18.03 Class 2, February 10, 2006

Numerical Methods

[1] The study of differential equations rests on three legs:

. Analytic, exact, symbolic methods

. Quantitative methods (direction fields, isoclines ....)

. Numerical methods

Even if we can solve symbolically, the question of computing values remains.
The number e is the value y(1) of the solution to
y' = y with y(0) = 1. But how do you find that in fact
e = 2.718282828459045.... ? The answer is: numerical methods.

As an example, take the first order ODE y' = x - y^2 = F(x,y)
with initial condition y(0) = 1. Question: what is y(1) ?

I revealed a picture of the direction field with this solution sketched.
This is what we had on Wednesday but upside down: then we considered
y' = y^2 - x . The funnel is at the top
this time. This solution seems to be one of those trapped in the funnel,
so for large x , the graph of y(x) is close to the graph of \sqrt(x) .
But what about y(1) ?

Here's an approach: use the tangent line approximation!

Since F(0,1) = -1 , the straight line best approximating the integral curve
at (0,1) has slope -1 , and goes through the point (0,1) : so this
gives the estimate y(1) is approximately 0 .

Well, we know that the integral curve is NOT straight. What to do?

Approximate it by a polygon!

So use the tangent line approximation to go half way, and then check
the direction field again:

y(.5) is approximately 1 + (slope)(run) = 1 + (-1)(.5) = .5
(.5,.50) is a vertex on this polygon. The direction field there has slope

F(.5,.5) = .5 - (.5)^2 = .25 .

We follow the line segment from there with this slope for a run of .5
to get to the point with x = 1 and
y = .5 + (slope)(run) = .5 + (.25)(.5) = .625 .

At this point I invoked the Mathlet Euler's Method. There t replaces x .

[2] This is a general method for computing y(b) from y(a) and the
direction field. In fact, it approximates y(x) for all x between a and b.

Pick a step size h (1 or 1/2 above). It should probably be so that $b = a + nh$ for some positive whole number n. We'll use n steps of size h to get from a to b .

Successively compute the vertices of the polygon. Do this in an organized way:
The x coordinates are easy:

$x_0 = a$, $x_1 = a + h$, $x_2 = a + 2h$, ... , $x_n = a + nh$, ... $x_N = b$.

The y cooridinates you compute, using the direction field.

Write $A_0 = F(x_0,y_0)$, $A_1 = F(x_1,y_1)$, and so on. Then

In the line $n = 1$, $y_1 = y_0 + h A_0$

In the line $n = 2$, $y_2 = y_1 + h A_1$

and in general $y(k+1) = y_k + h A_2$

This is "Euler's method."

[3] Keep this calculation organized in a talbe. We'll do our example one more time: take $n = 2$, so $h = 1/2$

k	xk	yk	Ak = xk - yk^2	h Ak
0	0	1	-1	-.5
1	.5	.5	.25	.125
2	1.0	.625		

I showed some more Euler polynomials for this equation.

Euler's method is rarely exact. Let's identify some sources for error. Much of numerical analysis is understanding and bounding potential error.

(1) Round off accumulation. Each computation involves a small error, from round off, and these errors accumulate. So you want to use the fewest
number of steps possible. On the other hand making the stepsize small keeps the polygonal approximation close to the actual solution. There is a tension here.

Usually more important than this is:

(2) The Euler polygon is not an exact solution, and the direction field at its vertices differs more and more from the direction field under the actual solution. At places where the direction field is changing rapidly,

this produces very bad approximations quickly.

For short intervals, at least, we can at least predict whether the Euler method will give an answer that is too large or too small. In this example, the direction field turns up secretly while the polygon is running;
so the actual answer is TOO BIG. This is a general thing.

I invoked the example F(t,y) = -ty with the initial condition (-2.2,0.10).

QUESTION: is the Euler approx for y(-1.2)

1. Too high

2. Too low

Blank. Not sure

A click shows: too small. [This was a poorly chosen example.]

If the integral curve is bending up, the Euler approximation is too low.
If it's bending down, the Euler approximation is too high.

[4] The way to address the the problem of variability of the direction field
is to poll the values of the direction field in the area that the next strut
in the Euler polygon plans to traverse, and use that information to give a
better derivative for the next strut. These polling methods can be very clever.

If you poll once at the midway point and simply take the average, you get
the "improved Euler method" (also known as Heun's method).

If you poll four times sequentially (in a certain very clever way) you get
"RK4."

It is interesting to compare these methods. Here's a comparison
using the ODE y' = y, y(0) = 1; the solution is y = e^x,
and we study y(1) = e.

Equal cost:

Method	Steps	Evaluations	Error
RK1 = Euler	1000	1000	1.35 x 10^{-3}
RK2 = Heun	500	1000	1.81 x 10^{-6}
RK4	250	1000	7.99 x 10^{-15}

Each evaluation of the direction field costs money - it takes time.
Heun polls twice per step, and RK4 polls 4 times, so the cost of

Euler with 1000 steps is about the same as the cost of Heun with 500 steps or RK4 with 250 steps.

The error for the Euler method is around 1/1000, even using 1000 steps of stepsize 1/1000. This reflects a general theorem: the expected error of Euler's method is proportional to the stepsize h.

For Heun the theory predicts error proportional to h^2. $h = 2 \times 10^{-3}$ here, so $h^2 = 4 \times 10^{-6}$; in the event we do better.
For RK4 it predicts error proportional to h^4, which is 16×10^{-12} here; in the event RK4 is even more accurate.

The moral is that for good accuracy Euler is essentially useless, and RK4 will always win. There are still higher order methods, but they involve more overhead as well and experience has shown that RK4 is a good compromise.

[5] There is a third source of error to keep in mind:

(3) Catastrophic overshoot.

Go back to the applet, with $F(t,y) = -ty$ and initial condition $(-2.2,.10)$.
Follow the Euler polygon for $h = 1$ a little further. The Euler polygon goes WILD, even though obviously the solutions are all asymptotically zero.

Another type of overshoot can be seen in the $y' = y^2 - t$ example. There, if you start at $(-1.5, -3)$, say, the actual solution enters the funnel, but Euler with h as small as $1/4$ will go off to infinity. In fact, the actual solutions that go to infinity get there in finite time, but these polygons never do that since the slope is finite everywhere.

Beware! ODE solvers are tricky and avoid things like this. One trick: when the direction field is steep, use smaller stepsizes.