Adaptive timesteps

Our problem -- a cannonball shot towards the Moon -- illustrates one of the big challenges for numerical methods: dealing with forces and accelerations which vary over wide ranges.

When the projectile is flying through the Earth's atmosphere at high speed, it encounters very large forces of air resistance. For example, if we use this simple approximation

$$F(\text{air}) = C * (\text{density}) * (\text{area}) * (\text{speed})^2$$

and some reasonable parameters, we end up with

$$F(\text{air}) = 0.02 * (1.21\ \text{kg/m}^3) * (12\ \text{m}^2) * (10{,}000\ \text{m/s})^2$$

$$= 29{,}000{,}000\ \text{Newtons}$$

Compare that to the force of gravity on our projectile, which has a mass of **10 tonnes**:

$$F(\text{grav}) = \text{mass} * g$$

$$= 98{,}000\ \text{Newtons}$$

Yow! The force of air resistance overwhelms the gravitational force.

On the other hand, consider the force on the same projectile once it has gone above the Earth's atmosphere. At an altitude of 200 km, for example, air resistance is negligible, and the force of gravity drops to

$$F(\text{grav}) = G * M(\text{earth}) * m(\text{ship}) / (\text{ total } R)^2$$

= 92,400 Newtons

When the cannonball is halfway to the Moon, this force is very much weaker still:

F(grav) = G * M(earth) * m(ship) / (1.92E8 m)^2

= 108 Newtons

The total force on the projectile varies by many orders of magnitude (by a factor of roughly 270,000, to be specific). What does this mean?

- When the forces are LARGE, the acceleration is large. The cannonball's velocity will change rapidly, so we must take small steps in time to avoid errors.
- When the forces are SMALL, the acceleration is small. The cannonball's velocity will change very gradually, so we *could* take large steps in time and still avoid errors.

If we use a fixed timestep for the entire simulation, we will have to pick a small one, like **dt = 0.01 seconds**, to avoid errors during the first few seconds of flight. But that's a pain in the neck:

Q: If we use a stepsize of 0.01 seconds,
and the total flight takes 5 days,
how many steps must we make?

One way to avoid such long execution times is to vary the size of the timestep: use a small step when the accelerations are large, and a large step when the accelerations are small. There are two questions we must answer:

1. How can we determine which sort of change -- if any -- we should make to the timestep?
2. How do we know if we picked too large a timestep and ruined the accuracy of the results?

What sort of a change should we make to the timestep?

The basic idea is to use the size of the acceleration of the cannonball as an indicator: big accelerations mean we need small timesteps. Actually, that's not quite true: if we use Heun's method, then we can handle even a large acceleration AS LONG AS IT IS CONSTANT. The trouble occurs if the acceleration changes quickly and significantly from one timestep to the next.

Fortunately, Heun's method provides a very simple way to check signficant changes: it includes the prediction of the acceleration we'll experience in a future timestep. So, we can simply compute a fractional change:

$$\text{frac\_change} = \text{abs}\left(\frac{\text{current accel} - \text{predicted accel}}{\text{current accel}}\right)$$

If the fractional change in acceleration is larger than some threshold, we might have to choose small timesteps; if it is smaller than some threshold, we might try using larger timesteps.

A simple algorithm for varying the timestep might look sort of like this:

```
compute frac_change in acceleration

if (frac_change > thresh_1)
   new_timestep = 0.5 * timestep;

if (frac_change < thresh_2)
   new_timestep = 2.0 * timestep;

otherwise, leave timestep as-is
```

Note that this algorithm allows the timestep to float into any range; in some circumstances, you might want to set minimum and maximum permitted values.

What are the appropriate threshold values? Good question. They will probably depend exactly on the nature of the calculations you are performing. The next section may help you to pick reasonable values.

How do we know if we picked too large a timestep?

If you know the exact solution to your problem, then you can compare the simulated results to the exact solution and decide if the errors are acceptable. In this week's case, however -- a projectile flying through the atmosphere -- we don't know the exact solution. Rats.

Could we use conserved quantities, such as total energy or angular momentum, as a check? In some situations, yes -- but in this situation, no: the total energy drops a bit as the projectile suffers air resistance.

Here are two things we can do:

- use fixed timesteps of various sizes on test cases, such as a cannonball with initial velocity 1000 m/s or 5000 m/s. Look at the change in the final results of those test cases as a function of the step size. When the final results change by a negligible amount, you know a safe upper bound on the stepsize for flight through the atmosphere.
- pick a section of the motion during which some quantity **should be** conserved. For example, once the projectile leaves the atmosphere, it should no longer lose energy to air resistance. So, if you compare the total energy of the projectile at some point above the atmosphere to the total energy at some other point above the atmosphere, there should be no difference.

Of course, just about any technique will yield some small change in energy. You must decide, for each particular situation, what sort of change is important, and what sort of change is not important. For example, you might decide that an error which changes the travel time by a few minutes -- out of a total of several days -- is not important. Once you've made this decision, you can start fiddling with the thresholds and limits on your timestep to make sure that your final result falls within the acceptable range.

Exercise in class: find appropriate limits on the timestep

Here's a version of the program for last week's assignment: it simulates the motion of a cannonball without air resistance, but including a changing gravitational force.

- cannonball_a.sci

You can call it like so:

```
cannonball_a(initial_speed, timestep, "heun", "test.out")
```

This will write diagnostic messages to the screen as the program runs, and also write a more compact set of output to the file "test.out".

1. use an initial speed of 1000 m/s
2. find a safe "largest good timestep" in the following manner: examine the maximum height reached by the ball, and require that it be within 1 percent of the TRUE height (which you can compute analytically)

Here's a slightly different program: it includes air resistance, but NOT the changing gravitational force from the Earth.

- cannonball_b.sci

You can call it like so:

```
cannonball_b(initial_speed, timestep, "heun", "test.out")
```

1. use an initial speed of 3000 m/s
2. find a safe "largest good timestep" in the following manner: examine the maximum height reached by the ball, and require that it be within 1 percent of the maximum height reached with a much smaller timestep

Source: http://spiff.rit.edu/classes/phys317/lectures/adaptive_step.html