Classification of o.d.e.s; aspects of relevance to numerics; not a course on o.d.e.s

1. Order	1st,2nd... highest derivative
2. Homogeneous/non	no forcing on r.h.s
3. Autonomous/non	independent variable not in coefficient xdx/dt not x/t dx/dt
4. Linear/non	$x, \dot{x}, \ddot{x}$... not $x\,\ddot{x}$, not $\sqrt{x^2}$
5. IVP/BVP	x(0), $\ddot{x}$(0)... ; x(t) y(0), y(L) ; y(x)

comments:

1. order = highest number of differentiations: $\dddot{y} = f(y, \ddot{y}, \dot{y}; t)$ is 3rd order nonautonomous. Or can write as set of 3 first order o.d.e.s. This is how R-K works.

$$y_1 = \frac{dy}{dt}$$
$$y_2 = \frac{dy_1}{dt} = \frac{dy^2}{dt^2}$$
$$\frac{dy_2}{dt} = \frac{dy^3}{dt^3} = f(y, y_1, y_2; t)$$

In vector form $$\frac{d}{dt}\begin{pmatrix} y \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ f \end{pmatrix}$$

For computing, vectors=array

$$d_t\boldsymbol{y} = \boldsymbol{F};\ \ \boldsymbol{y} = (y, y_1, y_2),\ \ \boldsymbol{F} = (y_1, y_2, f)$$

Note: number of integration constants = order. E.g., $y(0), \dot{y}(0), \ddot{y}(0)$

Or, more generally 3 pieces of data. E.g., $y(0), y(2), \int_0^2 y\,dx$

2. non-homogeneous, linear: homogeneous plus particular solution

$$\ddot{x} + x = \sin(\omega t)$$
$$x = \underbrace{A\sin t + B\cos t}_{homogeneous} + \frac{sin\omega t}{1-\omega^2}$$

note: variable coeff. can’t be solved except in special cases, like Bessel’s eqn.

4. Non-linear. $\ddot{x} + x^2 + x^3\dot{x}\ldots$ Can't add particular plus homogeneous.

E.g. $\dot{x} + x^2 = 1,\ x(0) = 0$

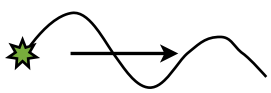
Particular: $x = \pm 1$, Homogeneous: $x = 1/(t+a)$.

But solution $\neq \dfrac{1}{t+1} - 1$

Correct solution

$$x = \frac{e^{2t} - 1}{e^{2t} + 1}$$

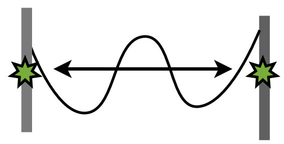
5. IVP



data t

damped
oscillator

BVP



0 x L

standing wave
hot / cold

Numerics: AB, R-K (explicit) IVP + shooting for BVP. Or matrix method for BVP.

E.g.

$$\ddot{x} = -x$$

$$x = A\sin(t) + B\cos(t)$$

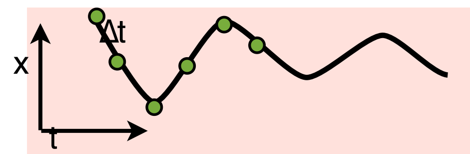
IVP: $x(0) = 0,\ \dot{x}(0) = 1$: $x = \sin(t)$

BVP: $x(0) = 0,\ x(1) = 1$: $x = \sin(t)/\sin(1)$

Numerical integration of o.d.e.s

Recall reduction to system of N (could be large) 1st order d.e.s

Integration provides basic perspective for o.d.e.s



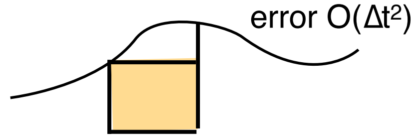
A. Integral over an interval (0,T):

1. Break (0,T) into (0,Δt),(Δt,2Δt)...(T-Δt,T)
2. Integrate over each interval by same method
3. For $\dot{x} = f(x,t)$; $x(t) - x(0) = \int_0^T f(x)\, dt$

$$x(T) - x(0) = \int_0^T f(x,t)dt = \sum \int_{n\Delta t}^{n+1\Delta t} f(x,t)dt$$

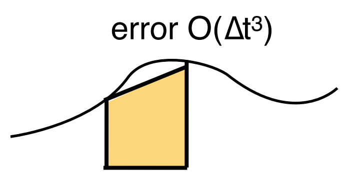
but x is fcn of t. How to evaluate integral?

