

# Modeling Liquids and Solids using Thermal Particles

David Tonnesen

Department of Computer Science  
University of Toronto  
10 King's College Road  
Toronto, Canada, M5S 1A4

## Abstract

This paper presents a new method for modeling fluids and solids for the purposes of computer graphics and animation. The model, based on molecular and Newtonian dynamics, describes the changes in geometry and movement of elemental volumes as a consequence of thermal energy and external forces. At "hot" temperatures objects exhibit fluid-like behavior with rapidly varying geometry. At "cold" temperatures objects behave like solids, retaining a stable shape but capable of deformation given sufficient external forces. To accommodate the continual changes in topology, we use particles to represent elemental volumes with potential energies defined between pairs of particles. The shape of the potential energy function is varied as a function of temperature. The objects simulated by these particle systems exhibit varying degrees of rigidity. We present a physically-based animation in a simulated world where a particle based object changes thermal states while interacting with external forces and obstacles.

**Keywords:** modeling, animation, deformation, heat, temperature, particle systems, dynamics, simulation.

## 1 Introduction

Both fluids and deformable solids have been modeled separately for computer graphics. For solids, physically based modeling has been used to model a wide variety of properties [19, 20, 21]. Elastic and inelastic deformations including viscoelasticity, plasticity, and fracturing have been modeled using connected systems of springs, damping elements, and slip units. Water has been modeled for a variety of special cases: As ocean waves using trigonometric functions [12, 4]; as a shallow water height field [10]; and visually as a waterfall using thousands of non-interacting point masses [17].

A two-pronged approach has been described for modeling deformable objects in both the solid and fluid states [21]. In the solid state an object is composed of hexahedral assemblies of point masses, springs, and conductive heat elements. In the fluid state it is represented by a molecular dynamics model using point masses with long

range attraction forces and short range repulsion forces. Going from the solid to the liquid model is simple, but it is unclear how one would regain the structure inherent in the solid model upon freezing the liquid.

Another approach presents particle based models for fluids, powders, and gelatinous solids [11]. In the solid state particles interact with a limited range attractive force, a repulsive force, and drag. For powders and fluids the particles interact with a short range repulsive force and a drag. According to molecular dynamic models physicists have shown that gases can be modeled using only repulsive forces, but for liquids both repulsive and attractive forces are needed unless there are external forces holding the material together [1].

We construct a single temperature dependent model using interacting particles based on molecular dynamics. Our particles interact according to pairwise potential energy functions which create short range repulsive forces and long range attractive forces. We have created a simple model which simulates both the liquid and solid states by varying the shape of the potential energy curve as a function of temperature.

The remainder of the paper is organized as follows. In Section 2 we review the basic models of heat and matter from the molecular dynamicist's view. In section 3 we discuss heat transfer at the macroscopic level. In Section 4 we draw on ideas presented in Sections 2 and 3 to design our temperature dependent model. Section 5 discusses the addition of external forces. In Section 6 we describe our discrete time integration procedure. Section 7 presents a simulation using the model. Section 8 summarizes and discusses possible extensions.

## 2 Molecular Dynamics

At the microscopic level there are a variety of forces acting on molecules. The attractive or binding components of intermolecular forces are valence and ionic or coulombic bonds, metallic forces, van der Waals (electrostatic and dispersion) forces. Repulsive forces result between molecules when full shells of electrons are brought close together. The resulting *sum* of these forces has been shown to be *repulsive* for small separations and *attractive* at larger separations.

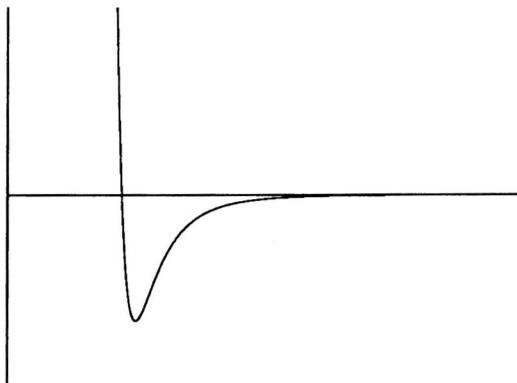


Figure 1: Lennard-Jones 12-6 potential energy as a function of separation.

