Euler's Method and Excel

Introduction

Euler's method allows us to find approximate solutions to differential equations. Developed by Leonhard Euler in the middle of the 18th century, it allowed mathematicians to find approximate solutions with simple hand calculations. Although simple, good solutions required hundreds, if not thousands, of computations, which today are easily handled by computers. In this activity, you will use Excel to perform these computations to simulate ballistic flight. We know the solution should be a parabolic trajectory so we can verify that we've implemented Euler's method correctly. We'll then modify the spreadsheet to solve a more complex problem numerically which would otherwise be extremely complex using purely analytical methods.

Lots of Linear Approximations

If we "zoom-in" to a continuous region of a function, or in other words, examine the function over a very small interval, the function will look essentially like a straight line. If it doesn't, all we have to do is zoom in even further until it does. This property is called *local linearity*. Over that small interval, we can approximate the function as a straight line. If we move to the right or the left of that interval, the function will again look like a line, but it may have a different slope. This means that a true curving function can be approximated as a sequence of short, connected line segments, with the slop of each segment varying from segment to segment. Euler's method makes use of this approximation.

Consider the differential equation

$$\frac{dy}{dx} = y$$

This DE tells us that the slope of the function at any point (*x,y*) is equal to the *y* coordinate. If the point is (1,3), then $\frac{dy}{dx} = 3$. So what is the value of the function at $x = 1.01$? The interval we're examining is $\Delta x = 0.01$. So if the slope of the the function is

$$\frac{\mathrm{dy}}{\mathrm{dx}} = 3$$

then the approximate change in the vertical direction is

$$\Delta y = m\Delta x$$

$$\Delta y = \frac{dy}{dx}\Delta x$$

$$\Delta y = (3)(0.01) = 0.03$$

Now that we know the next point is (1.01,3.03), then $\frac{dy}{dx} = y \approx 3.03$ so $\Delta y = 3.03\Delta x = .0303$, and we get (1.02, 3.0603). We can continue this process by using the current point to approximate the next point, then making that the current point, and so on. Since we're making only linear approximations, and then only keeping a certain number of digits, our accuracy will go down the further we project the approximation away from our starting point. However, if we want greater accuracy, we can just make our Δx smaller, and start over.

The complexity of this process is based on the complexity of $\frac{dy}{dx}$. Even if the derivative is complicated, all we have to do is plug-in the *x*- and *y*-values into the formula for the derivative get the slope of the linear approximation.

Ballistic Flight in a Vacuum

We will examing the flight path of an object in a gravitational field, neglecting the effect friction (air resistance). Think about throwing a ball on the moon. So what are our differential equations? Let's start with Newton's Second Law, $F = ma$. The acceleration in our case is due to gravity. This acceleration is only in the *y* direction. Lets define up as the positive direction so our vertical accelleration $a_y = g = -9.8\ m/s^2$. There is no horizontal acceleration, so $a_x = 0$.

Let's put our origin at the launch point, so our initial point is $(x_0, y_0) = (0,0)$. We want to plot *y* versus *x* (altitude versus downfield position), and we will track our x and y position both as a function of time. However, our accelleration is the *second* derivative of our position function , that is, $a_x = \frac{d^2x}{dt^2}$ and $a_y = \frac{d^2y}{dt^2}$. Euler's method uses the slope of a function, which is the first derivative, so what do we do?

We use the fact that the acceleration is the first derivative of the velocity, and the velocity is the first derivative of position. We'll use Euler's method once to find the velocity of the object from its acceleration, and then use it again to find its position from its velocity.

$$\frac{d^2y}{dt^2} + \text{Euler's Method} \rightarrow \frac{dy}{dt} + \text{Euler's Method} \rightarrow y$$

Since the derivatives are with respect to time, we will use small time-steps of $\Delta t = 0.2\ sec$. Finally, since there is no horizontal acceleration, the horizontal velocity won't change. Here are the initial conditions

Initial Position: $(x_0, y_0) = (0, 0)$

Launch Angle: $\theta = 45^\circ = \pi/4$ radians

Launch Velocity = v_0 = 75 m/sec → $(v_{0x}, v_{0y}) = (v_0\cos\theta, v_0\sin\theta)$

Mass of object = 1 kg

and the initial calculations.

t	a_y	v_y	y	v_x	x
0	-9.8	53.03301	0	53.03301	0
0.2	-9.8	$v_y(0.2) = v_y(0) + a_y(0)\Delta t$	$y(0.2) = y(0) + v_y(0)\Delta t$	53.03301	$x(0.2) = x(0) + v_x(0)\Delta t$
0.4	-9.8	$v_y(0.4) = v_y(0.2) + a_y(0.2)\Delta t$	$y(0.4) = y(0.2) + v_y(0.2)\Delta t$	53.03301	$x(0.4) = x(0.2) + v_x(0.2)\Delta t$
0.6	-9.8	$v_y(0.6) = v_y(0.4) + a_y(0.4)\Delta t$	$y(0.6) = y(0.4) + v_y(0.4)\Delta t$	53.03301	$x(0.6) = x(0.4) + v_x(0.4)\Delta t$

Now open the file \all access\joshinr\class notes\calc\eulermethod.xlsx and implement Euler's method. Graph *y* versus *x* to confirm your spreadsheet is working correctly. You can fit a trendline to your data to determine how accurate your calculations are.

