The Visual Computer

Deformable models

Demetri Terzopoulos and
Kurt Fleischer

Schlumberger Palo Alto Research
3340 Hillview Avenue, Palo Alto, CA 94304, USA

We develop physically-based models of deformable curves, surfaces, and solids for use in computer graphics. Our deformable models are governed by the mechanical laws of continuous bodies whose shapes can change over time. These laws, expressed in the form of dynamic differential equations, unify the description of shape and motion. By solving the equations numerically we are able to create realistic animations involving the interaction of deformable models with various applied forces, ambient media, and impenetrable obstacles in a simulated physical world. We develop deformable models capable of perfectly elastic behavior and more general inelastic behavior, including viscoelasticity, plasticity, and fracture.

Key words: Modeling – Animation – Deformation – Elasticity – Dynamics – Simulation

Methods to formulate and represent the shapes of objects are central to computer graphics modeling. Geometric methods have been particularly useful for modeling stationary, rigid objects whose shapes do not change over time. Unfortunately, purely geometric modeling primitives are inert. This paper develops a physically-based approach to modeling and animation founded on laws governing the dynamics of nonrigid bodies. These physical principles, formalized in continuum mechanics as partial differential equations, make our *deformable models* active – they respond in a natural manner to applied forces, ambient media, constraints, and collisions with other objects in a simulated physical environment. Our approach to modeling is fundamentally dynamic, and it unifies the description of shapes and their motions through space. Creating realistic animation becomes a matter of simulation; that is, of numerically solving the underlying differential equations of motion.

The following sections examine the implications of our physically-based approach to modeling and animation. We elaborate on the features that distinguish deformable models from conventional modeling techniques (Table 1), and describe the broad spectrum of behaviors that fall within the scope of deformable models.

Table 1. Features of deformable models

Deformable models	Conventional models
Physics & geometry	Geometry only
Active	Inert
Dynamic	Kinematic
Animation by simulation	Prescribed animation

1.1 Physically-based modeling and animation

Modeling based on physical principles is establishing itself as a potent technique in computer graphics. Physically-based models, while computationally more complex than many traditional models, offer unsurpassed realism in the modeling of natural phenomena. The repertoire of physically-based models continues to grow (see, e.g., Barr et al. 1987; Fournier et al. 1987). For instance, a theme currently gaining popularity is the use of Newtonian dynamics to model articulated or arbitrarily constrained assemblies of rigid primitives (Armstrong and Green 1985; Wilhelms and Barsky 1985; Girard and Maciejewski 1985; Wilhelms 1987; Barzel and Barr 1987; Hoffmann and Hopcroft 1987; Issacs and Cohen 1987).

Physically-based modeling offers critical advantages for computer animation. Conventional animation is kinematic; objects are set into motion by prescribing the positions of their geometric components at each instant in time, usually aided by key-frame interpolation techniques. Creating natural-looking motions kinematically requires expertise (Lassiter 1987). By contrast, in the dynamic approach to animation, we apply forces to objects while standard numerical procedures generate positions through time in accordance with Newton's laws. Thus, as computer animators, we may begin to think more like choreographers who remain rather unconcerned with the kinematic details of routines, knowing that physics will dictate the low-level motions of dancers. By incorporating basic physical principles into our models, we are well on the way toward modeling methods that allow computer animators to plan motions at the choreographic level, concentrating on abstract qualities such as timing, rhythm, and style.

In this paper we are concerned mainly with nonrigid, free-form objects. Conventional free-form design is, again, kinematic. For example, the degrees of freedom – typically, the control variables – of a mosaic of patches are adjusted to shape extended surfaces and to position them in space (De Boor 1978; Bartels et al. 1987). However, attempts at imitating the *natural* motion of extended, flexible objects by specifying hundreds of control variables kinematically through time seem hopelessly tedious. For this task especially, the physically-based approach is compellingly attractive.

Conceptually, our approach may be viewed as the application of nonrigid dynamics principles from continuum mechanics (Hunter 1983) to evolve shape and motion variables in response to applied forces. Technically, we employ concepts from the finite-element method (Zienkiewicz 1977), a computational technique that prescribes local polynomial "elements", each with specific "control variables" or degree of freedom, as piecewise approximations for the discrete solution of continuum mechanics equations. Thus, in principle, we can incorporate the physics of complex nonrigid motions into the free-form shape representations which have been used traditionally. In practice, however, we must pay close attention to computational complexity when choosing among candidate representations – simplicity, such as low polynomial order, tends to be a virtue.

A popular free-form shape representation in computer graphics is the spline patch. In applying splines to modeling the graphics community has unfortunately treated them as purely geometric entities, deemphasizing their physical underpinnings. The cubic interpolating spline, for instance, is an abstraction of the shape exhibited by a thin elastic beam (the elastica used in boat construction) whose minimal bending energy configuration may be characterized by a fourth-order differential equation (Faux and Pratt 1981). We have emphasized the physical basis of splines in our computational vision work, which applies multivariate generalizations of the classical spline equations to create models of flexible surfaces sensitive to forces derived from visual data (Terzopoulos 1983; Terzopoulos et al. 1987b; Kass et al. 1987; Terzopoulos 1988). Our deformable models offer a physically-based alternative to geometric techniques for distorting shapes in space, such as Barr's (1984) Jacobian matrix transforms for subjecting solid primitives to parameterized deformations, or Sederberg and Parry's (1986) Bernstein polynomial transforms for distorting free-form shapes. Somewhat less kinematic than the purely geometric methods is the scheme proposed by Wyvill et al. (1986) for animating soft objects as iso-surfaces in a 3D scalar field enveloping control points. Not surprisingly, however, since their control-point dynamics do not accurately describe the mechanics of deformable materials, animations created using this scheme will tend to look contrived unless care is taken to generate convincing motions.

The first physically-based models of flexible objects in computer graphics were concerned with static shape. Weil (1986) proposed an approach for interpolating surfaces between catenary curves to produce draped "cloth" effects. A firmer physical foundation underlies the static cloth model proposed by Feynman (1986). Subsequent efforts produced models for animating nonrigid objects in simulated physical environments (Terzopoulos et al. 1987a; Haumann 1987; Lundin 1987; Weil 1987, unpublished manuscript; Terzopoulos and Witkin 1988). The models of flexible objects proposed by Haumann and Weil can be viewed as dynamic extensions to Feynman's discrete approach. In Terzopoulos et al. (1987a), however, we employ continuous elasticity theory (Landau and Lifshitz 1959) to model the shapes and motions of deformable bodies.

By including distributed physical properties such as mass and damping, we can simulate the dynamics of deformable objects in response to the forces at work. The dynamic behavior inherent to our deformable models significantly simplifies the animation of complex objects, especially as they interact with other objects and constraints in a simulated physical environment. By basing our models on simplifications of elasticity theory, we can approximate a wide range of deformable materials, including string, rubber, cloth, paper, and flexible metals. This paper develops our approach to modeling deformable curves, surfaces, and solids, including *elastic deformation* as well as *inelastic deformation.*

1.2 Elastic and inelastic deformation

A deformation is termed "elastic" if, upon removal of all external forces, the undeformed reference shape restores itself completely. The basic assumption underlying the constitutive laws of classical elasticity theory is that the restoring force (stress) in a body is a single-valued function of the deformation (strain) of the body and, moreover, that it is independent of the history of the deformation. We express the restoring forces intrinsic to our elastically deformable models in terms of a deformation potential energy. An energy characterization is always possible for elastic models. As a generalization of the ideal spring, the elastic model stores potential energy during deformation, which it then releases as it recovers the reference shape.

The present paper also develops computer graphics models which venture into the diverse class of inelastic deformation phenomena intermediate between perfectly resilient elastic solids and viscous fluids. Unlike solids, fluids store no deformation energy and, hence, exhibit no tendency at all for deformational recovery. Generally, a deformation is known as "inelastic" if it does not obey the relatively stringent (Hookean) constitutive laws of classical elasticity. Inelastic deformations occur in real materials for temperatures and forces exceeding certain limiting values above which irreversible dislocations at the atomic level can no longer be neglected.

Why model inelastic behavior? Aside from an artistic motivation to achieve a rich variety of novel graphics effects, we wish to incorporate into our deformable models the mechanical behaviors commonly associated with high polymer solids – organic compounds containing a large number of recurring chemical structures – such as modeling clay, thermoplastic compound, or silicone putty (Alfrey 1947). These behaviors are responsible for the universal utility of these sorts of modeling compounds in molding complex shapes (e.g., in the design of automobile bodies). We are interested in assimilating some of the natural conveniences of this traditional art into the computer-aided design environment of the future. We envision users, aided by stereoscopic and haptic input-output devices, carving "computational plasticine" and applying simulated forces to it to create free-form shapes interactively.

Our physically-based models incorporate three canonical genres of inelastic behavior – *viscoelasticity, plasticity,* and *fracture.* Viscoelastic material behavior blends the characteristics of a viscous fluid together with elasticity. Silicone putty (Silly Putty[1]) exhibits unmistakable viscoelastic behavior; it flows under sustained force, but bounces like a rubber ball when subjected to quickly transient forces. Inelastic materials for which permanent deformations result from the mechanism of slip or atomic dislocation are known as plastic. Most metals, for instance, behave elastically only when the applied forces are small, after which they yield plastically, resulting in permanent dimensional changes. Our models can also simulate the behavior of thermoplastics, which may be formed easily into desired shapes by pressure at relatively moderate temperatures, then made elastic or rigid around these shapes by cooling. As materials are deformed beyond certain limits, they eventually fracture. Cracks develop wherever internal force or deformation distributions become excessive and their propagation is affected by local variations in material properties.

We conclude the introduction with a perspective on our work as it relates to the engineering analysis of materials and structures. First, here is a caveat: We make no particular attempt to model specific materials accurately. Usually the general behavior of a material will defy accurate mathematical description. Hence, engineering models tend to be complex and are usually implemented by sophisticated finite-element codes. These programs are suitable for analyzing the mechanics of structures constructed from specific materials such as steel and concrete (Kardestuncer and Norrie 1987). Com-

[1] Silly Putty is a trademark of Binney & Smith, Inc.

puter graphics has become indispensable for visualizing the large amount of numerical data produced during the preprocessing and postprocessing stages of finite-element analysis (Christiansen 1974; Christiansen and Benzley 1982; Shephard and Abel 1987).

Although we adopt certain numerical techniques from finite-element analysis, our computer graphics modeling work has a distinctly different emphasis. Our goal is to develop physically-based models with associated procedures that can be utilized to create realistic animations. To this end, our models idealize regimes of materials response under specific types of environmental conditions. Their parameters describe qualitatively familiar behaviors, such as stretchability, flexibility, resiliency, fragility, etc. Hence, deformable models are convenient for computer graphics applications, where a keen concern with computational tractability motivates mathematical abstraction and expediency.

1.3 Overview

The organization of the remainder of the paper is as follows: Section 2 formulates two types of elastically deformable models and their associated differential equations of motion. Section 3 describes external forces applicable to produce deformation and motion. Section 4 explains the basic inelastic phenomena in greater detail and describes how we incorporate inelastic behavior into our models. Section 5 overviews our numerical implementation of deformable models; it relegates to an appendix the implementation details for the case of elastic surfaces along with a procedure for solving the differential equations. Section 6 presents a variety of simulation examples using elastically and inelastically deformable models. In Sect. 7 we draw conclusions from our work.

2 Elastically deformable models

An elastically deformable model recovers its reference configuration as soon as all applied forces causing deformation are removed. We formulate two types of elastically deformable models: a *primal formulation*, which explicitly represents the configuration of the deformable model in space, and a *hybrid formulation*, which decomposes this configuration into a deformation component and a reference component capable of rigid-body motion. The primal and hybrid formulations complement each other in practice. This enables us to obtain numerically well-conditioned discrete models covering a wide range of deformable behaviors from highly elastic to nearly rigid. We compare the two formulations in more detail after developing the equations that govern the dynamics of the models under the action of applied forces.

The equations of motion stem from Newtonian mechanics and they balance the externally applied forces with the forces internal to the deformable model. In both formulations u denotes the intrinsic or material coordinates of points in a body domain Ω. For a solid body $\mathrm{u}=(u_1, u_2, u_3)$ has three coordinates. For a surface $\mathrm{u}=(u_1, u_2)$ and for a curve $\mathrm{u}=(u_1)$. In the three cases, and without loss of generality, Ω will be the unit cube $[0,1]^3$, the unit square $[0,1]^2$, and the unit interval $[0,1]$, respectively.

2.1 Primal formulation

The primal formulation of deformable models was presented originally by Terzopoulos et al. (1987a). It describes deformations using the positions

$$\mathbf{x}(\mathrm{u}, t)=[x_1(\mathrm{u}, t), x_2(\mathrm{u}, t), x_3(\mathrm{u}, t)]' \tag{1}$$

of constituent points in a body relative to an inertial frame of reference Φ in Euclidean 3-space, where the subscripts 1, 2, and 3 denote the X, Y, and Z axes, respectively, of the reference frame (Fig. 1). We refer to this vector-valued function of material coordinates and time as the configuration of the body. As the figure shows, we specify the body in its undeformed configuration by

$$\mathbf{x}^0(\mathrm{u})=[x_1^0(\mathrm{u}), x_2^0(\mathrm{u}), x_3^0(\mathrm{u})]'. \tag{2}$$

A deformable model is described completely by the positions $\mathbf{x}(\mathrm{u}, t)$, velocities $\partial\mathbf{x}/\partial t$, and accelerations $\partial^2\mathbf{x}/\partial t^2$ of its mass elements as functions of the material coordinates u and time t. Lagrange's equations of motion for $\mathbf{x}$ in the inertial frame Φ take the form (Goldstein 1980):

$$\mu\frac{\partial^2\mathbf{x}}{\partial t^2}+\gamma\frac{\partial\mathbf{x}}{\partial t}+\delta_{\mathbf{x}}\mathcal{E}=\mathbf{f}. \tag{3}$$

During motion, the net external forces $\mathbf{f}(\mathrm{u}, t)$ balance dynamically against (i) the inertial force due to the mass density $\mu(\mathrm{u})$, (ii) the velocity-dependent damping force with damping density $\gamma(\mathrm{u})$ (here a scalar, but generally a matrix), and (iii) the internal

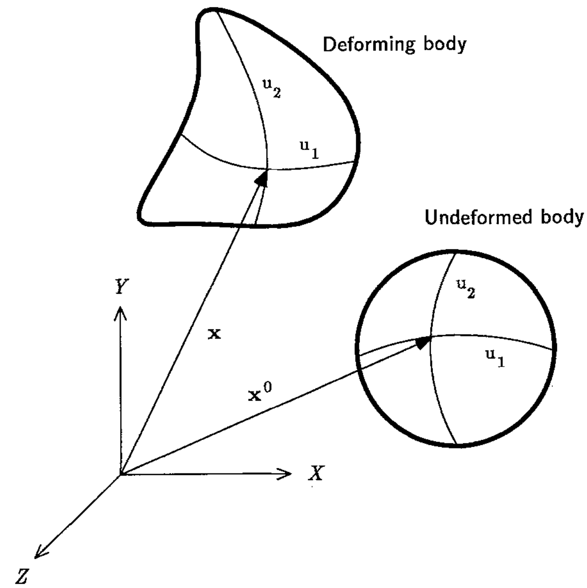


Fig. 1. Geometric representation of the primal model

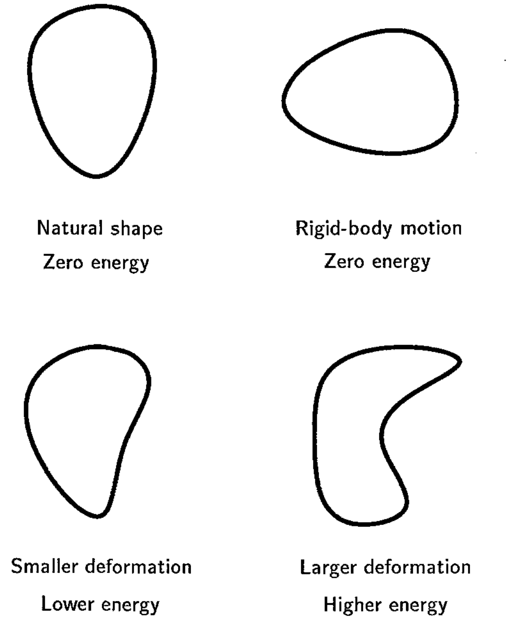


Fig. 2. Energy of deformation

elastic force $\delta_{\mathbf{x}}\mathcal{E}$ which resists deformation. The above is a partial differential equation (due to the dependence of $\delta_{\mathbf{x}}\mathcal{E}$ on $\mathbf{x}$ and its partial derivatives with respect to u – see below). Given (3) with appropriate conditions for $\mathbf{x}$ on the boundary of Ω and the initial configuration $\mathbf{x}(\mathrm{u},0)$ and velocity $\partial\mathbf{x}/\partial t|_{(\mathrm{u},0)}$, we are faced with solving an initial-boundary-value problem.