The concept of a pairwise intermolecular potential energy function has proven valuable in describing intermolecular forces quantitatively [1, 3, 18, 24]. At short ranges a potential corresponding to the extremely strong repulsive force exists due to the Pauli exclusion principle. Usually this potential energy function is represented as an exponential expression  $\phi_R(r) \propto r^{-n}$ , where  $r$  is the distance between the two particles. The long range potential functions which correspond to attractive forces, can be treated together as a single expression,  $\phi_A(r) \propto r^{-6}$ , where  $A$  and  $B$  are constants. The sum of the two energies gives a resultant energy between two molecules,

$$\phi(r) = \frac{B}{r^n} - \frac{A}{r^6}. \quad (1)$$

Lennard-Jones and Devonshire suggested the 12-6 energy function, where  $n = 12$  [24]. The angular dependence of interactions between molecules has been neglected by this formulation, thus it falls short of accurately modeling highly polarized molecules.

Figure 1 shows the 12-6 potential energy function as a function of distance. When two molecules are in equilibrium the potential energy between them is minimal. This distance is defined as the equilibrium separation  $r_o$ . The energy required to separate two such molecules is the *dissociation energy*  $\phi(r_o) = \epsilon$ . To increase the distance  $r$  between two molecules by a small amount  $dr$ , the work that would have to be performed is  $f(r)dr$  which can be equated to a loss in potential energy, that is

$$f(r)dr = -d\phi(r). \quad (2)$$

This is more commonly written as

$$f(r) = -\frac{d\phi(r)}{dr}. \quad (3)$$

It is usual to assume that the potential energy of the system is the sum of the potential energies over all pairs of molecules. Thus the inter-particle forces are the sum of all the pairwise forces.

A generalization of the 12-6 function is the Lennard-Jones bi-reciprocal function for  $n > m > 3$ ,

$$\phi(r) = \frac{-\epsilon}{n-m} \left( m \left( \frac{r_o}{r} \right)^n - n \left( \frac{r_o}{r} \right)^m \right). \quad (4)$$

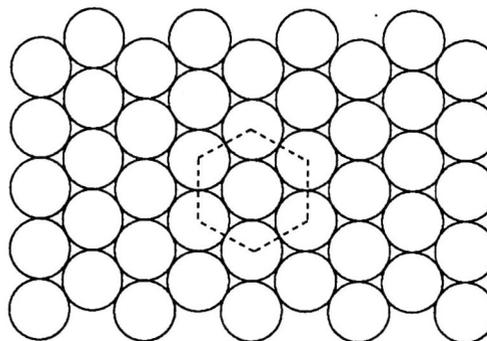


Figure 2: Hexagonal packing.

These functions have been used extensively in the literature for modeling spherical and quasi-spherical molecules, nonpolar molecules and weak polar molecules [18].

Liquids are qualitatively different from crystalline solids and the difference is a matter of geometry [1]. When external forces are insignificant, molecules arrange into closely packed structures to minimize their total energy. For spherically symmetrical potential energy functions in 2-D, the molecules arrange into hexagonal orderings as shown in figure 2. In 3-D the molecules arrange into hexagonal ordered 2-D layers. Solids are highly ordered while liquids tend to random ordering and gases have no ordering.

Heat in a system of molecules is the vibration or movement of individual molecules. The extent to which the intermolecular potential energy constrains the movement of the molecules depends on the kinetic energy (heat) of the molecules. At low temperatures the mean potential energy is greater than the mean kinetic energy, and the material is ordered as a solid. At high temperatures the mean kinetic energy is much greater than the mean potential energy and the material is in total disorder as a gas. In the liquid state the molecules are sufficiently close together for there to be local ordering, but the kinetic energy is large enough to prevent long-range ordering.

When discussing the movements of large numbers of molecules it is important to differentiate between the average speed and the average velocity of the molecules. In the cartesian coordinate reference frame, velocity is a vector in 3-space and speed is the magnitude of the vector. Kinetic energy of a system is based on the average speed of the molecules while rigid body motion is based on the average velocity of the molecules. Given a stationary object the average velocity of the molecules must be zero otherwise the object as a whole would be moving in the direction of the average velocity.

### 3 Heat Transfer

At the macroscopic level the heat  $\psi$  in a body  $A$  is given by integrating over the volume,

$$\psi = \iiint_A \rho \sigma \theta \, dx \, dy \, dz \quad (5)$$

where  $\rho(x, y, z)$  is the mass density of the body,  $\sigma(x, y, z)$  is the specific heat, and  $\theta(x, y, z)$  is the temperature.

