Constraint Methods for Flexible Models

John C. Platt
Alan H. Barr
California Institute of Technology
Pasadena, CA 91125

Abstract

Simulating flexible models can create aesthetic motion for computer animation. Animators can control these motions through the use of *constraints* on the physical behavior of the models. This paper shows how to use mathematical constraint methods based on physics and on optimization theory to create controlled, realistic animation of physically-based flexible models. Two types of constraints are presented in this paper: *reaction constraints* (RCs) and *augmented Lagrangian constraints* (ALCs). RCs allow the fast computation of collisions of flexible models with polygonal models. In addition, RCs allow flexible models to be pushed and pulled under the control of an animator. ALCs create animation effects such as volume-preserving squashing and the molding of taffy-like substances. ALCs are compatible with RCs. In this paper, we describe how to apply these constraint methods to a flexible model that uses finite elements.

KEYWORDS: Elasticity, Modeling, Dynamics, Constraints, Simulation
CR categories: G.1.6 — Constrained Optimization; I.3.7—Three-Dimensional Graphics and Realism (Animation)

1 Introduction

A primary goal of simulating flexible models is to animate physically realistic motions. Examples include simulating the musculature of a human body to create realistic walking; simulating the flow of viscous liquids, such as lava over volcanic rocks; or simulating a sculptor molding clay.

This paper takes a step towards these goals, by adding constraint properties to flexible models; and other properties, such as moldability and incompressibility. Using these properties, we can now simulate materials, such as clay, taffy, or putty, that have been very difficult to simulate using previous computer graphics models.

1.1 Desirable Properties of Flexible Models

In order to create pleasing and supple motions discussed above, we incorporate many of the following properties for our flexible models:

- *Physical Realism* — Flexible models should be able to move in natural, intuitive ways. Using the theory of elasticity to animate flexible models is very helpful in creating natural motion.

- *Controllability* — Flexible models should be able to follow an animation script. Models should be able to follow pre-defined paths exactly, while still wriggling in an interesting manner and interacting with other models.
- *Non-interpenetration* — Flexible models should be able to bounce off other models while using a small amount of computer time.
- *Limited Compressibility* — Flexible models should be able to have constant volume, even while being squashed. Models that squash without retaining their volume look as if they are made of sponge: they do not bulge out enough at the sides.
- *Moldability* — Flexible models should be moldable: external forces should mold the rest shape of the model. Models should follow the theory of *plasticity*, which describes materials that do not return to their rest shape after large deformation.

1.2 Force-Based Constraint Methods

Constraint methods that add external forces to physical systems yield physically realistic motion and allow simulation with simple, commercially available, differential equation solvers.

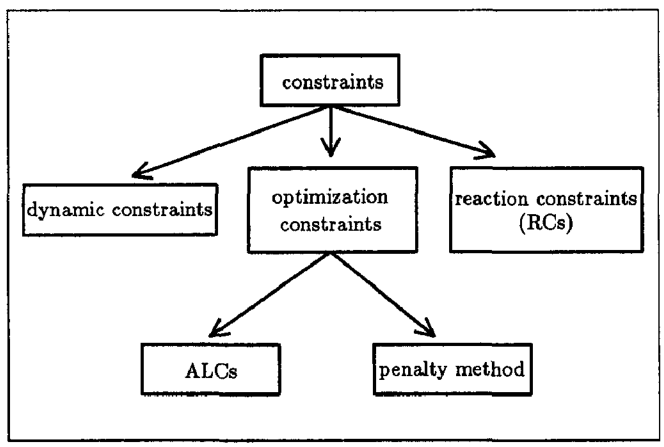

Figure 1: A hierarchy of constraint methods

There are at least three force-based constraint methods that allow the creation of flexible models with the properties listed in the last section.

- *Dynamic constraints* [3] use inverse dynamics to create critically damped forces which fulfill the constraints. Dynamic constraints are easy to use on systems which have simple dynamics. Elastic models have many state variables, however; this makes the dynamics hard to invert. We do not apply dynamic constraints to elastic models in this paper.

- *Reaction constraints*, presented in this paper, use a modified projection method for simple constraints, such as guiding flexible models along a path and preventing flexible models from penetrating a polygon. Reaction constraints supply reaction forces that cancel other forces that would violate the constraint. Reaction constraints require no extra differential equations, but they are limited in scope.
- *Optimization constraints* use ideas from optimization theory to constrain physical systems. Physical systems perform optimization, because the total energy of any physical system with dissipation decreases.

There are two types of optimization constraints. The simplest kind of optimization constraint is the well-known *penalty method*, where an extra energy that penalizes incorrect behavior is added to the physical system. The penalty method is analogous to adding rubber bands that attract the physical system to the constraints. One large disadvantage of the penalty method is that the constraints are enforced in the presence of external forces only as the ratios of the strengths of the rubber band to the external forces increases to infinity.

The ALC method is a constrained optimization method that adds differential equations that compute Lagrange multipliers of the physical system. These additional differential equations cause the system to eventually fulfill multiple constraints, even in the presence of external forces.

1.3 Previous Work

As discussed in the last section, this paper combines physically-based modeling techniques with constrained optimization methods.

There has been a growing interest in physical models in the field of computer graphics. Elastic models have been proposed previously [7] [11] [15] [17] that simulate deformable models quite well. Of these, [7] and [15] were based directly on variational principles, which are easily modified by constrained optimization techniques.

The physically-based elastic models are based on classical elasticity theory. A recommended explanation of elasticity may be found in Truesdell [16]; Fung [8] is another useful reference for both elasticity and plasticity.

In order to make controllable modeling and animation, researchers in computer graphics have previously studied constraint methods [2] [9] [18]. Witkin, et al. [20] applied the penalty method to parametrized constraints. Barzel and Barr [3] and Isaacs and Cohen [13] developed dynamic constraints. We extend their work to flexible models.

RCs are related to techniques that enforce boundary conditions of partial differential equations [21].

ALCs are based on the method of multipliers first developed by Arrow, et al. [1]. A comprehensive review paper was written by Bertsekas [4].

1.4 Preview

Sections 1–5 of this paper discusses optimization and various constraint methods. Section 2 explains why optimization theory is applicable to flexible models. Section 3 discusses the penalty method. Section 4 presents RCs and section 5 presents ALCs. Section 6 shows the application of the general constraint methods in the first part of the paper to flexible models. Section 7 shows various animation effects made by the constraints.

The appendices of this paper contain the mathematical details of how to apply RCs and ALCs to flexible models. The appendices describe the finite element flexible model, the equations necessary for animation control and collision, the equations for incompressibility and plasticity, and an explanation of why ALCs work.

2 Flexible Models Minimize Functions

This section illustrates that simulating physically-based flexible models is an optimization procedure. An optimization procedure finds a vector $\underline{x}$ to

$$\text{locally minimize } f(\underline{x}) \tag{1}$$

where $\underline{x}$ is a position in a high-dimensional space; and $f(\underline{x})$ is a scalar function, which can be imagined as the height of a landscape as a function of position $\underline{x}$ (see figure 2). In figure 2, the arrows represent the action of an optimization procedure, where $\underline{x}_0$ is the state of the system before the optimization procedure and $\underline{x}_{\min}$ is the state of the system afterwards.

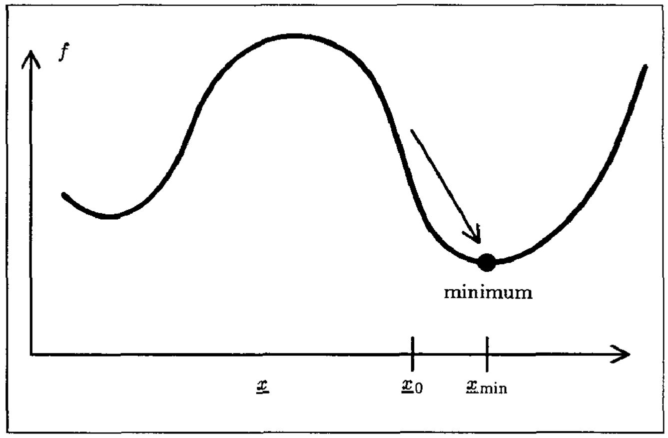

Figure 2: An optimization landscape

Physically-based flexible models minimize a particular function f. Consider the simplest flexible model, a spring. The energy of the spring comes in two forms: kinetic energy (the energy of motion) and potential energy (the energy stored in the tension of a spring). As a spring oscillates, the kinetic energy turns into potential energy, and back into kinetic energy. Because of friction, however, a spring eventually slows down and stops, with all of the energy having been converted into heat. The total energy of the spring always decreases. In general, any physical system with dissipation always loses energy, yet the total energy is always bounded below. Hence, physically-based flexible models will minimize their total energy as time increases. Even non-dissipative physical systems extremize energy over all paths in space-time.

