An Introduction to Physically Based Modeling: Energy Functions and Stiffness

Michael Kass
Pixar

Energy Functions and Stiffness

Michael Kass
Pixar

In earlier sections of these notes, we remarked that there are two main types of problems that arise in practical numerical solution of differential equations: accuracy and stability. Accuracy refers to the small errors introduced when trying to make a numerical approximation to the solution of a continuous differential equation. Stability refers to the problem of whether or not the approximations result in a method which converges at all. In some engineering contexts, high accuracy is very important, but for physically-based computer animation, accuracy tends to be much less important than stability. Small errors in simulations go largely un-noticed in computer graphics, but instabilities usually make the results of a simulation completely unusable. In addition, stability is very often the limiting factor in the computational speed of simulation because we can usually increase the time step of the differential equation solver until the limit of stability to make a simulation run as fast as possible. Here we will examine some of the causes of instability in differential equations resulting from physically-based modeling and discuss ways of dealing with it.

For illustrative purposes, instead of considering correct Newtonian mechanics governed by the law $F = ma$, we will analyze a simpler physical model guided by the law $F = mv$. The simpler model yields "physics" in which nothing moves unless a force is acting on it, in accordance with the theories of Aristotle in early Greece. The advantage of the simpler model is that we can construct interesting behavior directly from energy functions without the need to introduce frictional and damping forces. As a result, the model is particularly easy to analyze, yet it contains the important elements required to understand the source of instability.

The simplest physical system which exhibits the key problems that cause instability is a particle of unit mass in two dimensions affected by a potential energy $E(x, y)$. The force on the particle is given by the vector $(-\partial E/\partial x, -\partial E/\partial y)$ so using Aristotelian physics, the equations of motion of the particle are simply: $\partial X/\partial t = -(\partial E/\partial x)$, and $\partial Y/\partial t = -(\partial E/\partial y)$. The behavior of the system depends, of course, on the precise nature of the energy function.

Suppose we consider the behavior of the above system near an energy minimum. If we consider a small enough region, the energy function will be reasonably well approximated by the first few terms in its Taylor expansion. The constant term in the Taylor expansion does not affect the behavior of the particle since only relative changes in potential energy are significant. The gradient vanishes at the minimum, so the first derivative term has no effect. Thus the second derivative term is the first term which importantly affects the behavior near a minimum. Since the second derivative term gives rise to a quadratic energy, it is particularly interesting to look at how the system behaves when the energy function is a quadratic.

Let us begin by considering the quadratic energy function $E = (x^2 + y^2)/2$. The energy has a minimum at zero where its value is zero. The resulting force on the particle is simply $F = -\nabla E = (-x, -y)$. Hence the force on the particle will be a vector from the position of the particle back to the origin. Under Aristotelian dynamics, the particle will take a straight line path from wherever it starts to the origin. The particle will slow down as it approaches because the gradient gets smaller

