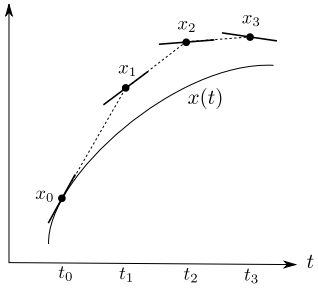
Numerical methods

Lecture3: Implicit methods for numerically solving ODEs

Oscar Benjamin and Lucia Marucci

Euler's method in pictures

$$x_{n+1} = x_n + f(x_n, t_n)h$$



Solving ODEs numerically: exercises

Using Euler's method, $x_{n+1} = x_n + hf(x_n)$, solve

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = -2x \qquad \text{with} \qquad x(0) = 1$$

for 5 steps with

1. $h = \frac{1}{4}$
2. $h = \frac{1}{2}$
3. $h = 1$
4. $h = 2$

What differences do you find? (What is the correct solution?)

Solving ODEs numerically: when things go wrong

As you can tell from the exercises, the choice of the step size h is critical

- If the stepsize is too large, Euler's method becomes *unstable* and the solution diverges (goes off to infinity).
- If the stepsize is too small, the calculation takes a long time; this is a problem if the ODE is very complicated.

To understand the stability of an integration method, test ODEs are used which we know the answer for, e.g.,

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = -Ax, \qquad \text{with} \qquad x(0) = 1$$

Solving ODEs numerically: when things go wrong

The *exact solution* to this equation is $x(t) = \mathrm{e}^{-At}$ and so $x \to 0$ as $t \to \infty$ when $A > 0$.

Euler's method gives us

$$x_{n+1} = x_n - hAx_n = x_n(1 - hA)$$

To ensure that $x \to 0$ as $n \to \infty$, we require $|1 - hA| < 1$. Since $h > 0$ and $A > 0$, this means $hA < 2$.

However, if $hA > 1$ then $1 - hA$ will be negative and so x_n will change sign at every iteration (it will *oscillate*). This does not happen in the true solution! So the final constraint on h is

$$h < \frac{1}{A}$$

Solving ODEs numerically: when things go wrong

The problem is that the critical value of h changes from ODE to ODE; we can calculate what it is for this simple, first-order, linear ODE but in general we cannot!

In fact, for some ODEs, Euler's method is always unstable regardless of the step size used! (Try the ODE for simple harmonic motion. . .)

So, although Euler's method is used in practice (*a lot!*) we should be extremely careful with it. One way around this instability is to use an *implicit method* such as *backwards Euler*.

[Note: no numerical method is perfect, even the best ones only work properly in certain situations.]

Solving ODEs numerically: implicit methods

Again, consider the general first-order ODE

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = f(x,t)$$

Euler's method is based on the *forward difference* approximation to a derivative, that is

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} \approx \frac{x(t+h)-x(t)}{h}$$

However, there are several different approximations to $\frac{\mathrm{d}\,x}{\mathrm{d}\,t}$. Another is the *backward difference* approximation

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} \approx \frac{x(t)-x(t-h)}{h}$$

We can rearrange this to find $x(t)$ to get

$$x(t) = x(t-h) + h\frac{\mathrm{d}\,x}{\mathrm{d}\,t} \qquad \Rightarrow \qquad x_{n+1} = x_n + hf(x_{n+1}, t_{n+1})$$

Solving ODEs numerically: implicit methods

Thus the *backwards Euler method* is

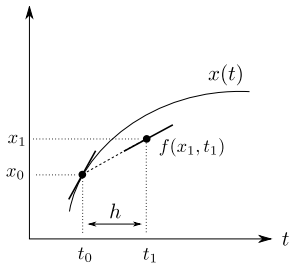
$$x_{n+1} = x_n + hf(x_{n+1}, t_{n+1})$$

Note that this is an *implicit* definition of x_{n+1}. If f is a nasty nonlinear function, we might not be able to find an *explicit* solution for x_{n+1}.

In practice, if we use an *implicit method* we must use a *numerical root finder* (e.g., Newton-Raphson - next section) at each step to find x_{n+1}. (Unless f is simple.)