The elastic force, which acts to restore deformed bodies to their natural shapes, is expressed as $\delta_{\mathbf{x}}$, a variational derivative[2] with respect to $\mathbf{x}$ of a nonnegative deformation potential energy functional $\mathcal{E}(\mathbf{x})$. It is always possible to characterize elastic restoring forces in terms of potential functionals which associate an energy with the net instantaneous deformation of the body.

A suitable energy functional must satisfy two requirements (Fig. 2). The first requirement is that the energy be zero for the model in its natural shape and that it grow as the total deformation of the model away from its natural shape increases. The second requirement is that the energy $\mathcal{E}(\mathbf{x})$ be invariant to rigid motions of the model in Φ, since rigid motions impart no deformation. To define deformation energies that meet these conditions for curves, surfaces, and solids, we employ concepts from differential geometry, which provides fundamental theorems regarding the equivalence of two shapes (Do Carmo 1974).

The shape of a 3-dimensional solid is determined by the Euclidean distances between nearby points. These distances change as the body deforms. Local distances (and angles) in the solid may be measured using the metric tensor $\mathbf{G}$, in this case a 3 × 3 matrix with entries (Faux and Pratt 1981)

$$G_{ij}(\mathbf{x}(\mathrm{u},t))=\frac{\partial \mathbf{x}}{\partial u_i}\cdot\frac{\partial \mathbf{x}}{\partial u_j}. \qquad (4)$$

A fundamental theorem of solids states that two solids in space will have the same instantaneous shape (i.e., differ by a rigid motion only) if their metric tensors are identical functions of $\mathrm{u}=(u_1,u_2,u_3)$ at time t.

If a body is infinitesimally thin in one or more of its dimensions, the distances between nearby points no longer suffice to determine shape. Hence, lengths between nearby points do not determine the shape of a surface, because the surface can be bent without perturbing these lengths. The fundamental theorem of surfaces states that two surfaces have the same shape if their metric tensors as well as their curvature tensors $\mathbf{B}$, both 2 × 2 matrices for a surface, are identical functions of $\mathrm{u}=(u_1,u_2)$. The components of the curvature tensor are (Faux and Pratt 1981)

[2] The variational derivative generalizes the concept of the derivative of a function to the case of functionals; i.e., mappings from functions to real numbers. Refer, e.g., to Courant and Hilbert (1953)

$$B_{ij}(\mathbf{x}(\mathrm{u},t)) = \mathbf{n}\cdot\frac{\partial^2\mathbf{x}}{\partial u_i\,\partial u_j}, \tag{5}$$

where

$$\mathbf{n}(\mathrm{u},t) = \frac{\partial\mathbf{x}/\partial u_1 \times \partial\mathbf{x}/\partial u_2}{|\partial\mathbf{x}/\partial u_1 \times \partial\mathbf{x}/\partial u_2|} \tag{6}$$

gives the unit surface normal over the surface at time t.

For the case of space curves, $\mathrm{u}=u$, and the metric and curvature tensors are the well-known scalar arc-length $s(\mathbf{x}(u,t))$ and curvature $\kappa(\mathbf{x}(u,t))$ functions. Again, arc-length and curvature do not entirely determine the shape of a space curve, since the curve can twist in space without affecting these quantities. Thus, the fundamental theorem of curves states that two curves have the same shape if their arc-length, curvature, and torsion $\tau(\mathbf{x}(u,t))$ are identical functions of $\mathrm{u}=u$ (refer to Faux and Pratt (1981) for the definitions of these functions).

Using the above differential measures of shape, we define potential energies that meet the necessary requirements for elastic curves, surfaces, and solids. A reasonable energy for elastic bodies is a Euclidean (L_2) norm of the difference between the fundamental tensors of the deformed body and the fundamental tensors of the body in its natural shape. Consistently with (2), we distinguish the fundamental forms associated with the natural shapes of deformable bodies with a superscript 0; e.g,

$$\mathbf{G}^0_{ij} = \frac{\partial\mathbf{x}^0}{\partial u_i}\cdot\frac{\partial\mathbf{x}^0}{\partial u_j}. \tag{7}$$

For a deformable curve, we define the energy

$$\mathcal{E}(\mathbf{x}) = \int_\Omega w^1(s-s^0)^2 + w^2(\kappa-\kappa^0)^2 + w^3(\tau-\tau^0)^2\,du \tag{8}$$

where $w^1(u)$, $w^2(u)$, and $w^3(u)$ are weighting functions. The analogous energy for a deformable surface in space is

$$\mathcal{E}(\mathbf{x}) = \int_\Omega |\mathbf{G}-\mathbf{G}^0|^2_{\mathbf{w}^1} + |\mathbf{B}-\mathbf{B}^0|^2_{\mathbf{w}^2}\,du_1\,du_2, \tag{9}$$

where the weighted matrix norms $|\cdot|_{\mathbf{w}^1}$ and $|\cdot|_{\mathbf{w}^2}$, involve the weighting functions $w^1_{ij}(u_1,u_2)$ and $w^2_{ij}(u_1,u_2)$ respectively[3]. Similarly, an energy for a deformable solid is

$$\mathcal{E}(\mathbf{x}) = \int_\Omega |\mathbf{G}-\mathbf{G}^0|^2_{\mathbf{w}^1}\,du_1\,du_2\,du_3, \tag{10}$$

where the weighted matrix norm $|\cdot|_{\mathbf{w}^1}$ involves the weighting functions $w^1_{ij}(u_1,u_2,u_3)$.

The deformation energies (8–10) are invariant with respect to rigid motions, and they include the fewest partial derivatives necessary to restore the natural shapes of non-rigid curves, surfaces, and solids, respectively. However, we can include in them higher-order derivatives to further constrain the smoothness of the deformations (Terzopoulos 1986).

The weighting functions in the above energies determine the properties of the simulated deformable material. In the curve energy (8), $w^1(u)$, $w^2(u)$, and $w^3(u)$ determine the resistance to stretching, bending, and torsion, respectively, along the curve. In the energy for the solid (10), w^1_{ij} determine the resistance to stretching along u_1, u_2, and u_3, as well as shearing across planes perpendicular to these axes.

To better understand the influence of the weighting functions, consider the surface energy (9). Here $w^1_{ij}(u_1,u_2)$ control surface tensions and shear strengths which minimize the deviation of the surface's actual metric coefficients G_{ij} from its natural coefficients G^0_{ij}. As w^1_{ij} is increased, the material becomes more resistant to length deformation, with w^1_{11} and w^1_{22} determining this resistance along u_1 and u_2, and $w^1_{12}=w^2_{21}$ determining the resistance to shear deformation. The functions $w^2_{ij}(u_1,u_2)$ control surface rigidities which act to minimize the deviation of the surface's actual curvature coefficients B_{ij} from its natural coefficients B^0_{ij}. As w^2_{ij} is increased, the material becomes more resistant to bending deformation, with w^2_{11} and w^2_{22} determining this resistance along u_1 and u_2, and $w_{12}=w_{21}$ determining the resistance to twist deformation.

To simulate a stretchy rubber sheet, for example, we make w^1_{ij} relatively small and set $w^2_{ij}=0$. To simulate relatively stretch resistant cloth, we increase the value of w^1_{ij}. To simulate paper, we make w^1_{ij} relatively large and we introduce a modest value for w^2_{ij}. Springy metal may be simulated by increasing the value of w^2_{ij}. Since $w^1_{ij}(\mathrm{u})$ and $w^2_{ij}(\mathrm{u})$ are functions of material coordinates u, we may vary the material properties over the surface, and we may introduce local singularities such as fractures and creases (see Sect. 4).

[3] By a (L_2) *weighted matrix norm* of an $n\times n$ matrix **A** with components a_{ij}, we mean $|\mathbf{A}|_{\mathbf{w}} = \left(\sum_{i,j=1}^{n} w_{ij}\,a_{ij}^2\right)^{1/2}$

2.2 Hybrid formulation

An alternative formulation (Terzopoulos and Witkin 1988) leads to "hybrid" deformable models. Hybrid models include explicit deformable and rigid dynamics operating in concert. To define these models, we represent the positions of mass elements in the body relative to ϕ by

$$\mathbf{q}(\mathrm{u}, t)=\mathbf{r}(\mathrm{u}, t)+\mathbf{e}(\mathrm{u}, t), \tag{11}$$

the sum of a reference component $\mathbf{r}(\mathrm{u}, t)=[r_1(\mathrm{u}, t), r_2(\mathrm{u}, t), r_3(\mathrm{u}, t)]'$ and a deformation component $\mathbf{e}(\mathrm{u}, t)=[e_1(\mathrm{u}, t), e_2(\mathrm{u}, t), e_3(\mathrm{u}, t)]'$. Here, $\mathbf{r}$ and $\mathbf{e}$ are expressed relative to a reference frame ϕ, with the subscripts 1, 2, and 3 denoting, respectively, the x, y, and z axes of ϕ (Fig. 3).
The origin of frame ϕ coincides with the deformable body's center of mass

$$\mathbf{c}(t)=\int_{\Omega} \mu(\mathrm{u})\, \mathbf{x}(\mathrm{u}, t)\, d\mathrm{u}. \tag{12}$$

It is a noninertial frame which is conveyed along with the body in accordance with the laws of rigid-body dynamics (Goldstein 1980). Define the linear and angular velocities of ϕ relative to Φ as

$$\mathbf{v}(t)=\frac{d\mathbf{c}}{dt}; \quad \boldsymbol{\omega}(t)=\frac{d\boldsymbol{\theta}}{dt}, \tag{13}$$

where $d\boldsymbol{\theta}$ is a quantity whose magnitude equals the infinitesimal rotation angle and whose direction is along the instantaneous axis of rotation of ϕ relative to Φ. Then, the velocity of mass elements of the model relative to Φ, given their velocities $\partial \mathbf{e}(\mathrm{u}, t)/\partial t$ relative to ϕ, is

$$\frac{\partial \mathbf{x}}{\partial t}(\mathrm{u}, t)=\mathbf{v}+\boldsymbol{\omega}\times\mathbf{q}+\frac{\partial \mathbf{e}}{\partial t}. \tag{14}$$

We use Lagrangian mechanics to transform the kinetic energy of the primal model (3) according to the above decomposition (Terzopoulos and Witkin 1988). Assuming small deformations, this yields the following three coupled differential equations for the unknown functions $\mathbf{v}$, $\boldsymbol{\omega}$, and $\mathbf{e}$ under the action of an applied force $\mathbf{f}(\mathrm{u}, t)$:

$$m\frac{d\mathbf{v}}{dt}+\frac{d}{dt}\int_{\Omega}\mu\frac{\partial \mathbf{e}}{\partial t}d\mathrm{u}+\int_{\Omega}\gamma\frac{\partial \mathbf{x}}{\partial t}d\mathrm{u}=\mathbf{f}^{\mathrm{v}}; \tag{15a}$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega})+\frac{d}{dt}\int_{\Omega}\mu\mathbf{q}\times\frac{\partial \mathbf{e}}{\partial t}d\mathrm{u} + \int_{\Omega}\gamma\mathbf{q}\times\frac{\partial \mathbf{x}}{\partial t}d\mathrm{u}=\mathbf{f}^{\omega}; \tag{15b}$$

$$\mu\frac{\partial^2 \mathbf{e}}{\partial t^2}+\mu\frac{d\mathbf{v}}{dt}+\mu\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\mathbf{q}) +2\mu\boldsymbol{\omega}\times\frac{\partial \mathbf{e}}{\partial t}+\mu\frac{d\boldsymbol{\omega}}{dt}\times\mathbf{q}+\gamma\frac{\partial \mathbf{x}}{\partial t}+\delta_{\mathbf{e}}\mathcal{E}=\mathbf{f}. \tag{15c}$$

Here $m=\int_{\Omega}\mu\, d\mathrm{u}$ is the total mass of the body, and the inertia tensor $\mathbf{I}$ is a 3×3 symmetric matrix with entries

$$I_{ij}=\int_{\Omega}\mu(\delta_{ij}\mathbf{q}^2-q_i q_j)\, d\mathrm{u}, \tag{16}$$

where $\mathbf{q}=[q_1, q_2, q_3]'$ and δ_{ij} is the Kronecker delta. The applied force $\mathbf{f}(\mathrm{u}, t)$ contributes to elastic deformation, as well as to a net translational force $\mathbf{f}^{\mathrm{v}}$ and a net torque $\mathbf{f}^{\omega}$ on the center of mass:

$$\mathbf{f}^{\mathrm{v}}(t)=\int_{\Omega}\mathbf{f}\, d\mathrm{u}; \quad \mathbf{f}^{\omega}(t)=\int_{\Omega}\mathbf{q}\times\mathbf{f}\, d\mathrm{u}. \tag{17}$$

The ordinary differential equations (15a, b) describe $\mathbf{v}$ and $\boldsymbol{\omega}$, the translational and rotational motion of the body's center of mass. The terms on the left hand sides of these equations pertain to the total moving mass of the body as if concentrated at $\mathbf{c}$, the total (vibrational) motion of the mass elements about the reference component $\mathbf{r}$, and the total damping of the moving mass elements.
The partial differential equations (15c) describes

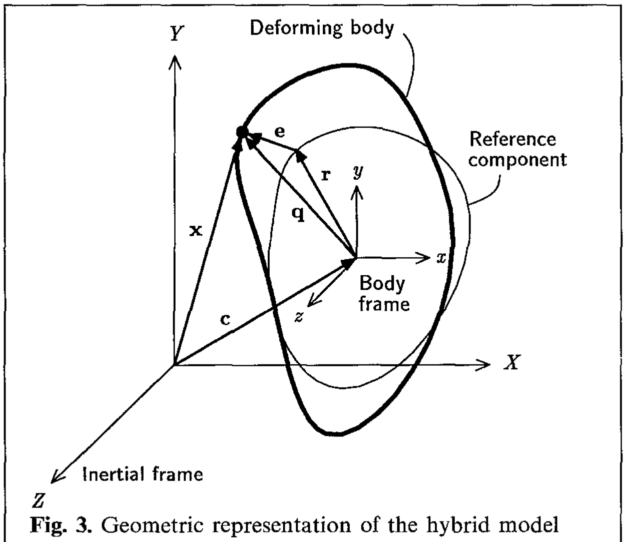


Fig. 3. Geometric representation of the hybrid model

the deformation $\mathbf{e}$ of the model away from $\mathbf{r}$ relative to ϕ. Each of its terms is a dynamic force per mass element: (i) the basic inertial force, (ii) the inertial force due to linear acceleration of ϕ, (iii) the centrifugal force due to the rotation of ϕ, (iv) the Coriolis force due the velocity of the mass elements in ϕ, (v) the transverse force due to the angular acceleration of ϕ, (vi) the damping force, and (vii) the elastic restoring force due to deformation away from $\mathbf{r}$.