4. Rectangular rule (motivates Euler explicit)



$$x(t + \Delta t) = x(t) + f(x(t),t)\Delta t$$

5. Trapezoidal, or midpoint, rule: motivates Runge-Kutta and Adams Bashforth



$$x(t + \Delta t) = x(t) + f\left(x(t + \tfrac{1}{2}\Delta t), t + \tfrac{1}{2}\Delta t\right)\Delta t$$

but how?

B. Euler explicit (t = nΔt)

$$x_{n+1} = x_n + \int_0^{\Delta t} f dt'' \sim \; x_{n+1} = x_n + f(x_n,t)\Delta t$$

1. Example: $\dot{x} = \alpha x$

$$f = \alpha x, \quad \alpha = \alpha_r + i\alpha_i, \; \text{Let } x_0 = A \text{ be i.c. :}$$

Exact: $f = Ae^{\alpha t}$

Numerical $x_{n+1} = x_n + (\alpha x_n)\,\Delta t = (1+\alpha\Delta t)\,x_n$.

$$
\begin{aligned}
x_0 &= A \\
x_1 &= (1+\alpha\Delta t)x_0 = (1+\alpha\Delta t)A \\
x_2 &= (1+\alpha\Delta t)x_1 = (1+\alpha\Delta t)^2 A \\
x_n &= (1+\alpha\Delta t)^n A
\end{aligned}
$$

Example $\alpha_i = 0,\ \alpha_r\Delta t = -0.1,\ A = 1$

$$x_1 = 0.90 \text{ exact } 0.904$$

$$x_2 = 0.81 \text{ exact } 0.819$$

etc.

Error over 1 interval $Error = Ae^{\alpha\Delta t} - A(1+\alpha\Delta t) \approx \tfrac{1}{2} A(\alpha\Delta t)^2$

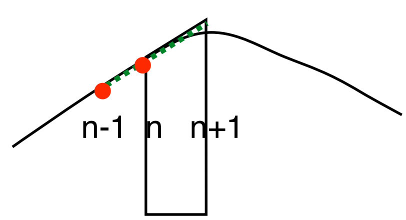
First order accurate: error per time step $\tfrac{1}{2}\,(\alpha\Delta t)^2 = 0.005$

How can we do a 2nd order integration?

C. Adams-Bashforth (A-B2: 2nd order, explicit)
(def'n of *explicit*: uses present and past data, n,n-1,n-2...)

Trapezoidal rule applied to $\dot{x} = f(x,t)$, extrapolating f from previous time-step

Explicit because we know n and n-1 data.



$$x_{n+1} = x_n + \int_0^{\Delta t} f(t')dt'$$

$$f(t') = f_n + \frac{f_n - f_{n-1}}{\Delta t}t' + O(\Delta t^2); -\Delta t < t' < \Delta t$$

$$
\begin{aligned}
x_{n+1} &= x_n + \int_0^{\Delta t} f_n + \frac{f_n - f_{n-1}}{\Delta t}t'dt' + O(\Delta t)^3 \\
&= x_n + f_n\Delta t + \frac{1}{2}(f_n - f_{n-1})\Delta t
\end{aligned}
$$

A-B2 $$\boxed{x_{n+1} = x_n + \left(\tfrac{3}{2} f_n - \tfrac{1}{2} f_{n-1}\right) \Delta t}$$

- Start up: given x_0, use Euler explicit for one step, then A-B
- Multi-level methods, A-B3... (explicit), A-M (implicit)
- Explicit is simple because it uses existing data and marches:

```
t=0 ; x(0)=0 ; Δt=0.01
x(1) = x(0) + f(x(0),0)Δt      ! Euler, startup
DO n=1,N
  x(n+1) = x(n) + [ 1.5*f(x(n)) - .5*f(x(n-1)) ]*Δt   ! A-B2 time advancement
  t = t + Δt
ENDDO
```

E.g.