Since simulating a flexible model is an optimization procedure, we can use optimization concepts to modify the flexible model. A useful concept is that optimization procedures, like computer graphics models, can be constrained.

A *constrained optimization procedure* finds a minimum of a function on a specified subspace. The prototypical constrained optimization problem can be stated as

$$\text{locally minimize } f(\underline{x}), \text{ subject to } g(\underline{x}) = 0, \tag{2}$$

where $g(\underline{x}) = 0$ is a scalar equation describing a subspace of the state space. During constrained optimization, the state vector $\underline{x}$ should be attracted to the subspace $g(\underline{x}) = 0$, then slide along the subspace until it reaches the locally smallest value of $f(\underline{x})$ on $g(\underline{x}) = 0$ (see figure 9). Solutions to a constrained optimization problem are restricted to a subset of the solutions of the corresponding unconstrained optimization problem.

Since physically-based flexible models minimize a function, we use constrained optimization algorithms as physical constraint methods. Applying constrained optimization algorithms to a physical system still decreases the total energy of the system, while enforcing external constraints; thus, optimization constraints do not destabilize physical systems.

There are other optimization procedures than simply simulating a physical system. The simplest optimization algorithm is *gradient descent*, where the values of $\underline{x}$ ski downhill, in the opposite direction of the gradient ∇f (see figure 3). ∇f points in the direction of the maximum increase in f.

$$\dot{x}_i = -\frac{\partial f}{\partial x_i} \tag{3}$$

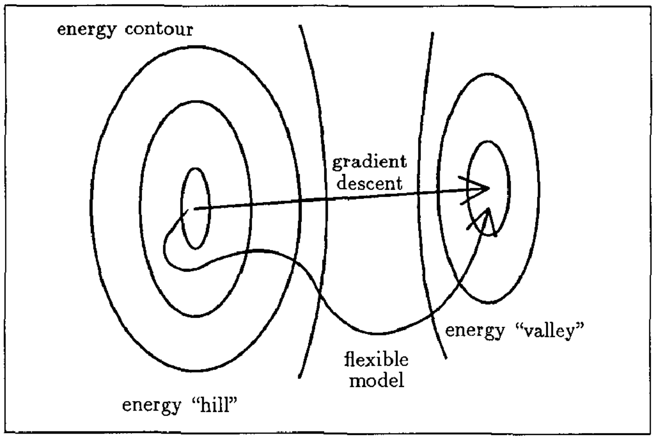

Figure 3: Both Gradient Descent and Flexible Models Minimize a Function

3 The Penalty Method

This section discusses a traditional constrained optimization technique called the penalty method; the method has previously been used in constraining computer graphics models [15][20].

The physical interpretation of the penalty method is a rubber band that attracts the physical state to the subspace $g(\underline{x}) = 0$. The penalty method adds a quadratic energy term that penalizes violations of constraints [12]. Thus, the constrained minimization problem (2) is converted to the following unconstrained minimization problem:

$$\text{minimize } \mathcal{E}_{\text{penalty}}(\underline{x}) = f(\underline{x}) + c(g(\underline{x}))^2. \quad (4)$$

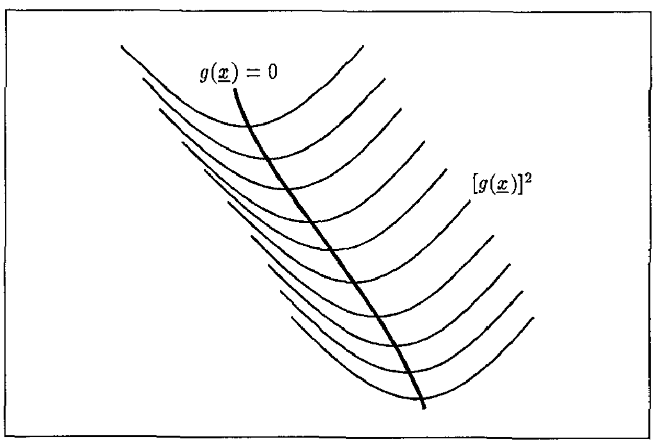

Figure 4: The penalty method makes a trough in state space

The penalty method can be extended to fulfill multiple constraints by using more than one rubber band. Namely, the constrained optimization problem

$$\text{minimize } f(\underline{x}), \text{ subject to } g_\alpha(\underline{x}) = 0; \quad \alpha = 1, 2, \ldots, n; \quad (5)$$

is converted into unconstrained optimization problem (see figure 4)

$$\text{minimize } \mathcal{E}_{\text{penalty}}(\underline{x}) = f(\underline{x}) + \sum_{\alpha=1}^{n} c_\alpha (g_\alpha(\underline{x}))^2. \quad (6)$$

The penalty method has a few convenient features.

- *Inexact Constraints* — There are situations in which it is not necesssary to exactly fulfill constraints; sometimes it is desirable to compromise between constraints.
- *Ease of Use* — Adding a rubber band to a physical system is simple and requires no extra differential equations.

However, the penalty method has number of disadvantages.

- *Inexact Constraints* — For finite constraint strengths c_α, the penalty method does not fulfill the constraints precisely. Under many circumstances, however, constraints should be fulfilled exactly. Using multiple rubber band constraints is like building a machine out of rubber bands; the machine would not hold together perfectly.
- *Stiffness of Equations* — Second, as the constraint strengths increase, the differential equations become *stiff*; that is, there are widely separated time constants. Most numerical methods must take time steps on the order of the fastest time constant, while most modelers are interested in the behavior at the slowest time constant. As a result of stiffness, the numerical differential equation solver takes very small time steps, using a large amount of computing time without getting much done.

4 Reaction Constraints

When flexible models are constrained to be on the outside of another model, or when they are constrained by an animator, they should fulfill these constraints quickly and exactly. As discussed in the last section, the penalty method has difficulties with swiftly fulfilling precise constraints.

RCs are a constraint method that retains the advantages of the penalty method while avoiding many of the disadvantages. RCs can force a point to follow a path, or to lie on the outside of a polygonal model. RCs are fast and simple to use, and do not require additional differential equations to be added to the physical system. However, only one RC can be applied to a mass point at any time.

RCs cancel forces that violate constraints and add forces that would critically damp the distance from the state to the constraint surface. RCs are a combination of the projection method [12] and dynamic constraints.

RCs work on individual mass points. Since elastic models are frequently discretized into mass points, RCs are applicable to constraining elastic models on a point-by-point basis.

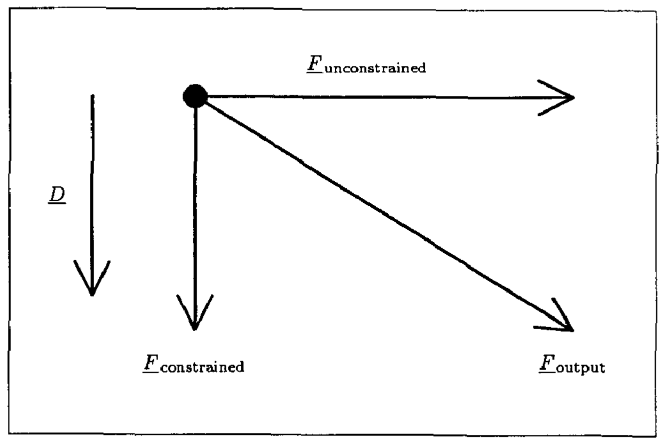

Figure 5: The reaction constraint cancels undesirable force components.