Once again we represent the restoring force in (15c) as a variational derivative of a elastic potential energy functional $\mathcal{E}$. Contrary to the primal formulation, this energy need not be rigid-motion invariant, since it characterizes $\mathbf{e}(\mathrm{u}, t)$ which is merely a deformational displacement from the reference shape $\mathbf{r}$ relative to the moving frame ϕ. This permits us to use simple, linear restoring forces stemming from controlled-continuity spline energies (Terzopoulos 1986) of the form

$$\mathcal{E}(\mathbf{e}) = \frac{1}{2} \int_{\Omega} \sum_{m=0}^{p} \sum_{|j|=m} \frac{m!}{j_1! \dots j_d!} w_j |\partial_j^m \mathbf{e}|^2, \qquad (18)$$

where $j = (j_1, \dots, j_d)$ is a multi-index with $|j| = j_1 + \dots + j_d$, and

$$\partial_j^m = \frac{\partial^m}{\partial u_1^{j_1} \dots \partial u_d^{j_d}} \qquad (19)$$

($d=1$ for curves, $d=2$ for surfaces, and $d=3$ for solids). Thus the energy density under the integral is a weighted sum of the magnitude of the deformation $\mathbf{e}$ and of its partial derivatives with respect to material coordinates. Generally, the order p of the highest partial derivative included in the sum determines the order of smoothness of the deformation.

As in the primal formulation, the weighting functions $w_j(\mathrm{u})$ in (18) control the properties of the deformable model over the material coordinates. In the case of surface ($d=2$), the function w_{00} penalizes the total magnitude of the deformation; w_{10} and w_{01} penalize the magnitude of its first partial derivatives; w_{20}, w_{11}, and w_{02} penalize the magnitude of its second partial derivatives; etc.

2.3 Comparison of the two formulations

In practice, the primal and hybrid formulations offer specific benefits and drawbacks which become significant at extreme limits of deformable behavior.

The primal formulation handles free motions implicitly, but at the expense of a non-quadratic energy functional $\mathcal{E}(\mathbf{x})$ (nonlinear restoring forces). For highly nonrigid models such as rubber sheets the equation of motion (3) is numerically well conditioned, hence soluble without much difficulty. However, our experiments indicate a deterioration of the conditioning of the discrete equations with increasing rigidity, evidently due to their exacerbated nonlinearity with increasing stiffness. The consequence is severe difficulty in solving the equations, making the primal formulation impractical for stiff models.

The hybrid formulation permits the use of a quadratic energy functional $\mathcal{E}(\mathbf{e})$ (linear restoring forces). Although they appear more complicated, the equations of motion (15) offer a significant practical advantage for fairly rigid models, and especially so when complex reference shapes are desired. Numerical conditioning improves as the model becomes more rigid, tending in the limit to well-conditioned, rigid-body dynamics. However, attempting extremely flexible models like cloth or stretchy rubber sheets with the hybrid formulation may result in rather unnatural deformations due to the simple connection of the deformation to a rigid reference shape through linear forces.

3 Applied forces

Applying external forces to deformable models yields realistic dynamics. This section lists representative examples of external forces. A plethora of complicated force functions is conceivable, but we include only simple instances of gravitational forces, aerodynamic forces, and repulsive forces due to collisions with impenetrable objects. The net external force $\mathbf{f}(\mathrm{u}, t)$ in (3) or (15c) is the sum of the individual external forces.

A gravitational force acting on the deformable model is given by

$$\mathbf{f}_g(\mathrm{u}) = \mu(\mathrm{u})\, \mathbf{g}, \qquad (20)$$

where $\mu(\mathrm{u})$ is the mass density and $\mathbf{g}$ is the gravitational field.

A force that connects a material point u_0 of a deformable model to a point $\mathbf{p}_0$ in space by an ideal Hookean spring is

$$\mathbf{f}_s(\mathrm{u}, t) = k(\mathbf{p}_0 - \mathbf{x}(\mathrm{u}_0, t))\, \delta(u - u_0), \qquad (21)$$

where k is the spring constant and δ is the unit delta function. Point $\mathbf{p}_0$ acts as a soft constraint on the model.

The force on the surface of a deformable model due to a viscous fluid is

$$\mathbf{f}_v(\mathrm{u}, t) = c(\mathbf{n} \cdot (\mathbf{s}(\mathbf{x}, t) - \partial \mathbf{x}/\partial t))\, \mathbf{n}, \qquad (22)$$

where c is the strength of the fluid force and $\mathbf{n}(\mathrm{u}, t)$ is the unit surface normal. Here, the dot product is taken with respect to the velocity of the surface relative to some stream function $\mathbf{s}(\mathbf{p}, t)$, in general, a time-varying driving function that represents the stream velocity at all points $\mathbf{p}$ in space. The force of the flow field projected normal to the surface is linear in the velocity, and it models a viscous medium. It is possible to define more complex forces including a variety of aerodynamic effects.

We can simulate collision dynamics between deformable models and immobile impenetrable obstacles by creating a potential function $c \exp(-f(\mathbf{x})/\varepsilon)$ around each obstacle, where f is the obstacle's implicit function. The constants c and ε determine the shape of the potential. We choose them such that the resulting repulsive force, expressed as the gradient of the potential, grows quickly if the model attempts to penetrate the obstacle. The repulsive force is

$$\mathbf{f}_c(\mathrm{u}, t) = -c((\nabla f(\mathbf{x})/\varepsilon) \exp(-f(\mathbf{x})/\varepsilon) \cdot \mathbf{n})\, \mathbf{n}, \qquad (23)$$

where $\mathbf{n}$,(u, t) is the unit normal vector of the deformable body's surface.

For most applications, deformable models should not self- or inter-penetrate as they deform. To avoid such intersections, we can surround the surfaces of nonrigid models with a repulsive collision force as well. The repulsive force requires an implicit description of the surface of each model, which is only available locally in our models. Each model decomposes into many small patches (see Sect. 5) and the repulsive force computation can become expensive. To enhance efficiency, we may organize the patches into hierarchical bounding-volume data structures.

Surrounding objects with force fields is analogous to barrier methods in constrained optimization (Luenberger 1973). The virtue of the technique is its simplicity, but rarely can it exactly enforce the geometric constraints associated with obstacles. Depending on the speed of collision, among the consequences may be poor approximation to the physics. Perhaps more disastrously, high-momentum collisions will generate colossal forces which wreck havoc with the numerical stability of the simulation (unless adaptive time stepping is employed to mitigate the problem; see Sect. 5). An alternative approach to dealing with obstacles is to use the analogue of Lagrange multiplier type optimization methods for constrained problems. To this end, we have used geometric projection techniques to ensure that no point on a deformable model ever penetrates an obstacle. Depending on the representation and geometric complexity of obstacles, projection techniques can require sophisticated numerical approximations that significantly increase the computational cost of the animation.

The above discussion evades nonlinear contact phenomena such as friction. We approximate friction incrementally in rather simple albeit effective ways. For instance, when the differential movement of deformable model patches indicates contact with other surfaces, we apply velocity-dependent damping (as in the second term in (3)) to the component of the velocity $\partial \mathbf{x}/\partial t$ lying in the tangent plane of the obstacle at the point of contact. At the same time, we nullify the velocity component along the inward normal to the surface or reflect a fraction of it to cause rebound from the collision. In the real world, however, friction is a complex nonconservative effect with nontrivial dependence on the initial point of contact and on the sequence of normal forces.

4 Modeling inelastic deformation

This section describes the basic inelastic behaviors of materials and how we incorporate them into our models. A formal treatment of inelastic deformation is beyond the scope of this paper. For the theory of viscoelasticity, plasticity, and fracture, refer to, e.g., Christensen (1982), Mendelson (1968), and Sih (1981).

Basic inelastic behaviors are easily understood in terms of assemblies of idealized uniaxial (one-dimensional) mechanical units. The ideal linear elastic unit is the spring (Fig. 4a). The spring satisfies Hooke's law – elongation or contraction e is proportional to applied tension or compression force: $ke = f$, where k is the spring constant. The elastic unit is supplemented by two other uniaxial units,

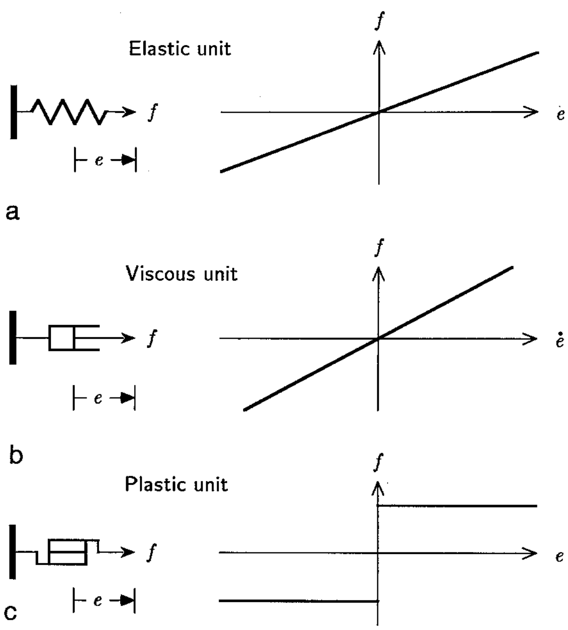


Fig. 4a–c. Uniaxial deformation units and their response to applied forces. **a** Elastic spring. **b** Viscous dashpot. **c** Plastic slip unit

the viscous and plastic units (Fig. 4b, c). By assembling these units in specific configurations, we can simulate simple, uniaxial viscoelasticity and plasticity. The basic laws governing these units generalize over an extended multidimensional continuum within our inelastically deformable models. The generalizations also specify how deformations and stresses in one direction couple with deformations and stresses in other directions.

4.1 Viscoelasticity

Viscoelasticity is a generalization of elasticity and viscosity. It is characterized by the phenomenon of creep which manifests itself as a time dependent deformation under constant applied force. In addition to instantaneous deformation, creep deformations develop which generally increase with the duration of the force. Whereas an elastic model, by definition, is one which has memory only of its reference shape, the current deformation of a viscoelastic model is a function of the entire history of applied force. Conversely, the current restoring force is a function of the entire history of deformation.

The ideal linear viscous unit is the dashpot (Fig. 4b). The rate of increase in elongation or contraction e (strain) is proportional to applied force f (stress): $\eta \dot{e} = f$, where η is the viscosity constant. The elastic and viscous units are combined to model linear viscoelasticity. Figure 5a illustrates a four-unit viscoelastic model, a series assembly of the so called Maxwell and Voigt viscoelastic models. The stress-strain relationship for this assembly has the general form

$$a_2 \ddot{e} + a_1 \dot{e} + a_0 e = b_2 \ddot{f} + b_1 \dot{f} + b_0 f, \qquad (24)$$

with the coefficients depending on the spring and viscosity constants. The response of the models to an applied force (Fig. 5b) is shown graphically in Fig. 5c.

4.2 Plasticity

In plasticity, unique relationships between displacement and applied force do not exist in general.

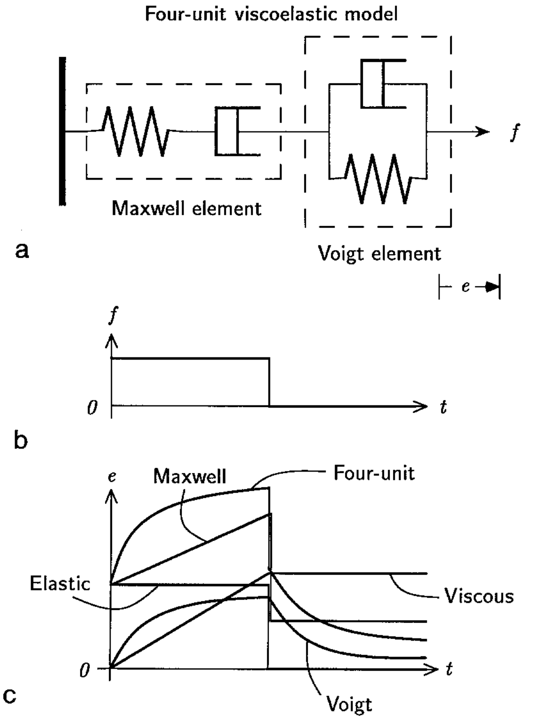


Fig. 5a–c. Uniaxial viscoelastic model. **a** The four-element model is a series connection of a Maxwell viscoelastic unit and a Voigt viscoelastic unit. **b** Force applied to the model. **c** Responses of various components (not to relative scale)

The ideal plastic unit is the slip unit (Fig. 4c). It is capable of arbitrary elongation or contraction when the applied force exceeds a yield force. During plastic yield, the apparent instantaneous elastic constants of the material are smaller than those in the elastic state. Removal of applied force causes the material to unload elastically with its initial elastic constants. This behavior may be termed "elastoplastic," since it generalizes simple elasticity and plasticity.

Elastoplasticity may be modeled by assembling springs with plastic units. Figure 6b illustrates the response of the simple elastoplastic model in Fig. 6a. The model is linearly elastic from O to A. After reaching the yield point A, the model exhibits linear hardening toward B. Upon unloading from any point B, the model moves down toward C, with force amplitude $f_B - f_C = 2f_A$ defining the elastic region. Subsequent loads now move the model back up towards B. Loading past point B causes further plastic deformation along BE. The reverse plastic deformation occurs along CD. After a closed cycle in force and displacement OABCDO, the material returns to its initial state and subsequent behavior is not affected by the cycle.

It is possible to model viscoplasticity, a generalization of plasticity and viscosity, analogously by assembling dashpots with plastic units.

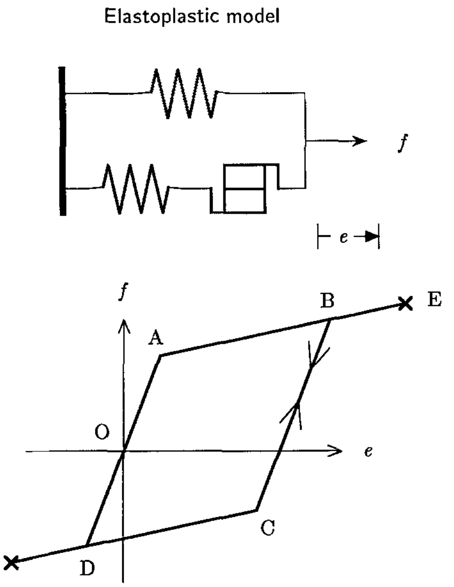


Fig. 6a, b. Uniaxial elastoplastic model. **a** The three-unit model. **b** Response to applied force (see text)

4.3 Fracture

Solid materials cannot sustain arbitrarily large stresses without failure. In the elastoplastic model of Fig. 6, fracture is indicated at point E; elongation beyond this limit causes the elastoplastic material to fail. Fractures are localized position discontinuities that arise due to the breaking of atomic bonds in materials. Discontinuities usually initiate from stress singularities that arise at imperfections or corners of cavities ubiquitous in solids.

Solids exhibit three modes of fracture opening: a tensile mode and two shear modes, one planar and one normal to a plane. As fractures develop they release the potential energy of deformation (strain energy) stored by the material. For fractures to propagate, the energy release rate as the fracture lengthens must exceed a critical value. For brittle materials such as glass, fracture will develop unstably if the energy released is equal to the energy used to create the free surface associated with the fracture. In this case, minor variations in material properties in the continuum can greatly influence the propagation. In general, fracture propagation is not well understood. One of the many research topics in fracture mechanics (see the large large body of literature; e.g., Sih 1981) is the analysis of plasticity effects at fracture tips, a phenomenon which cannot be neglected in materials such as steel.

