Initial value problems

Euler’s method, Runge-Kutta, and Runge-Kutta-Fehlberg

David A.W. Barton

University of Bristol

Initial value problems

Integration of differential equations over time starting from some *initial value*

$$\frac{dx}{dt} = f(t, x), \qquad x(0) = A$$

It gives $x(t)$ over a desired time interval, e.g., $0 \leq t \leq T$

Visualise as a time series or a phase portrait

Initial value problems – example

Van der Pol oscillator

Second Order

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0$$

Or in first-order form

Two State variables

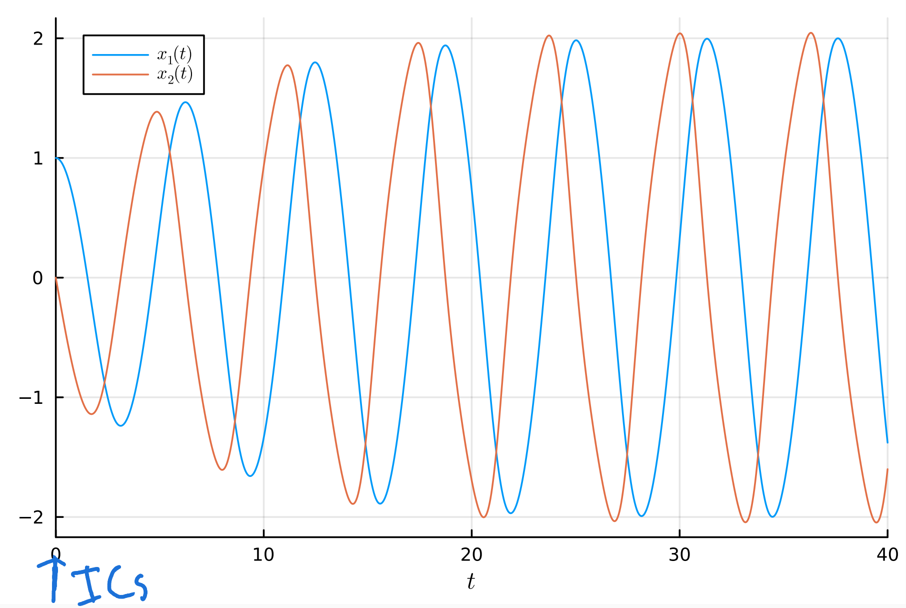
$$x_1' = x_2$$
$$x_2' = \mu(1 - x_1^2)x_2 - x_1$$

With $x_1(0) = 1$ and $x_2(0) = 0$

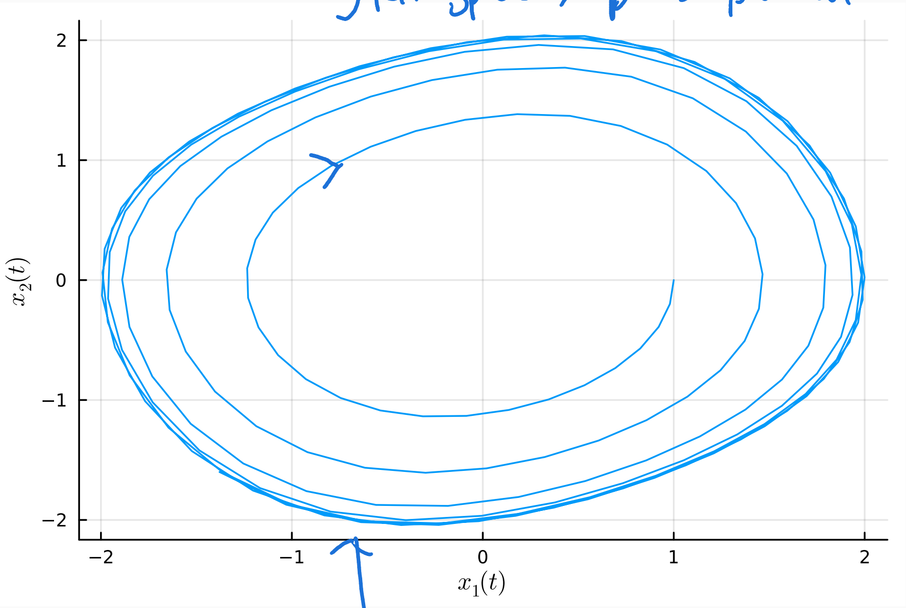
Van der Pol – time series and phase portrait

