

The Development of a Three-Dimensional Material Point Method Computer Simulation Algorithm for Bullet Impact Studies

M.J. Connolly, E. Maldonado and M.W. Roth
Department of Physics
University of Northern Iowa
Cedar Falls, Iowa 50614-0150 USA

Received November 20, 2008

Accepted March 27, 2009

ABSTRACT

The two-dimensional Material Point Method (MPM) algorithm outlined by Chen and Brannon has been extended to three dimensions. The development of the code is discussed as well as applications for simulating bullet impact on biological and non-biological systems.

I. INTRODUCTION

The study of bullet impacts on various systems is a currently active research field whose results have important meaning for the health and safety of people. Due especially to the destructive nature of bullet insult, computer simulations prove to be very useful in understanding and predicting the behavior of systems involved in such impacts. It is of particular importance to study systems whose components are elastic and can fail. This involves the bullet and target being composed of either a finite element grid or actual particles. In order to study material deformation and failure, we chose the latter scenario.

One significant challenge in simulating impact with systems composed of particles is that internal forces need to be communicated throughout the object when it deforms. In order to achieve such communication it is normally thought that a collision algorithm between the particles constituting the system is required which, in three dimensions is quite time consuming. There exists, however a two-dimensional (2D) Material Point Method (MPM) algorithm evaluated by Chan and Brannon [1] which allows modeling of elastic behavior and failure with no contact algorithm, as discussed later. The purpose of this work is to develop a three-dimensional (3D) MPM

algorithm and to demonstrate its utility for impact simulations.

II. THE MATERIAL POINT METHOD

We began with the 2D Material Point Method (MPM) algorithm outlined and evaluated by Z. Chen and R. Brannon [1]. MPM is advantageous because it doesn't require a complex time consuming algorithm to simulate collisions or material failure. Instead, MPM maps the mass and momentum of groups of particles in the simulation to a background grid and utilizes the conservation equation of mass as well as conservation of momentum to ultimately advance the system through time. The use of the background grid also avoids entanglement associated with other contact – based algorithms.

To begin with, the initial conditions for the simulation include a collection of particles with initial positions $\{\bar{x}_p^0\}$, velocities

$\{\bar{v}_p^0\}$ and boundary conditions chosen so as to reflect the important physics of the actual system as much as possible. Next, the background grid is defined, where computational space is separated into a defined number of grid cells in which nodes are placed in each corner of each cell. The mass M_p of each particle p at time t is

mapped to the nodes i of the grid cells based on the shape function $N_i(\bar{x}_p^t)$, resulting in the mass m_i^t of node i at time t :

$$m_i^t = \sum_{p=1}^{Np} M_p N_i(\bar{x}_p^t). \quad (1)$$

The same is done for the momentum $(Mv)_p^t$ of the particles:

$$(m\vec{v})_i^t = \sum_{p=1}^{Np} (M\vec{v})_p^t N_i(\bar{x}_p^t). \quad (2)$$

The 2D program uses rectangular cells to divide up space, with for nodes – one at each vertex of the square. Shape functions then are used to provide weighted portions of a particle's mass and momentum to the grid nodes. Our shape functions $N_i(\bar{x}_p^t)$ are calculated in the following way, using eight vertices of a box. The coordinates of the box containing a particle p are normalized so that they take on extreme values of 0 or 1 at vertices of the box, as shown in Fig. 1. So if a particular box has a vertex at (x_j, y_j, z_j) and another vertex at $(x_j + \Delta x, y_j + \Delta y, z_j + \Delta z)$ across its diagonal, the normalized coordinates are

$$\begin{aligned} \zeta &= \frac{(x_p^t - x_j)}{\Delta x} \\ \eta &= \frac{(y_p^t - y_j)}{\Delta y} \\ \chi &= \frac{(z_p^t - z_j)}{\Delta z} \end{aligned} \quad (3)$$

and the eight shape functions used in the particle – to – node mappings are calculated from the normalized coordinates as shown in Table 1 so that they are equal to unity for a vertex (node) where the particle is at and is zero for a given node if the particle is at any other node. Once the mappings in equations (1) and (2) is completed then the forces at each node must be calculated. The internal force (f_i^t) arises from elastic deformation of the object and is calculated as