4.4 Incorporating inelastic behavior

This section describes how we incorporate inelastic behavior using the hybrid formulation, then briefly indicates how we obtain such behavior in the primal formulation.

Recall that the hybrid formulation expresses deformation $\mathbf{e}(\mathrm{u}, t)$ with respect to a reference component $\mathbf{r}(\mathrm{u}, t)$. We elicit viscoelastic, plastic, and fracture behavior from our hybrid model by designing internal processes that update $\mathbf{r}$ and modify material properties according to applied force and instantaneous deformation.

As a simple case of viscoelasticity, consider the Maxwell unit depicted in Fig. 5. We allow $\mathbf{e}(\mathrm{u}, t)$, as governed by (18), to play the role of a multidimensional elastic spring in the continuum generalization of this unit, while $\mathbf{r}(\mathrm{u}, t)$ plays the role of the dashpot. An internal process which evolves the reference component according to the simple rule

$\dot{\mathbf{r}}(\mathrm{u}, t) = (1/\eta(\mathrm{u}))\, \mathbf{e}(\mathrm{u}, t)$ simulates the viscous behavior of the dashpot.

Thus, the viscoelastic process establishes feedback from $\mathbf{e}$ into $\mathbf{r}$. During each time interval, a portion of the instantaneous elastic displacement is transferred into the reference component, which maintains a deformation history. This is analogous to the incremental strain theory or flow theory of elasticity. We can extend this to simulate the four-element viscoelastic model shown in Fig. 5, according to a suitable generalization of (24). More complex viscoelastic behaviors including nonlinearities are readily incorporated by introducing arbitrary functions into the feedback loop. We have witnessed bizarre but nonetheless interesting behavior by choosing physically unrealizable parameters, such as negative viscosity η.

We have incorporated into our models a multidimensional extension of the uniaxial elastoplastic element of Fig. 6. Here, the reference component $\mathbf{r}$ absorbs the extension of the plastic unit as soon as the applied force exceeds the yield limit. In the multidimensional case, this limit generalizes to a "yield condition" expressed as a function which may depend on stresses internal to the model (such as the Tresca or von Mises yield conditions (Mendelson 1968)) or on the internal deformation $\mathbf{e}$. The model behaves elastically until the local yield condition is exceeded. Beyond the yield point the material parameters w_j are reduced locally to simulate linear strain hardening.

The controlled-continuity spline energy (18) allows our models to simulate the piece-wise continuous deformations characteristic of fractures, creases, curvature discontinuities, etc. As was mentioned earlier, the distributed parameter functions w_j offer local continuity control throughout the deformable model's material domain Ω. Discontinuities in the deformation of order $0 \leq k < p$ will occur freely at any material point u_0 such that $w_j(\mathrm{u}_0) = 0$ for $|j| > k$ (Terzopoulos 1986). An internal fracture process automatically monitors stress and deformation distributions over Ω. When local stress or deformation exceeds prescribed fracture limits, the process nullifies the w_j as necessary to introduce the appropriate discontinuities.

We have experimented with several simple schemes for propagating fractures in our models. For instance, at each time step we can insert a position discontinuity (order $k = 0$) at the material point u_* (or some small set of points) at which there occurs the maximum elastic displacement over Ω exceeding the limiting elongation. The yield limit may vary greatly over position in real materials, especially if there happen to be localized weaknesses due to imperfections. We have experimented successfully with yield functions that vary stochastically around some mean yield limit. Interesting possibilities abound.

We can extend our primal formulation of elastically deformable models to include inelastic behavior. Recall that in this formulation the deformation energy functionals involve the fundamental tensors of curves, surfaces, and solids. According to (10), for instance, the elastic energy density for a solid model is the squared normed difference between the metric tensor $\mathbf{G}(\mathbf{x})$ of the deformed body and its reference metric tensor counterpart $\mathbf{G}^0$ associated the undeformed body. Our approach to eliciting inelastic behavior is analogous to that for the hybrid model. We incorporate a feedback process which evolves the reference tensor $\mathbf{G}^0$ (as well as any higher-order reference tensors that may be present in $\mathcal{E}(\mathbf{x})$ for deformable surface or curve models) according to the model's instantaneous internal stresses or deformations. Furthermore, internal processes associated with plasticity and fracture dynamically adjust the material property functions $w^1_{i,j}(\mathrm{u})$ in the weighted norm $|\cdot|_{\mathbf{w}^1}$.

5 Implementation

To create animation with deformable models, we simulate their differential equations of motion numerically. After each time step (or every few time steps) in the simulation, we render the models' state data to create successive frames of an animation sequence. This section explains the steps in our numerical implementation of deformable models.

The first step is to discretize the partial differential equations (3) or (15c) in material coordinates, possibly after some simplification. This step, known as semidiscretization, may be carried out using finite-difference or finite-element approximations on a discrete mesh of nodes (Lapidus and Pinder 1982; Zienkiewicz 1977). The result is a large system of simultaneous ordinary differential equations.

The second step is to integrate the semidiscrete system through time, thus simulating the dynamics of deformable models. We use a semi-implicit time integration procedure which evolves the elastic displacements (and rigid-body dynamics in the hybrid model) from prescribed initial conditions. In essence, the evolving deformation yields a recursive sequence of (dynamic) equilibrium problems, each requiring solution of a *sparse*, linear system whose dimensionality is proportional to the number of nodes comprising the discrete model.

The size of these linear systems can vary widely depending on the application. In the simulations presented in the next section, which have been selected to convey the broad scope of deformable models, the systems range in size from hundreds to tens of thousands of equations. Since deformable models involve many state variables (significantly more than for typical rigid or articulated body simulations) it is crucial to choose applicable numerical solution methods judiciously in order to achieve efficiency ((Press et al. 1986) is a nice survey of standard numerical techniques).

For up to moderately-sized problems, we have used direct methods; especially, a Choleski-type matrix factorization procedure with forward-reverse resolution. We use an efficient, profile storage scheme (Zienkiewicz 1977) which exploits the sparsity of the linear system. For large problems involving surfaces or solids, we must resort to iterative methods such as successive over-relaxation (SOR) or the conjugate gradient (CG) method. We have also made use of an alternating-direction-implicit method (ADI) which iterates fast, one-dimensional Choleski solvers (Press et al. 1986). Multigrid methods based on SOR have served well in the largest of our simulations (Hackbusch 1985).

Appendix A gives the mathematical details of our implementation for the case of elastic surfaces. Curves (solids) represent a straightforward restriction (extension) of the discrete equations given therein.

6 Simulations

This section presents a selection of deformable model simulations. First we describe two systems that allow a user to interact with deformable models in real time. We then describe non-interactive simulations of elastic and inelastic models. Tables 2 and 3 facilitate comparison among the details of various examples. The color graphics images have been rendered using the modeling testbed system described by Fleischer and Witkin (1988).

6.1 Interactive models

Figure 7a–c uses strobe frames to depict "Flatland," a simplified physical world. Flatland models are deformable planar curves capable of rigid-body dynamics as well as elastic and inelastic dynamics combining viscoelasticity, plasticity, and fractures. The simulations in Fig. 7a–c involve a 50-node discrete model. We compute the collisions using a simple projection method. This method does not conserve the area of the model. Flatland simulations are efficient enough to run at interactive rates on Symbolics 3600 series Lisp Machines[4]. This enables us to manipulate the models using a mouse, subjecting them to user-controlled forces, gravity, collision forces, etc.

Figure 7d shows another simulation in Flatland which features the interactive molding of inelastic models by applying forces under interactive control. The user starts in Fig. 7d with a circular viscoelastic model fixed at its center. The model can be thought of as thermoplastic material. The user applies a sustained spring force from point A. The spring (under position control at one end from a mouse) is depicted in the figure as a line between two points. The spring force deforms the model, stretching it to the left, an effect known as "stress relaxation". Next, the spring is released from A, then reactivated at B and swept through C, D, and E, pulling the model along. The final free-form shape is set by "cooling" the thermoplastic. The model then becomes perfectly elastic and it may be bounced (Fig. 7e). Finally the model is made inelastic and bounced again (Fig. 7f). We have animated diverse inelastic behaviors in Flatland, including buckling and collapse under load, swelling after impact (negative viscoelasticity), and numerous others.

It is possible to interact with a 3D elastic surface model on a Sun 3/160 workstation supported by a TAAC-1 coprocessor to accelerate the numerical

Table 2. Interactive model simulation details

Example	Formulation	Number of Equations	Numerics	Computer
Figure 7	Hybrid	100	Choleski	Lispm
Figure 8	Primal	300	ADI	SUN/TAAC

[4] Symbolics 3600 is a trademark of Symbolics, Inc.

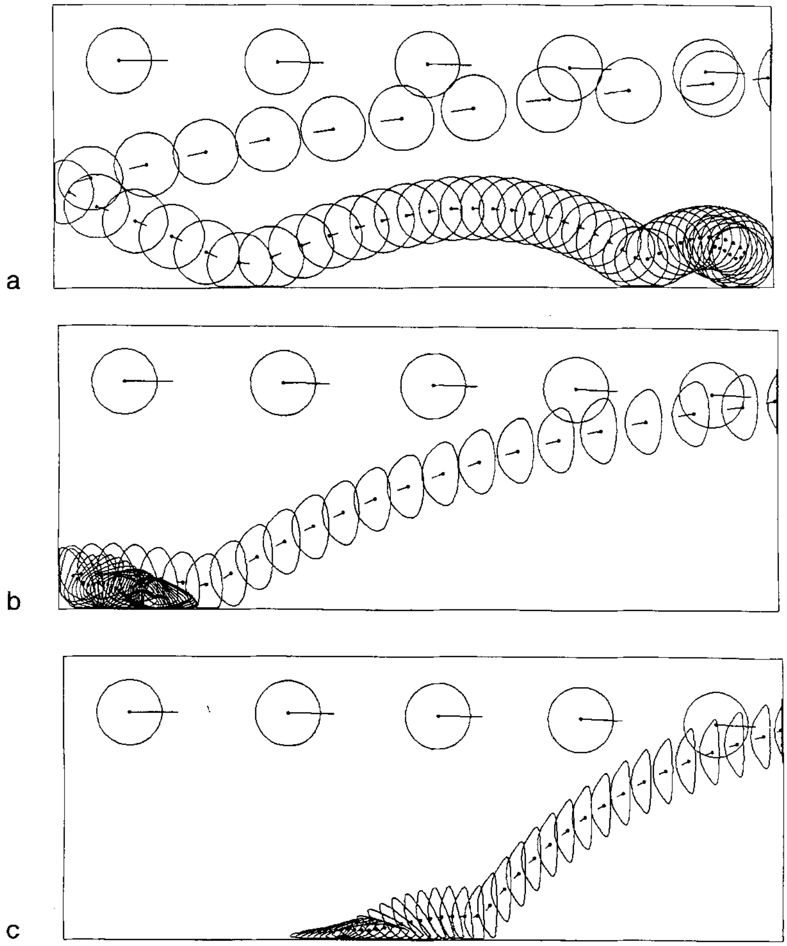


Fig. 7a–f. Interactive simulations in "Flatland". Models are "strobed" while undergoing motion subject to gravity, aerodynamic drag, and collisions against frictionless walls. Velocity vector of the center of mass (*dot*) is indicated. **a** Elastic model. **b** Viscoelastic model. **c** Highly viscoelastic model. **d** A viscoelastic circle is deformed into a free-form shape. **e** The shape is turned elastic and bounced. **f** Same shape made viscoelastic and bounced

solution[5]. The deformable surface is in a viscous ambient medium, and it reacts to an external time-varying force (Fig. 8). The numerical computations and visualization can be performed at interactive rates for a 10 × 10 surface model with shaded rendering, or with larger models rendered as wireframes. One version of this physically-based "3D-World" features a stereo head-mounted display and a Polhemus 3SPACE[6] device as a six degree-of-freedom mouse ("bat") for applying forces to the surface.

6.2 Elastic models

Figure 9 shows two different static behaviors of an elastic surface discretized on a 23 × 23 grid. The surface is lifted by a spring attached to the rightmost corner and constrained at the remaining corners. In Fig. 9a we specified the material property functions so as to simulate a thin plate, whose rest state is flat ($\mathbf{B}=\mathbf{0}, \mathbf{w}^1=\mathbf{0}, \mathbf{w}^2>\mathbf{0}$). In Fig. 9b we adjusted the material properties to simulate a membrane resistant to elongation or contraction away from the prescribed metric $\mathbf{G}^0$ and not resistant to bending ($\mathbf{w}^1>\mathbf{0}, \mathbf{w}^2=\mathbf{0}$); i.e., a thin cloth-like material.

Figure 10 shows a static simulation of cloth-like mesh material draped over a mannequin stand. The material is "pinned" at the shoulders and constrained by a "belt" around the waist. The realistic drapes and folds have not been specified kinematically. Rather, they are among the features that emerge naturally from the physically-based model.

Figure 11 illustrates what is in essence a physically-based method for constructing a convex hull. Fig-

[5] SUN 3/160 and TAAC-1 are trademarks of Sun Microsystems, Inc.

[6] 3SPACE is a trademark of McDonnell Douglas, Inc.

