
Density	1·0	0·5
Young's modulus	1×10^4	1×10^4
Poisson's ratio	0·3	0·0
Initial velocity	1·0 and $-1{\cdot}1$	0·0

Copyright © 1999 John Wiley & Sons, Ltd.

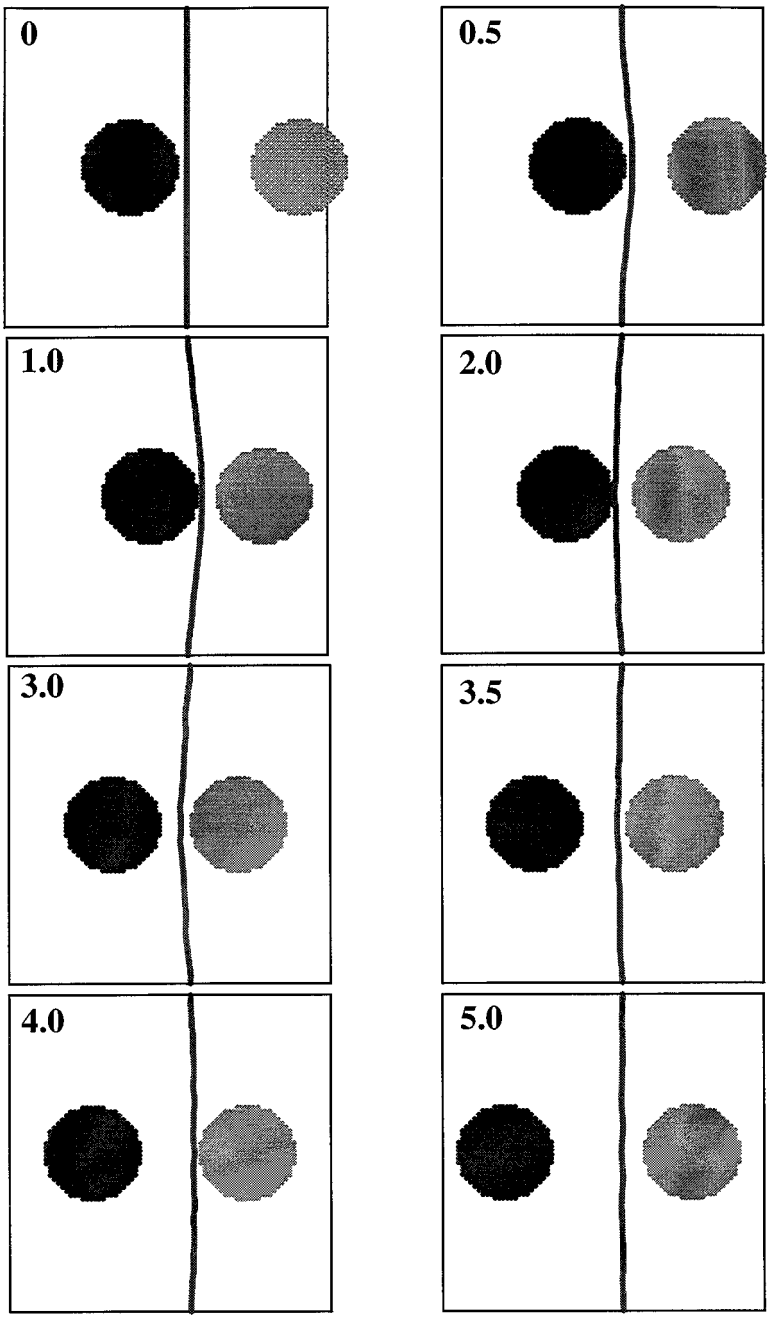


Figure 22. Material point positions at various times

Copyright © 1999 John Wiley & Sons, Ltd.

deforms the wire up until about $t = 1{\cdot}0$. This is when Disk II begins to pinch the wire and push Disk I and the membrane to the left. Finally, Disk I moves off to the left, and Disk II moves to the right of the wire as seen in the last four frames from $t = 3{\cdot}0$ to $5{\cdot}0$.

