Ch. 3 Runge-Kutta Methods — continued

Key Points:
- Explicit Runge-Kutta Methods (ERK)
  - Taylor Expansion Methods
- Implicit Runge-Kutta Methods (IRK)

Collocation + IRK methods

Runge-Kutta Fehlberg

Runge-Kutta Methods

Recall that so far we have seen both single step methods (e.g. Euler, Backward Euler, Trapezoid, ...) where solving $\vec{y}' = \vec{f}(t, \vec{y})$ approximately required a knowledge of the solution $\vec{y}_k$ to get the next approximation $\vec{y}_{k+1}$.

Runge-Kutta Methods are popular single step methods. These methods, which can be designed with different orders, are the common means of obtaining the necessary ~~starting~~ values for multistep methods. Note that typically you want to use a method for obtaining these starting values that has the same order as the multistep scheme you plan to use.

Brief intro to Runge-Kutta Methods

- Single step (usually explicit but implicit versions too..)
- tend to require a lot of function evaluations: $f(t, y)$

Explicit Runge-Kutta Schemes

Recall our representation of $\vec{y}' = f(t,\vec{y})$ as a quadrature problem

$$\vec{y}(t_{n+1}) = \vec{y}(t_n) + \int_{t_n}^{t_{n+1}} \vec{f}(t,\vec{y}(t))\,dt$$

$t = t_n + h\tau$
$h = t_{n+1} - t_n$

$$\boxed{\vec{y}(t_{n+1}) = \vec{y}(t_n) + h\int_0^1 \vec{f}(t_n + h\tau, \vec{y}(t_n + h\tau))\,d\tau}$$

we could then apply some quadrature rule to the integral to get an approximation

$$\int_0^1 \vec{f}(t_n + h\tau, \vec{y}(t_n + h\tau))\,d\tau$$

$$\approx \sum_{j=1}^{\nu} w_i\, \vec{f}(t_n + h\tau_j, \vec{y}(t_n + h\tau_j))\,d\tau$$

where w_j are the weights
τ_j are the nodes (with respect to integration variable τ)
and ν is the number of nodes

Comment on notation: I'm currently dragging my feet and not coming around to Iserles choice of notation for weights (b_j) and nodes (c_j) although have given in to using ν as the number of nodes (formerly n).

Unlike a quadrature rule applied to

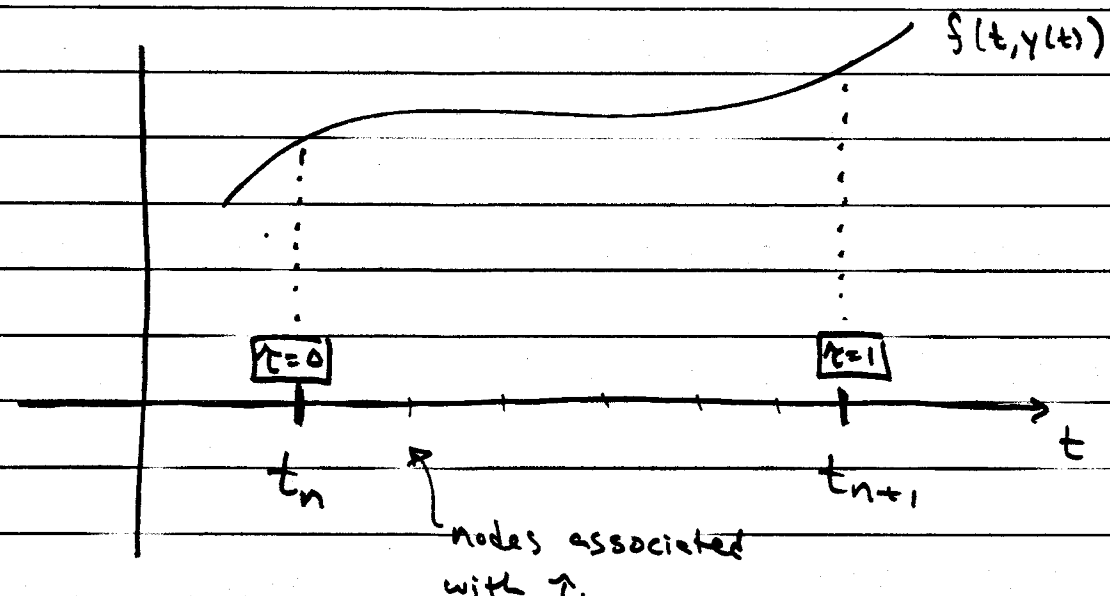
$$\int_0^1 f(\tau)\,d\tau$$

where ~~the function f~~ it is known exactly where to evaluate f, in the present context we have

$$\int_0^1 f(t_n+h\tau,\ \vec{y}(t_n+h\tau))\,d\tau$$

where in general we do not know $\vec{y}(t_n+h\tau)$ exactly. So, in our quadrature rule ~~approx~~ we need to come up with a way to approximate the quantities

$$\vec{\xi}_j = \vec{y}(t_n+h\tau_j) \qquad j=1,2,\dots,\nu$$



Recall: $0 \le \tau_1 < \tau_2 < \dots < \tau_{\nu-1} < \tau_\nu \le 1$

e.g. open Newton-Cotes (equally-spaced $\tau_j \in (0,1)$)
closed Newton-Cotes (" " $\tau_j \in [0,1]$)
$\tau_1 = 0, \ \tau_\nu = 1$

⋮

zeros of orthogonal polynomials

⋮

The idea is that we have an approximation to the solution $\vec{y}(t)$ at t_n ($\tau = 0$) ... $\vec{y}_n$ and we want to get an approximation at $\vec{y}_{n+1}$.