A reaction constraint is a procedure that processes the net force at a point, $\underline{F}_{\text{input}}$ created by physics or other constraint techniques, in order to yield a constrained force at a point $\underline{F}_{\text{output}}$, needed to fulfill a particular constraint. The RC first projects out undesirable components of $\underline{F}_{\text{input}}$ to yield $\underline{F}_{\text{unconstrained}}$ (see figure 5). Next, $\underline{F}_{\text{constrained}}$ is computed to yield critically damped motion that fulfills the constraint. Finally, the control force $\underline{F}_{\text{output}}$ is the sum of the constrained and unconstrained forces:

$$\underline{F}_{\text{output}} = \underline{F}_{\text{constrained}} + \underline{F}_{\text{unconstrained}} \quad (7)$$

To fulfill Newton's second law, the reaction force $\underline{F}_{\text{input}} - \underline{F}_{\text{output}}$ should be applied to the object that is interacting with the flexible model.

Let the vector $\underline{D}$ be the deviation in the position of the mass point. That is, the vector $\underline{D}$ points from the mass point towards where the mass point should be. The constrained force that eventually sets $\underline{D}$ to zero is

$$\underline{F}_{\text{constrained}} = k\underline{D} + c\frac{d}{dt}\underline{D} \tag{8}$$

where k is the strength of the constraint and c is the damping. If $c = \sqrt{2}k$, then the mass point fulfills the constraint with critically damped motion. If the damping is too low, then the constraint force overshoots. For critically damped motion, if k is increased, then the time needed to fulfill the constraint is decreased.

In the appendices, we describe the equations necessary for implementing two useful reaction constraints (see figure 6):

- *Path Following* — In constraining flexible models, we frequently want to constrain a mass point to follow a specified spatial path parameterized by time, without speeding up or slowing down. The pre-defined path is a useful constraint in animation, where flexible models need to be picked up and moved around. If only a few mass points of the flexible models are constrained, then the rest of the model is free to wriggle in a physically realistic manner. The equations for the path-following reaction constraint are contained in Appendix A.
- *Attraction to a Plane* — Another useful constraint is to force a mass point to lie on a plane. A mass point inside of a polygonal model can be forced outside of the polygonal model by using a planar reaction constraint. The equations for the planar reaction constraint are contained in Appendix B.

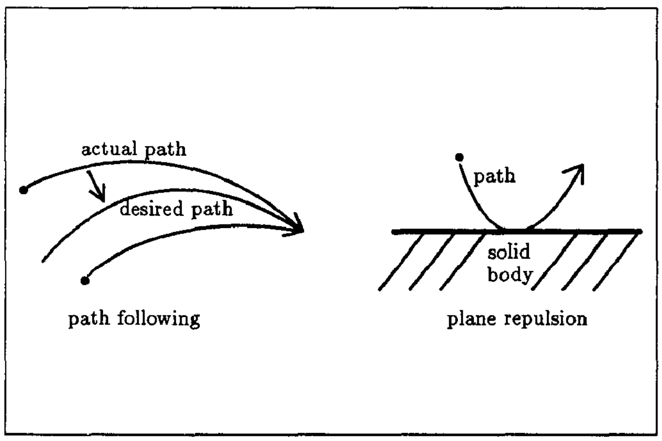

Figure 6: Examples of Reaction Constraints

Using reaction constraints is an easy way to implement simple constraints. Similar to the penalty method, no extra differential equations are required. Unlike the penalty method, the constraint is fulfilled in the presence of outside forces. If a flexible model is being lifted by a reaction force against gravity, then the lifting path is followed, even if gravity increases by a factor of ten. The reaction constraint thus reduces the amount of parameter adjustment needed in modeling elastic objects.

Reaction constraints are an extension of the projection method of constrained optimization, where any motion outside an allowed region is projected back into the region. Reaction constraints are more appropriate for physical models than the projection method, because the projection method needs to manipulate the physical state variables directly. Reaction constraints manipulate only forces, hence are compatible with both dynamic constraints [3] and with ALCs. In addition, reaction constraints do not need special numerical routines.

Reaction constraints are much faster than the penalty method for collisions. The penalty method tries to cancel a large penetration force by adding a force that is a rapidly changing function of position. Small numerical step sizes are needed for the penalty method in order to prevent unstable oscillation. However, reaction constraints cancel a penetration force, independent of the depth of the penetration. Reaction constraints, therefore, can take much larger step sizes.

5 Augmented Lagrangian Constraints

In the animation of flexible models, more than one constraint per mass point is needed. Constraints may be more complex than simple path following or repulsion from a plane. In addition, we wish to enforce real properties of flexible models, such as incompressibility and moldability.

This section presents a type of constraint, called an augmented Lagragian constraint, that enforces the complex, multiple constraints needed for flexible models. The differential equations used in ALCs were first developed by Arrow in 1958 [1].

5.1 Lagrange Multipliers

Lagrange multiplier methods, like the penalty method, convert constrained optimization problems into unconstrained extremization problems. Namely, a solution to the equation (2) is also a critical point of the energy

$$\mathcal{E}_{\text{Lagrange}}(\underline{x}) = f(\underline{x}) + \lambda g(\underline{x}). \tag{9}$$

λ is called the Lagrange multiplier for the constraint $g(\underline{x}) = 0$ [12].

A direct consequence of equation (9) is that the gradient of f is collinear to the gradient of g at the constrained extrema (see Figure 7). The constant of proportionality between ∇f and ∇g is $-\lambda$:

$$\nabla \mathcal{E}_{\text{Lagrange}} = 0 = \nabla f + \lambda \nabla g. \tag{10}$$

We use the collinearity of ∇f and ∇g in the design of the ALC.

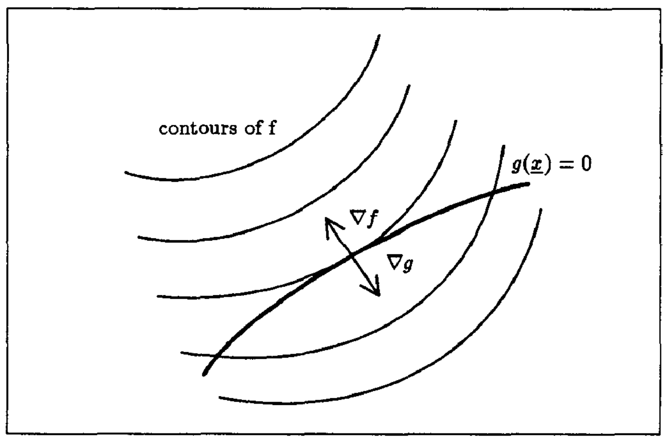

Figure 7: At the constrained minimum, $\nabla f = -\lambda \nabla g$

A simple example shows that Lagrange multipliers provide the extra degrees of freedom necessary to solve constrained optimization problems. Consider the problem of finding a point (x, y) on the line $x+y=1$ that is closest to the origin. Using Lagrange multipliers,

$$\mathcal{E}_{\text{Lagrange}} = x^2 + y^2 + \lambda(x + y - 1) \tag{11}$$

Now, take the derivative with respect to all variables, x, y, and λ.

$$\frac{\partial \mathcal{E}_{\text{Lagrange}}}{\partial x} = 2x + \lambda = 0 \tag{12}$$

$$\frac{\partial \mathcal{E}_{\text{Lagrange}}}{\partial y} = 2y + \lambda = 0 \tag{13}$$

$$\frac{\partial \mathcal{E}_{\text{Lagrange}}}{\partial \lambda} = x + y - 1 = 0 \tag{14}$$

With the extra variable λ, there are now three equations in three unknowns. In addition, the last equation is precisely the constraint equation.

5.2 Gradient Descent Does Not Work with Lagrange Multipliers

Applying gradient descent in equation (3) to the energy in equation (9) yields

$$\dot{x}_i = -\frac{\partial \mathcal{E}_{\text{Lagrange}}}{\partial x_i} = -\frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i}, \quad (15)$$

$$\dot{\lambda} = -\frac{\partial \mathcal{E}_{\text{Lagrange}}}{\partial \lambda} = -g(\underline{x}). \quad (16)$$

Note that there is an auxiliary differential equation for λ, which is necessary to apply the constraint $g(\underline{x}) = 0$. Also, recall that when the system is at a constrained extremum, $\nabla f = -\lambda \nabla g$, hence, $\dot{x}_i = 0$.