The amount of heat leaving a body per unit time is given by,

$$\iint\int_A \nabla \cdot (K \nabla \theta) \, dx dy dz \quad (6)$$

where  $K$  is a 3x3 symmetric matrix known as the thermal conductivity matrix.

Setting the rate of decrease of heat in the body equal to the amount of heat leaving the body we arrive at the partial differential equation called the heat equation,

$$\frac{\partial \rho \sigma \theta}{\partial t} = \nabla \cdot (K \nabla \theta). \quad (7)$$

For a homogenous and isotropic material  $K = kI$  where  $I$  is the identity matrix the equation reduces to the familiar form,

$$\frac{\partial \rho \sigma \theta}{\partial t} = k \nabla^2 \theta, \quad (8)$$

where  $\nabla^2$  is the Laplacian.

## 4 Discrete Model

An obvious brute force approach to modeling solids and liquids for computer graphics would be to follow the molecular dynamicists and model each of the individual molecules. Given that an ounce of water contains over  $10^{25}$  molecules, it is also obvious that this approach is not computationally feasible. However this does not prevent us from borrowing some ideas from molecular dynamics.

To accommodate continual changes in topology, we have chosen to use a system of interacting particles. Each particle of the model represents a volume of matter; that is, a very large number of molecules. The interactions between the particles are based on the presence of intermolecular forces and thermal energy. The sum of the interactions model the object as a whole.

To maintain the cohesive properties found in the solid and fluid states we define a potential energy function between pairs of particles: a function that yields short range repulsive forces and long range attractive forces. The attractive forces are necessary to bind the particles together for both the solid and liquid states. It may not appear necessary to have attractive forces for fluids, but it is the attractive forces that produce surface tension [24]. The repulsive forces are necessary to keep particles from occupying the same space.

As our model heats up we do not increase the kinetic energy of the particles as does Greenspan [7]. Instead, we model heat and the inter-molecular potential energies together by a continuous family of inter-particle potential energy functions  $\phi_\psi(r)$ . Figure 3 shows three curves from a family of curves. "Cold" temperatures map to curves with low potential energy minimal (deeper energy wells). While "hot" temperatures map to curves with higher potential energy minimal (shallow wells). As thermal energy is added to a system of particles or molecules, the total energy should increase accordingly. In molecular dynamics, the addition of thermal energy increases the kinetic energy and the inter-molecular potential energy function is fixed. In thermal particles the inter-particle potential energy is increased and the kinetic energy is unchanged.

The velocity of a particle directly parallels the average velocity of the corresponding volume of molecules.

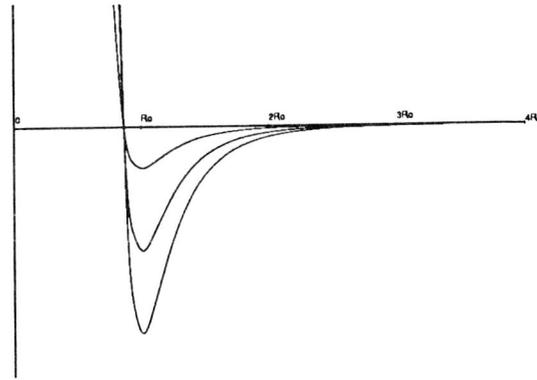


Figure 3: Family of interparticle potential energy functions.

The average speed of a molecular volume is modeled by addition of heat energy to the corresponding particle as described above. This increases the total energy of the particle without directly changing the particle's velocity. The less shallow "well" in the energy curve will result in particles being able to move past other particles more easily in response to external forces.

### 4.1 Inter-Particle Potential Function

We impose two conditions on our family of potential energy functions. The dissociation energy of a pair of particles should be equal to some constant inter-molecular potential energy, say  $-\epsilon_m$ , plus a varying thermal energy  $\psi$ . That is,

$$\phi_\psi(r_0) = -\epsilon_m + \psi. \quad (9)$$

Second, to avoid significant expansion or contraction of matter for different temperatures the equilibrium separation spacing  $r_0$  should be constant across the family of potential energy functions,

$$\frac{d\phi_\psi(r_0)}{dr} = -f_\psi(r_0) = 0. \quad (10)$$

We modify the Lennard-Jones bi-reciprocal function to create a family of inter-particle potential energy curves  $\phi_\psi$  for varying values of  $\psi$ ,

$$\phi_\psi(r) = \frac{-\epsilon_m + \psi}{n - m} \left( m \left( \frac{r_0}{r} \right)^n - n \left( \frac{r_0}{r} \right)^m \right) \quad (11)$$

where  $\epsilon_m$  is the constant inter-molecular potential energy, and  $\psi$  is the thermal energy. The corresponding function for force is,

$$f_\psi(r) = \frac{-\epsilon_m + \psi}{n - m} \left( nm \frac{r_0^n}{r^{n+1}} - nm \frac{r_0^m}{r^{m+1}} \right) \quad (12)$$

By setting  $r = r_0$  and reducing it is easy to show that these basis functions satisfy our two conditions.