For explicit Runge-Kutta, we need an explicit way to generate the intermediate steps at the nodes τ_j leading us towards $\vec{y}_{n+1}$. ~~Focus on the "closed" case with~~

~~$0 < \tau_1 < \tau_2 < \tau_3 < \ldots < \tau_{\nu-1} < \tau_\nu$~~

plan... • find an explicit way to get from ~~[illegible]~~ $\tau_1 = 0$ to $\tau = \tau_2$

• then find an explicit way to get from τ_2 to τ_3 (using available previous information)

• etc.

Define

$$\vec{\xi}_1 = \vec{y}_n \qquad \approx \vec{y}(t_n)$$

$$\vec{\xi}_2 = \vec{y}_n + h\, a_{2,1} \vec{f}(t_n, \vec{\xi}_1) \qquad \approx \vec{y}(t_n + h\tau_2)$$

$$\vec{\xi}_3 = \vec{y}_n + h\, a_{3,1} \vec{f}(t_n, \vec{\xi}_1) + h\, a_{3,2} \vec{f}(t_n + h\tau_2, \vec{\xi}_2)$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \qquad \ddots$$

$$\vec{\xi}_\nu = \vec{y}_n + h \sum_{i=1}^{\nu-1} a_{\nu,i} \vec{f}(t_n + h\tau_i, \vec{\xi}_i)$$

where the $a_{j,i}$ are components of an RK Matrix A

$$A = \begin{bmatrix} 0 & & & & & \\ a_{2,1} & 0 & & & & \\ a_{3,1} & a_{3,2} & 0 & & & \\ a_{4,1} & a_{4,2} & a_{4,3} & \ddots & & \\ \vdots & & & \ddots & \ddots & \\ a_{\nu,1} & a_{\nu,2} & \cdots & & a_{\nu,\nu-1} & 0 \end{bmatrix}$$

is a strictly lower triangular matrix

(zeros fill in the 'missing' elements)

Then the explicit Runge-Kutta scheme (ERK) is

$$\boxed{\vec{y}_{n+1} = \vec{y}_n + h\sum_{j=1}^{\nu} w_j \vec{f}(t_n + \tau_j h, \vec{\xi}_j)}$$

where the values $\vec{\xi}_j$ ~~are~~ are the approximations to $\vec{y}(t_n + \tau_j h)$ that go with each node τ_j. So

$$\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_\nu \end{bmatrix} = \text{'RK weights'}, \quad \vec{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \vdots \\ \tau_\nu \end{bmatrix} = \text{"RK nodes"}$$

In this context ~~~~ we have ν-stage ERK.

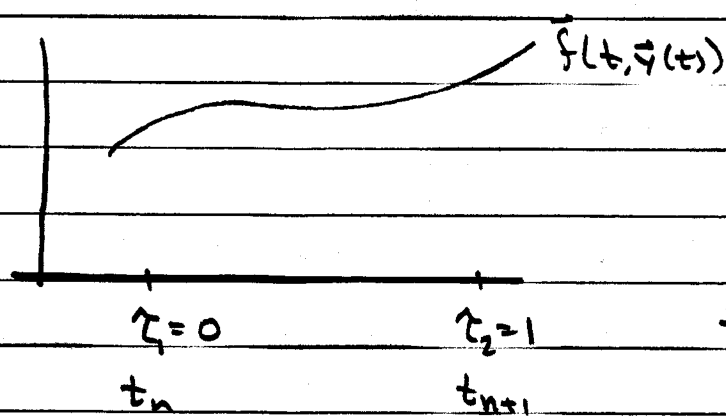
That is, for $\nu = 2$ we have 2-stage ERK

$\nu = 3$ " " 3-stage ERK

$\nu = 4$ " " 4-stage ERK.

Before pursuing the question of how to choose the RK Matrix A along with the weights $\vec{w}$ and nodes $\vec{\tau}$ let's consider an example.

EXAMPLE 1 (2-stage ERK scheme) ... one possible 2-stage ERK.