Ballistic Flight with Air Resistance

Now that you have Euler's method working in Excel, we can make our ballistic model more accurate. Usually we ignore friction effects because they're quite complicated and make analytical (formula based) solutions extremely difficult, if not impossible. That is, friction can make the differential equation extremely complicated and difficult to solve. However, with Euler's method, we can make numerical approximations regardless of the complexity of the differential equation. We can then plot those approximations to understand the nature of the solution. We will now add air resistance to our ballistic model.

Adding drag from air resistance is actually quite simple. Air resistance slows an object down. More accurately, friction is a force, and in flight causes drag acting in the direction opposite of the object's velocity. It turns out that air resistance is proportional to the square of the object's velocity. We call the proportionality constant C_d the *drag coefficient.*

$$F_{drag} = C_d v^2$$

Now, remember that $F = ma$, so if we divide the drag force by the object's mass, we can determine the acceleration that force is causing on the object. We add this acceleration to all the other accelerations the object feels, giving us a net acceleration. We then use this net acceleration in our Euler calculations.

$$F_{drag} = ma_{drag} \rightarrow \frac{F_{drag}}{m} = a_{drag} \rightarrow a_{net} = a + a_{drag}$$

In air, the drag coefficient is quite small and by convention, $C_d > 0$. It's important to remember that the drag force always acts in the opposite direction from the velocity, so when the object is rising, drag is pointing down, and when the object is falling, the drag is pointing up. Because of this we need to be careful to keep track of our signs when adding the acceleration from drag to other accelerations in the model.

Here are the adjustments we need to make to our spreadsheet.

t	a_y	a_{drag}	a_{y-net}	v_y	y
0	-9.8	0	-9.8	53.03301	0
0.2	-9.8	$C_d[v_y(0)]^2/m$	$a_{ynet}(0) = a_y + a_{drag}(0)$	$v_y(0.2) = v_y(0) + a_{ynet}(0)\Delta t$	$y(0.2) = y(0) + v_y(0)\Delta t$
0.4	-9.8	$C_d[v_y(0.2)]^2/m$	$a_{ynet}(0.4) = a_y + a_{drag}(0.2)$	$v_y(0.4) = v_y(0.2) + a_{ynet}(0.2)\Delta t$	$y(0.4) = y(0.2) + v_y(0.2)\Delta t$
0.6	-9.8	$C_d[v_y(0.4)]^2/m$	$a_{ynet}(0.6) = a_y + a_{drag}(0.4)$	$v_y(0.6) = v_y(0.4) + a_{ynet}(0.4)\Delta t$	$y(0.6) = y(0.4) + v_y(0.4)\Delta t$

It's also important to remember that air resistance affects not only the vertical velocity, but also the horizontal velocity. In the previous model, there was no horizontal acceleration. In this model, air resistance now creates an acceleration opposite the horizontal velocity.

t	a_x	a_{drag}	a_{xnet}	v_x	y
0	0	0	0	53.03301	0
0.2	0	$C_d[v_y(0)]^2/m$	$a_{xnet}(0.2) = a_x + a_{drag}(0)$	$v_x(0.2) = v_x(0) + a_{xnet}(0)\Delta t$	$x(0.2) = x(0) + v_x(0)\Delta t$
0.4	0	$C_d[v_y(0.2)]^2/m$	$a_{xnet}(0.4) = a_x + a_{drag}(0.2)$	$v_x(0.4) = v_x(0.2) + a_{xnet}(0.2)\Delta t$	$x(0.4) = x(0.2) + v_x(0.2)\Delta t$
0.6	0	$C_d[v_y(0.4)]^2/m$	$a_{xnet}(0.6) = a_x + a_{drag}(0.4)$	$v_x(0.6) = v_x(0.4) + a_{xnet}(0.4)\Delta t$	$x(0.6) = x(0.4) + v_x(0.4)\Delta t$

Now copy your calculation block and paste it next to itself. Add columns to calculate the acceleration due to air resistance and the new net acceleration in the *x* and *y* directions. Then propagate these changes to your velocities and positions. Plot this second set of x and y data alongside your first plot to see the effect of air resistance.