The total force  $F_i$  on particle  $i$ , due to the system of particles, is the sum of the individual interactions of particle  $i$  with each neighboring particle  $j$ .

$$F_i = \sum_{j \in \mathcal{N}_i} f_\psi(r_{ij}) \quad (13)$$

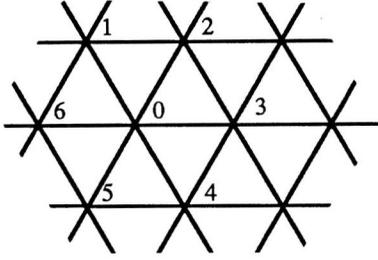


Figure 4: Finite difference grid for the Laplacian

where  $\psi = \frac{\psi_i + \psi_j}{2}$  is the average thermal energy of the two particles, and  $\mathcal{N}_i$  is the set of particles neighboring particle  $i$ . We have limited the range to allow larger variations in local geometry than would otherwise occur. As the range of neighbors is increased the local geometry is smoothed. This provides another way for animators to vary the properties of their simulated materials.

## 4.2 Discrete Heat Transfer

From the general heat equation (7) we use a discrete approximation to solve for the heat and temperature of each particle over time.

The heat of a particle is related to the temperature as follows. We assume the specific heat  $\sigma_i$  and temperature  $\theta_i$  are constant over the interior of a given particle  $i$ . The mass  $m_i$  for the particle is equal to integrating the mass density  $\rho_i$  over the particle's volume. From equation (5) the heat  $\psi_i$  in particle  $i$  simplifies to,

$$\psi_i = \sigma_i \theta_i m_i. \quad (14)$$

The change in heat of particle  $i$  over a time interval  $\Delta t$ ,

$$\frac{\psi_i^{t+\Delta t} - \psi_i^t}{\Delta t} \quad (15)$$

approximates the left side of the heat equation.

To evaluate the  $\nabla \cdot (K \nabla)$  term in the heat equation we introduce a thermal conductivity  $k_{ij}$  between each pair of particles  $i$  and  $j$ . We compute the approximation on a pair-wise basis,

$$\sum_{j \in \mathcal{N}_i} \frac{\frac{1}{n} k_{ij} (\theta_j - \theta_i)}{\frac{1}{4} r_{ij}^2} \quad (16)$$

where  $r_{ij}$  is the distance between  $i$  and  $j$ , and  $n$  is the number of neighbors.

The approximation is based on the finite difference method. To see this we look at the hexagonal configuration of particles shown in Figure 4. The Laplacian of the temperature at the center particle  $i = 0$  is,

$$\begin{aligned} \nabla^2 \theta_0 &\approx \frac{\frac{1}{6} (\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6) - \theta_0}{\frac{1}{4} r^2} \\ &= \sum_{j=1}^{n=6} \frac{\frac{1}{n} k_{ij} (\theta_j - \theta_0)}{\frac{1}{4} r^2} \end{aligned}$$

when approximated by finite differences [15].

## 5 External Forces

The model becomes more interesting when we place it in an environment with other objects. This requires a method to calculate the collisions of particles with obstacles and for the particles to respond accordingly.

We introduce obstacles into our environment and create forces so that particles will not penetrate these objects. A repulsive force is defined between each particle and object surface similar to the repulsive force between particles. The repulsive force is limited to a short range so that particles are only repelled when they are very close to the surface. The force is based on inverse powers of the distance between the object and particle. As a particle and object collide the particle will slow down due to the repulsive force and gain potential energy relative to the obstacle.

