Ordinary Differential Equations (I)

Caroline Colijn

C.Colijn@bristol.ac.uk

Strategies for deriving schemes (I)

- Basic idea: derive formula for $x(t+h)$ in terms of x at earlier times, e.g. $x(t)$
- One possibility: integrate

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

to give

$$x(t+h) \simeq x(t) + \int_t^{t+h} f(x(\tilde{t}))\mathrm{d}\tilde{t}.$$

- Then, many approaches e.g.

$$x(t+h) \simeq x(t) + \frac{h}{6}\left[f(x(t)) + 4x(t+h/2) + f(x(t+h))\right].$$

(Simpson's rule: a multi-step formula)

Strategies for deriving schemes (II)

- Use difference formulae to approximate derivative
- For example:

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

goes to

$$\frac{x(t+h) - x(t)}{h} \simeq f(x(t))$$

- So

$$x(t+h) \simeq x(t) + hf(x(t))$$

(so called *Euler method*)

- Generally better (as in approximate integration) to use lots of small steps rather than one big one

Explicit Euler method

- Differential equation problem: find $x(t^*)$, $t^* > t_0$, when

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), t), \qquad x(t_0) = x_0.$$

- Notation:
  - h is time-step (assumed constant here)
  - $t_n = t_0 + nh$, suppose $t^* = t_0 + Nh$
  - $\hat{x}(t)$ is approx to $x(t)$
  - shorthand: $x_n = \hat{x}(t_n)$
- (Forward, Explicit) Euler method gives

$$x(t+h) \simeq x(t) + hf(x(t), t)$$

Explicit Euler Method

- Formulate this in steps:

$$\hat{x}(t+h) = \hat{x}(t) + hf(\hat{x}(t), t)$$

$$\hat{x}(t_n+h) = \hat{x}(t_n) + hf(\hat{x}(t_n), t_n)$$

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

- where $x_n = \hat{x}(t_n)$

Example (1998 Q7(c))

- Use Euler's method to solve the differential equation

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

to obtain an approximation to $x(0.3)$, using step size $h = 0.1$.

Other step sizes?

- N.B. an exact solution is known:

$$x(0.3) = 1.399359864 \qquad \text{exactly.}$$

- Error analysis for different step sizes

h	$\hat{x}(0.3)$	E
0.003	1.398155460	.001204404
0.0003	1.399239185	.000120679
0.00003	1.399347793	.000012071
0.000003	1.399358656	.000001208

- Smaller step size gives more accurate answer, but then integrating up to a given point (e.g. $t = 0.3$) takes more computational work. Global error is $O(h)$

Error analysis of Euler method

- Autonomous ODE $\mathrm{d}x/\mathrm{d}t = f(x)$ for simplicity
- One step of the method, assuming $x(t_n)$ is known exactly:

$$\hat{x}(t_n + h) = x(t_n) + hf(x_n),$$

and also, by Taylor's Theorem

$$x(t_n + h) = x(t_n) + hx'(t_n) + \frac{1}{2!}h^2 x''(\xi)$$

Subtracting:

$$E = \hat{x}(t_n + h) - x(t_n + h) = h \underbrace{\left[f(x_n) - x'(t_n)\right]}_{0,\text{ by ODE}} - \frac{1}{2!}h^2 x''(\xi)$$

- Euler method makes $O(h^2)$ error in a single step
  - i.e., "second order" local truncation error

Local and global error

- *Local truncation error* of a numerical ODE solver is error committed in a single step:

$$E = \hat{x}(t_n + h) - x(t_n)$$

  - Euler method has $O(h^2)$ "second order" local truncation error

- *Global truncation error* is the error committed in integrating up to a certain fixed point in time, possibly taking many steps
  - to integrate from t_0 to t^* takes $\frac{t^* - t_0}{h}$ steps
  - Euler method has $\frac{t^* - t_0}{h} O(h^2) = O(h)$ global error, we say it is a *first order method*

- General principle: global order one less than local

Instability and how to fix it

- Example: use the Euler method to solve the DE problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = 4x - x^3, \qquad x(0) = 1,$$

Instability

- ... is "qualitatively wrong" growing oscillation!
- Formal definition: a numerical ODE solver is *unstable* if it gives *growing solutions* when applied to the ODE

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda x, \qquad \lambda < 0.$$

- Note: exact solution is $c\mathrm{e}^{\lambda t}$
  - decays for large t if $\lambda < 0$

Stability analysis of the Euler method

- Apply Euler to

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda x, \qquad \lambda < 0.$$

What happens to numerical solutions?
(They should decay.)

$$\begin{aligned} x_{n+1} &= x_n + hf(x_n) \\ &= x_n + h\lambda x_n \\ &= (1 + h\lambda)x_n \end{aligned}$$

- So $x_n = (1 + h\lambda)^n x_0$, grows if $1 + h\lambda < -1$,
  - method unstable if $h > \dfrac{2}{|\lambda|}$

Solution of linear difference equations

- *First order:*

$$x_{n+1} = cx_n.$$

Solution is $x_n = c^n x_0$

- *Second order:*

$$ax_{n+2} + bx_{n+1} + cx_n = 0.$$

Substitute $x_n = c\,\mu^n$, yields

$$a\mu^2 + b\mu + c = 0 \qquad \text{to solve for } \mu$$

- Higher order problems similar
c.f. linear ordinary differential equations

Exercise

- Analyse the stability properties of the midpoint rule

$$\frac{x_{n+1} - x_{n-1}}{2h} = f(x_n).$$

- *Note:* this is a very simple example of a *multi-step* scheme
- Compare to trapezium rule for integration
- More next time (time permitting) or in posted notes

Implicit methods (I)

Don't want instability to occur. How to fix it?
Answer: two new methods

- 1. *backward Euler method*:

$$\frac{x_{n+1} - x_n}{h} = f(x_{n+1})$$

usually written

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

or (for non-autonomous ODEs)

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

but *CANNOT* usually isolate x_{n+1}

Implicit methods (II)

- 2. *Trapezoidal method*:

$$\frac{x_{n+1} - x_n}{h} = \frac{1}{2} f(x_n) + \frac{1}{2} f(x_{n+1})$$

usually written

$$x_{n+1} = x_n + \frac{h}{2} f(x_n) + \frac{h}{2} f(x_{n+1})$$

or (for non-autonomous ODEs)

$$x_{n+1} = x_n + \frac{h}{2} f(x_n, t_n) + \frac{h}{2} f(x_{n+1}, t_{n+1})$$

but *CANNOT* usually isolate x_{n+1}

Stability of implicit methods

Take $f(x) = \lambda x$ so x_{n+1} can be isolated.

- *Backward Euler* gives

$$x_{n+1} = \left(\frac{1}{1 - \lambda h} \right) x_n.$$

- *Trapezoidal* gives

$$x_{n+1} = \left(\frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}} \right) x_n$$

Both methods: $\lambda < 0$ gives multiplier $\in (-1, +1)$ for any h

This known as *unconditional stability*

Accuracy...

Quite hard to analyse, as the one step solution of the scheme itself needs to be written as a Taylor series!

Upshot:

- *Backward Euler* method
  - local error $O(h^2)$
  - global error $O(h)$
  - i.e. is a *first order* scheme

- *Trapezoidal* method
  - local error $O(h^3)$
  - global error $O(h^2)$
  - i.e. is a *second order* scheme