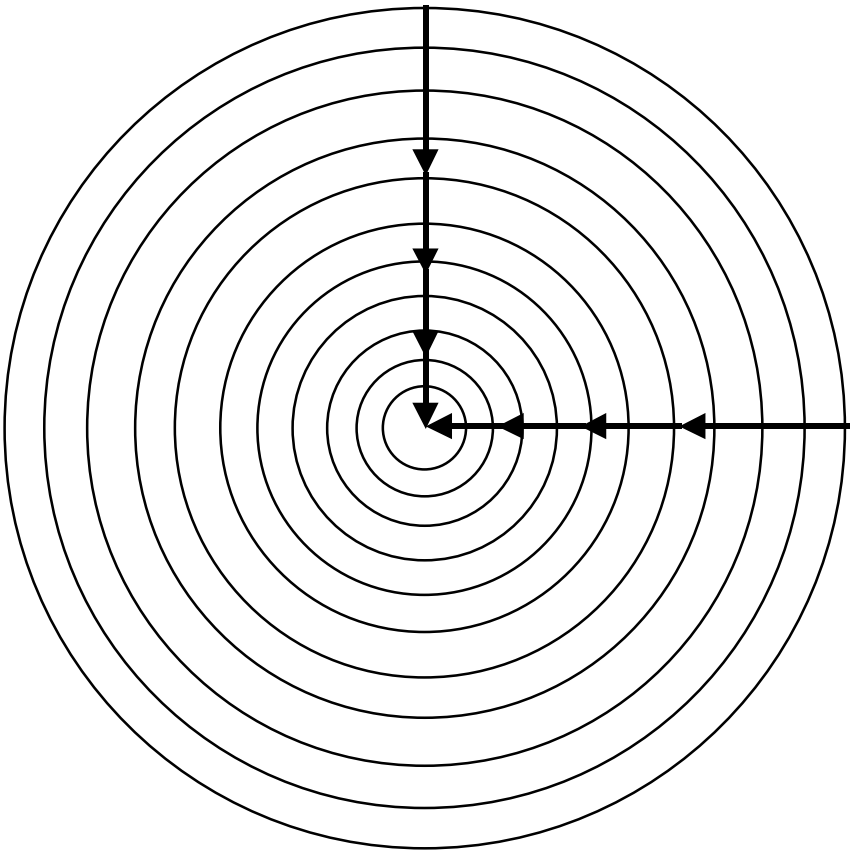

Figure 1: Symmetric energy function $E = (x^2 + y^2)/2$

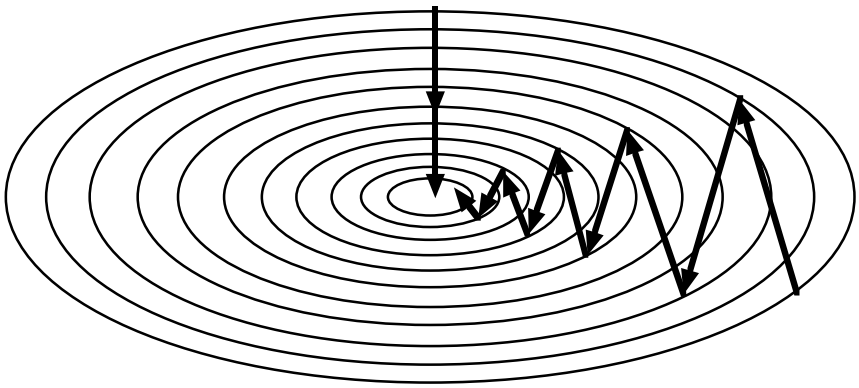

Figure 2: Stretched energy function $E = (\epsilon x^2 + y^2)/2$

and smaller. If the particle is placed at the origin, its velocity will be zero since the gradient vanishes.

Suppose we try to compute the motion of the particle using Euler's method. If P_0 represents the starting position of the particle, its position at the end of the first time step is given by $P_1 = P_0 + h\nabla E = (1-h)P_0$ where h is the time step. Obviously, if h is very small, it will take a large number of iterations before the particle moves close to the minimum at $(0, 0)$. As h gets larger, the particle moves further in one iteration, so the simulation proceeds more rapidly. In the range $0 < h < 1$, the iteration yields exponential decay with time constant proportional to $1/h$. If $h = 1$, the particle moves directly to the minimum in one step, which is the fastest possible simulation. If h is increased beyond this point, the simulation oscillates, flipping from one side of the origin to the other on each iteration. Nonetheless, as long as $h < 2$, the position of the simulated particle converges towards the energy minimum at $(0, 0)$. On each iteration, the sign of the vector from the origin to the particle changes, but the magnitude decrease. If $h > 2$, however, the iteration diverges. On each iteration, the position of the particle reflects around the origin and gets further away. Figure 1 shows some level curves for the energy function and some paths to the minimum.

If the energy function is a symmetric quadratic as above and the step size is set appropriately, Euler's method works quite well. How could we complain about being able to jump to the minimum in a single step? Even if we do not know the correct step size to use, it should be possible to get to the minimum in a relatively small number of step. It would be even better if the iteration were stable for all step sizes, but this can nonetheless be regarded as a success for Euler's method.

Something very different occurs if we take the original symmetric energy function and stretch it out in one direction. Suppose we have $E = (\epsilon x^2 + y^2)/2$. Then the resulting force on the particle is $F = -\nabla E = (-\epsilon x, -y)$ and the force no longer points to the minimum unless $x = 0$, $y = 0$, or $\epsilon = 1$. Now consider what happens if we use Euler's method. We begin at some point $P_0 = (x_0, y_0)$. The next point is $P_1 = ((1 - h\epsilon)x_0, (1 - h)y_0)$. Note that the y component of the Euler iterates in this case will diverge if $h > 2$. Thus, even at the largest step size that will still maintain stability, the value of x decreases each iteration by only a factor of $(1 - 2\epsilon)$, which describes exponential decay with a time constant proportional to $1/\epsilon$.

Note the profound change that resulted from just a simple scaling of the energy function. With the symmetric energy, we can get to the minimum in a single iteration if we pick the right step size. Otherwise, if we pick a stable step size, we approach the minimum with a time constant proportional the the inverse of the step size. With the stretched energy, however, even if we pick the largest pos-

sible stable step size, we approach the minimum with time constant proportional to $1/\epsilon$. How bad is this? Arbitrarily bad. In real problems, $1/\epsilon$ can be on the order of 10^6 or 10^9 or worse. Clearly this is a serious difficulty.