$t_{n+1} = t_n + h$

$$\vec{\xi}_1 = \vec{Y}_n$$

$$\vec{\xi}_2 = \vec{Y}_n + h\vec{f}(t_n, \vec{\xi}_1)$$

$$\boxed{\vec{Y}_{n+1} = \vec{Y}_n + \tfrac{1}{2}h\left[\vec{f}(t_n, \vec{\xi}_1) + \vec{f}(t_n + h, \vec{\xi}_2)\right]}$$

Also sometimes called "Modified Euler method"

i.e. Heun's Method (2nd order)

error ~ $O(h^3)$

Here

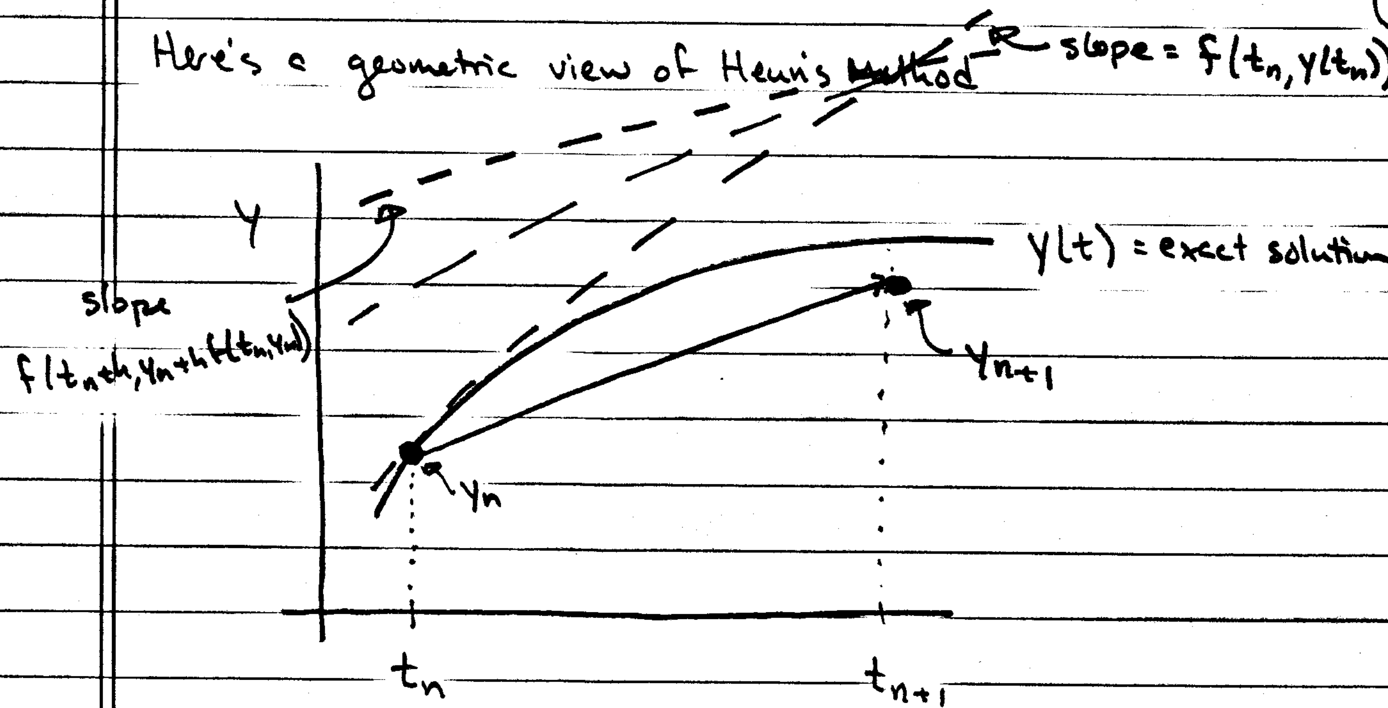
$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\vec{w} = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \qquad \vec{\tau} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

=> RK Tableaux $\begin{array}{c|c} \vec{\tau} & A \\ \hline & \vec{w}^T \end{array}$

$$= \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

Here's a geometric view of Heun's method



- first get the slope $f(t_n, y(t_n))$ — — — —
- second, get the slope $f(t_n+h, y_n+hf(t_n,y_n))$ — — — —
- third, average these two slopes : $\frac{1}{2}\left[f(t_n, \ldots) + f(t_n+h, \ldots) \right]$ — — —
- use the average slope to advance from t_n to t_{n+1}.

(see also Atkinson, p. 421)

EXAMPLE 2 (another 2-stage ERK)

In RK Tableaux form

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array} \qquad \vec{\tau} = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} \quad \vec{w} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 \\ 1/2 & 0 \end{bmatrix}$$

So

$$\vec{y}_{n+1} = \vec{y}_n + h\left\{ w_1 \vec{f}(t_n + \tau_1 h, \vec{\xi}_1) + w_2 \vec{f}(t_n + \tau_2 h, \vec{\xi}_2) \right\}$$

$$\vec{\xi}_1 = \vec{y}_n$$

$$\vec{\xi}_2 = \vec{y}_n + h a_{2,1} \vec{f}(t_n, \vec{\xi}_1)$$

Filling in details...

$$\boxed{\vec{y}_{n+1} = \vec{y}_n + h\, \vec{f}\left(t_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}h\, f(t_n, \vec{y}_n)\right)}$$

This is a midpoint rule with the value of $y(t_n + \frac{1}{2}h)$ approximated by an Euler step to the midpoint

That is, Heun's method is to trapezoid rule

as

This method is to midpoint rule

(see also example 2 in Atkinson, p. 420-421)