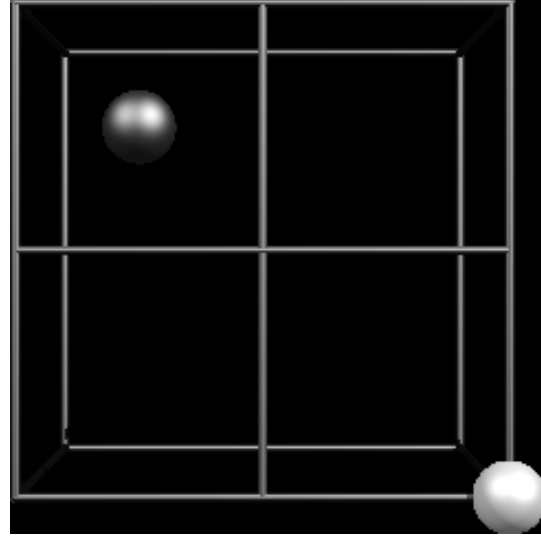


Figure 1. A section of the MPM grid. The gray particle is in the middle of its box and so its properties are portioned out to the box vertices with weighting functions but the white particle has the entirety of its properties assigned to the vertex it is sitting on.

$$(\tilde{f}_i^t)^{\text{int}} = - \sum_{p=1}^{Np} \mathbf{G}_i(\mathbf{x}_p^t) \bullet \mathbf{s}_p^t \frac{M_p}{\rho_p^t}. \quad (4)$$

The external force (if implemented) is then calculated at each node

$$\tilde{f}_i^{\text{ext}} = \tilde{c}_i^t + M_p \tilde{g} m_i^t \quad (5)$$

note that gravity can be explicitly taken into account in equation 5. Then the nodal

accelerations are obtained using $a_i^t = \frac{f_i^t}{m_i^t}$

where the total force is given by $f_i^t = (f_i^t)^{\text{int}} + (f_i^t)^{\text{ext}}$. The node momenta are then updated in time

$$(mv)_i^{t+\Delta t} = (mv)_i^t + f_i^t \Delta t, \quad (6)$$

and subsequently the kinematic variables associated with each particle p in the simulation are updated using node – to – particle mappings:

Node Coordinates	Shape Function
(0,0,0)	$(1-\xi)(1-\eta)(1-\chi)$
(1,0,0)	$\xi(1-\eta)(1-\chi)$
(1,1,0)	$\xi\eta(1-\chi)$
(0,1,0)	$(1-\xi)\eta(1-\chi)$
(0,0,1)	$(1-\xi)(1-\eta)\chi$
(1,0,1)	$\xi(1-\eta)\chi$
(1,1,1)	$\xi\eta\chi$
(0,1,1)	$(1-\xi)\eta\chi$

Table 1. Expressions for the eight shape functions used to map particle properties to each of the eight vertices of the grid cube in which it resides.

$$\bar{a}_p^t = \sum_{i=1}^{N_n} \frac{\bar{f}_i^t}{m_i^t} N_i(\bar{x}_p^t) \quad (7)$$

$$\bar{v}_p^t = \sum_{i=1}^{N_n} \frac{(m\bar{v})_i^{t+\Delta t}}{m_i^t} N_i(\bar{x}_p^t) \quad (8)$$

$$\bar{x}_p^{t+\Delta t} = \bar{x}_p^t + \bar{v}_p^{t+\Delta t} \Delta t \quad (9)$$

The new particle kinematic variables in eqs. (7-9) are mapped back onto the nodes to obtain updated nodal velocities.

Now because the simulated material(s) may have deformed in being updated, the elastic properties of the system must be taken into account. The rate of strain tensor \bar{e} is now calculated from the velocity gradients:

$$e_{ij} = \frac{1}{2} (\partial_j x_i' + \partial_i x_j').$$

Here the primed variables represent velocity components and, explicitly in 3D we have

$$\bar{e} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial v}{\partial y} + \frac{\partial v}{\partial x} \right) & \frac{\partial v}{\partial y} & \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\ \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial y} \right) & \frac{\partial w}{\partial z} \end{bmatrix} \quad (10)$$

Subsequently, the 3D elastic stress – strain relationship is utilized to calculate the updated stress tensor $\bar{s}^{t+\Delta t} = \bar{s}^t + \Delta \bar{s}$, where $\Delta s_{ij} = \sigma_{ij} = \sum_{ijkl} C_{ijkl} \varepsilon_{kl}$. Different

types of strain in 3D are visualized in Fig. 2 and stress is shown in Fig. 3. There are several special cases of the elastic tensor as well as strain tensor which will be discussed. The algorithm then returns to mapping particle properties to the grid (eq. 1) and is repeated until a desired time limit is

reached.