Solutions to the constrained optimization problem (2) are saddle points of the energy in equation (9), which has no lower bound [1]. If the vector $\underline{x}$ is held fixed where $g(\underline{x}) \neq 0$, the energy can be decreased to $-\infty$ by sending λ to $+\infty$ or $-\infty$.

Gradient descent does not work with Lagrange multipliers, because a critical point of the energy in equation (9) need not be an attractor for equations (15) and (16). A stationary point must be a local minimum in order for gradient descent to converge.

5.3 The Basic Lagrange Constraint

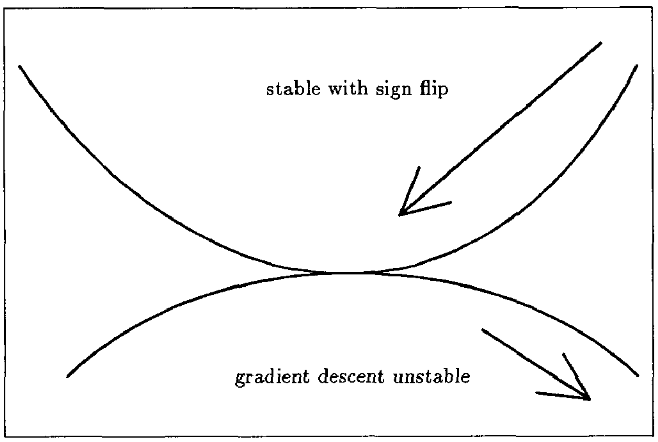

Figure 8: The sign flip from equation (16) to equation (18) makes Lagrange multipliers stable

We present an alternative to differential gradient descent that estimates the Lagrange multipliers, so that the constrained minima are attractors of the differential equations, instead of "repulsors." The differential equations that solve (2) are

$$\dot{x}_i = -\frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i}, \quad (17)$$

$$\dot{\lambda} = +g(\underline{x}). \quad (18)$$

Equations (17) and (18) are similar to equations (15) and (16). As in equations (15) and (16), solutions to problem 2 are stationary points of equations (17) and (18). Notice, however, the sign inversion in the equation (18), as compared to equation (16). The equation (18) is performing gradient *ascent* on λ. The sign flip makes the method stable, as shown in Appendix G (see figure 8)

The system of differential equations (17) and (18) gradually fulfills the constraints. Notice that the function $g(\underline{x})$ can be replaced by $kg(\underline{x})$, without changing the location of the constrained minimum. As k is increased, the state begins to undergo damped oscillation about the constraint subspace $g(\underline{x}) = 0$. As k is increased further, the frequency of the oscillations increase, and the time to convergence increases.

5.4 Extensions to the Algorithm

One extension to equations (17) and (18) is an algorithm for constrained minimization with multiple constraints. Adding an extra differential equation for every equality constraint and summing all of the constraint forces creates the energy

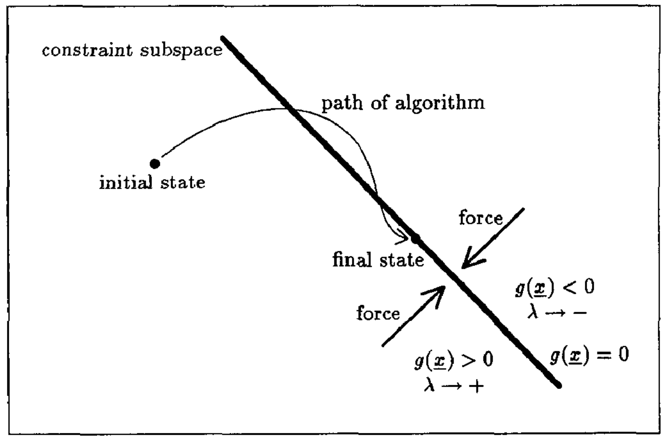

Figure 9: The state is attracted to the constraint subspace

$$\mathcal{E}_{\text{multiple}} = f(\underline{x}) + \sum_{\alpha} \lambda_\alpha g_\alpha(\underline{x}), \quad (19)$$

which yields differential equations

$$\dot{x}_i = -\frac{\partial f}{\partial x_i} - \sum_{\alpha} \lambda_\alpha \frac{\partial g_\alpha}{\partial x_i}, \quad (20)$$

$$\dot{\lambda}_\alpha = +g_\alpha(\underline{x}). \quad (21)$$

Another extension is constrained minimization with inequality constraints. As in traditional optimization theory [12], one uses additional slack variables to convert inequality constraints into equality constraints. Namely, a constraint of the form $h(\underline{x}) \geq 0$ can be expressed as

$$g(\underline{x}) = h(\underline{x}) - z^2. \quad (22)$$

Since z^2 must always be positive, then $h(\underline{x})$ is constrained to be positive. The slack variable z is treated like a component of $\underline{x}$ in equation (17). An inequality constraint requires two extra differential equations, one for the slack variable z and one for the Lagrange multiplier λ.

Alternatively, the inequality constraint can be represented as an equality constraint. For example, if $h(\underline{x})$ is constrained to be greater than zero, then the optimization can be constrained with

$$g(\underline{x}) = \begin{cases} [h(\underline{x})]^2, & \text{if } h > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

Combining the basic Lagrangian constraints with the penalty method yields augmented Lagrangian constraints (ALCs). ALCs have better convergence properties than basic Lagrangian constraints, as shown in Appendix G. The basic Lagrangian constraints are completely compatible with the penalty method. If one adds a penalty force to equation (17) that corresponds to an quadratic energy

$$E_{\text{penalty}} = \frac{c}{2}(g(\underline{x}))^2, \quad (24)$$

then the set of differential equations for an ALC is

$$\dot{x}_i = -\frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i} - cg \frac{\partial g}{\partial x_i}, \quad (25)$$

$$\dot{\lambda} = g(\underline{x}). \quad (26)$$

The extra force from the penalty does *not* change the position of the stationary points of the differential equations, because the penalty force is zero when $g(\underline{x}) = 0$, independent of the value of c.

There is a minimum necessary penalty strength c required in some cases for the ALC to converge (see appendix G). The minimum penalty strength in the ALC is usually much less than the strength needed by the penalty method for an accurate solution [4]. ALCs are applicable to more general constraints than RCs, especially when more than one non-linear constraint is associated with each mass point.

6 Constraining Flexible Models with Augmented Lagrangian Constraints

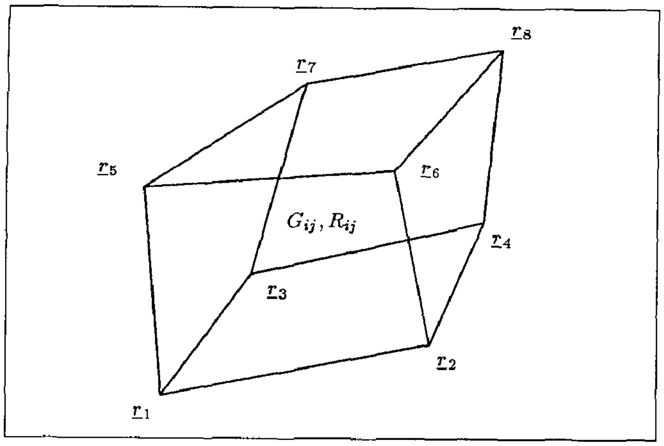

Figure 10: An element of flexible material

ALCs are ideal for the non-linear constraints that arise from adding new properties to flexible models. The augmented Lagrangian constraints are applied to the differential equations that govern an element of material. Flexible models are created by aggregating these elements in a grid, which may be difficult in the case of complex rest shapes [19].

The internal forces on a element are fully derived in Appendix C. The forces depend on the average metric tensor, $G_{ij}(\underline{r}_1, \underline{r}_2, \ldots \underline{r}_8)$, which describes the current shape of an element, and is computed for each element of material using the finite element method (see figure 10) [19]. Each element of material also has a rest state, which is described by R_{ij}. For a Hookean elastic material, the internal force encourages the metric tensor of each element to be close to the metric tensor of the rest state [15].