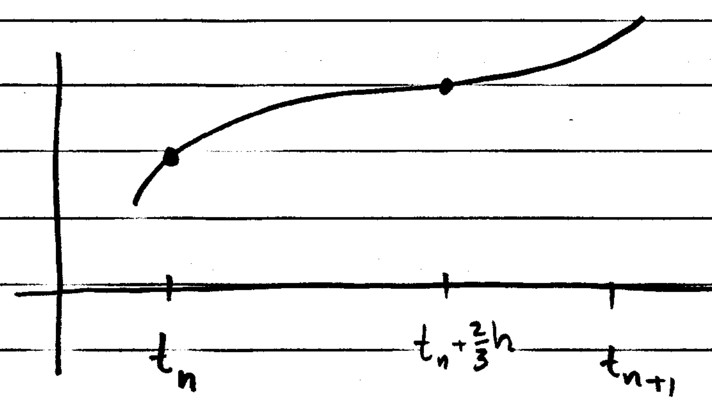
EXAMPLE 3 (another 2-stage ERK)

In RK Tableaux form

$$\begin{array}{c|cc} 0 & & \\ 2/3 & 2/3 & \\ \hline & 1/4 & 3/4 \end{array} \qquad \vec{w} = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} \quad \vec{\tau} = \begin{bmatrix} 0 \\ 2/3 \end{bmatrix}$$

Here

$$\vec{y}_{n+1} = \vec{y}_n + h\left\{ \frac{1}{4}\vec{f}(t_n, \vec{y}_n) + \frac{3}{4}\vec{f}\Big(t_n + \tfrac{2}{3}h,\ \underbrace{\vec{y}_n + \tfrac{2}{3}h f(t_n, \vec{y}_n)}_{\substack{\text{Forward} \\ \text{Euler step} \\ \text{out to} \\ t_n + \frac{2}{3}h}}\Big) \right\}$$



This is a weighted average of slope values (i.e. f values) at t_n and $t_n + \frac{2}{3}h$.

Comment on names/labels for these three examples

I called Example $\boxed{1}$, $\boxed{2}$, $\boxed{3}$

Heun's, Modified Midpoint, alternative weighting

Heath also calls [redacted] Example 1 – Heun's method.

However, Burden + Faires refer to (6th edition, p. 279-280)

EXAMPLE 1 as 'Modified Euler Method'

EXAMPLE 2 as 'Midpoint Method'

EXAMPLE 3 as 'Heun's Method'

To motivate these examples lets consider what are sometimes called

Taylor Series Methods (e.g. see Atkinson, p. 418-419)

Consider our initial value problem

$$\begin{cases} y' = f(t, y) \\ y(t_0) = y_0 \end{cases} \Rightarrow y(t) = \text{exact solution.}$$

A Taylor Expansion of $y(t_1)$ is about $y(t_0)$, e.g. is

$t_1 = t_0 + h$

$$y(t_1) = y(t_0) + h\, y'(t_0) + \tfrac{1}{2} h^2 y''(t_0) + \tfrac{1}{3!} h^3 y'''(t_0)$$

$$+ \dots + \frac{1}{k!} h^k y^{(k)}(t_0) + \frac{h^{k+1}}{(k+1)!} y^{(k+1)}(\theta)$$

For some $\theta \in [t_0, t_1]$.

Note:
$$\left.\begin{aligned} y'(t) &= f(t, y(t)) \\ y''(t) &= \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f \equiv g \end{aligned}\right\} \text{ see also HW (ch.1) exercise 1.5}$$

$$y'''(t) = \frac{\partial^2 f}{\partial t^2} + \frac{\partial^2 f}{\partial t \partial y} f + \left(\frac{\partial^2 f}{\partial t \partial y} + \frac{\partial^2 f}{\partial y^2} f \right) f$$

$$+ \frac{\partial f}{\partial y} \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f \right)$$

$$= \frac{\partial^2 f}{\partial t^2} + 2 \frac{\partial^2 f}{\partial t \partial y} f + \frac{\partial^2 f}{\partial y^2} f^2 + \frac{\partial f}{\partial y} \frac{\partial f}{\partial t} + \left(\frac{\partial f}{\partial y}\right)^2 f$$

$\vdots$

So for example, one could define the Taylor Scheme

$$(*) \quad \boxed{\vec{y}_{n+1} = \vec{y}_n + h \underbrace{\vec{f}(t_n, \vec{y}_n)}_{\approx \vec{y}'(t_n)} + \tfrac{1}{2}h^2 \underbrace{\vec{g}(t_n, \vec{y}_n)}_{\approx \vec{y}''(t_n)}}$$

This is a second order scheme (see Ch.1 HW Exercise 1.5)

Taylor Series Methods can be useful if the function $f(t,y)$ can be (easily) differentiated analytically. Often, however, this is not the case ... e.g. large systems of ODEs.

So in the scheme (*) we are using both derivative information (like Forward Euler) and second derivative information ($g = y''$) to move forward one time step.

Runge-Kutta ~~attemp~~ methods attempt to match this higher order approach via carefully chosen function evaluations (of the original function f) rather than higher derivative information at t_n.