4.4. *Offset disks*

A variation to the problem described above is made where the two disks are offset vertically from one another and move normally toward the wire at the same velocity. Four position-plots at increasing times are shown in Figure 23. The disks pinch the wire and cause it to deform which induces tension in the wire. The no-slip contact causes the disks to rotate clockwise as seen in frame one. The arrows indicate the direction of rotation. When the motion of the disks is stopped, the tension in the wire is relieved by pushing the disks away as seen in frame three of Figure 23.

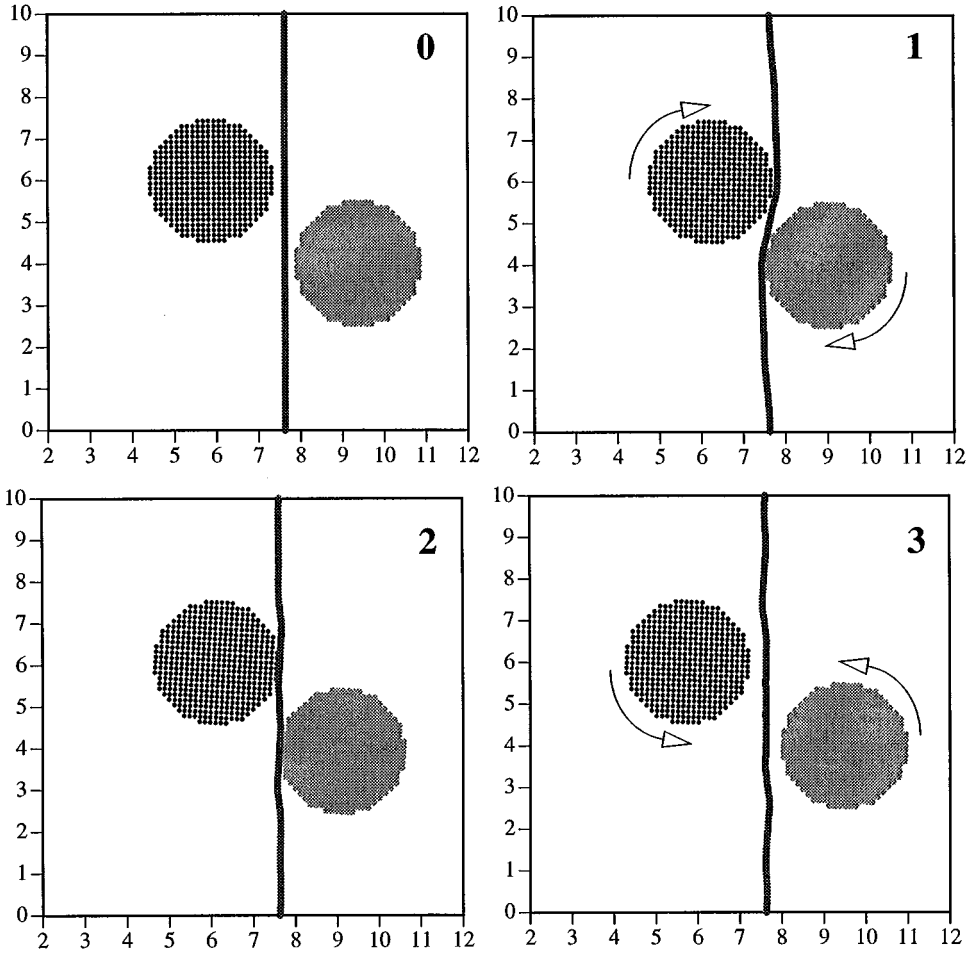


Figure 23. Material point positions for offset disk impact

Copyright © 1999 John Wiley & Sons, Ltd.

After release, the disks are rotating counterclockwise due to the torque induced by the no-slip contact.