III. SPECIAL CASES OF THE ELASTICITY TENSOR

There form of the elastic tensor is related to the symmetry properties of the material being modeled [3]. In the most general case, the material we are dealing with may have no planes of symmetry at all. In that case the elastic tensor requires 21 unique elements to describe the elastic

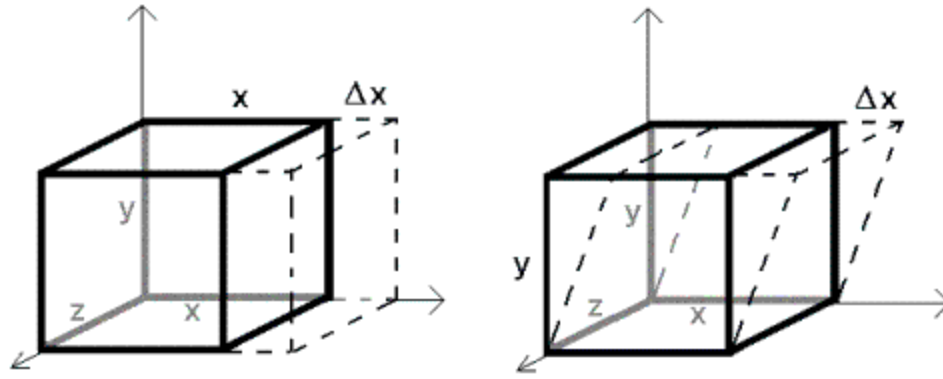


Figure2. Different types of material deformation. Normal compression or extension (left) results in a volume change for the material and is related to diagonal elements of the strain tensor; shear strain (right) conserves volume and relates to off-diagonal elements [2].

response of the material. The tensor is necessarily symmetric and so the lower diagonal block is omitted but implied:

$$\tilde{C} = \begin{bmatrix} A & B & C & D & E & F \\ & G & H & I & J & K \\ & & L & M & N & O \\ & & & P & Q & R \\ & & & & S & T \\ & & & & & U \end{bmatrix}. \quad (11)$$

Now suppose the material has one plane of symmetry. It is termed monoclinic and the number of elements of the elastic tensor needed are now only 13. This arises from the fact that the material properties do not change upon a mirror reflection about the (x,y) plane so the stresses are identical for the strains $\tilde{\epsilon}$ and $\tilde{\epsilon}' = \tilde{L}^T \tilde{\epsilon} L$, where

$$\tilde{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

The result is

$$\tilde{C} = \begin{bmatrix} A & B & C & 0 & 0 & F \\ & G & H & 0 & 0 & K \\ & & L & 0 & 0 & O \\ & & & P & Q & 0 \\ & & & & S & 0 \\ & & & & & U \end{bmatrix}. \quad (12)$$

An orthotropic material has 3 planes of symmetry and then the elastic tensor is further reduced to nine elements:

$$\tilde{C} = \begin{bmatrix} A & B & C & 0 & 0 & 0 \\ & G & H & 0 & 0 & 0 \\ & & L & 0 & 0 & 0 \\ & & & P & 0 & 0 \\ & & & & S & 0 \\ & & & & & U \end{bmatrix}. \quad (13)$$

Now further assume that the material has 3 planes of symmetry and one is isotropic. Then only five independent constants are needed