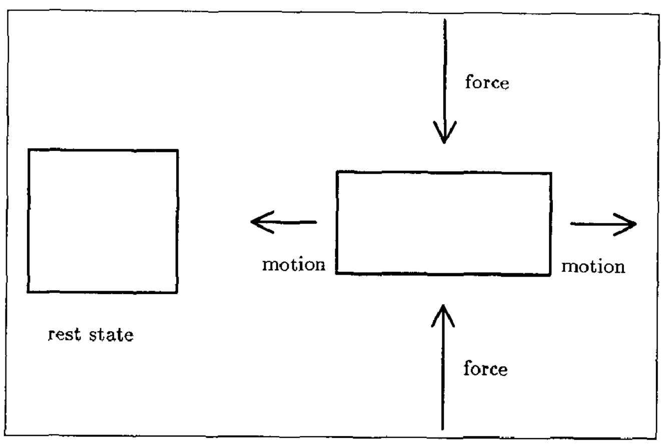

Figure 11: Incompressibility preserves the volume of an element

Hookean elasticity, however, does not fully describe the range of materials that are desirable to animate. For example, a Hookean elastic model can be easily compressed. If an elastic model undergoes violent deformation, as is common in computer graphics, then it will behave more like a sponge than like gelatin. If an incompressible material is desired (see figure 11), then ALCs are added to the equations for an elastic element.

The volume squared of one element is the determinant of the metric tensor G_{ij} of that element [6]. To constrain the volume of an element to be a constant V_0, we apply the augmented Lagrangian method, using the constraint

$$g = \det G_{ij} - V_0^2 = 0. \tag{27}$$

The complete differential equation for an incompressible element is given in Appendix E.

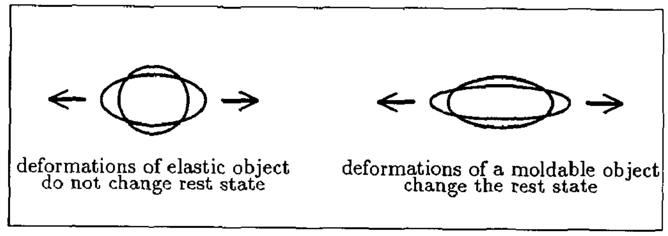

Figure 12: The rest shape of plastic materials changes after strong deformation.

Many materials, such as taffy and putty, are moldable. Moldable materials do not return to their rest shape after being strongly deformed (see figure 12). Augmented Lagrangian constraints can be applied to each element's rest state so that it roughly approximates the theory of strongly deformed materials.

A moldable element has a rest metric R_{ij} that is constrained to be close to the metric G_{ij} [8]. Mathematically, there is an inequality constraint, based on the von Mises' yield criterion from the theory of plasticity [8],

$$P = (G_{ij} - R_{ij})(G_{ij} - R_{ij}) - P_0 < 0. \tag{28}$$

Using the method described in equation (23), we use the constraint function

$$\eta = \begin{cases} 1/2P^2; & \text{if } P > 0, \\ 0; & \text{if } P \leq 0. \end{cases} \tag{29}$$

For plasticity, there are differential equations for R_{ij} derived from applying equations (20) and (21) to the constraint in equation (29). The general differential equations for a moldable element are given in Appendix F.

The general equations for applying an ALC to a flexible model are given in appendix D. To apply an ALC to a flexible model, forget that the position and velocity are related, and simply apply equations (25) and (26) directly. In general, using ALCs on flexible models results in equations of the form

$$\dot{x}_i = v_i - u_i(\underline{x}, \underline{v}) \tag{30}$$

$$\dot{v}_i = F_i - \epsilon v_i - w_i(\underline{x}, \underline{v}) \tag{31}$$

where u_i and v_i are functions determined by applying various ALCs. Equations (30) and (31) do not appear to be in the form of a standard physical system. However, we can change the differential equations in (30) and (31) into one second-order differential equation:

$$\ddot{x}_i + \epsilon \dot{x}_i = F_i - \epsilon u_i - w(\underline{x}, \underline{x} + \underline{u}) - \frac{d}{dt} u_i(\underline{x}, \underline{v}). \tag{32}$$

The left-hand side of equation (32) is a standard form for a physical system; therefore, ALCs add only forces to flexible models.

7 Results

We have simulated all of the constraints discussed in this paper using standard differential equations solvers [14]. Since differential equations are simulated over a time interval, the results are in the form of animation. The figures in this section are individual frames from a sequence.

Figures 13 and 14 show frames from an animation of a compressible elastic cube of gelatin which is lifted up and then bounced off a table. The lifting of the cube is done with a path-following reaction constraint, and the table is implemented with a reaction constraint that keeps the cube above a plane. Notice that since the cube in compressible, its volume can vary through the course of the simulation.

Figure 15 shows a compressible seat cushion being squashed with a sphere. The sphere is a physical model with mass. An RC prevents the sphere from penetrating the cushion.

Figure 16 shows an incompressible moldable cube striking a surface. Instead of bouncing off the surface, the moldable cube sticks to the surface, with its sides near the surface bulging out. Incompressiblity forces the sides to bulge, and the moldability updates the rest shape so that the shape is no longer a cube. Both the incompressibility and the moldability are enforced with augmented Lagrangian constraints.

Figures 17–20 illustrate the moldability of the models. A sphere squashes the model in figure 17; but the elastic models bounces back to its rest shape in figure 18. In figure 19, a moldable model starts with the same rest shape, and is squashed by the sphere; but in figure 20, the moldable model has a dented edge.

8 Conclusions

In the past, researchers have made models that simulate the behavior of flexible materials. These models automatically move in a physically realistic way, without specifying the exact positions and velocities of the model at all times. The "hands-off" nature of the physically-based models, however, makes them hard for an animator to control.

By adding physical modeling constraints to the elastic models, a compromise can be reached between completely specifying the motion of a model and allowing a simulation package to run freely. Constraint methods are useful for controlling the flexible models, while retaining the physically realistic motion created by the physics.

This paper presents two constraint techniques, based physics and optimization theory, for constraining the physical simulation of flexible models: reaction constraints and augmented Lagrange constraints. Both reaction constraints and augmented Lagrange constraints eventually fulfill specified constraints exactly, unlike the penalty method.

Reaction constraints, based on the projection method, are a simple way of enforcing path following or repulsion from a polygon. Reaction constraints require no extra differential equations, because they project away undesirable components of the force. Only one reaction constraint can be applied to a mass point at a time. Reaction constraints are useful for guiding flexible models along a path and for reducing the amount of computation time needed for collisions.

ALCs are a differential version of the method of multipliers from optimization theory. ALCs are a general technique for constrained optimization. In this paper, we use ALCs for constraining flexible models to be incompressible and moldable.

Compressible elastic models look as if they are made out of sponge. To simulate other materials, such as rubber, an augmented Lagrange incompressibility constraint should be added to the elastic model.

Many natural substances, such has clay and taffy, do not return to their rest shape after strong deformations. Purely elastic models are inadequate for these substances. Using ALCs to keep the rest shape near the current shape is an effective model for these moldable substances. In addition, by applying forces to these plastic substances, we can mold interesting shapes without numerically specifying the rest shape.

Acknowledgements

We wish to thank Jed Lengyel for rest state models and John Snyder for rendering software. This paper was supported by grants from Apple Computer, Hewlett-Packard Company, Symbolics Inc., and an AT&T Bell Labs Fellowship.

References

[1] Arrow, K., Hurwicz, L., Uzawa H., *Studies in Linear and Nonlinear Programming,* (Stanford University Press, Stanford, CA, 1958).

[2] Badler, N., "Multi-Dimensional Input Techniques and Articulated Figure Positioning By Multiple Constraints," *1986 Workshop on Interactive 3D Graphics* (Chapel Hill, NC, 1986).

[3] Barzel, R., Barr, A., "Modeling with Dynamic Constraints," in *Topics in Physically Based Modeling,* SIGGRAPH Tutorial 17 Notes, (1987).

[4] Bertsekas, D., "Multiplier Methods: a Survey," *Automatica,* **12**, 133–145, (1976).

[5] de Boor, C., *A Practical Guide to Splines,* (Springer-Verlag, NY, 1978).

[6] do Carmo, M., *Differential Geometry of Curves and Surfaces,* (Prentice-Hall, Englewood Cliffs, NJ, 1974).