In general, one can identify the order ~~[illegible]~~ of an ERK by examining

$$Q \equiv \vec{y}(t_{n+1}) - \left\{ y(t_n) + h \sum_{j=1}^{\nu} w_j \vec{f}\left(t_n + \tau_j h, \vec{\xi}_j(t_n + \tau_j h)\right) \right\}$$

where $\vec{\xi}_j(t_n + \tau_j h)$ is interpretted as the previously-defined $\vec{\xi}_j$ expressions evaluated at the exact solution.

In principle, ~~we~~ the plants to make good choices for w_j, τ_j and a_{ji} (RK weights, RK nodes, RK Matrix) for a choice of ν.

Comments

- We'll work through the details of this for $\nu = 2$ (2-stage ERK)
- We'll note some 3-stage and 4-stage ERK methods
- Elements of Graph Theory can (and should, by Iserles recommendation) be used to derive higher order ERK methods (e.g. beyond order 4).
- Note ν-stage ERK methods of order ν exist only for $\nu \leq 4$ ... so the cost rises considerably for these higher order methods. (see Iserles, p. 41).

So for the 2-stage ERK case $(\nu = 2)$ examine

$$Q \equiv \vec{y}(t_{n+1}) - \left\{ \vec{y}(t_n) + h w_1 \vec{f}(t_n, \underbrace{\vec{y}_n}_{\vec{\xi}_1}) + h w_2 \vec{f}(t_n + \tau_2 h, \underbrace{\vec{y}_n + a_{2,1} h \vec{f}(t_n, \vec{y}_n)}_{\vec{\xi}_2}) \right\}$$

Note: we are assuming from the start that $\tau_1 = 0$ so that we make use of the information already available at t_n ... i.e. we know $\vec{y}_n$.

- the goal here will be to determine how to optimally place the other node (τ_2), how to weight the two 'slope' values (w_1, w_2) and how to select the RK Matrix component $a_{2,1}$.

Our first task is to Taylor Expand – assuming sufficient smoothness — the expression

$$\vec{f}(t_n + \tau_2 h, \vec{y}_n + a_{2,1} h \vec{f}(t_n, \vec{y}_n))$$

$$\approx \vec{f}(t_n, \vec{y}_n) + \tau_2 h \frac{\partial \vec{f}}{\partial t}(t_n, \vec{y}_n) + a_{2,1} h \vec{f}(t_n, \vec{y}_n) \cdot \frac{\partial \vec{f}}{\partial \vec{y}}(t_n, \vec{y}_n) + O(h^2)$$

Now,

$$Q = \cancel{\vec{y}(t_n)} + h\underbrace{\vec{y}'(t_n)}_{=f} + \tfrac{1}{2}h^2 \underbrace{\vec{y}''(t_n)}_{=\frac{\partial f}{\partial t}+\frac{\partial f}{\partial y}\cdot f} + O(h^3)$$

$$- \Bigg\{ \cancel{\vec{y}(t_n)} + hw_1 \vec{f}(t_n,\vec{y}_n)$$

$$+ hw_2 \left[\vec{f}(t_n,\vec{y}_n) + \tau_2 h \frac{\partial \vec{f}}{\partial t}(t_n,\vec{y}_n) + a_{2,1} h\, \vec{f}(t_n,\vec{y}_n) \frac{\partial \vec{f}}{\partial \vec{y}}(t_n,\vec{y}_n) + O(h^2) \right] \Bigg\}$$

$$= h\left\{ \vec{f}(t_n,\vec{y}) \left[1 - w_1 - w_2 \right] \right\}$$

$$+ h^2 \left\{ \frac{1}{2}\left(\frac{\partial \vec{f}}{\partial t}(t_n,\vec{y}_n) + \frac{\partial \vec{f}}{\partial \vec{y}}(t_n,\vec{y}_n)\, \vec{f}(t_n,\vec{y}_n) \right) - \tau_2 w_2 \frac{\partial \vec{f}}{\partial t}(t_n,\vec{y}_n) - a_{2,1} w_2\, \vec{f}(t_n,\vec{y}_n) \frac{\partial \vec{f}}{\partial \vec{y}}(t_n,\vec{y}_n) \right\}$$

$$+ O(h^3)$$

So, examining these terms, we require

$\boxed{w_1 + w_2 = 1}$ eliminates $O(h)$ term

$\boxed{\tau_2 w_2 = \frac{1}{2}}$ eliminates $\frac{\partial \vec{f}}{\partial t}$ at $O(h^2)$

$\boxed{a_{2,1} w_2 = \frac{1}{2}}$ eliminates $\frac{\partial \vec{f}}{\partial \vec{y}} \cdot \vec{f}$ at $O(h^2)$

i.e. $a_{2,1} = \tau_2$

Note:

- we should really make sure we know more about the $O(h^3)$ term.
  - Iserles notes that applying the method to the scalar problem $y' = y$ $(f = y)$ easily shows the $O(h^3)$ term remains.

- our three requirements for 4 variables

  $w_1, w_2, \tau_2, a_{2,1}$

  allow for a family of 2-stage 2nd order ERK methods.