$\dot{x} = 2x \quad x_0=1; \;\; \Delta t = 0.01$

Euler startup : $x_1 = 1.02$ exact: 1.0202

AB-2 : $x_2 = 1.02+[1.5\times(2\times1.02)-0.5\times(2\times1)] \times 0.01=1.0406$ exact: 1.0408

D. Runge-Kutta (very popular explicit (marching) method)

- RK1 = Euler explicit
- RK2 = midpoint rule, or predictor-corrector
- RK3,4.. = multi-point, or iterated predictor-corrector
- Multiple RHS evaluations

1. Predictor-corrector, RK2

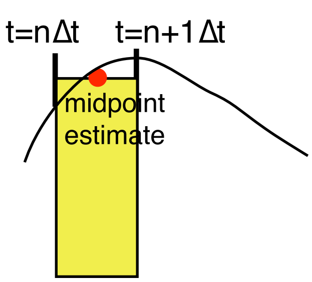
Mid-point predictor:

$$\tilde{x}_{n+1/2} = x_n + {}^{1}\!/_{2}\, f_n \Delta t$$

(or $\tilde{x}_{n+1} = x_n + \Delta t f_n$, $\tilde{x}_{n+1/2} = (x_n + x_{n+1})/2$)

Corrector:

$$x_{n+1} = x_n + \Delta t f(\tilde{x}_{n+1/2}) + O(\Delta t^3)$$



`Pseudo-code' for $\dot{x} = F(x,t)$

```
x(:)=x0(:) ;  t = 0 ; Δt=0.01
FOR n=1,N
!- midpoint estimate
   xdot(:)=F(x,t)
   xhalf(:)=x(:)+xdot(:)*Δt/2
!- fullstep
   xdot(:)=F(xhalf,t+Δt/2)
   x(:)=x(:)+xdot(:)*Δt
END

Subroutine F(x,t)           ẍ  = G(x,ẋ,t)
XP(1) = X(2)                ẋ1 = x2
XP(2) = G(X(:),t)           ẋ2 = G(x1,x2,t)
Return XP
```

Proof: algorithm is

$x_1 = x_0 + \Delta t\, f(x_0 + ½\Delta t\, f_n) = x_0 + \Delta t\, f_n + ½\Delta t^2\, f_n\, f_n' + O(\Delta t^3)$

Using $\dot{x} = f$; by chain rule, $\ddot{x} = f'\,\dot{x} = ff'$. So Taylor series gives RK2:

$x_1 = x_0 + \Delta t\, \dot{x}_0 + ½\Delta t^2\, \ddot{x}_0 + O(\Delta t^3) = = x_0 + \Delta t\, f_n + ½\Delta t^2\, f_n\, f_n' + O(\Delta t^3)$

I.e., RK2 is Taylor series to second order.

Or, conversely, via Taylor series

$$\dot{x} = f(x),$$

$$x(t) = x(0) + \dot{x}(0)t + {}^1\!/\!_2\,\ddot{x}(0)t^2 + O(t^3)$$

$$= x(0) + f[x(0)]t + {}^1\!/\!_2\,\dot{f}[x(0)]t^2 + O(t^3)$$

$$\dot{f} = f'\dot{x} = f'f$$

$$x(t) = x_0 + f_0 t + {}^1\!/\!_2\, f_o' f_0 t^2$$

$$= x_0 + (f_0 + {}^1\!/\!_2\, f_o' f_0 t)t$$

Rearrange $$x(t) = x_0 + f(x_0 + {}^1\!/\!_2\, f_0 t)t + O(t^3)$$

i.e., we know it is second order by its Taylor-series (proof of RK in general)

Multiple RHS evaluations: 4th order R-K requires 4 RHS evaluations

```
Δt    = tend-t
tmid  = t+Δt/2.
CALL RHS(n,t,y,yp)
   y1 = y+ Δt*yp/2.
CALL RHS(n,tmid,y1,y1p)
   y2 = y+ Δt*y1p/2.
CALL RHS(n,tmid,y2,y2p)
   y3 = y+Δt*y2p
CALL RHS(n,tend,y3,y3p)
   y = y+Δt*(y3p+2.*y2p+2.*y1p+yp)/6.
t = t + Δt
```