[7] Feynman, C., *Modeling the Appearance of Cloth,* MSc Thesis, EECS Dept. , (MIT, Cambridge, MA, 1986).

[8] Fung, Y., *Foundations of Solid Mechanics,* (Prentice-Hall, Englewood Cliffs, NJ, 1965).

[9] Girard, M., Maciejewski, A. "Computational Modelling for the Computer Animation of Legged Figures," *Proc. SIGGRAPH 1985* 263–270, (1985).

[10] Goldstein, H, *Classical Mechanics,* (Addison-Wesley, Reading, MA, 1950).

[11] Haumann, D., "Modeling the Physical Behavior of Flexible Objects", in *Topics in Physically-Based Modeling,* SIGGRAPH Tutorial 17 Notes, 1987.

[12] Hestenes, M., *Optimization Theory,* (Wiley & Sons, NY, 1975).

[13] Isaacs, P., Cohen, M., "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics," *Proc. SIGGRAPH 1987,* 215–224, (1987).

[14] Press, W., Flannery, B., Teukolsky, S., Vetterling W., *Numerical Recipes,* (Cambridge University Press, Cambridge, 1986).

[15] Terzopoulos, D., Platt, J., Barr, A., Fleischer, K., "Elastically Deformable Models," *Proc. SIGGRAPH 1987,* 205–214, (1987).

[16] Truesdell, C., "The Non-Linear Field Theory of Mechanics," in *Encyclopedia of Physics,* S. Flügge, ed., **III/3** (Springer-Verlag, Berlin, 1965).

[17] Weil, J., "The Synthesis of Cloth Objects," *Proc. SIGGRAPH 1986,* 49–54 (1986),

[18] Wilhelms, J., Barsky, B., "Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots," *Proc. Graphics Interface '85,* 97–104 (Montreal, 1985)

[19] White, R., *An Introduction to the Finite Element Method with Applications to Non-linear Problems,* (John Wiley & Sons, NY, 1985).

[20] Witkin, A., Fleischer, K., Barr, A., "Energy Constraints on Parametrized Models," *Proc. SIGGRAPH 1987,* 225–232, (1987).

[21] Zienkiewicz, O., *The Finite Element Method,* Third Edition, (McGraw-Hill, London, 1977).

Appendices

A Equations for a Path Following RC

The deviation vector $\underline{D}$ to a path is the difference between where the mass point is and where it should be on the path at that time. Let $(\underline{x}(t), \underline{v}(t))$ be the current position and velocity of the mass point, and $(\underline{x}^*(t), \underline{v}^*(t))$ be the desired position and velocity of the mass point. Then,

$$\underline{D} = \underline{x}^*(t) - \underline{x}(t) \tag{33}$$

$$\frac{d}{dt}\underline{D} = \underline{v}^*(t) - \underline{v}(t) \tag{34}$$

Since we want to control the velocity along the path, we do not allow any unconstrained force:

$$\underline{F}_{\text{unconstrained}} = 0 \tag{35}$$

The final control force is:

$$\underline{F}_{\text{output}} = c(\underline{v}^*(t) - \underline{v}(t)) + k(\underline{x}^*(t) - \underline{x}(t)), \tag{36}$$

Notice how the control force in this case is independent of the input force, $\underline{F}_{\text{input}}$.

B Equations for a Planar RC

Consider the plane with normalized plane equation $P(\underline{x}(t)) = Ax(t) + By(t) + Cz(t) + D = 0$. Let the homogeneous operator be $Q(\underline{x}(t)) = Ax(t) + By(t) + Cz(t)$. The normal, $\hat{n}$, to the plane is $(A\ B\ C)^T$. We want the distance of the mass point to the plane to be zero:

$$\begin{aligned} \underline{D} &= -\hat{n}P(\underline{x}), && (37) \\ \frac{d}{dt}\underline{D} &= -\hat{n}Q(\underline{v}). && (38) \end{aligned}$$

where the vector $\underline{x}$ is the position of the mass point and the vector $\underline{v}$ is the velocity.

The components of the input force normal to the plane need to be controlled. The force tangent to the plane should be unconstrained.

$$\underline{F}_{\text{unconstrained}} = F_{\text{input}} - (F_{\text{input}} \cdot \hat{n})\hat{n}. \tag{39}$$

Using equation (8) yields

$$\underline{F}_{\text{constrained}} = -(kP(\underline{x}) + cQ(\underline{v}))\hat{n}. \tag{40}$$

The output of a planar RC is

$$\underline{F}_{\text{output}} = F_{\text{input}} - (kP(\underline{x}) + cQ(\underline{v}) + \underline{F}_{\text{input}} \cdot \hat{n})\,\hat{n}. \tag{41}$$

To constrain a point to lie on one side of the plane, $P(\underline{x}) < 0$, we apply the reaction constraint only if the mass point is on the wrong side of the plane and if the input force is not lifting the point away from the plane:

$$P(\underline{x}) < 0 \quad \text{and} \quad \underline{F}_{\text{input}} \cdot \hat{n} > \underline{F}_{\text{constrained}} \cdot \hat{n}. \tag{42}$$

The one-sided planar RC can be extended to prevent any mass points from entering a solid polygonal model. From inside of the model, choose the closest polygon, then apply the one-sided planar RC to force the mass point to the surface of that polygon.

C Finite Elements for Elasticity

Following [15], there is a potential energy for each flexible element that encourages the metric tensor to be near the rest metric:

$$U = s\sum_{i,j}(G_{ij} - R_{ij})^2, \tag{43}$$

where s is the stiffness of the material. The energy in equation (43) describes an isotropic material with a Poisson ratio of zero. The force on the the points that make up the element is the derivative of the potential energy [10]:

$$\underline{F}_k^{\text{elastic}} = s\sum_{i,j}(G_{ij} - R_{ij})\frac{\partial G_{ij}}{\partial \underline{r}_k}, \tag{44}$$

where $\underline{r}_k$ is the position of the kth corner. In addition, there is a viscous damping force that resists changes in the metric tensor:

$$\underline{F}_k^{\text{viscous}} = l\sum_{i,j}\dot{G}_{ij}\frac{\partial G_{ij}}{\partial \underline{r}_k} = l\sum_{i,j,m}\frac{\partial G_{ij}}{\partial \underline{r}_m}\cdot \underline{v}_m \cdot \frac{\partial G_{ij}}{\partial \underline{r}_k}, \tag{45}$$

where $\underline{v}_m$ is the velocity of the mth corner, and l is the viscous damping of the element. If $s \gg l$, then the material acts like a solid. If $l \gg s$, then the material acts like a fluid [16]. Using Newton's Second Law, the differential equations for an unconstrained viscoelastic element is

$$\begin{aligned} \frac{d}{dt}\underline{r}_i &= \underline{v}_i && (46) \\ \frac{d}{dt}\underline{v}_i &= \underline{F}^{\text{elastic}} + \underline{F}^{\text{viscous}} && (47) \end{aligned}$$

The viscoelastic forces and the constraint force depend on G_{ij}. Following the finite element method, the G_{ij} in each element is assumed to be the integrated average of G_{ij} over the entire element. Let $\underline{a}$ be the material coordinates of a point in the element and let $\underline{r}(\underline{a})$ be the position of the points $\underline{a}$. Then, from the definition of metric tensor,

$$G_{ij} = \int \frac{\partial \underline{r}}{\partial a_i}\frac{\partial \underline{r}}{\partial a_j}.dV \tag{48}$$

Assuming a position in the element is a linear interpolation of the positions of the corners of the element (see figure 11), the average G_{ij} can be analytically computed from the positions of the corners. To compute G_{ij}, estimates of the spatial derivatives are required:

$$\begin{aligned} &\underline{\alpha}_i = \underline{r}_{2i} - \underline{r}_{2i-1}, && i = 1,2,3,4 && (49) \\ &\underline{\beta}_1 = \underline{r}_3 - \underline{r}_1, \underline{\beta}_2 = \underline{r}_4 - \underline{r}_2, && \underline{\beta}_3 = \underline{r}_7 - \underline{r}_5, \underline{\beta}_4 = \underline{r}_8 - \underline{r}_6, && (50) \\ &\underline{\gamma}_i = \underline{r}i + 4 - \underline{r}_i, && i = 1,2,3,4 && (51) \end{aligned}$$