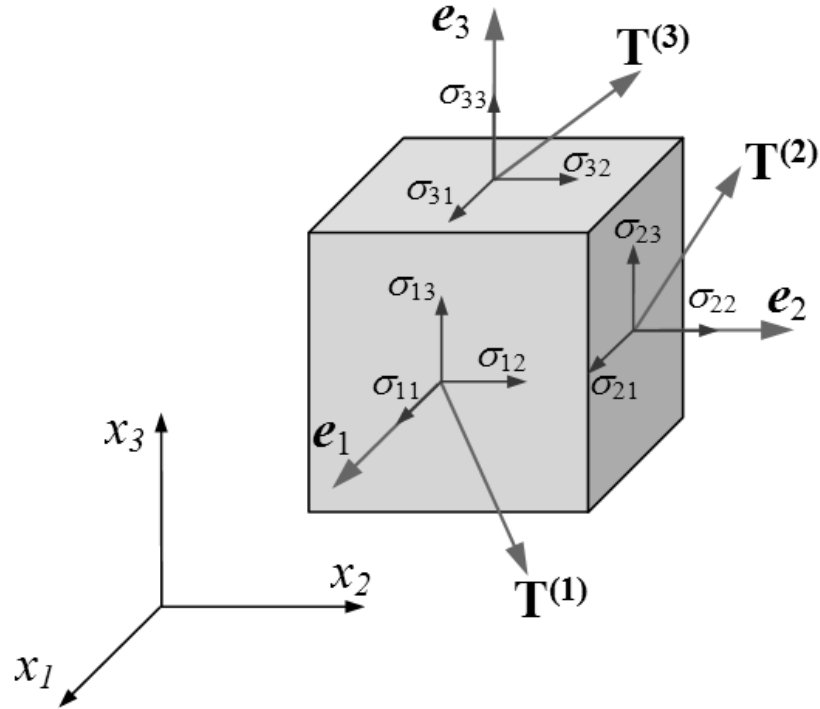


Figure 3. Visualization of material stress. [3] The \mathbf{e}_i are unit outward normals to the cube of material and the elements of the stress tensor σ_{ij} are the stresses in direction j acting on the face of the cube spanned by the tangent vectots $T^{(j)}$ normal to the unit vectors \mathbf{e}_i with coordinate $x_i = \text{constant}$.

$$\tilde{C} = \begin{bmatrix} A & B & C & 0 & 0 & 0 \\ & A & C & 0 & 0 & 0 \\ & & L & 0 & 0 & 0 \\ & & & P & 0 & 0 \\ & & & & P & 0 \\ & & & & & (A-B) \end{bmatrix}. \quad (14)$$

Now suppose that the material is entirely isotropic. Then the number of constants reduces to two:

$$\tilde{C} = \begin{bmatrix} A & B & B & 0 & 0 & 0 \\ & A & B & 0 & 0 & 0 \\ & & A & 0 & 0 & 0 \\ & & & (A-B) & 0 & 0 \\ & & & & (A-B) & 0 \\ & & & & & (A-B) \end{bmatrix}. \quad (15)$$

In our simulations we assume isotropic media and so the elastic tensor takes for form of equation (15) and the stress-strain

relationship can be expressed explicitly in terms of relevant material properties:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{bmatrix}. \quad (16)$$

Here E is the elastic modulus and ν is Poisson's ratio. In the case where a fluid is being modeled, shear forces are completely absent and the elastic tensor is written as

$$\vec{C} = \begin{bmatrix} A & 0 & 0 & 0 & 0 & 0 \\ & A & 0 & 0 & 0 & 0 \\ & & A & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix}. \quad (17)$$

SPECIAL CASES OF STRAIN

There are also times when the geometry of structures being considered has a coordinate whose behavior can be ignored [4]. This is usually the case, for example, when the dimensions of an object on one dimension are much larger than in two others. The results is plane strain, where the strain tensor takes the form

$$\vec{\epsilon} = \begin{bmatrix} A & B & 0 \\ C & D & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (18)$$

and the corresponding stress tensor is written as

$$\vec{\sigma} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & e \end{bmatrix}. \quad (19)$$

For certain symmetries there may be antiplane strain where

$$\vec{\epsilon} = \begin{bmatrix} 0 & 0 & c \\ 0 & 0 & f \\ g & h & 0 \end{bmatrix}. \quad (20)$$

Note for antiplane strain that the response of the material is purely shear.

SIMULATIONS

The 3D simulations developed here were validated by monitoring total mechanical energy conservation of two colliding elastic spheres. As shown in Figure 4, the algorithm conserves total mechanical energy and nicely describes the collision quite well. The first major application of the 3D MPM method is bullet impact on a muscle / bone composite structure. Figure 5 shows a preliminary simulation where a projectile insults a simulated muscle / bone complex.