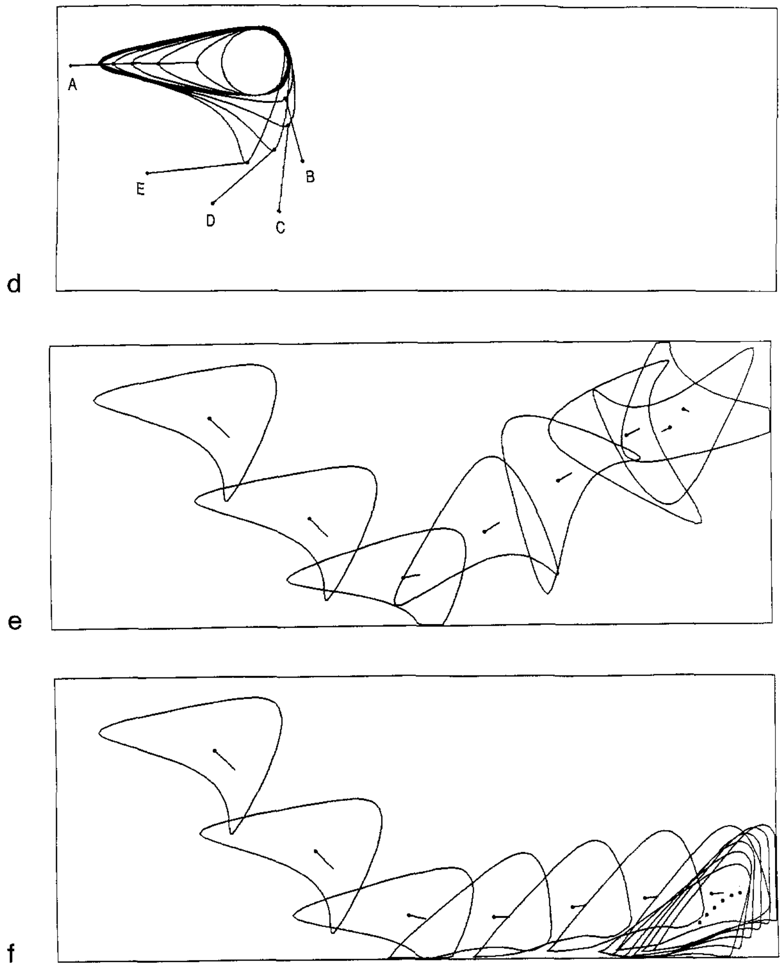


Fig. 7d–f. For legend see page 319

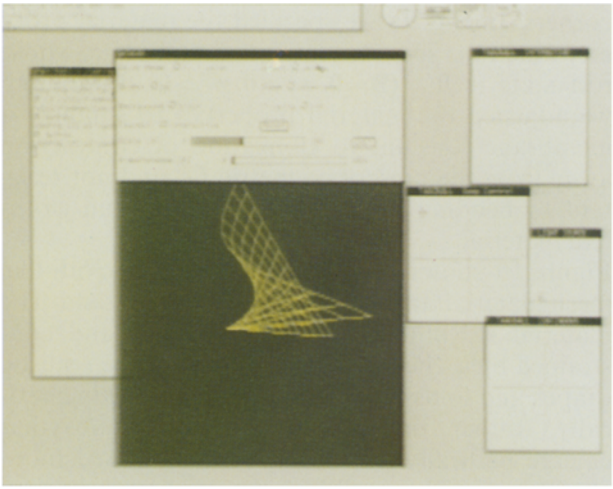
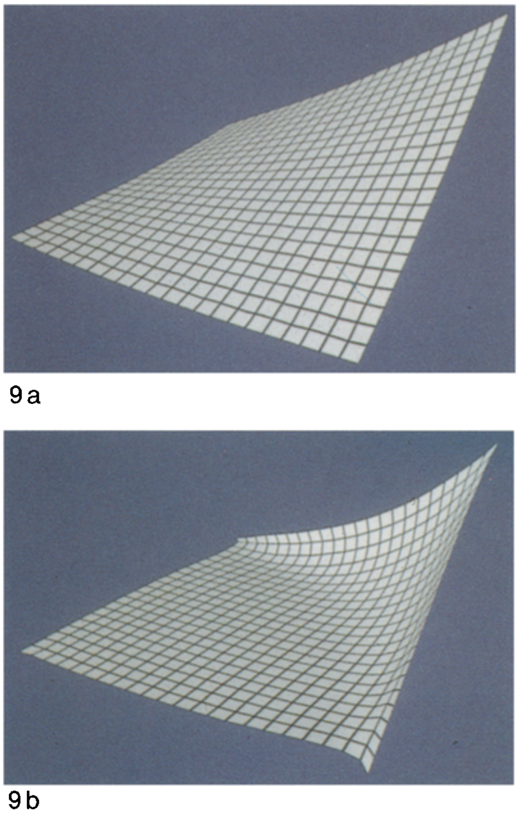


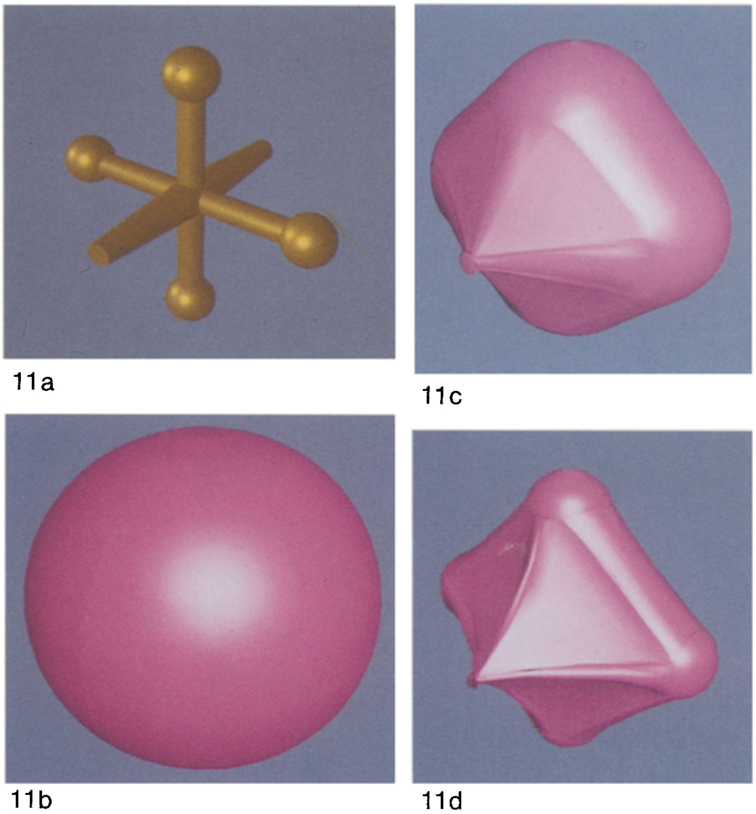
Fig. 8. 3D surface model simulations at interactive rates. A 10 × 10 elastic surface is subjected to a point force. The user sees continuous motion

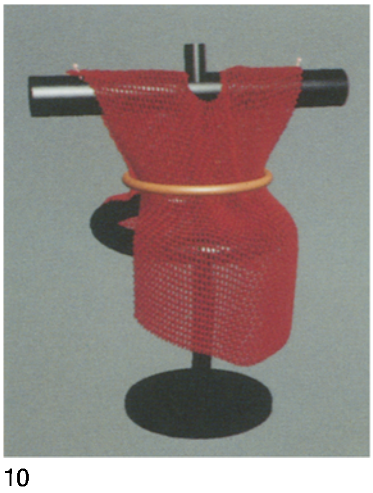
ure 11a shows a model of a rigid jack obstacle. In Fig. 11b, a membrane shrink-wrap $(\mathbf{G}=\mathbf{0}, \mathbf{w}^1 > \mathbf{0}, \mathbf{w}^2 = \mathbf{0})$ is stretched into a sphere around the jack. Figure 11c, d are stills from a motion sequence showing the membrane shrinking during the physical simulation. Shrinkage ceases when the membrane's surface tension comes into balance with the collision force surrounding the jack which prevents penetration.

Figure 12 illustrates a simulation of a flag waving in the wind. The flag material is a fixed metric membrane $(\mathbf{w}^1 > \mathbf{0}, \mathbf{w}^2 = \mathbf{0})$. The wind is constant and its effect on the flag is modeled by the aerodynamic force in (22). The flag is fixed to a rigid flagpole along one of its edges by imposing a fixed-position constraint (introduced as a Dirichlet boundary condition).

Figure 13 illustrates a simulation of a carpet falling







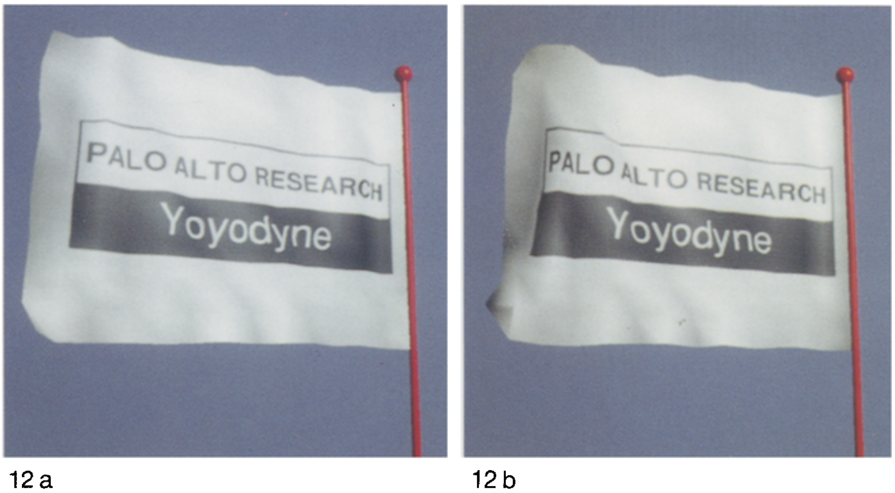


Fig. 9a, b. Lifting elastic surfaces

Fig. 10. Robe draped on a mannequin

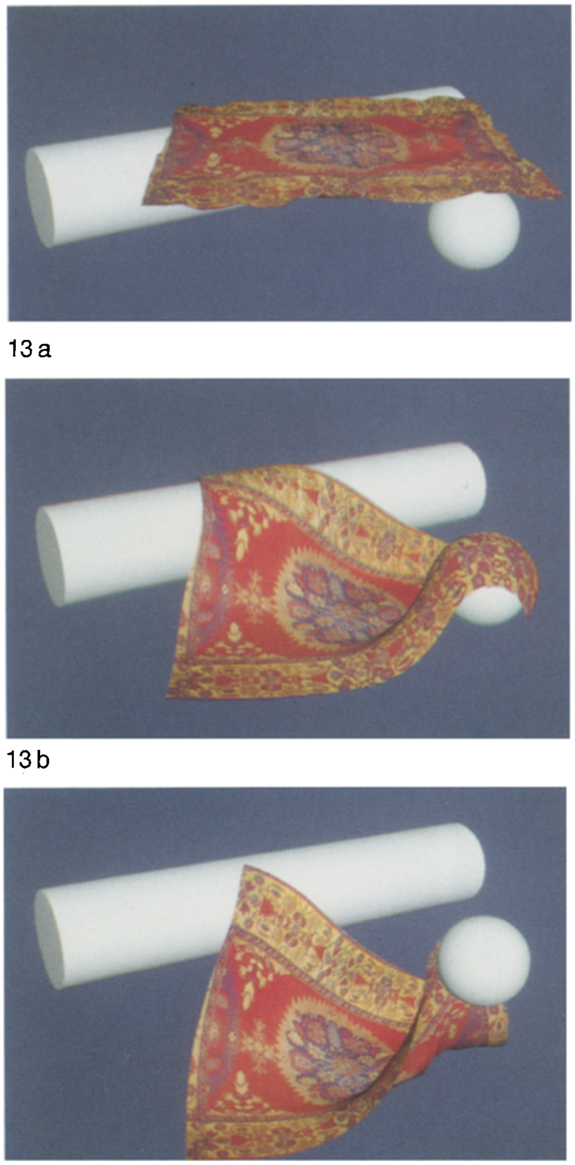
Fig. 11a–d. Membrane shrinking around a jack

Fig. 12a, b. Flag waving in a wind

onto two rigid bodies in a gravitational field. The carpet material is a prescribed-metric membrane with a small amount of plate rigidity ($\mathbf{w}^1 > \mathbf{0}, \mathbf{w}^2 > \mathbf{0}$). The carpet slides off the bodies due to the interaction between gravity and the repulsive collision forces.

The above simulations employed the primal formulation of deformable models. SOR-type iterative methods were used. The following few simulations employed the hybrid formulation.

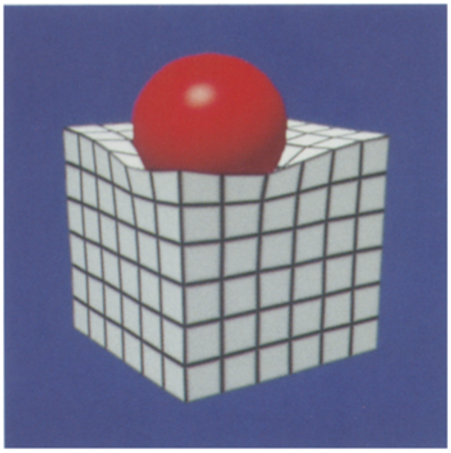
Figure 14 illustrates a heavy ball resting on a supporting elastic solid. The internal elastic force inter-



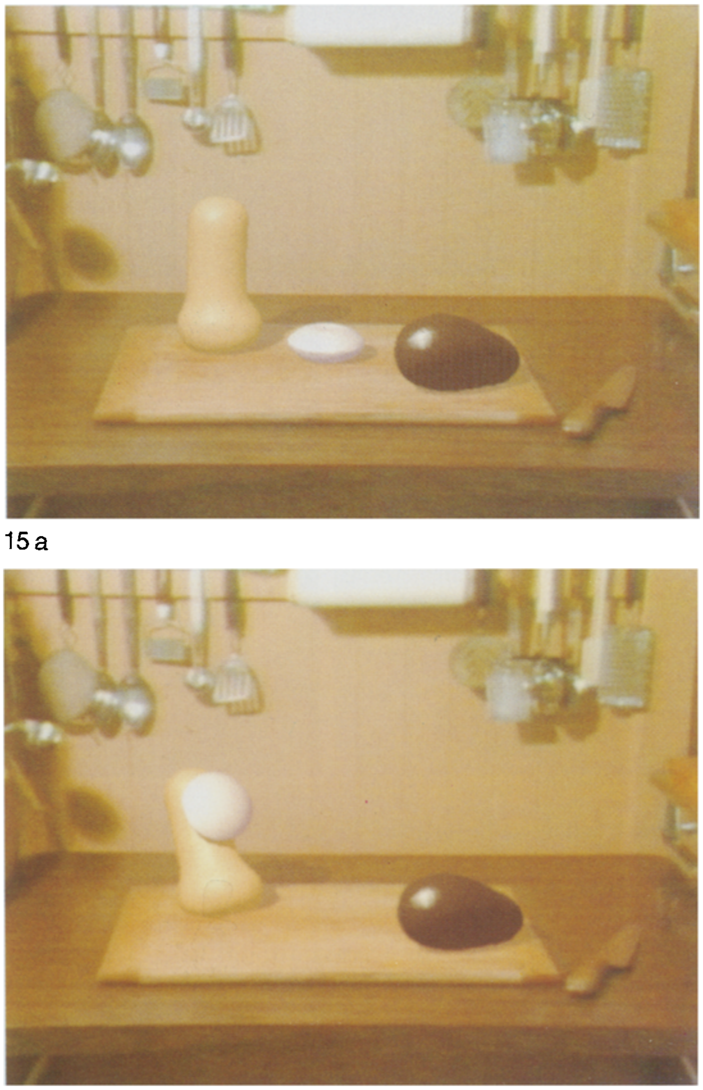
13 a

13 b

13 c



14



15 a

15 b

Fig. 13a–c. "Persian carpet" falling over immobile obstacles

Fig. 14. Heavy ball on a deformable solid

Fig. 15a, b. Stills from *Cooking with Kurt* acts with the collision force to deform the solid. The size of discrete model is 7 × 7 × 7.

Figure 15 shows frames from a physically-based animation entitled *Cooking with Kurt* (Fleischer et al. 1987). The animation starts with live video footage of Kurt placing several vegetables on a cutting board. The vegetables "come to life", bouncing around the kitchen table, colliding elastically with each another and with the table-top. The synthetic vegetables are deformable surface models. Their reference shapes were reconstructed from an image of the real vegetables using computer vision techniques, most of which are described by Terzopoulos et al. (1987b) and Kass et al. (1987). Our vision techniques exploit the "modeling clay" properties of deformable models. They provide principled ways of transforming raw image data into synthetic force fields that sculpt deformable models into shapes consistent with the imaged objects. After the models have captured the shapes of the real objects, we animate them in a physically-based, synthesized table-top world. The applied forces include driving forces (thrusters), attitude control forces (gyros), and interaction forces (collisions with friction). The models exhibit deformations, accelerations, collisions, tumbling, and other realistic physical motions. The figure shows an elastic collision in progress.

6.3 Inelastic models

Figure 16 simulates an interaction with a model of a "plasticine" bust. We initialized the reference component of the hybrid model formulation with 3D range data from a laser scanned sculpture of Victor Hugo (data source: Hansen and Henderson 1986). The figure shows the undeformed model, and a simulated robot hand pinching Hugo's plastic cheek with sticky fingers, pulling away, and releasing to show the residual plastic deformation. Because of the relatively large size of the discrete model, 180 × 127 we resorted to a multigrid solution method similar to the one described by Terzopoulos (1983).

The remaining examples feature fracture propagation in deformable models.

Figure 17 shows a 30 × 30 surface model sheared by opposing forces. In this simulation, we perturbed stochastically the fracture tolerance around the material's mean tolerance in order to encourage some unpredictability in the propagation of fractures.