Averages of the spatial derivatives are also required:

$$\underline{a} = \sum_{i=1}^{4}\alpha_i, \quad \underline{b} = \sum_{i=1}^{4}\beta_i, \quad \underline{c} = \sum_{i=1}^{4}\gamma_i. \tag{52}$$

Finally, the various components of G_{ij} can be computed, assuming the element has unit length, width, and height in material coordinates.

$$\begin{aligned} G_{00} &= \frac{1}{18}(2\underline{a}\cdot\underline{a} - \underline{\alpha}_1\cdot\underline{\alpha}_4 - \underline{\alpha}_2\cdot\underline{\alpha}_3) && (53) \\ G_{11} &= \frac{1}{18}(2\underline{b}\cdot\underline{b} - \underline{\beta}_1\cdot\underline{\beta}_4 - \underline{\beta}_2\cdot\underline{\beta}_3) && (54) \\ G_{22} &= \frac{1}{18}(2\underline{c}\cdot\underline{c} - \underline{\gamma}_1\cdot\gamma_4 - \underline{\gamma}_2\cdot\underline{\gamma}_3) && (55) \\ G_{01} = G_{10} &= \frac{1}{24}[\underline{a}\cdot\underline{b} - (\underline{\alpha}_1 + \underline{\alpha}_2)\cdot(\underline{\beta}_1 + \underline{\beta}_2) \\ &\quad +(\underline{\alpha}_3 + \underline{\alpha}_4)\cdot(\underline{\beta}_3 + \underline{\beta}_4)] && (56) \\ G_{02} = G_{20} &= \frac{1}{24}[\underline{a}\cdot\underline{c} - (\underline{\alpha}_1 + \underline{\alpha}_3)\cdot(\underline{\gamma}_i + \underline{\gamma}_2) \\ &\quad +(\underline{\alpha}_2 + \underline{\alpha}_4)\cdot(\underline{\gamma}_3 + \underline{\gamma}_4)] && (57) \\ G_{12} = G_{21} &= \frac{1}{24}[\underline{b}\cdot\underline{c} - (\underline{\beta}_1 + \underline{\beta}_3)\cdot(\underline{\gamma}_1 + \underline{\gamma}_3) \\ &\quad +(\underline{\beta}_2 + \underline{\beta}_4)\cdot(\underline{\gamma}_2 + \underline{\beta}_4)] && (58) \end{aligned}$$

As in the continuous case, the diagonal terms of the metric tensor G_{ij} in equations (53)–(58) depend on various distances in the cube, while the off-diagonal terms depend on angles. Also, the G_{ij} are quadratic functions of the $\underline{r}_i$. Thus, $\partial G_{ij}/\partial r_i$ are complicated, although linear, functions of r_i.

The finite element is equivalent to a set of mass points with nonlinear springs between them.

D Equations for a Flexible Model ALC

This appendix illustrates how to apply ALCs to physical systems. As stated in section 2, physical systems perform optimization, but not gradient descent. ALCs, however, are easily added to physical systems.

Consider a typical flexible model, with forces $F_i(\underline{x})$ and damping ϵ. The differential equation for this system is

$$\begin{aligned} \dot{x}_i &= v_i, && (59) \\ \dot{v}_i &= F_i - \epsilon v_i. && (60) \end{aligned}$$

Let us constrain the flexible model in equations (59) and (60) to lie on the subspace $g(\underline{x}) = 0$. There are $2N$ optimizing state variables: $\underline{x}_i$ and $\underline{v}_i$. We can apply an augmented Lagrangian λ to the equation for x to fulfill $g(\underline{x}) = 0$. We can also add a penalty term $(dg/dt)^2$ to the v equation to provide extra damping in the direction of violation of the constraint. (Notice that this extra damping force is zero when the constraint is fulfilled.) The final form of an ALC applied to a physical model is

$$\begin{aligned} \dot{x}_i &= v_i - (\lambda + kg)\frac{\partial g}{\partial x_i}, && (61) \\ \dot{v}_i &= F_i - \epsilon v_i - c\frac{\partial g}{\partial x_i}\frac{\partial g}{\partial x_j}v_j, && (62) \\ \dot{\lambda} &= g(\underline{x}). && (63) \end{aligned}$$

As section in 4, multiple constraints are performed by creating an auxiliary differential equation for each constraint and summing all of the constraint forces.

E Equations for Incompressibility

The constraint for an incompressible element is

$$g = \det(G_{ij}) - V_0^2 = 0. \tag{64}$$

The derivative of the constraint g with respect to the spatial variables $\underline{r}_i$ is needed for an ALC. Let C_{ij} be the matrix of cofactors of G_{ij}. Then, the derivative is

$$\frac{\partial g}{\partial r_l} = C_{ij}\frac{\partial G_{ij}}{\partial r_l}. \tag{65}$$

Then, the differential equations for an incompressible element with other forces $\underline{F}_l$ are

$$\frac{d}{dt}\underline{r}_l = \underline{v}_l - (\lambda + kg)\frac{\partial g}{\partial r_l}, \tag{66}$$

$$\frac{d}{dt}\underline{v}_l = \underline{F}_l - \frac{\partial g}{\partial r_l}\frac{\partial g}{\partial r_i}\underline{v}_i, \tag{67}$$

$$\dot{\lambda} = g. \tag{68}$$

F Equations for a Moldability

The constraint for a moldable element is:

$$P = (G_{ij} - R_{ij})(G_{ij} - R_{ij}) - P_0 < 0 \tag{69}$$

$$\eta = \begin{cases} 1/2P^2; & \text{if } P > 0, \\ 0; & \text{if } P \le 0. \end{cases} \tag{70}$$

Again, the derivative of the constraint function with respect to the state variables is needed by an ALC. For stretchable models, however, the rest metric is also a function of time. We thus need the derivative of η with respect to R_{ij}. Let $Q = P$ if $P > 0$ and $Q = 0$, otherwise. Then,

$$\frac{\partial \eta}{\partial r_l} = Q(G_{ij} - R_{ij})\frac{\partial G_{ij}}{\partial r_l} \tag{71}$$

$$\frac{\partial \eta}{\partial R_{ij}} = -Q(G_{ij} - R_{ij}). \tag{72}$$

Using these derivatives yield the differential equations for a moldable element with other forces $\underline{F}_l$:

$$\frac{d}{dt}\underline{r}_l = \underline{v}_l - (\sigma + c\eta)\frac{\partial \eta}{\partial r_l} \tag{73}$$

$$\frac{d}{dt}v_l = \underline{F}_l - \frac{\partial \eta}{\partial r_l}\frac{\partial \eta}{\partial r_i}\underline{v}_i \tag{74}$$

$$\dot{R}_{ij} = -\frac{\partial \eta}{\partial R_{ij}} \tag{75}$$

$$\dot{\sigma} = \eta. \tag{76}$$

G Why ALCs Work

The damped oscillations of equations (20) and (21) can be explained by differentiating equation (20) and then substituting (21):

$$\ddot{x}_i + \sum_j \left(\frac{\partial^2 f}{\partial x_i \partial x_j} + \sum_\alpha \lambda_\alpha \frac{\partial^2 g_\alpha}{\partial x_i \partial x_j}\right)\dot{x}_j + \sum_\alpha g_\alpha \frac{\partial g_\alpha}{\partial x_i} = 0. \tag{77}$$

Equation (77) is the equation for a damped mass system, with an inertia term, $\ddot{x}_i$; a damping matrix,

$$A_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} + \sum_\alpha \lambda \frac{\partial^2 g_\alpha}{\partial x_i \partial x_j}; \tag{78}$$

and an internal force, $\sum_\alpha g_\alpha \partial g_\alpha / \partial x_i$, which is the derivative of the internal energy,

$$U = \frac{1}{2}\sum_\alpha (g_\alpha(\underline{x}))^2. \tag{79}$$

If the system is damped and the state remains bounded, the state falls into a constrained minima.