Backward Euler in pictures

$$x_{n+1} = x_n + hf(x_{n+1}, t_{n+1})$$



Find a point x_{n+1} where the gradient $f(x_{n+1}, t_{n+1})$ points back towards (x_n, t_n).

Solving ODEs numerically: implicit methods

Again, we can use a test ODE to work out when the backwards Euler method is stable and when it is unstable; consider again

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = -Ax, \qquad \text{with} \qquad x(0) = 1$$

In this case, the backwards Euler method gives

$$x_{n+1} = x_n - hAx_{n+1} \qquad \Rightarrow \qquad x_{n+1} = \frac{x_n}{1+hA}$$

Again, the absolute value of the coefficient of the x_n term must be less than one for the method to converge correctly.

In this case, we always have $hA > 0$ and so $0 < \frac{1}{1+hA} < 1$ always!

Thus, the backwards Euler method converges *unconditionally*. (But only for this problem...)

Solving ODEs numerically: recap

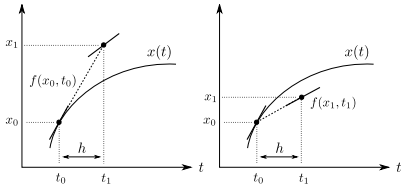
Recap

- Euler's method is a numerical method for integrating ODEs (that is, calculating the solution for a given set of initial conditions).
- It generates a sequence of data points $\{x_0, x_1, x_2, \ldots, x_n\}$ at a discrete set of times $\{t_0, t_1, t_2, \ldots, t_n\}$.
- The spacing between the time points is determined by the *step size* h.
- For a well-behaved numerical method, as $h \to 0$ the error between the discrete approximation and the true solution should also tend to zero.
- Higher-order ODEs can be transformed into *state-space form* and solved as a set of *coupled* first-order ODEs.
- *Great care must be used when choosing* h*!*
- Implicit methods (e.g., *backwards Euler*) can help with stability but are more computationally expensive.

Solving ODEs numerically: implicit methods

So far we've consider the *forwards Euler* and *backwards Euler* methods. We now consider higher-order schemes.

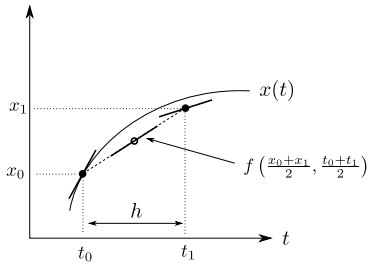
Backward and forward Euler both use the slope at the *end-points* of the interval to calculate the step. Remember the extra accuracy gained from using the *midpoint* in the explicit midpoint method...



Solving ODEs numerically: implicit midpoint method

One step of the implicit midpoint method is given by

$$x_{n+1} = x_n + hf\left(\frac{x_n + x_{n+1}}{2}, \frac{t_n + t_{n+1}}{2}\right)$$



Using the midpoint gives a more accurate estimate of the average slope over the interval $[t_n, t_{n+1}]$.

Solving ODEs numerically: implicit methods

Thus the *implicit midpoint method* for ODEs

$$x_{n+1} = x_n + hf\left(\frac{x_n + x_{n+1}}{2}, \frac{t_n + t_{n+1}}{2}\right)$$

is an *implicit method* for solving ODEs (since we have to solve to find x_{n+1}, which we can often only do numerically).

The implicit midpoint method for ODEs is $O(h^2)$ accurate; where both the forward and backward Euler methods are only $O(h)$ accurate.

Both the backward Euler and implicit midpoint methods are more *stable* than any of the explicit methods.

Stiff and non-stiff ODEs

Rule number 1: For most ODEs we will just use Runge-Kutta with a small enough step-size h.

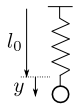
Some ODEs are *stiff* and in this case explicit methods (forward Euler, explicit midpoint, Runge-Kutta, etc) will often be unstable even for very small h.

Rule number 2: For stiff ODEs use a special stiff ODE solver (usually an implicit method like backward Euler, implicit midpoint etc).

Some ODEs have additional geometric properties that need further care...