For an object  $k$  and particle  $i$ , separated by the distance  $d_{ki}$  we define a collision potential energy function to be

$$e_c(d_{ki}) = \begin{cases} \frac{\alpha}{d_{ki}^\eta} + \beta d_{ki} + \gamma & \text{when } d_{ki} \leq d_o \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

and the resulting force

$$f_c(d_{ki}) = \begin{cases} \frac{n\alpha}{d_{ki}^{\eta+1}} - \beta & \text{when } d_{ki} \leq d_o \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

The distance  $d_o$  is the distance from the obstacle surface at which the particle gains potential energy relative to the obstacle. At this distance the particle starts to feel the obstacle collision force. By constraining  $e_c(d_o) = 0$  and  $f_c(d_o) = 0$ , the potential energy and force functions are continuous for  $d_{ki} > 0$ . For a constant value of  $\eta$ , the user need only specify the distance  $d_o > 0$  and the scaling factor  $\alpha > 0$ , and the remaining constants are uniquely determined.

The obstacles can be non-particle based objects such as planes and spheres or the obstacles can be objects composed of thermal particles. Particles are defined to belong to different objects and only particles belonging to the same object interact according to the inter-particle potential energy function. Particles from differing objects interact using the collision function. For particle-particle collisions the distance measure is the distance between colliding particles.

An alternate method of constraining particles outside of objects is to use reaction-constraint forces [21, 13]. This approach is good for constraining particles outside of surface geometries. During a dynamic simulation we are not calculating the continuous surface geometry of a particle based object. The method we describe, closer in spirit to the penalty method [20], works for this case.

We have added gravity,

$$f_{g_i} = g m_i \quad (19)$$

where  $g$  is a gravitational constant and  $m_i$  is the mass of particle  $i$ , to our simulated world so that we can drop objects on to tables and pour fluids into cups.

The energy functions mentioned so far are conservative and hence result in perfectly elastic collisions. By introducing a damping force,

$$f_{d_i} = -b v_i \quad (20)$$

where  $v_i$  is the velocity of particle  $i$ , we can create inelastic collisions and can model physical phenomena such as air resistance.

## 6 Time Integration

The movement of particle  $i$  in time is governed by Newton's laws of motions:

$$\begin{aligned} a_i &= f_i/m_i \\ v_i &= \frac{da_i}{dt} \\ x_i &= \frac{dv_i}{dt} \end{aligned} \quad (21)$$

where  $f_i$  is sum of all the forces on particle  $i$ ,  $m_i$  is the mass,  $a_i$  is the acceleration,  $v_i$  is the velocity, and  $x_i$  is the position of particle  $i$ .

Given the initial position  $x_i(t_0)$  and the initial velocity  $v_i(t_0)$  of particle  $i$ , for  $i = 1, 2, \dots, N$  we can simulate the Newtonian dynamics. At each time step  $t_{j+1} = t_j + \Delta t$  we sum all of the forces acting on each particle  $i$  and integrate over the time interval. The forces include the inter-particle forces, collision, gravity, and damping forces.

We use the modified midpoint method [14] to advance the current velocity and position over the time step. The time interval  $\Delta t$  is divided into  $n$  equal substeps of size  $h = \Delta t/n$ . The explicit formulation for our system is given by

$$\begin{aligned} z_0 &\equiv v_i(t_j) \\ z_1 &= z_0 + ha_i(t_j) \\ z_{q+1} &= z_{q-1} + 2ha_i(t_j + qh) \\ &\quad \text{for } q = 1, 2, \dots, n-1 \\ v_i(t_j + \Delta t) &= \frac{1}{2}[z_n + z_{n-1} + ha_i(t_j + \Delta t)] \end{aligned} \quad (22)$$

for calculating the new velocity and,

$$\begin{aligned} y_0 &\equiv x_i(t_j) \\ y_1 &= y_0 + hv_i(t_j) \\ y_{q+1} &= y_{q-1} + 2hv_i(t_j + qh) \\ &\quad \text{for } q = 1, 2, \dots, n-1 \\ x_i(t_j + \Delta t) &= \frac{1}{2}[y_n + y_{n-1} + hv_i(t_j + \Delta t)] \end{aligned} \quad (23)$$

for calculating the new position.

We must take care when evaluating the numerics to avoid instabilities due to large forces. A large force can occur if the distance between a pair of particles approaches zero. An simple way to avoid this problem is to add a small offset  $\delta$  to the distance  $r_{ij}$  before the divisions in (12). The same approach can be applied to the collision force equation (18).