As in physics, we can construct a total energy of the system, which is the sum of the kinetic and potential energies.

$$E = T + U = \sum_i \frac{1}{2}(\dot{x}_i)^2 + \sum_\alpha \frac{1}{2}(g_\alpha(\underline{x}))^2. \tag{80}$$

If the total energy is decreasing with time, and the state remains bounded, then the system will dissipate any extra energy, and will settle down into the state where

$$g_\alpha(\underline{x}) = 0, \tag{81}$$

$$\dot{x}_i = \frac{\partial f}{\partial x_i} + \sum_\alpha \lambda \frac{\partial g_\alpha}{\partial x_i} = 0, \tag{82}$$

which is a constrained extremum of the original problem in equation (2).

The time derivative of the total energy in equation (80) is

$$\dot{E} = \sum_i \ddot{x}_i \dot{x}_i + \sum_\alpha g_\alpha(\underline{x})\frac{\partial g_\alpha}{\partial x_i}\dot{x}_i = -\sum_{i,j} \dot{x}_i A_{ij} \dot{x}_j. \tag{83}$$

If damping matrix A_{ij} is positive definite, the system converges to fulfill the constraints [1].

ALC always converges for quadratic programming, a special case of constrained optimization. A quadratic programming problem has a quadratic function $f(\underline{x})$ and piecewise linear continuous functions $g_\alpha(\underline{x})$, such that

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \text{ is positive definite and } \frac{\partial^2 g_\alpha}{\partial x_i \partial x_j} = 0. \tag{84}$$

Under these circumstances, the damping matrix A_{ij} is positive definite for all $\underline{x}$ and λ, so that the system converges to the constraints.

It is possible, however, to pose a problem that has contradictory constraints. For example,

$$g_1(x) = x = 0 \text{ and } g_2(x) = x - 1 = 0. \tag{85}$$

In the case of conflicting constraints, the ALC compromises, trying to make each constraint g_α as small as possible However, the Lagrange multipliers λ_α go to $\pm\infty$ as the constraints oppose each other. It is possible, however, to arbitrarily limit the λ_α at some large absolute value.

For a given constrained optimization problem, it is frequently necessary to alter the ALC to have a region of positive damping surrounding the constrained minima. Arrow [1] combines the multiplier method with the penalty method to yield a modified multiplier method that is locally convergent around constrained minima [1].

The damping matrix is modified by the penalty force to be

$$A_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} + \sum_\alpha \lambda_\alpha \frac{\partial^2 g_\alpha}{\partial x_i \partial x_j} + c_\alpha \frac{\partial g_\alpha}{\partial x_i}\frac{\partial g_\alpha}{\partial x_j} + cg\frac{\partial^2 g_\alpha}{\partial x_i \partial x_j}. \tag{86}$$

Arrow [1] proves a theorem that states that there exists a $c^* > 0$, such that if $c > c^*$, the damping matrix in equation (86) is positive definite at constrained minima. Using continuity, the damping matrix is positive definite in a region R surrounding each constrained minimum. If the system starts in the region R and remains bounded and in R, then the convergence theorem is applicable, and the augmented Lagrangian method converges to a constrained minimum.

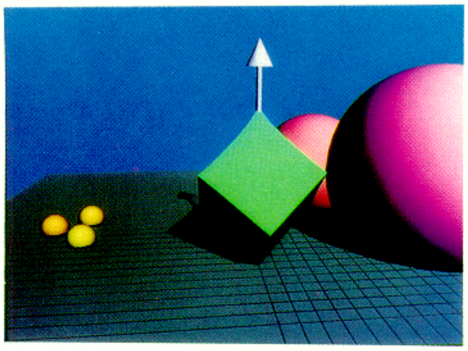

Figure 13: A compressible gelatinous cube is picked up with an RC.

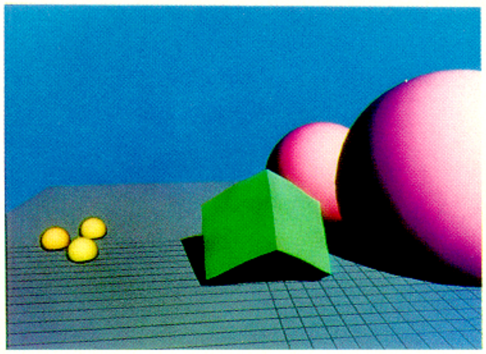

Figure 14: A compressible gelatinous cube hits the table.

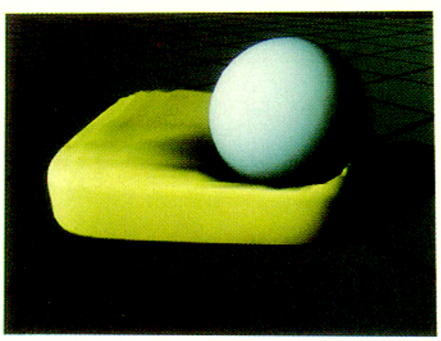

Figure 15: A sphere squashes a seat cushion.

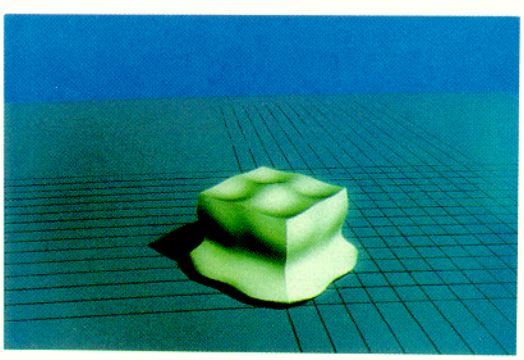

Figure 16: A lump of moldable incompressible clay hits the table.

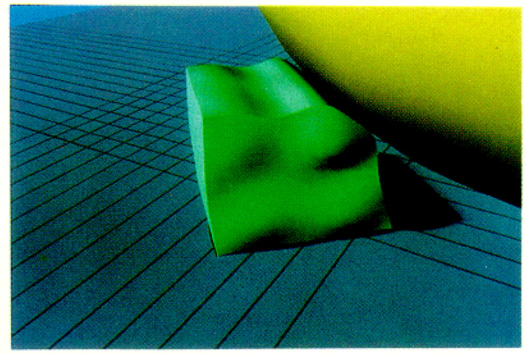

Figure 17: An elastic model is squashed.

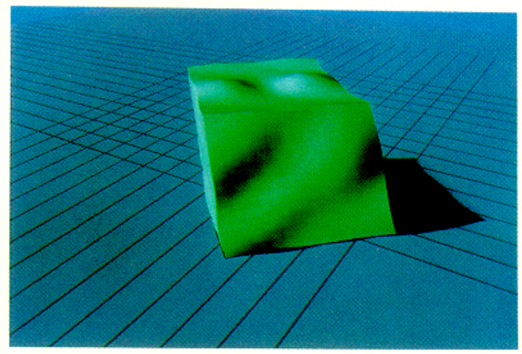

Figure 18: An elastic model returns to its rest shape.

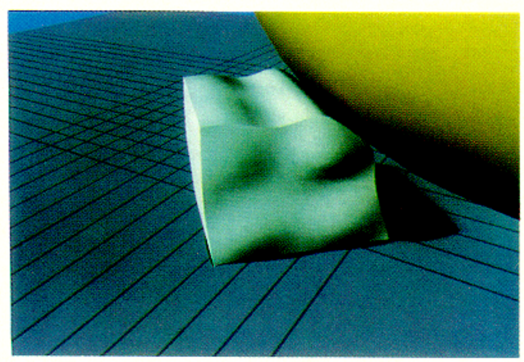

Figure 19: A moldable model is squashed.

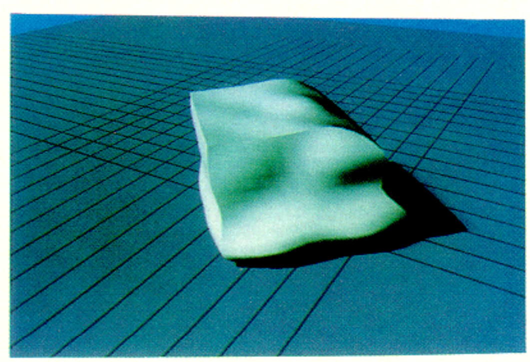

Figure 20: A moldable model assumes a new rest shape after strong deformation.