  ↳ Recall our 3 examples ... also shown on p. 39 of Iserles
  - Heun
  - modified midpoint
  - alternate weighting

What about higher order?

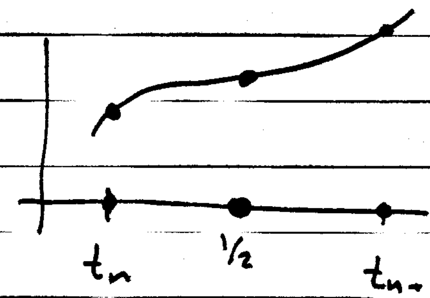
EXAMPLE 4+5

3-stage, 3rd order ERK Methods

$$\begin{array}{c|c} \vec{c} & A \\ \hline & \vec{w}^T \end{array}$$

Classic RK

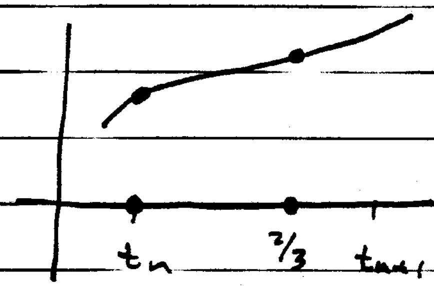
$$\begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$



-

Nystrom

$$\begin{array}{c|ccc} 0 & & & \\ 2/3 & 2/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 3/8 & 3/8 \end{array}$$



Think about how many function evaluations are required.

'Classic RK'

$$y_{n+1} = y_n + h\left[\frac{1}{6}\vec{k}_1 + \frac{2}{3}\vec{k}_2 + \frac{1}{6}\vec{k}_3\right]$$

where

	function evals
$\vec{k}_1 = \vec{f}(t_n, \vec{y}_n)$	1
$\vec{k}_2 = \vec{f}(t_n + \frac{1}{2}h, \vec{y}_n + \frac{1}{2}h\vec{k}_1)$	1
$\vec{k}_3 = \vec{f}(t_n + h, \vec{y}_n - h\vec{k}_1 + 2h\vec{k}_2)$	1
	3

Nystrom

$$y_{n+1} = y_n + h\left[\frac{1}{4}\vec{k}_1 + \frac{3}{8}\vec{k}_2 + \frac{3}{8}\vec{k}_3\right]$$

where

	function evals
$\vec{k}_1 = \vec{f}(t_n, \vec{y}_n)$	1
$\vec{k}_2 = \vec{f}(t_n + \frac{2}{3}h, \vec{y}_n + \frac{2}{3}h\vec{k}_1)$	1
$\vec{k}_3 = \vec{f}(t_n + \frac{2}{3}h, \vec{y}_n + \frac{2}{3}h\vec{k}_2)$	1
	3

EXAMPLE 6

Another 3-stage 3rd order ERK

$$\begin{array}{c|ccc} 0 & & & \\ 1 & 1 & & \\ 1/2 & 1/4 & 1/4 & \\ \hline & 1/6 & 1/6 & 2/3 \end{array} \qquad \begin{array}{c|c} \vec{\tau} & A \\ \hline & \vec{w}^T \end{array}$$

That is,

$$\vec{y}_{n+1} = \vec{y}_n + h\left[\frac{1}{6}\vec{k}_1 + \frac{1}{6}\vec{k}_2 + \frac{2}{3}\vec{k}_3\right]$$

← Simpson-like method (used as part of ode23)

$$\vec{k}_1 = \vec{f}(t_n, \vec{y}_n)$$

$$\vec{k}_2 = \vec{f}(t_n + h, \vec{y}_n + h\vec{k}_1)$$

3 function evaluations

$$\vec{k}_3 = \vec{f}\left(t_n + \tfrac{1}{2}h, \vec{y}_n + \tfrac{1}{2}h\left(\frac{\vec{k}_1 + \vec{k}_2}{2}\right)\right)$$

Note that these values ~~[illegible]~~ of $w_1, w_2, w_3, \tau_1, \tau_2, \ldots$ etc. satisfy the conditions derived by Iserles

$w_1 + w_2 + w_3 = 1$ ✓

$w_2\tau_2 + w_3\tau_3 = 1/2$ ✓ $\qquad \sum_{i=1}^{j-1} a_{j,i} = \tau_j \quad j = 2,3$ ✓

$w_2\tau_2^2 + w_3\tau_3^2 = 1/3$ ✓

$w_3\, a_{3,2}\, \tau_2 = 1/6$ ✓

EXAMPLE 7 (3-stage, 3rd order ERK)

What about

$$\vec{y}_{n+1} = \vec{y}_n + h\left[\frac{1}{6}\vec{k}_1 + \frac{2}{3}\vec{k}_2 + \frac{1}{6}\vec{k}_3\right]$$

where

$$\vec{k}_1 = f(t_n, y_n)$$

$$\vec{k}_2 = f(t_n + \tfrac{1}{2}h,\ \vec{y}_n + \tfrac{1}{2}h\vec{k}_1)$$

$$\vec{k}_3 = f(t_n + h,\ \vec{y}_n + h\vec{k}_2)$$

This is another ~~[illegible]~~ variation of simpson's rule.