5. CONCLUSION

The material point method uses Lagrangian material points and an Eulerian or spatial mesh to define the computational domain. The material points move through the Eulerian mesh, on which the momentum equation is solved. A review of the equations is given as they are developed for general solid materials. This paper presents the modifications necessary to simulate thin membranes. The modifications require that strains be projected onto the local normal-tangential co-ordinate system for each material point. A one-way constitutive equation can be used to approximate a wrinkle in a membrane without requiring fine resolution of the Eulerian mesh. The membrane formulation uses only a single material point through the membrane thickness. The time step constraint of the explicit formulation is based on the mesh size and not the spacing between material points. The simulation of the swinging pendulum demonstrates that adequate resolution of membrane forces can be achieved on a square Cartesian mesh. Numerical convergence is also observed. The simple example of a string and mass illustrates the way the one-way constitutive equation allows a wrinkle to be approximated without resolving the physical geometry of the wrinkle. The disks and wire simulation shows how the MPM allows pinching of a membrane which is generally difficult to handle with traditional finite element methods. These examples indicate how complex simulations can be performed with the relatively simple MPM algorithm. The new capability of simulating membranes has extended the generous suite of problems solvable with the MPM. Work is currently ongoing to combine the membrane formulation with a compressible fluids formulation in the MPM to simulate fluid-structure interaction.

ACKNOWLEDGEMENTS

The authors wish to thank J. U. Brackbill for useful discussions regarding the contact algorithm. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

REFERENCES

1. J. U. Brackbill and H. M. Ruppel, 'FLIP: a method for adoptively zoned, particle-in-cell calculations in two dimensions', *J. Comput. Phys.*, **65,** 314–343 (1986).
2. J. U. Brackbill, D. B. Kothe and H. M. Ruppel, 'FLIP: a low-dissipation, particle-in-cell method for fluid flow', *Comput. Phys. Commun.*, **48,** 25–38 (1988).
3. D. Sulsky, Z. Chen and H. L. Schreyer, 'A particle method for history-dependent materials', *Comput. Meth. Appl. Mech. Engng.*, **118,** 179–196 (1994).
4. D. Sulsky, S.-J. Zhou and H. L. Schreyer, 'Application of a particle-in-cell method to solid mechanics', *Comput. Phys. Commun.*, **87,** 136–252 (1995).
5. D. Sulsky and H. L. Schreyer, 'Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems', *Comput. Meth. Appl. Mech. Engng.*, **139,** 409–429 (1996).
6. H. Koivurova and A. Pramila, 'Nonlinear vibration of axially moving membrane by finite element method', *Comput. Mech.*, **20,** 573–581 (1997).
7. P. S. Han and M. D. Olson, 'Interactive analysis of wind-loaded pneumatic membrane structures', *Comput. Struct.*, **25**(5), 699–712 (1987).

Copyright © 1999 John Wiley & Sons, Ltd.

8. J. J. Nieboer *et al*., 'Airbag modeling techniques', Proc 34th Stapp Car Crash Conference, SAE Paper 902322 (1990).
9. K. Yamamoto, M. Okada and J. Kameyama, 'Flow of a viscous fluid past a flexible membrane at low Reynolds numbers', *Fluid Dyn. Res*., **9,** 289–299 (1992).
10. R. Smith and W. Shyy, 'Computation of unsteady laminar flow over a flexible two-dimensional membrane wing', *Phys. Fluids*, **7**(9), 2175–2184 (1995).
11. M. P. Rast, 'Simultaneous solution of the Navier–Stokes and elastic membrane equations by a finite element method', *Int. J. Numer. Meth. Fluids*, **19,** 1115–1135 (1994).
12. D. Burgess, D. Sulsky and J. U. Brackbill, 'Mass matrix formulation of the FLIP particle-in-cell method', *J. Comput. Phys*., **103,** 1–15 (1992).
13. C. S. Peskin, 'Numerical analysis of blood flow in the heart', *J. Comput. Mech*., **25,** 220–252 (1977).
14. C. S. Peskin and D. M. McQueen, *A General Method for the Computer Simulation of Biological Systems Interacting with Fluids*, Symposium-Society for Experimental Biology, UK, ISSN/ISBN 0081–1386, 1995.
15. C. DeBoor, A *Practical Guide to Splines*, Springer, New York, 1978.
16. A. R. York, 'The development of modifications to the material point method for the simulation of thin membranes, compressible fluids, and their interactions', *Ph.D. Dissertation*, The University of New Mexico, 1997.

Copyright © 1999 John Wiley & Sons, Ltd.