State space / phase portrait

Time Series



ICs



Limit cycle

Euler's method

Euler's method is simplest ODE integrator; for any ODE

$$\frac{dx}{dt} = f(t, x)$$

Fist Order in time

Not very accurate/robust

We have

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

x can be a vector and $f(t, x)$ a vector-valued function

Classic Runge-Kutta

Fouth Order

Most common version

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = f(t_n, x_n), \qquad k_2 = f\left(t_n + \frac{h}{2}, x_n + h\frac{k_1}{2}\right),$$

$$k_3 = f\left(t_n + \frac{h}{2}, x_n + h\frac{k_2}{2}\right),$$

$$k_4 = f(t_n + h, x_n + hk_3)$$

Butcher Tableau

Explicit Runge-Kutta methods are given by their Butcher Tableau

$$x_{n+1} = x_n + h \sum_{i=1}^{s} b_i \, k_i$$

where s is the number of stages and

$$k_1 = f(t_n, x_n)$$

$$k_2 = f\big(t_n + c_2 h, x_n + (a_{2,1} k_1) h\big)$$

$$k_3 = f\big(t_n + c_3 h, x_n + (a_{3,1} k_1 + a_{3,2} k_2) h\big)$$

$$\vdots$$

$$k_s = f\big(t_n + c_s h, x_n + (a_{s,1} k_1 + \cdots + a_{s,s-1} k_{s-1}) h\big)$$

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Runge-Kutta family of methods

Fourth Order

Euler's method

$$
\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}
$$

Classical Runge-Kutta

$$
\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array}
$$

Sum to 1

3/8 rule

$$
\begin{array}{c|cccc}
0 & & & & \\
1/3 & 1/3 & & & \\
2/3 & -1/3 & 1 & & \\
1 & 1 & -1 & 1 & \\
\hline
 & 1/8 & 3/8 & 3/8 & 1/8
\end{array}
$$

Writing out the 3/8 rule

Following the pattern of classic Runge-Kutta (RK4)

$$x_{n+1} = x_n + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4)$$

where

$$k_1 = f(t_n, x_n), \qquad k_2 = f\left(t_n + \frac{1}{3}, x_n + \frac{h}{3}k_1\right)$$

$$k_3 = f\left(t_n + \frac{2}{3}, x_n - \frac{h}{3}k_1 + hk_2\right)$$

$$k_4 = f(t_n + h, x_n + hk_1 - hk_2 + hk_3)$$

0				
$1/3$	$1/3$			
$2/3$	$-1/3$	1		
1	1	-1	1	
	$1/8$	$3/8$	$3/8$	$1/8$

Slightly better error properties than RK4

Checking the error

- Could run two separate integrations to determine accuracy
  - One with stepsize h
  - One with stepsize $^{h}/_{2}$

  This is very inefficient!

- Instead, use an extended Butcher Tableau
  - Two integration methods that use (almost) the same tableau
  - The higher-order method has an extra line in the tableau
  - Calculate the difference and adjust the stepsize accordingly

Runge-Kutta-Fehlberg (ode45)

c

0						
$1/4$	$1/4$					
$3/8$	$3/32$	$9/32$				
$12/13$	$1932/2197$	$-7200/2197$	$7296/2197$			
1	$439/216$	-8	$3680/513$	$-845/4104$		
$1/2$	$-8/27$	2	$-3544/2565$	$1859/4104$	$-11/40$	
4th order	$25/216$	0	$1408/2565$	$2197/4104$	$-1/5$	0
5th order	$16/135$	0	$6656/12825$	$28561/56430$	$-9/50$	$2/55$

b

Adaptive step sizes

→ 4th order solution

Calculate the difference between the base solution x and the higher accuracy solution x^*

$$\epsilon = |x - x^*|$$

Set the new step size to be

desired tolerance

↳ $10^{-6} \ldots 10^{-10}$

$$h_{\text{new}} = \beta h \left(\frac{\epsilon_{\text{tol}}}{\epsilon}\right)^{1/5}$$

where $\beta \approx 1$ is a safety parameter (often $\beta = 0.9$)

If the step size is decreased, the step just calculated should be performed again with the smaller step size

Summary

- Recapped Euler’s method and classic Runge-Kutta
- Introduced Butcher Tableau
- Shown that extended tableau can give efficient ways for estimating error
- Runge-Kutta-Fehlberg is an accurate adaptive step size method
  - Takes bigger steps when error is low
  - Takes smaller steps when error is high