3 function evaluations

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

$w_1 + w_2 + w_3 = 1$ ✓ $\qquad \frac{1}{6} + \frac{2}{3} + \frac{1}{6} = 1$

$w_2 \tau_2 + w_3 \tau_3 = 1/2$ ✓ $\qquad \frac{2}{3}\cdot\frac{1}{2} + \frac{1}{6}\cdot 1 = 1/2$

$w_2 \tau_2^2 + w_3 \tau_3^2 = 1/3$ ✓ $\qquad \frac{2}{3}\left(\frac{1}{2}\right)^2 + \frac{1}{6}\cdot 1^2 = 1/3$

$w_3 a_{3,2} \tau_2 = \frac{1}{6}$ $\qquad \frac{1}{6}\cdot 1\cdot 1 = 1/6$

$\sum_{i=1}^{j-1} a_{j,i} = \tau_j \quad j = 2,3$ ✓

EXAMPLE 8 (4-stage 4^{th} order ERK)

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

i.e. local error [redacted]) $\sim O(h^5)$

One way to express this method is

$$\vec{y}_{n+1} = \vec{y}_n + \frac{h}{6}\left[\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4\right]$$

where

	function evals
$\vec{k}_1 = \vec{f}(t_n, \vec{y}_n)$	1
$\vec{k}_2 = \vec{f}(t_n + \frac{1}{2}h, \vec{y}_n + \frac{1}{2}h\vec{k}_1)$	1
$\vec{k}_3 = \vec{f}(t_n + \frac{1}{2}h, \vec{y}_n + \frac{1}{2}h\vec{k}_2)$	1
$\vec{k}_4 = \vec{f}(t_n + h, \vec{y}_n + h\vec{k}_3)$	1
	4

Interpretation in terms of Simpson's Rule

Recall that Simpson's ~~rule for~~ quadrature rule is

$$\int_a^b f(t)\,dt = (b-a)\left[\frac{1}{6}f(a) + \frac{2}{3}f\left(\frac{a+b}{2}\right) + \frac{1}{6}f(b)\right]$$

The 3-stage 3rd order methods
.. 'classic RK' + at least two others (EX 4, 6, 7)
use this basic form with functions evaluated
in various ways.

The 4-stage 4th order method also has a (EX 8)
similar form but splits the $\frac{2}{3}f\left(\frac{a+b}{2}\right)$ into
an average over two function evaluations.

Runge-Kutta Fehlberg Methods

- Atkinson, p. 429
- Ascher + Petzold, p. 90-91
- Sauer
- Iserles, Ch. 6, p. 113

- variable-step size RK methods
- error control
  - truncation error computed by comparing the result of a lower order RK method to a higher order one (see Atkinson, p. 429)
    (due to E. Fehlberg (1970) Computing 6, 61-71)

EXAMPLE (Matlab's 'ode23')

- runs 2-stage 2nd order ERK in parallel with 3-stage 3rd order ERK

$$z_{n+1} = z_n + \cancel{\tfrac{1}{2}}h\left[\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2\right] \qquad \text{(Heun) order 2}$$

$$y_{n+1} = y_n + h\left[\tfrac{1}{6}k_1 + \tfrac{1}{6}k_2 + \tfrac{2}{3}k_3\right] \qquad \text{Simpson-like (see Ex 6)}$$

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

$$k_2 = f(t_n + h, y_n + hk_1)$$

$$k_3 = f\left(t_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}h\left(\tfrac{k_1+k_2}{2}\right)\right)$$

Comments:

- uses 3 function evaluations to do both 2nd + 3rd order methods
- $\|z_{n+1} - y_{n+1}\|$ is an error approx that can be used to adjust stepsize — adaptive.

Implicit Runge-Kutta Schemes (IRK)

Recap: (ERK)

$$\begin{cases} \vec{y}_{n+1} = \vec{y}_n + h \displaystyle\sum_{j=1}^{\nu} w_j \vec{f}(t_n + \tau_j h, \vec{\xi}_j) \\ \text{~~where~~} \\ \vec{\xi}_j \equiv y_n + h \underbrace{\displaystyle\sum_{i=1}^{j-1} a_{j,i} \vec{f}(t_n + \tau_i h, \vec{\xi}_i)}_{} \qquad j = 1, 2, \ldots, \nu \end{cases}$$

ξ_j depends on previously determined values.

In implicit RK ~~there is one minor change to~~ we have

$$\begin{cases} \vec{y}_{n+1} = \vec{y}_n + h \displaystyle\sum_{j=1}^{\nu} w_j \vec{f}(t_n + \tau_j h, \vec{\xi}_j) \qquad \text{(same)} \\ \text{but now} \\ \vec{\xi}_j = \vec{y}_n + h \underbrace{\displaystyle\sum_{i=1}^{\nu} a_{j,i} \vec{f}(t_n + \tau_i h, \vec{\xi}_i)}_{} \qquad j = 1, 2, \ldots, \nu \end{cases}$$

ξ_j depends on all other ξ_i values

In IRK the RK Matrix A is a 'full' matrix whereas in ERK A was strictly lower triangular.

