Explicit RK methods:

put this common term in storage

$$y(t_n) \approx \xi_1 = y_n \qquad k_1 = f(t_n + c_1 h, \xi_1)$$

$$y(t_n + c_2 h) \approx \xi_2 = y_n + h a_{21} f(t_n + c_1 h, \xi_1) \qquad k_2 = f(t_n + c_2 h, \xi_2)$$

$$y(t_n + c_3 h) \approx \xi_3 = y_n + h a_{31} f(t_n + c_1 h, \xi_1) + h a_{32} f(t_n + c_2 h, \xi_2)$$

$$\vdots$$

$$y(t_n + c_\nu h) \approx \xi_\nu = y_n + h a_{\nu 1} f(t_n + c_1 h, \xi_1) + h a_{\nu 2} f(t_n + c_2 h, \xi_2) + \cdots + h a_{\nu \nu-1} f(t_n + c_{\nu-1} h, \xi_{\nu-1})$$

$$c_1 = 0, \quad c_2 = a_{21}, \quad c_3 = a_{31} + a_{32}, \quad c_\nu = \sum_{j=1}^{\nu-1} a_{\nu j}$$

$$y_{n+1} = y_n + h b_1 f(t_n + c_1 h, \xi_1) + \cdots + h b_\nu f(t_n + c_\nu h, \xi_\nu)$$

In general when writing code, define also

$$k_1 = f(t_n + c_1 h, \xi_1)$$

$$k_2 = f(t_n + c_2 h, \xi_2)$$

$$\vdots$$

$$k_\nu = f(t_n + c_\nu h, \xi_\nu)$$

Note after using ξ_i to compute k_i then ξ_i is never used again in the computation, so it doesn't need to be stored...

Then

$$y_{n+1} = y_n + h(b_1 k_1 + b_2 k_2 + \cdots + b_\nu k_\nu)$$

The next lab will explore this in practice...

Generalize, but first... tune the parameters, a_{ij}'s and b_j's to optimize something ... typically order ...

Explicit midpoint method

The (explicit) midpoint method is a

0	0	0
1/2	1/2	0
	0	1

Generic second-order method

0	0	0
α	α	0
	$1-\frac{1}{2\alpha}$	$\frac{1}{2\alpha}$

all 2nd order 2-stage methods...

c_1	a_{11}	a_{12}
c_2	a_{21}	a_{22}
	b_1	b_2

Explicit method

c_1	0	0
c_2	a_{21}	0
	b_1	b_2

Consistency

$c_i = \sum_j a_{ij}$

$b_1 + b_2 = 1$

so there are only 2 free parameters is a general 2-stage Explicit RK method

0	0	0
a_{21}	a_{21}	0
	$(1-b_2)$	b_2

for example a_{21} and b_2

Optimize the order to be 2nd order... we get

$$b_2 = \frac{1}{2a_{21}}$$

Now can tune for something else after order...

Kutta's third-order method

0	0	0	0
1/2	1/2	0	0
1	−1	2	0
	1/6	2/3	1/6

Heun's third-order method

0	0	0	0
1/3	1/3	0	0
2/3	0	2/3	0
	1/4	0	3/4

Textbook Nystrom's third order

0	0	0	0
2/3	2/3	0	0
2/3	0	2/3	0
	1/4	3/8	3/4

Classic fourth-order method

The "original" Runge–Kutta method.[3]

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	1/3	1/3	1/6

3/8-rule fourth-order method [

This method doesn't have as much notori
paper (Kutta, 1901).[3]

0	0	0	0	0
1/3	1/3	0	0	0
2/3	−1/3	1	0	0
1	1	−1	1	0
	1/8	3/8	3/8	1/8

Ralston's fourth-order method [edit]

This fourth order method[1] has minimum truncation error.

0	0	0	0	0
.4	.4	0	0	0
.45573725	.29697761	.15875964	0	0
1	.21810040	−3.05096516	3.83286476	0
	.17476028	−.55148066	1.20553560	.17118478

EXTRA CREDIT: are these decimal numbers exact? And under the assumption they are, is the method really 4th order?