Figure 18 presents an animation of a 23 × 23 net falling over an impenetrable obstacle in a gravitational field. Here, however, the mesh incorporates a fracture limit based on the deformation in the material. When a fiber stretches beyond the fracture limit, the fracture process breaks it by applying the discontinuity insertion technique described in Sect. 4. The yield limit is uniform over the mesh, which causes linear tears, as one might obtain with coarse cloth.

Figure 19 shows still frames from an animation of a 20 × 20 cloth-like deformable surface being torn gently at first, then violently, by opposing forces. In this simulation the material tolerance included some randomness. The surface is rendered as threads (with several interpolated threads per element).

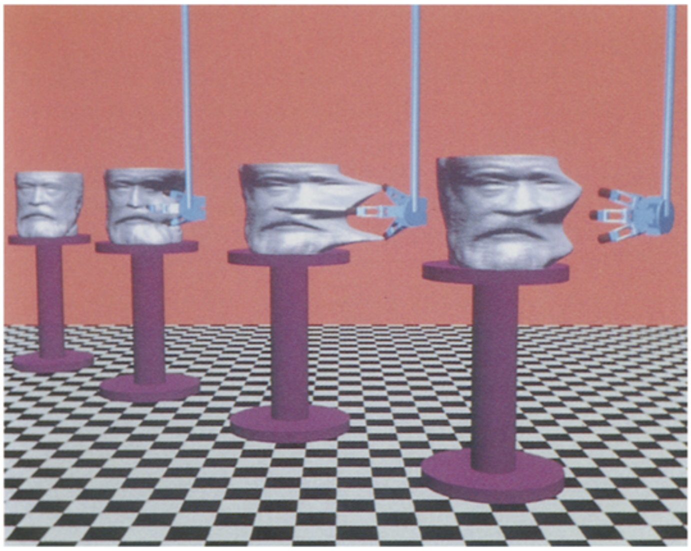
Figure 20 illustrates an implementation of inelastic deformable models on an 8K-processor CM-2 Connection Machine[7]. Local, parallel, iterative techniques such as relaxation are most suitable for achieving maximum performance on this massively parallel computer. Eight 16 × 16 × 4 inelastic solids were dropped simultaneously over identical spherical obstacles. The figure shows one of the solids fracturing into many fragments as it is strained by the "hot" obstacle.

Figure 21 shows a 3 × 3 × 7 inelastic "semi-liquid" mass propelled into a room. It collides with a spherical obstacle and the back wall before coming to rest asunder on the floor. This simulation is a serial version of the Connection Machine implementation.

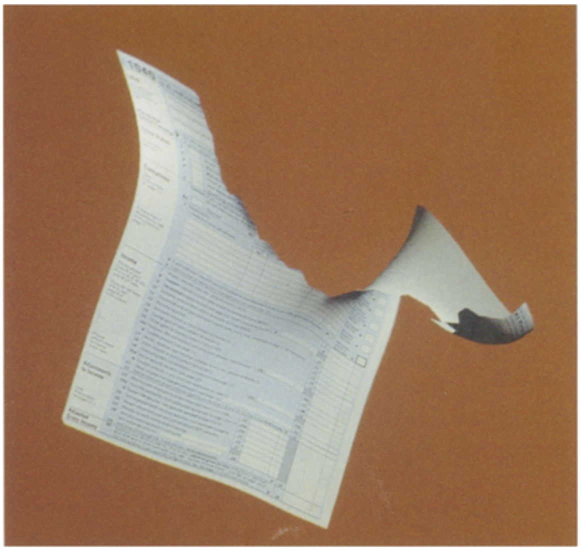
Table 3. Details about selected simulations

Example	Formulation	Number of Equations	Numerics	Computer
Figure 9	Primal	1587	Relaxation	Lispm
Figure 10	Primal	1800	Relaxation	Lispm
Figure 14	Hybrid	1029	Choleski	Lispm
Figure 15	Hybrid	507	ADI	Lispm
Figure 16	Hybrid	68580	Multigrid	Lispm
Figure 17	Primal	2700	ADI	Lispm
Figure 18	Primal	1587	ADI	Lispm
Figure 19	Primal	1200	ADI	Lispm
Figure 20	Primal	24576	Relaxation	CM-2
Figure 21	Primal	189	Relaxation	Lispm

[7] Connection Machine is a trademark of Thinking Machines, Inc.



16



17

Fig. 16. Hugo (*back to front*) A "plasticine" bust of Victor Hugo. Grabby hand pinches. Grabby hand pulls. Deformed Hugo

Fig. 17. April 15, 1988

The ("blobbies") technique described by Blinn (1982) was applied to render the simulation data in the previous two examples. We associate an exponential potential function with each node, and render the iso-potential surface of the resulting field. This rendering technique (similar to the one applied by Wyvill et al. (1986)) gives the inelastic solids a goopy appearance.

7 Summary and conclusion

We have proposed deformable models for use in computer graphics. Our goal has been to create physically-based models for nonrigid curves, surfaces, and solids that inherit the essential behavior of real-world deformable materials, while maintaining computational tractability for the purposes of computer animation. First we formulated elastic models as well as hybrid models that combine elastic deformation with rigid-body dynamics. Then we developed techniques for incorporating inelastic behavior into our models and applied them to create interesting elasticity, viscoelasticity, plasticity, and fracture effects in simulated physical worlds.

Deformable models are active: they respond to external influences and interact with other computer graphics objects in a natural way. Their underlying differential equations unify the description of shape and motion, and the numerical solution of these equations yields realistic statics and dynamics. Our results highlight the power of this approach – we are able to synthesize complex motions arising from the interaction of deformable models with diverse forces, ambient media, constraints, and impenetrable obstacles.

The addition of deformable models to the growing arsenal of computer graphics models that incorporate basic physical principles takes us to an opportune point. From here, high-level computer animation systems that can cope with both rigid and nonrigid objects seem attainable. In the near future, such systems will enable animators to plan physically accurate motions very conveniently, not through mere kinematic key-framing, but also according to abstract qualities such as timing, rhythm, and style. To realize such systems, we must develop methods for controlling and coordinating the motions of our deformable models in ways that appeal to animators. A branch of optimal control theory that deals with the control of nonrigid structures offers mathematical techniques applicable to this problem. This opens up an interesting area for future research.

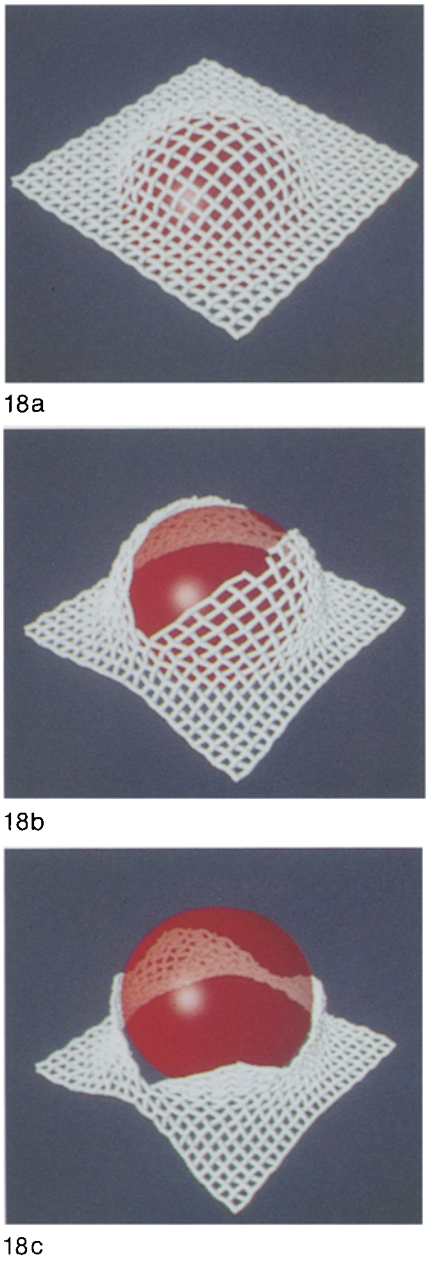


Fig. 18a–c. Net tearing over a spherical obstacle

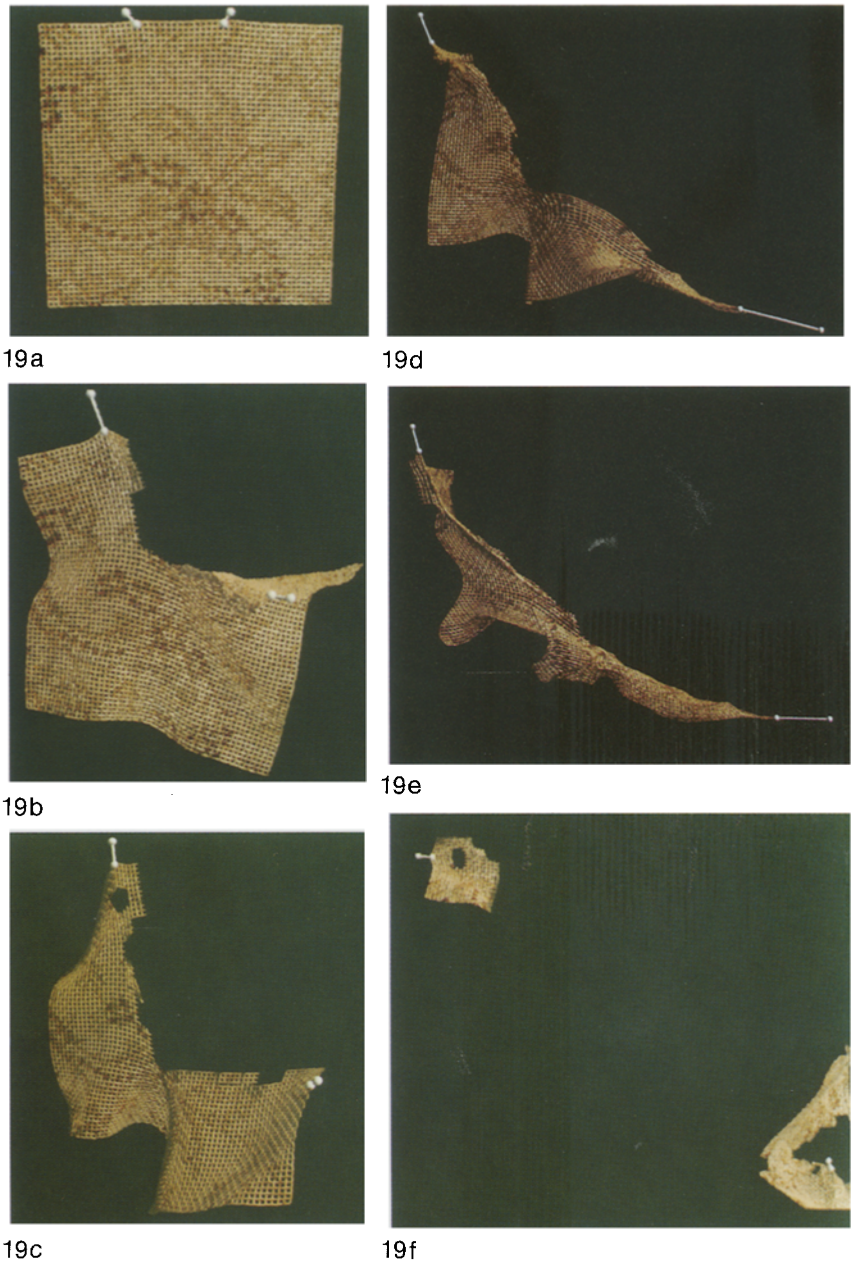
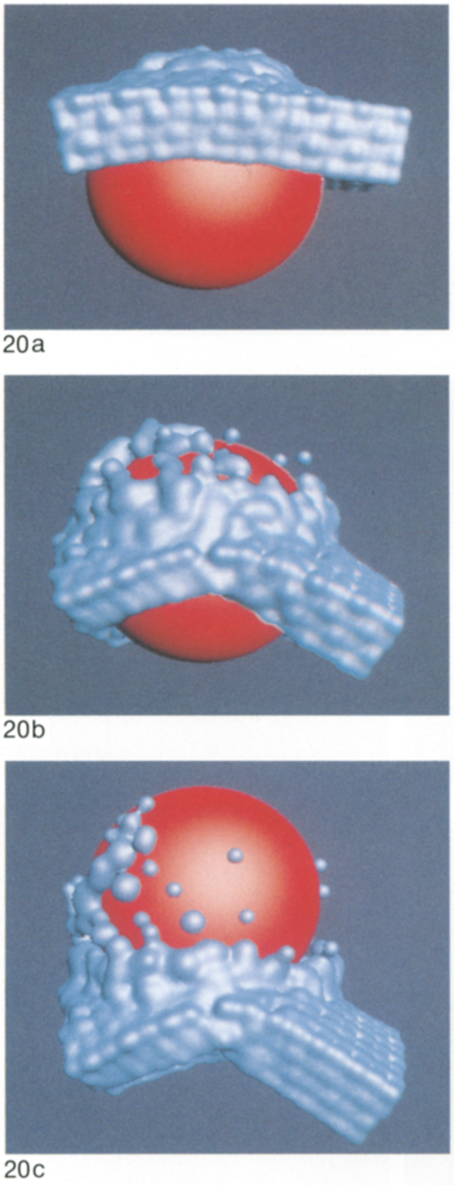


Fig. 19a–f. Opposing forces tearing a "woven cloth"

Acknowledgements. We thank our colleagues for their contributions to this research. John Platt and Alan Barr collaborated with us to develop the primal formulation of elastically deformable models. Andrew Witkin collaborated to develop the hybrid formulation. He and Michael Kass played major roles in the creation of *Cooking with Kurt*. Jay Freeman facilitated the porting of numerics written on the Lisp Machine to the SUN/TAAC workstation and implemented the Polhemus 3SPACE-based user interface to interactive deformable surfaces. To create audio accompaniment for some of our video animations, Tony Crossley synthesized digital soundtracks driven by deformable model simulation data. Thanks to Lewis Tucker and Thinking Machines, Inc., for the use of their facilities, and especially to Carl Feynman for helping to conceive CM-Goop and for implementing it on the Connection Machine. Robert Howe created the (Stanford-JPL) robot hand model that tortured "Hugo".