EXAMPLE 9 (2-stage, 3rd order IRK)

$$\begin{cases} \vec{y}_{n+1} = \vec{y}_n + \frac{1}{4}h\left[\vec{f}(t_n, \vec{\xi}_1) + 3\vec{f}(t_n + \frac{2}{3}, \vec{\xi}_2)\right] \\ \text{where } \vec{\xi}_1 \text{ and } \vec{\xi}_2 \text{ are determined by} \\ \vec{\xi}_1 = \vec{y}_n + \frac{1}{4}h\left[\vec{f}(t_n, \vec{\xi}_1) - \vec{f}(t_n + \frac{2}{3}h, \vec{\xi}_2)\right] \\ \vec{\xi}_2 = \vec{y}_n + \frac{1}{12}h\left[3\vec{f}(t_n, \vec{\xi}_1) + 5\vec{f}(t_n + \frac{2}{3}h, \vec{\xi}_2)\right] \end{cases}$$

see Iserles, p. 42 for a detailed derivation that shows this scheme is order 3 (error $\sim O(h^4)$)

Comments

- For every $\nu \geq 1$ there is a unique IRK method of order 2ν (compare Gaussian Quadrature Thm 3.3) [Iserles, p. 42]

- the RK Tableaux for this is

$$\vec{c}\left\{\begin{array}{c|cc} 0 & 1/4 & -1/4 \\ 2/3 & 3/12 & 5/12 \\ \hline & \underbrace{1/4 \quad\quad 3/4}_{\vec{w}^T} & \end{array}\right.\Bigg\} A$$

~~note the dependence of ξ_1 on~~
note the interdependence of ξ_1 and ξ_2

Collocation and IRK Methods

Define the collocation method: (for $\vec{y}' = \vec{f}(t,\vec{y})$)

Given $(t_n, \vec{y}_n)$ advance to $(t_{n+1}, \vec{y}_n)$ $t_{n+1} = t_n + h$

by finding a ν^{th} degree polynomial $\vec{u}(t)$ such that

(**) $$\vec{u}'(t_n + \tau_j h) = \vec{f}(t_n + \tau_j h, \vec{u}(t_n + \tau_j h)) \quad j = 1, 2, \ldots, \nu$$

by choosing collocation parameters $\tau_1, \tau_2, \ldots, \tau_\nu$

so $\vec{u}(t_n) = \vec{y}_n$.

That is, choose τ_j so that the ODE is exactly satisfied at ν distinct points.

Idea:

$$\vec{u} = \vec{a}_0 + \vec{a}_1 t + \vec{a}_2 t^2 + \ldots + \vec{a}_\nu t^\nu$$

have unknowns $\vec{a}_0, \vec{a}_1, \ldots, \vec{a}_\nu$ $(\nu+1)$

Along with unknowns $\tau_1, \tau_2, \ldots, \tau_\nu$ (ν)

we have $2\nu + 1$ degrees of freedom. We should be able to reach order 2ν in this method (i.e. error $\sim h^{2\nu+1}$)

(at least if we make the right choices... see Corollary p.47)

... details ...

Lemma 3.5 shows the connection between this collocation method and the IRK method.

Lemma 3.5

Set

$$g(t) \equiv \prod_{j=1}^{\nu} (t - \tau_j)$$

$$g_\ell(t) \equiv \frac{g(t)}{t-\tau_\ell} \qquad \ell = 1,2,\ldots,\nu$$

let

$$a_{j,i} = \int_0^{\tau_j} \frac{g_i(\tau)}{g_i(\tau_i)}\, d\tau \qquad j,i = 1,2,\ldots,\nu$$

$$w_j = \int_0^1 \frac{g_j(\tau)}{g_j(\tau_j)}\, d\tau \qquad j = 1,2,\ldots,\nu$$

The collocation method (**) is identical to the IRK method with

$$\begin{array}{c|c} \vec{\tau} & A \\ \hline & \vec{w}^T \end{array}$$

Proof:

see Iserles, p.43-44

Comments: (Iserles, p.44)

- Collocation Methods are special cases of IRK methods
- IRK preferred for actual computation

Two more ~~results~~ results to state (w/o proof ... see Iserles)

Thm 3.7

Suppose

$$\int_0^1 g(\tau)\tau^j d\tau = 0 \qquad j = 0, 1, \ldots, m-1$$

orthogonality condition

for some $m \in \{0, 1, \ldots, \nu\}$ where

$$g(t) \equiv \prod_{\ell=1}^{\nu} (t - \tau_\ell)$$

Then the collocation method (**) is of order $\nu + m$.

Corollary (Iserles, p. 47)

Let $\tau_1, \tau_2, \ldots, \tau_\nu$ be the zeros of the polynomials $\tilde{P}_\nu \in \mathbb{P}_\nu$ that are orthogonal with respect to the weight ~~functions~~ ~~$\omega(t) \equiv 1$~~, $0 \le t \le 1$.

Then the underlying collocation method (**) is order 2ν.

$\mathbb{P}_\nu$ = set of all real polynomials of degree ν.