ODEs with geometric properties

Consider a mass on a spring which is stretched an amount y and has natural length l_0 and stiffness k:



Let's just ignore gravity in this example (it doesn't change the result much). If we also have no damping then the equation describing this is

$$m\frac{\mathrm{d}^2 y}{\mathrm{d}\,t^2} = -ky$$

Since there is no damping *energy* is conserved which means that

$$\frac{1}{2}mv^2 + \frac{1}{2}ky^2 = E = const.$$

ODEs with geometric properties

For simplicity take $m = 1$ and $k = 1$ so that we have

$$\frac{\mathrm{d}^2 y}{\mathrm{d}\,t^2} = -y, \qquad \frac{1}{2}v^2 + \frac{1}{2}y^2 = E$$

We can rewrite this in state-space form as two first order ODEs

$$\frac{\mathrm{d}\,y}{\mathrm{d}\,t} = x \qquad \frac{\mathrm{d}\,x}{\mathrm{d}\,t} = -y$$

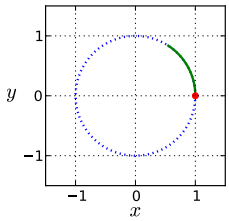
Now since $v = \frac{\mathrm{d}\,y}{\mathrm{d}\,t} = x$ we have that

$$x^2 + y^2 = 2E$$

which is the equation of a circle of radius $\sqrt{2E}$ in the xy-plane...

ODEs with geometric properties

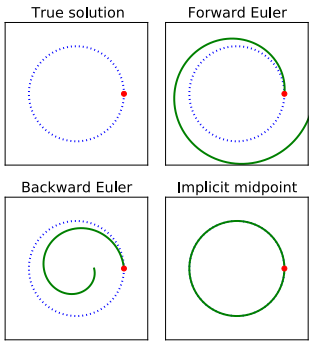
Given initial conditions $y(0) = 0$ and $y'(0) = x(0) = 1$ we have the solution $x = \cos t$, $y = \sin t$ which lies on a circle of radius 1.



So what happens if we try to solve these ODEs numerically?

ODEs with geometric properties

Which methods conserve energy by giving a solution that lies on the circle?



University of BRISTOL
Numerical methods
March 15th 2018

ODEs with geometric properties

Other ODEs have different geometric properties e.g. *conservation of energy* or *angular momentum* etc..

Need to use special methods to ensure that the numerical solution matches the qualitative behaviour of the true solution.

Such methods are known as *geometric* or *symplectic* integration methods.

We won't cover any except from the implicit midpoint method here but there are many...

University of BRISTOL
Numerical methods
March 15th 2018

Solving ODEs numerically: special ODEs

There are many different sub-types of ODEs with different properties

- *Stiff ODEs* — use an *implicit integrator* for stability.
- *Energy preserving ODEs* — use a *geometric integrator* to make sure the energy of the system remains constant.
- *Hamiltonian systems* — use a *symplectic integrator* to preserve the Hamiltonian structure.
- *Differential algebraic equations (DAEs)* — ...

All of these can be extended with *variable step-size* modifications to increase computational performance.

Rule of thumb: if in doubt, try *Runge-Kutta* and vary the step-size until the output doesn't change when you halve the step-size. If that fails, try an implicit method.

University of BRISTOL
Numerical methods
March 15th 2018

Exercises

- Using a stepsize of $h = 0.5$ numerically solve $\frac{\mathrm{d}x}{\mathrm{d}t} = x$ with $x(0) = 1$ to estimate $x(1)$ (true answer is e). What is the global error using the Euler, explicit midpoint or Runge-Kutta methods?
- Redo the example from the start of this lecture using the *backwards-Euler* method i.e. 5 steps to solve $\frac{\mathrm{d}x}{\mathrm{d}t} = -2x$ with $x(0) = 1$ and stepsizes $h = \frac{1}{4}, \frac{1}{2}, 1, 2$. What is the true solution and how does the *local error* seem to change as h changes?
- Numerically solve $\frac{\mathrm{d}^2y}{\mathrm{d}t^2} = -y$ with initial conditions $y(0) = 0$ and $y'(0) = 1$ using the explicit-midpoint and implicit-midpoint methods and a stepsize of $h = 0.1$ (only for two steps). How do the errors compare after 1 step or after 2 steps? Which methods give a solution with constant energy? Hint: $2E = y^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2$