## 7 Simulation

Figure 5 and 6 shows two free standing structures. The bonds between the particles are strong enough so that the objects are solid at cold temperatures. In figure 5 the material is flexible and bends under the influence of gravity, but does not break apart. In figure 6 we have increased the exponents of the potential energy function

creating a stiffer material. The particles composing the gables have mass  $m_i = 1$ , specific heat  $\sigma = 1$ , and a temperature of  $0^\circ$ . The gravitational constant is  $g = 1.5$ . The collision force constants are  $\beta = 25$ ,  $d_o = 1$ , and  $\eta = 4$ . The damping constant is  $b = 1.5$ . The equilibrium separation is  $r_o = 1.192$ , and minimum energy at  $r_o$  is for a temperature of  $0^\circ$  is  $\epsilon = 200$ . The values of  $n$  and  $m$  are (4, 2) and (8, 6) for the figures 5 and 6 respectively. Each gable is composed of 69 particles.

Figure 7 presents a selection of frames from an animation. In the animation a rectangular shaped beam composed of particles is dropped from a height under the influence of gravity. The right side of the beam strikes the sphere while the left side continues towards the floor, rotating and slightly flexing the beam. The object bounces back up into the air from the nearly elastic collision. The beam bounces a few more times and slides off of the sphere towards the camera. The beam is initially "cold" and the floor is "hot". Heat is transferred from the floor into the beam at the points of contact. The boundary particles gain heat and in turn transfer heat throughout the object. As the beam heats, it softens, until it exhibits fluid like properties. The parameters for the animation are  $g = 1$ ,  $n = 8$ ,  $m = 6$ ,  $b = .18$  with the remaining parameters the same as above. The beam consists of 192 particles.

We rendered each particle in the beam as a sphere and the gable as an iso-surface. For the iso-surface we assign each particle a field that is maximum at the particle's center and decreases as a function of distance. The iso-surface is all points in space for which the sum of the fields equals a threshold constant. We have rendered the iso-surface using a direct ray-intersection technique [2]. Algorithms exist for approximating the iso-surface with polygonal patches [25, 9]. The resulting polygons may be rendering using traditional techniques. A simple approximation to the iso-surface is to render a sphere for each particle and shade the sphere according to the gradient of the summed fields at that point [11]. If the fields are defined as polynomials there exists an algorithm to find all of the intersections of the iso-surface with a ray in time linear in the number of particles [23].

## 8 Discussion and Extensions

We have developed a discrete model to simulate temperature dependent deformable objects. The model is based on molecular dynamics and is computed according to classical Newtonian dynamics. The objects interact within a simulated environment responding to external forces such as gravity and collisions with other obstacles.

A novel feature of our work is that we have modeled both the solid and liquid states using a single unified model. The material properties continuously change as a function of temperature. The model consists of particles with position, velocity, and thermal energy attributes. Each particle models a volume of the given object and interacts with the other particles via a potential energy function that varies with the separation and the thermal energy of the particles. At temperature extremes the object can act like a nearly rigid solid or a non-viscous liquid. At temperatures in between the model exhibits varying degrees of deformability. Heat is transferred between particles and between particles and non-particle objects.

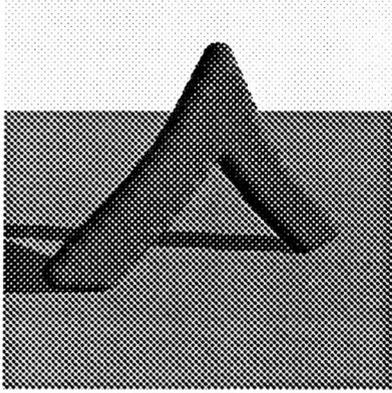


Figure 5: Flexible solid

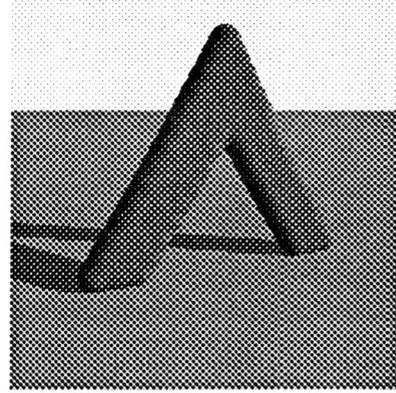


Figure 6: Stiff solid

This allows an object to gain or lose heat from the surrounding environment and exhibit temperature gradients within the object.