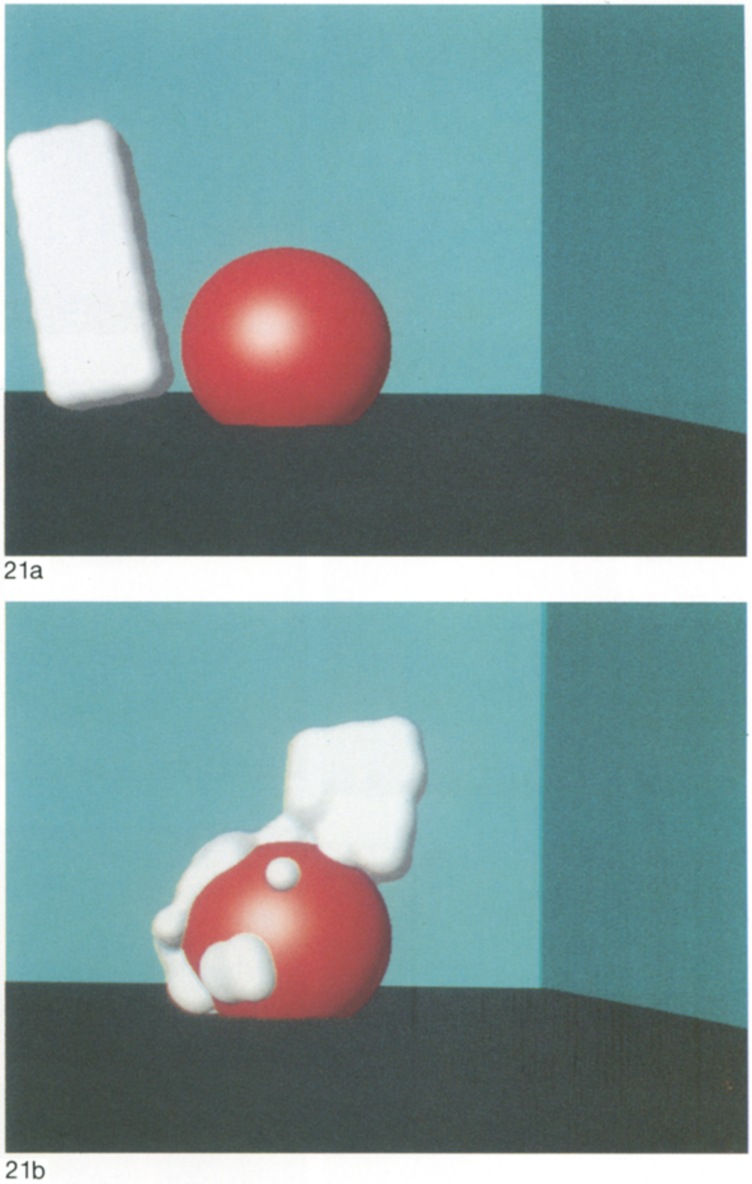
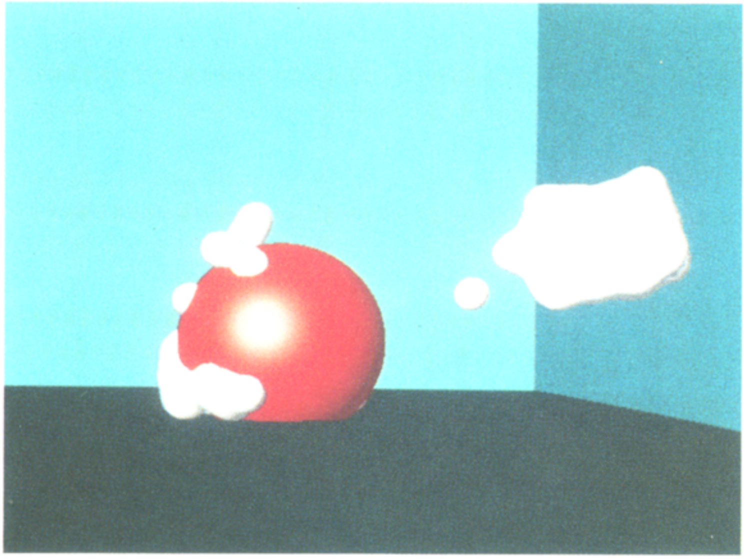


Fig. 20. Inelastic solid simulation on a massively parallel computer

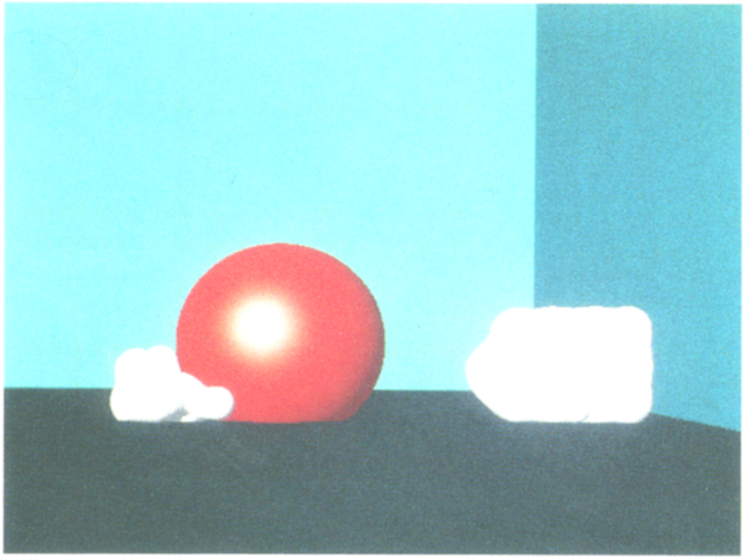
Fig. 21. Block of goop flung into a room

Appendix Implementation of deformable surfaces

This appendix presents the details of our implementation of deformable models for the case of elastic surfaces (i.e., $\mathrm{u}=(u_1, u_2)$ and $p=2$). We consider the primal formulation and the hybrid formulation in turn. In each case, we first give the elastic forces arising from the deformation energy $\mathcal{E}$, we discretize the continuous equations of motion in material coordinates and, finally, we offer procedures for integrating the resulting differential equations through time.



21c



21d

A.1 A simplified elastic force for the primal formulation

We will use a weighted matrix norm in (9) to obtain the following simplified deformation energy for a surface:

$$\mathcal{E}(\mathbf{x}) = \int_{\Omega} \sum_{i,j=1}^{2} (w_{ij}^{1}(G_{ij} - G_{ij}^{0})^2 + w_{ij}^{2}(B_{ij} - B_{ij}^{0})^2)\, du_1\, du_2. \tag{25}$$

The first variational derivative $\delta_{\mathbf{x}} \mathcal{E}(\mathbf{x})$ of (25) can be approximated by the vector expression:

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \sum_{i,j=1}^{2} -\frac{\partial}{\partial u_i}\left(\alpha_{ij} \frac{\partial \mathbf{x}}{\partial u_j}\right) + \frac{\partial^2}{\partial u_i\, \partial u_j}\left(\beta_{ij} \frac{\partial^2 \mathbf{x}}{\partial u_i\, \partial u_j}\right), \tag{26}$$

where the functions $\alpha_{ij}(\mathrm{u}, \mathbf{x})$ and $\beta_{ij}(\mathrm{u}, \mathbf{x})$ determine the elastic properties of the material. We use

$$\alpha_{ij}(\mathrm{u}, \mathbf{x}) = w_{ij}^{1}(\mathrm{u})\,(G_{ij} - G_{ij}^{0}) \tag{27}$$

and after linearization

$$\beta_{ij}(\mathrm{u}, \mathbf{x}) = w_{ij}^{2}(\mathrm{u})\,(B_{ij} - B_{ij}^{0}). \tag{28}$$

Note that for the special case where $\alpha_{12}=\alpha_{21}=0$ and where α_{11}, α_{22}, and the β_{ij} are linearized so as to be independent of $\mathbf{r}$, we obtain the "thin plate surface under tension" (Terzopoulos 1986). The thin plate surface under tension further reduces to the traditional "spline under tension" in the case of curves.

A.2 Semidiscretization

Expression (26) for the elastic force is continuous in the material coordinates of the deformable surface. To simulate the dynamics of the model, we can discretize the expression by applying finite-element or finite-difference approximation methods. Discretization transforms the partial differential equation of motion (3) into a system of ordinary differential equations. We will illustrate the discretization step using standard finite-difference approximations.

We discretize the unit square domain $\Omega = 0 \leq u_1, u_2 \leq 1$ of the surface as a regular $M \times N$ discrete grid Ω^h of nodes. The internode spacings are $h_1 = 1/(M-1)$ and $h_2 = 1/(N-1)$ in the u_1 and u_2 coordinate directions respectively. Nodes are indexed by integers $[m, n]$ where $0 \leq m \leq M$ and $0 \leq n \leq N$. We discretize an arbitrary continuous vector function of $\mathbf{u}(\mathrm{u}, t)$ by arrays of continuous-time vector-valued nodal variables: $\mathbf{u}_t[m, n] = \mathbf{u}(mh_1, nh_2, t)$. For notational convenience we will denote the collection of nodal variables $\mathbf{u}_t[m, n]$ by the MN dimensional grid vector $\underline{\mathbf{u}}_t$. For now, we shall suppress the time-dependence notation until we consider integration through time in the next section.

The elastic force requires approximations to the first and second derivatives of the nodal variables. Given a grid function $\mathbf{u}[m, n]$, we define the forward first difference operators

$$D_1^+ \mathbf{u}[m, n] = (\mathbf{u}[m+1, n] - \mathbf{u}[m, n])/h_1$$
$$D_2^+ \mathbf{u}[m, n] = (\mathbf{u}[m, n+1] - \mathbf{u}[m, n])/h_2 \qquad (29)$$

and the backward first difference operators

$$D_1^- \mathbf{u}[m, n] = (\mathbf{u}[m, n] - \mathbf{u}[m-1, n])/h_1$$
$$D_2^- \mathbf{u}[m, n] = (\mathbf{u}[m, n] - \mathbf{u}[m, n-1])/h_2. \qquad (30)$$

Using (29) and (30), we can define the forward and backward cross difference operators

$$D_{12}^+ \mathbf{u}[m, n] = D_{21}^+ \mathbf{u}[m, n] = D_1^+ D_2^+ \mathbf{u}[m, n],$$
$$D_{12}^- \mathbf{u}[m, n] = D_{21}^- \mathbf{u}[m, n] = D_1^- D_2^- \mathbf{u}[m, n], \qquad (31)$$

and the central second difference operators

$$D_{11} \mathbf{u}[m, n] = D_1^- D_1^+ \mathbf{u}$$
$$D_{22} \mathbf{u}[m, n] = D_2^- D_2^+ \mathbf{u}. \qquad (32)$$

Now, using the grid functions $\mathbf{x}[m, n]$, $w_{ij}^1[m, n]$, $w_{ij}^2[m, n]$ to represent their continuous counterparts, we apply the above difference operators to discretize the functions (27) and (28) as follows:

$$a_{ij}[m, n] = w_{ij}^1[m, n]\,(D_i^+ \mathbf{x}[m, n] \cdot D_j^+ \mathbf{x}[m, n] - G_{ij}^0[m, n]),$$
$$b_{ij}[m, n] = w_{ij}^2[m, n]\,(\mathbf{n}[m, n] \cdot D_{ij}^{(+)} \mathbf{x}[m, n] - B_{ij}^0[m, n]), \qquad (33)$$

where the $(+)$ superscript indicates that the forward cross difference operator is used when $i \neq j$, and, from (6),

$$\mathbf{n}[m, n] = \frac{D_1^+ \mathbf{x}[m, n] \times D_2^+ \mathbf{x}[m, n]}{|D_1^+ \mathbf{x}[m, n] \times D_2^+ \mathbf{x}[m, n]|} \qquad (34)$$

is the surface normal grid function. The elastic force (26) can then be approximated by

$$\boldsymbol{\varepsilon}[m, n] = \sum_{i,j=1}^{2} -D_i^- (a_{ij} D_j^+ \mathbf{x}[m, n]) + D_{ij}^{(-)}(b_{ij} D_{ij}^{(+)} \mathbf{x}[m, n]). \qquad (35)$$

We introduce free (natural) boundary conditions by nullifying the inner difference operators in (35). To introduce fractures in the surface, a free boundary condition can be inserted by setting to zero the value of any difference operator D_i^+ or $D_{ij}^{(+)}$ in (35) involving $\mathbf{x}[m, n]$ on opposite sides of a fracture. Such boundary conditions are also established on the free edges of the surface, where these operators would attempt to access nodal variables $\mathbf{e}[m, n]$ outside the discrete domain Ω^h.

Expressing the grid functions $\mathbf{x}[m, n]$ and $\boldsymbol{\varepsilon}[m, n]$ as $\underline{\mathbf{x}}$ and $\underline{\boldsymbol{\varepsilon}}$ in our grid vector notation, (35) may be written in the vector form

$$\underline{\boldsymbol{\varepsilon}} = \mathbf{K}(\underline{\mathbf{x}})\,\underline{\mathbf{x}}, \qquad (36)$$

where $\mathbf{K}(\underline{\mathbf{x}})$ is an $MN \times MN$ matrix known as the *stiffness matrix*. Due to the local support of the finite-difference discretization on the $[m, n]$ grid, $\mathbf{K}$ has the desirable computational properties of sparseness and bandedness.

Next we discretize the mass density $\mu(u_1, u_2)$ and damping density $\gamma(u_1, u_2)$ as grid functions $\mu[m, n]$ and $\gamma[m, n]$. Let $\mathbf{M}$ be the *mass matrix*, a diagonal $MN \times NM$ matrix with the $\mu[m, n]$ variables as diagonal components, and let $\mathbf{C}$ be the *damping matrix* constructed similarly from $\gamma[m,n]$. The discrete counterpart to (3) can be expressed in grid vector form using (36) by the simultaneous system of second-order ordinary differential equations

$$\mathbf{M}\frac{d^2\underline{\mathbf{x}}}{dt^2}+\mathbf{C}\frac{d\underline{\mathbf{x}}}{dt}+\mathbf{K}(\underline{\mathbf{x}})\,\underline{\mathbf{x}}=\underline{\mathbf{f}}(\underline{\mathbf{x}}), \tag{37}$$

where the next external force on the surface $\mathbf{f}(u_1,u_2)$ has been discretized into the grid vector $\underline{\mathbf{f}}$ which represents the grid function $\mathbf{f}[m,n]$.

A.3 Numerical integration through time

To simulate the dynamics of an elastic model, we integrate the system of ordinary differential equations (37) through time using a step-by-step numerical procedure.

Evaluating $\underline{\boldsymbol{\varepsilon}}$ at $t+\Delta t$ and $\underline{\mathbf{f}}$ at t, and substituting the discrete-time approximations

$$\frac{d^2\underline{\mathbf{x}}}{dt^2}\approx(\underline{\mathbf{x}}_{t+\Delta t}-2\underline{\mathbf{x}}_t+\underline{\mathbf{x}}_{t-\Delta t})/\Delta t^2$$

$$\frac{d\underline{\mathbf{x}}}{dt}\approx(\underline{\mathbf{x}}_{t+\Delta t}-\underline{\mathbf{x}}_{t-\Delta t})/2\Delta t \tag{38}$$

into (37), we obtain the semi-implicit integration procedure

$$\mathbf{A}_t\,\underline{\mathbf{x}}_{t+\Delta t}=\underline{\mathbf{g}}_t, \tag{39}$$

where the $MN\times MN$ matrix

$$\mathbf{A}_t(\underline{\mathbf{x}}_t)=\mathbf{K}(\underline{\mathbf{x}}_t)+\left(\frac{1}{\Delta t^2}\mathbf{M}+\frac{1}{2\Delta t}\mathbf{C}\right) \tag{40}$$

and the effective force vector

$$\begin{aligned}\underline{\mathbf{g}}_t=\underline{\mathbf{f}}_t&+\left(\frac{1}{\Delta t^2}\mathbf{M}+\frac{1}{2\Delta t}\mathbf{C}\right)\underline{\mathbf{x}}_t\\&+\left(\frac{1}{\Delta t}\mathbf{M}-\frac{1}{2\Delta t}\mathbf{C}\right)\dot{\underline{\mathbf{x}}}_t,\end{aligned} \tag{41}$$

with

$$\dot{\underline{\mathbf{x}}}_t=(\underline{\mathbf{x}}_t-\underline{\mathbf{x}}_{t-\Delta t})/\Delta t. \tag{42}$$

Applying the above semi-implicit procedure, we can evolve the dynamic solution from given initial conditions $\underline{\mathbf{x}}_0$ and $\dot{\underline{\mathbf{x}}}_0$ at $t=0$. During each time step, we solve a sparse linear algebraic system (39) for the instantaneous configuration $\underline{\mathbf{x}}_{t+\Delta t}$ using the preceding solution $\underline{\mathbf{x}}_t$ and $\dot{\underline{\mathbf{x}}}_t$.