In order to understand the nature of the difficulty, consider what the stretched energy looks like. As ϵ gets very small, the energy function stops looking like a symmetric ball, and looks increasingly like a steep ravine. At the bottom of the ravine, there is a very small slope towards the minimum, but anywhere else, the dominant slope is towards the bottom of the ravine. As a result, a simulation using Euler's method will primarily send the particle back and forth, up and down the sides of the ravine, while very, very slowly drifting towards the minimum. This is shown in fig. 2. The problem is that Euler's method is based on a local planar approximation of the energy function. When the energy function begins to look like a highly curved ravine, that approximation becomes very poor unless the neighborhood and hence the step size is very small. As a result, Euler's method becomes extremely slow in these cases.

Note that when we stretch the energy function, we change the time constants on the true solutions to the differential equations. In the symmetric case, the solution is $x = x_0 e^{-t}$, $y = y_0 e^{-t}$. In the stretched case, the solution is $x = x_0 e^{-\epsilon t}$, $y = y_0 e^{-t}$. The time constants for the symmetric case are the same, but in the stretched case, the time constants have a ratio of e. This is the usual root of "stiffness" in differential equations. As we saw, the maximum step size is set by the process with the fastest time constant, and this can make simulation of the longer time constant unbearably slow.

In order to deal with the difficulties of stiffness, we have two main choices. We can either use a method based on a more accurate local model of the energy landscape, or we can try to formulate a different but related set of differential equations capable of solving our original problem. Reformulating the problem to eliminate the stiffness is preferred when possible, but very often there is no choice besides simulating the stiff equation.

In many uses of physically-based modeling, people have been tempted to use springs of various sorts to enforce constraints. The problem is that keeping constraints satisfied with springs requires making the spring constants very large, which gives rise to stiff systems of equations. In many cases, Lagrange multipliers or changes of variables can make it possible to achieve the same results with differential equations that are much less stiff.

If it is not possible to reformulate the differential equations to remove the stiffness, the only practical way to solve them is to use a technique which combats stiffness using a better local model of the forces than Euler's method. In general, these are known as implicit methods, one of which is treated in detail in "Continuum Dynamics for Computer Graphics" elsewhere in these notes.

The details of implicit differential equation solvers are treated in a wide variety of texts. Here we attempt to give a little intuition about how they work. Remember that the gradient of the symmetric energy points towards the minimum, but not the gradient of the stretched energy. This is a large part of the difficulty. If we keep moving in the gradient direction, we will get to the minimum very slowly. In order to take a very large step, therefore, a differential equation solver should move directly towards the minimum, even when faced with the stretched energy. How can a differential equation solver know which way to go to get to the minimum? In the general case, this can be arbitrarily difficult, but for quadratic energies, it is particularly simple. Quadratic energies have linear gradients. The minimum occurs when the gradient vanishes, so finding the point where this occurs can be done by solving a set of linear equations. For the case of the stretched energy, the gradient is $(\epsilon x, y)$ which clearly vanishes only at the point $(0, 0)$. Thus, by solving an appropriate linear equation, a differential equation solver can move towards the minimum of a quadratic energy function.

In general, the best linear estimate for the gradient of an energy function near a point x_0 is given

by the expression $\nabla E(x) \approx \nabla E(x_0) + H \cdot (x - x_0)$ where H is the matrix of second derivatives or Hessian matrix given by $H = \partial^2 E / \partial x_i \partial x_j$. Since H is symmetric, by a suitable orthogonal change of basis, it is always possible to turn H into a diagonal matrix. For the symmetric energy function, H is already diagonal: it is simply a two by two identity matrix. For the stretched energy, H is also already diagonal with entries $(\epsilon, 1)$. In the basis that diagonalizes H, the condition number of H is defined to be the ratio between the magnitudes of the largest and smallest diagonal entries of H. If the condition number is small, then we have a nearly symmetric energy function and the equations will not be stiff. If the condition number is large, it means that if we consider the two dimensions which gave rise to the smallest and largest diagonal entries of H, we have the situation of the stretched energy function with $1/\epsilon$ corresponding to the condition number. Clearly, if the condition number is large, we are potentially in a troublesome situation.

The symmetric and stretched energy functions, point out a canonical difficulty in solving differential equations as well as a canonical solution. If we look at a suitable neighborhood of any smooth energy function, we can approximate its behavior as a quadratic. If we pick the right coordinates, the Hessian matrix will be diagonal, so we can characterize the local behavior in those coordinates by the diagonal entries of the Hessian. If the entries are all of approximately the same magnitude, then the energy function is reasonably symmetric at least locally, and our numerical problems are relatively minor. If the entries are of wildly different magnitudes, then if we pick the two coordinates corresponding the the smallest and largest magnitude entries, we have an energy function that looks locally like a steep ravine. In order to take large steps in an energy function of this shape, we need to use an implicit method which will examine the matrix H to find the axis of the ravine, so it can travel down the ravine directly towards the minimum. We will see an example of one such method in "An Introduction to Continuum Dynamics for Computer Graphics."