The computation time of this algorithm is  $O(N \log N)$ , where  $N$  is the number of particles. The computation of forces is  $O(N)$ , since for each particle we evaluate the interaction with a limited range of neighboring particles. We are using a kd-tree to find the nearest neighbors of each particle [16]. This spatial subdivision algorithm takes  $O(N \log N)$  time. Thus these two steps result in the  $O(N \log N)$  nature. The model falls in the class of systems referred to as N-body systems. Approaches to solving the discretized N-body problem have been investigated by Greenspan [8, 7], Gould [5], Greengard [6], and Zhao [26].

It would be interesting to include friction and convert the energy lost to friction into thermal energy. Thus an object sliding over a rough surface would heat up. We could simulate inelastic materials with the addition of damping elements, as described by Terzopoulos and Fleischer [19], between neighboring particles. Another idea is to use a distance measure other than Euclidean distance for the potential energy function. This might lead to objects that shear or bend easier along a given direction, similar to many objects in the real world.

## 9 Acknowledgements

I am grateful to Demetri Terzopoulos for inspiring work on this subject and discussions. I would like to gratefully acknowledge the University of Toronto for financial support. I would like to thank the good folks at DGP who wrote *Optik*, the renderer used to generate the pictures, including Jos Stam who wrote the “blobbie” primitive.

## References

- [1] Allan F. M. Barton. *The Dynamic Liquid State*. Longman Group Limited, New York, 1974.
- [2] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, July 1982.
- [3] Robert W. Christy and Agnar Pytte. *The Structure of Matter: An Introduction to Modern Physics*. W. A. Benjamin, New York, 1965.
- [4] Alain Fournier and William T. Reeves. A simple model of ocean waves. In *Computer Graphics*, volume 20. SIGGRAPH, August 1986.
- [5] Harvey Gould and Jan Tobochnik. *An Introduction to Computer Simulation Methods*. Addison-Wesley, Reading Massachusetts, 1988.
- [6] Leslie F. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. The MIT Press, Boston, Massachusetts, 1988.
- [7] Donald Greenspan. *Discrete Models*. Addison-Wesley, Reading, Massachusetts, 1973.
- [8] Donald Greenspan. *Arithmetic Applied Mathematics*. Pergamon Press, Oxford, England, 1980.
- [9] Jay Hersh. Soft object extensions to the clockworks system. Technical Report TR-87054, Rensselaer Design Research Center, Rensselaer Polytechnic Institute, Troy, New York, 1987.
- [10] Michael Kass and Gavin Miller. Rapid, stable fluid dynamics for computer graphics. In *Computer Graphics*. SIGGRAPH, August 1990.
- [11] Gavin Miller and Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. In *SIGGRAPH '89, Course 30 notes: Topics in Physically-based Modeling*. SIGGRAPH, August 1989. Boston, Massachusetts.
- [12] Darwyn R. Peachy. Modeling waves and surf. In *Computer Graphics*, volume 20. SIGGRAPH, August 1986.
- [13] John C. Platt and Alan H. Barr. Constraint methods for flexible models. *Computer Graphics*, 22(4), August 1988.
- [14] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1988.
- [15] Karel Rektorys, editor. *Survey of Applicable Mathematics*, chapter 27: Solution of Partial Differential Equations by the Finite-Difference Method. MIT Press, 1969.