The MPM method has also been used to simulate bullet impact on

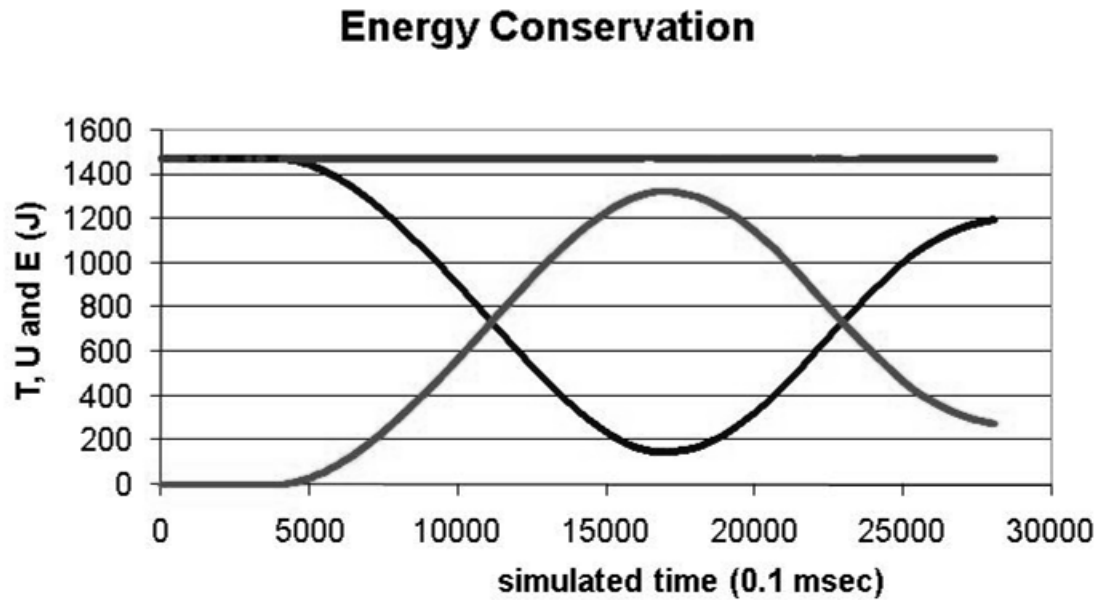


Figure 4. Kinetic energy (sinusoidal curve starting at upper left), elastic potential energy (sinusoidal curve starting at lower left) and total mechanical energy (horizontal line at top) for the two-sphere collision validation run.

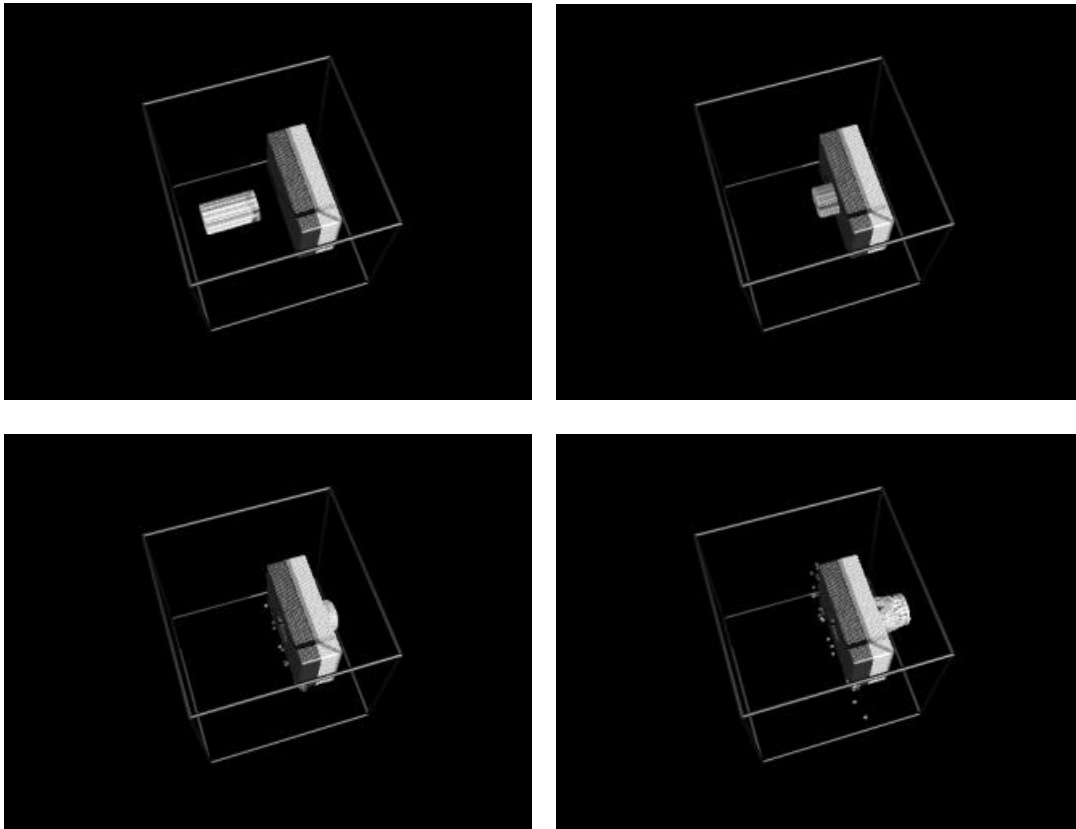


Figure 5. Sequence is top left to bottom right. Snapshots from a preliminary muscle/bone impact simulation. The bullet enters at left; muscle tissue is dark gray and bone is white; all materials have different elastic properties.