Obvious generalization Implicit RK methods ...

for example these numbers might not be zero...

Before implicit ... Embedded methods ...

c_1	a_{11}	a_{12}	$\dots$	a_{1s}
c_2	a_{21}	a_{22}	$\dots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\dots$	a_{ss}
	b_1	b_2	$\dots$	b_s
	b_1^*	b_2^*	$\dots$	b_s^*

← all zero for explicit method ..

$$\begin{cases} y_{n+1} = y_n + h\left(b_1 k_1 + b_2 k_2 + \cdots + b_s k_s\right) \\ y^*_{n+1} = y_n + h\left(b^*_1 k_1 + b^*_2 k_2 + \cdots + b^*_s k_s\right) \end{cases}$$

Idea Compare y_{n+1} to y^*_{n+1} :

① If they are very close, then good!

② If not, the there is an estimate of the error...

This idea is used for controlling the error in automatic libraries built in to Julia and Matlab and in practical numeric packages ...

Implicit .. method ... midpoint Quadrature rule

$$y' = f(t,y)$$

$$\int_{t_n}^{t_{n+1}} y'(t)\,dt = \int_{t_n}^{t_{n+1}} f(t, y(t))\,dt$$

$$y(t_{n+1}) - y(t_n) \approx h\, f\left(t_n + \frac{h}{2},\ y\left(t_n + \frac{h}{2}\right)\right)$$

how to approximate this ...

Two ideas so far:

Interpolate: $y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, \frac{1}{2}(y_n + y_{n+1})\right)$

Use the ODE: $y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n)\right)$

Implicit RK method

$y(t_n + \frac{h}{2}) \approx \xi_1 = y_n + \frac{h}{2} f\left(t_n + \frac{h}{2}, \xi_1\right)$

Third Idea

$$y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, \xi_1\right)$$

RK Tableaux (2^{nd} order 1-stage implicit RK method)

$\frac{1}{2}$	$\frac{1}{2}$
	1

very small, but no zeros in the corner so implicit.

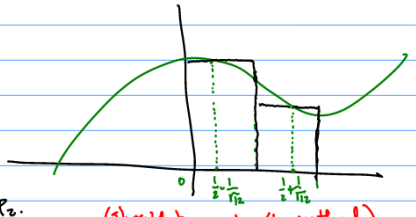
One more time for higher order...

$$\int_{t_n}^{t_{n+1}} y'(t)\,dt = \int_{t_n}^{t_{n+1}} f(t, y(t))\,dt$$

$$y(t_{n+1}) - y(t_n) \approx \frac{h}{2}\left(f\left(t_n + \left(\tfrac{1}{2} - \tfrac{1}{\sqrt{12}}\right)h, y\left(t_n + \left(\tfrac{1}{2} - \tfrac{1}{\sqrt{12}}\right)h\right)\right) + f\left(t_n + \left(\tfrac{1}{2} + \tfrac{1}{\sqrt{12}}\right)h, y\left(t_n + \left(\tfrac{1}{2} + \tfrac{1}{\sqrt{12}}\right)h\right)\right)\right)$$

from the lecture on Gauss quadrature...

$\int_0^1 f(x)\,dx \approx \frac{1}{2} f\left(\frac{1}{2} - \frac{1}{\sqrt{12}}\right) + \frac{1}{2} f\left(\frac{1}{2} + \frac{1}{\sqrt{12}}\right)$



$\nu = 2$ since we used the roots of P_2.

(should be order 4 method)

The RK Tableau for this is

$$\frac{\sqrt{3}}{6} = \frac{\sqrt{3}}{2\sqrt{3}\sqrt{3}} = \frac{1}{2\sqrt{3}} = \frac{1}{r_{12}}$$

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$y\left(t_n + \left(\tfrac{1}{2} - \tfrac{1}{r_{12}}\right)h\right) \approx \xi_1$$

$$y\left(t_n + \left(\tfrac{1}{2} + \tfrac{1}{r_{12}}\right)h\right) \approx \xi_2$$

So this comes directly from the quadrature formulas.