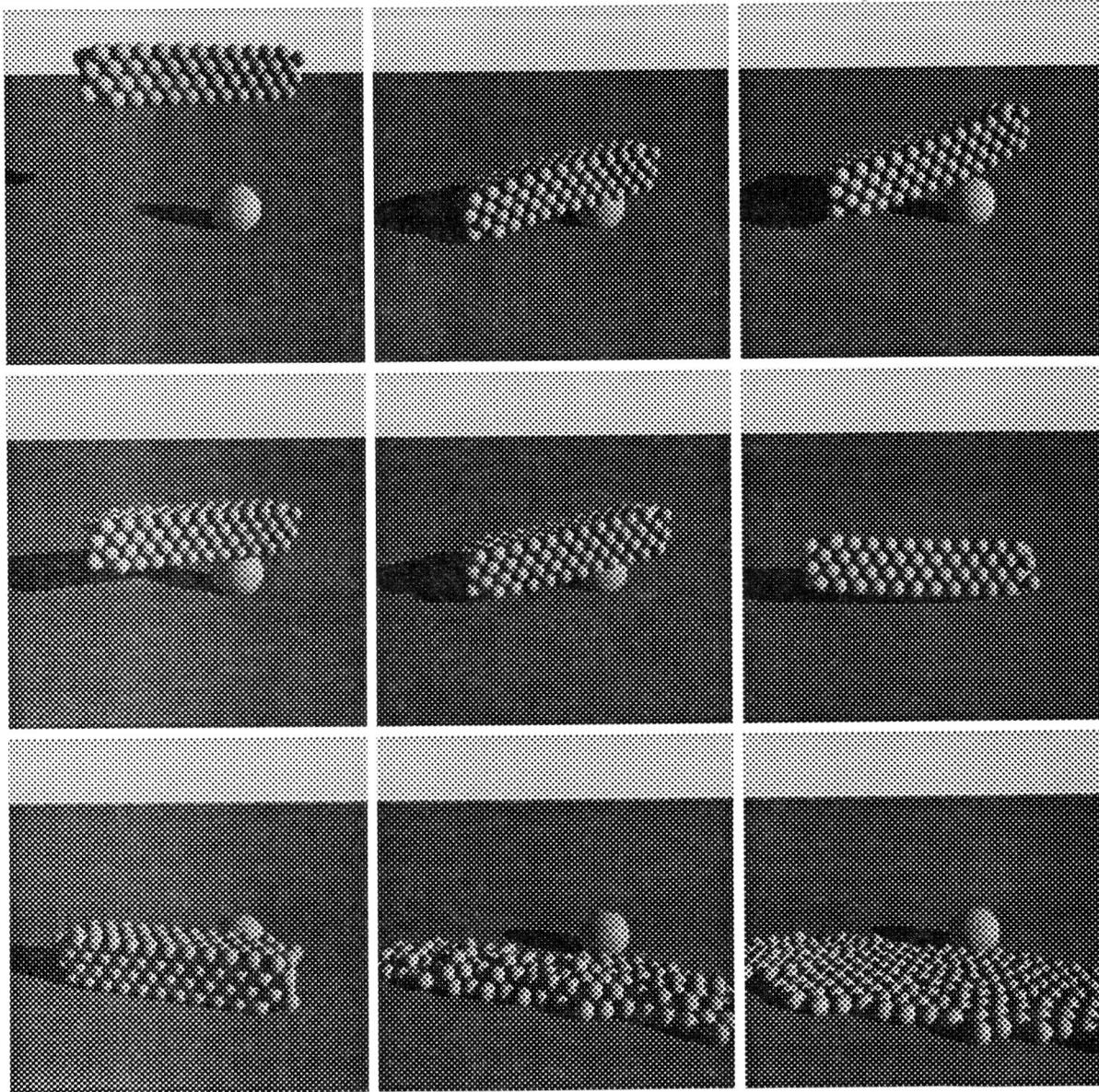


Figure 7: A beam colliding with obstacles and melting. The frames were taken from an animation at the following times:  $t = 0, 3, 4, 5, 9, 12, 25, 70, 83$ .

- [16] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Massachusetts, 1989.
- [17] Karl Sims. Particle animation and rendering using data parallel computation. In *Computer Graphics*, volume 24. SIGGRAPH, August 1990.
- [18] H. N. V. Temperley. *Liquids and their properties*. Ellis Horwood Limited, Chichester, England, 1978.
- [19] Demetri Terzopoulos and Kurt Fleischer. Deformable models. *The Visual Computer*, 1988.
- [20] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Computer Graphics*, volume 21. SIGGRAPH, Aug 1987.
- [21] Demetri Terzopoulos, John Platt, and Kurt Fleischer. Heating and melting deformable models (From Goop to Glop). In *Proceedings Graphics Interface*. Graphics Interface, June 1989.
- [22] Paul A. Tipler. *Physics*. Worth Publishers, New York, New York, 1976.
- [23] David Tonnesen. Ray-tracing implicit surfaces resulting from the summation of bounded polynomial functions. Technical Report TR-89003, Rensselaer Design Research Center, Rensselaer Polytechnic Institute, Troy, New York, 1989.
- [24] D. H. Trevena. *The liquid phase*. Wykeham Publications, London, England, 1975.
- [25] B. Wyvill, C. McPheeters, and G. Wyvill. Data structures for soft objects. *The Visual Computer*, 2, 1986.
- [26] F. Zhao. An  $O(N)$  algorithm for three-dimensional N-body simulations. Technical Report AI-TR-995, MIT Artificial Intelligence Laboratory, Cambridge, Massachusetts, 1987.