A.4 Implementing the hybrid formulation

Implementation of the hybrid formulation follows the same steps described above. For a surface the variational derivative of (18) is

$$\begin{aligned}\delta_{\mathbf{e}}\mathcal{E}(\mathbf{e})=w_{00}\,\mathbf{e}&-\frac{\partial}{\partial u_1}\left(w_{10}\frac{\partial\mathbf{e}}{\partial u_1}\right)-\frac{\partial}{\partial u_2}\left(w_{01}\frac{\partial\mathbf{e}}{\partial u_2}\right)\\&+\frac{\partial^2}{\partial u_1^2}\left(w_{20}\frac{\partial^2\mathbf{e}}{\partial u_1^2}\right)\\&+2\frac{\partial^2}{\partial u_1\,\partial u_2}\left(w_{11}\frac{\partial^2\mathbf{e}}{\partial u_1\,\partial u_2}\right)\\&+\frac{\partial^2}{\partial u_2^2}\left(w_{02}\frac{\partial^2}{\partial u_2^2}\right),\end{aligned} \tag{43}$$

where $\mathrm{u}=(u_1,u_2)$ are the surface's material coordinates.

Using the finite-difference operators (29–32) the discrete form of the above expression is

$$\begin{aligned}\boldsymbol{\varepsilon}[m,n]=w_{00}\,\mathbf{e}[m,n]&-D_1^-(w_{10}\,D_1^+\,\mathbf{e})\,[m,n]\\&-D_2^-(w_{01}\,D_2^+\,\mathbf{e})\,[m,n]\\&+D_{11}(w_{20}\,D_{11}\,\mathbf{e})\,[m,n]\\&+2D_{12}^-(w_{11}\,D_{12}^+\,\mathbf{e})\,[m,n]\\&+D_{22}(w_{02}\,D_{22}\,\mathbf{e})\,[m,n].\end{aligned} \tag{44}$$

As before, this may be written in the grid vector form

$$\underline{\boldsymbol{\varepsilon}}=\mathbf{K}\underline{\mathbf{e}}. \tag{45}$$

Unlike (36), however, the sparse, banded stiffness matrix $\mathbf{K}$ is constant.

Using (45), the equations of motion (15) can be expressed in semidiscrete form by the following system of coupled ordinary differential equations:

$$m\frac{d\mathbf{v}}{dt}=\mathbf{g}^{\mathbf{v}}, \tag{46a}$$

$$\frac{d}{dt}(\mathbf{I}\omega)=\mathbf{g}^{\boldsymbol{\omega}}, \tag{46b}$$

$$\mathbf{M}\frac{d^2\underline{\mathbf{e}}}{dt^2}+\mathbf{C}\frac{d\underline{\mathbf{e}}}{dt}+\mathbf{K}\underline{\mathbf{e}}=\underline{\mathbf{g}}^{\mathbf{e}}, \tag{46c}$$

where

$$\mathbf{g}^{\mathbf{v}}=h_1\,h_2\left(\sum_{m,n}\underline{\mathbf{f}}-\frac{d}{dt}\sum_{m,n}\mu\frac{d\underline{\mathbf{e}}}{dt}-\sum_{m,n}\gamma\frac{d\underline{\mathbf{x}}}{dt}\right), \tag{47a}$$

$$\begin{aligned}\mathbf{g}^{\boldsymbol{\omega}}=h_1\,h_2\Bigg(&\sum_{m,n}\underline{\mathbf{q}}\times\underline{\mathbf{f}}-\frac{d}{dt}\sum_{m,n}\mu\underline{\mathbf{q}}\times\frac{d\underline{\mathbf{e}}}{dt}\\&-\sum_{m,n}\gamma\underline{\mathbf{q}}\times\frac{d\underline{\mathbf{x}}}{dt}\Bigg),\end{aligned} \tag{47b}$$

$$\underline{\mathbf{g}}^{\mathbf{e}} = \underline{\mathbf{f}} - \mu \frac{d\mathbf{v}}{dt} - \mu \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \underline{\mathbf{q}})$$

$$-2\mu\boldsymbol{\omega} \times \frac{d\underline{\mathbf{e}}}{dt} + \mu \frac{d\boldsymbol{\omega}}{dt} \times \underline{\mathbf{q}}. \qquad (47\,\mathrm{c})$$

Note that the integrals in (15–17) have been approximated by sums over nodal variables. Some of the terms in (15c) have been brought to the right hand side in order to simplify the final step of the solution process.

To simulate the dynamics of our model, we integrate the semidiscrete system (46) through time, evolving the rigid-body and deformation dynamics from given initial conditions $\mathbf{v}_0$, $\boldsymbol{\omega}_0$, $\underline{\mathbf{e}}_0$, and $(d\underline{\mathbf{e}}/dt)_0$ at $t=0$. Each time-step requires the solution of two algebraic equations for $\mathbf{v}$ and $\boldsymbol{\omega}$ for the motion of the body frame ϕ, in tandem with a linear algebraic system for the displacement component $\underline{\mathbf{e}}$. Substituting the discrete time approximation $(d\mathbf{v}/dt) \approx (\mathbf{v}_{t+\Delta t} - \mathbf{v}_t)/\Delta t$ into (46a), we obtain the integration procedure

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \Delta t\, \mathbf{g}_t^{\mathbf{v}}/m \qquad (48)$$

for the linear velocity of ϕ at the next time instant. Similarly, we obtain from (46b)

$$\boldsymbol{\omega}_{t+\Delta t} = \mathbf{I}_{t+\Delta t}^{-1}(\mathbf{I}_t\, \boldsymbol{\omega}_t + \Delta t\, \mathbf{g}_t^{\boldsymbol{\omega}}) \qquad (49)$$

for the angular velocity of ϕ for the next time instant. At each time step, the body is translated by $\mathbf{d} = \Delta t\, \mathbf{v}_t$, and it is rotated by an angle of $\theta = \Delta t |\boldsymbol{\omega}_t|$ about the unit vector $\mathbf{a} = [a_1, a_2, a_3]' = \boldsymbol{\omega}_t / |\boldsymbol{\omega}_t|$ using the transformation matrix

$$\mathbf{R} = \begin{pmatrix} a_1 a_1 \operatorname{vers}\theta + \cos\theta & a_1 a_2 \operatorname{vers}\theta - a_3 \sin\theta & a_1 a_3 \operatorname{vers}\theta + a_2 \sin\theta \\ a_2 a_1 \operatorname{vers}\theta + a_3 \sin\theta & a_2 a_2 \operatorname{vers}\theta + \cos\theta & a_2 a_3 \operatorname{vers}\theta - a_1 \sin\theta \\ a_3 a_1 \operatorname{vers}\theta - a_2 \sin\theta & a_3 a_2 \operatorname{vers}\theta + a_1 \sin\theta & a_3 a_3 \operatorname{vers}\theta + \cos\theta \end{pmatrix}, \qquad (50)$$

where $\operatorname{vers}\theta = (1 - \cos\theta)$ denotes the versine of θ. The reference and displacement components are transformed as follows:

$$\underline{\mathbf{r}}_{t+\Delta t} = \mathbf{R}\,\underline{\mathbf{r}}_t + \mathbf{d}; \qquad \underline{\mathbf{e}}_{t+\Delta t} = \mathbf{R}\,\underline{\mathbf{e}}_t. \qquad (51)$$

Next, substituting the discrete-time approximations (38) into (46c), we obtain the semi-implicit recursive procedure

$$\mathbf{A}\,\underline{\mathbf{e}}_{t+\Delta t} = \underline{\mathbf{g}}_t, \qquad (52)$$

where the matrix $\mathbf{A}$ is given by the right hand side of (40), but with constant $\mathbf{K}$ from (45), while the effective force vector $\underline{\mathbf{g}}_t$ is given by the right hand side of (41) with $\underline{\mathbf{g}}_t^{\mathbf{e}}$ from (47c) replacing $\underline{\mathbf{f}}_t$ and $\underline{\mathbf{e}}$ replacing $\underline{\mathbf{x}}$.

In the hybrid formulation, matrix $\mathbf{A}$ is constant (also in the primary formulation for the special case of quadratic energies); hence, the direct solution method need factorize it only once at $t=0$, then merely resolve the vector $\underline{\mathbf{g}}_t$ at succeeding time steps, thereby saving substantial computation. Further savings are possible by neglecting some of the interaction terms on the right hand sides of (47); for example, the centrifugal force may be neglected unless large angular velocities $\boldsymbol{\omega}$ are expected, while the Coriolis force may be neglected unless significant $d\underline{\mathbf{e}}/dt$ is expected.

References

Alfrey T (1947) Mechanical behavior of high polymers. Interscience, New York, NY

Armstrong WW, Green M (1985) The dynamics of articulated rigid bodies for purposes of animation. The Visual Computer 1:231–240

Barr AH (1984) Global and local deformations of solid primitives. Computer Graphics (Proc SIGGRAPH) 18(3):21–29

Barr A, Barzel R, Haumann D, Kass M, Platt J, Terzopoulos D, Witkin A (1987) Topics in physically-based modeling. ACM SIGGRAPH '87 Course Notes, vol 17, Anaheim, CA

Bartels RH, Beatty JC, Barsky BA (1987) An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann, Los Altos, CA

Barzel R, Barr A (1987) Dynamic Constraints. In: Barr A., et al., (eds) (1987) Topics in physically-based modeling ACM SIGGRAPH '87 Course Notes, vol 17, Anaheim, CA

Blinn JF (1982) A generalization of algebraic surface drawing. ACM Trans Graph 1:235–256

De Boor C (1978) A practical guide to splines. Springer, Berlin Heidelberg New York

Do Carmo MP (1974) Differential geometry of curves and surfaces. Prentice-Hall, Englewood Cliffs, NJ

Christiansen HN (1974) Computer generated displays of structures in vibration. The Shock and Vibration Bulletin 44(2):185–192

Christiansen HN, Benzley SE (1982) Computer graphics displays of nonlinear calculations. Computer methods in applied mechanics and engineering 34:1037–1050

Christensen RM (1982) Theory of viscoelasticity, 2nd ed. Academic Press, New York

Courant R, Hilbert D (1953) Methods of mathematical physics, vol 1. Interscience, London

Faux JD, Pratt MJ (1981) Computational geometry for design and manufacture. Halstead Press, Horwood, NY

Feynman CR (1986) Modeling the Appearance of Cloth, MSc thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA

Fleischer K, Witkin A (1988) A modeling testbed. Proc Graphics Interface '88. Edmonton, Canada, pp 127–137

Fleischer K, Witkin A, Kass M, Terzopoulos D (1987) Cooking with Kurt. An animated video. Schlumberger Palo Alto Research, Palo Alto, CA

Fournier A, Bloomenthal J, Oppenheimer P, Reeves WT, Smith AR (1987) The modeling of natural phenomena. ACM SIGGRAPH '87 Course Notes, vol 16. Anaheim, CA

Girard M, Maciejewski AA (1985) Computational modeling for the computer animation of legged figures. Computer Graphics (Proc SIGGRAPH) 19(3):263–270

Goldstein H (1980) Classical mechanics, 2nd edn. Addison-Wesley, Reading, MA

Hackbusch W (1985) Multigrid methods and applications. Springer, Berlin Heidelberg New York

Hansen C, Henderson T (1986) UTAH Range Database, Dept Comput, Univ Utah, Salt Lake City, Utah, TR No. UUCS-86-113

Haumann D (1987) Modeling the physical behavior of flexible objects. In: Barr A., et al., (eds) (1987) Topics in physically-based modeling ACM SIGGRAPH '87 Course Notes, vol 17, Anaheim, CA

Hoffmann CM, Hopcroft JE (1987) Simulation of physical systems from geometric models. IEEE J Robotics and Automation, RA-3(3):194–206

Hunter SC (1983) Mechanics of Continuous Media, 2nd edn. Ellis Horwood, Chishter, England

Issacs PM, Cohen MF (1987) Controlling dynamic simulation with kinematic constraints, behavior functions, and inverse dynamics. Computer Graphics (Proc SIGGRAPH) 21(4):215–224

Kardestuncer H, Norrie DH (eds) (1987) Finite element handbook. McGraw-Hill, New York

Kass M, Witkin A, Terzopoulos D (1987) Snakes: Active contour models. Int J Comput Vision 1:321–331

Landau LD, Lifshitz EM (1959) Theory of elasticity. Pergamon Press, London, UK

Lapidus L, Pinder GF (1982) Numerical solution of partial differential equations in science and engineering. Wiley, New York

Lassiter J (1987) Principles of traditional animation applied to 3D computer animation. Computer Graphics (Proc SIGGRAPH) 21(4):35–44

Luenberger DG (1973) Introduction to linear and nonlinear programming. Addison-Wesley, Reading, MA

Lundin D (1987) Ruminations of a model maker. IEEE Comput Graph Appl 7(5):3–5

Mendelson A (1968) Plasticity – theory and application. Macmillan, New York

Press WH, Flannery BP, Teukolsky SA , Vetterling WT (1986) Numerical recipes: the art of scientific computing. Cambridge University Press, Cambridge, UK

Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. Comput Graph (Proc SIGGRAPH) 20(4):151–160

Shephard MS, Abel JF (1987) Interactive computer graphics for CAD/CAM. In: Kardestuncer H, Norrie DH, (eds) (1987) Finite Element Handbook McGraw-Hill, New York, pp 4.233–4.254

Sih GC (1981) Mechanics of fracture. Martinus Nijhoff, The Hague

Terzopoulos D (1983) Multilevel computational processes for visual surface reconstruction. Computer Vision, Graphics, and Image Processing 24:52–96

Terzopoulos D (1986) Regularization of inverse visual problems involving discontinuities. IEEE Trans Pattern Anal Mach Intell PAMI-8:413–424

Terzopoulos D (1988) The computation of visible-surface representations. IEEE Trans Pattern Anal Mach Intell PAMI-10:417–438

Terzopoulos D, Platt J, Barr A, Fleischer K (1987a) Elastically deformable models. Comput Graph (Proc. SIGGRAPH) 21(4):205–214

Terzopoulos D, Witkin A (1988) Physically-based models with rigid and deformable components. IEEE Comput Graph Appl 8(6):41–51

Terzopoulos D, Witkin A, Kass M (1987b) Symmetry-seeking models and 3D object reconstruction. Int J Comput Vision 1:211–221

Weil J (1986) The synthesis of cloth objects. Comput Graph (Proc SIGGRAPH) 20(4):49–54

Wilhelms J (1987) Using dynamic analysis for realistic animation of articulated bodies. IEEE Comput Graph Appl 7(6):12–27

Wilhelms J, Barsky BA (1985) Using dynamic analysis to animate articulated bodies such as humans and robots. Proc Graphics Interface '85. Montreal, Canada, pp 97–104

Wyvill B, McPheeters C, Wyvill G (1986) Animating soft objects. The Visual Computer 2:235–242

Zienkiewicz OC (1977) The finite element method, 3rd edn. McGraw-Hill, London

Note added in proof

Recently we have extended our formulation of deformable models to include heat conduction. The extended models feature thermoelasticity, melting, and fluid-like behavior in the molten state. In addition to their nonrigid dynamics, governed by Lagrangian equations of motion, the models transfer heat through their interiors according to the heat equation for nonhomogeneous, nonisotropic conductive media. We have implemented the models in their solid state as hexahedral finite-element assemblies in which thermoelastic elements interconnect mass particles situated in a lattice. As the temperature increases, the stiffness of a thermoelastic element decreases and eventually the element fractures when its temperature exceeds the melting point. In its molten state the deformable model involves a many-body simulation in which "fluid" particles that have broken free from the lattice interact through long-range attraction forces and short-range repulsion forces. We have created physically-based computer animations involving a simulated physical world populated by hot constraint surfaces and thermoelastic deformable models. Reference: Terzopoulos D., Platt J., Fleischer K. (Nov. 1988) From goop to glop: Heating and melting deformable models. Technical Report, Schlumberger Palo Alto Research, Palo Alto, CA.