Kevlar Breakpoint Variation Series

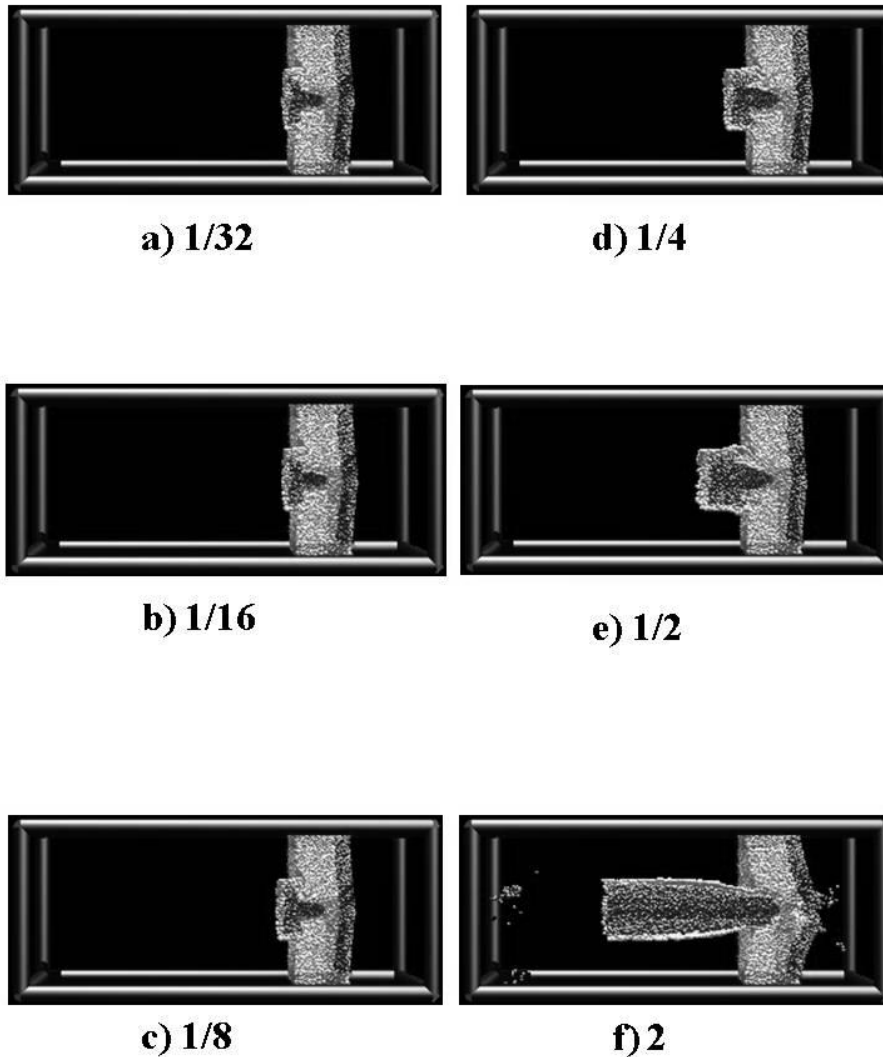


Figure 6. Bullet impact on ceramic / Kevlar composite vest for various fractions of material breaking points.

ceramic/Kevlar composite vests; a typical run is shown below where the sensitivity of the results to the materials' elastic breaking points is examined.

REFERENCES

1. Z Chen, RM Brannon, Sandia National Laboratories (SAND2002-0482), 2002
2. *Mathematics of Strain and Strain Rate*, edited by Asbjørn Støylen, Norwegian University of Science and Technology: <http://folk.ntnu.no/stoylen/strainrate/mathematics/>
3. Infinitesimal Strain Theory, Wikipedia article: http://en.wikipedia.org/wiki/Strain_tensor
4. High Strength Composites 1998-2005 by Pichai Rusmee, webpage University of Utah, Department of Mechanical Engineering: <http://www.mech.utah.edu/~rusmeeha/labNotes/composites.html>