2.20 One-step methods - RK review

$$x_{n+1} := x_n + h\phi_f(t_n, x_n, h)$$

with a method dependent *increment function* ϕ_h.

The basic construction scheme of an explicit method is

$$\begin{aligned} X_1 &= x_n \\ X_i &= x_n + h\sum_{j=1}^{i-1} a_{ij} f(t_n + c_j h, X_j) \quad i = 2, \ldots, s \\ x_{n+1} &= x_n + h\sum_{i=1}^{s} b_i f(t_n + c_i h, X_i). \end{aligned}$$

s number of stages, X_i stage values, b_i (quadrature) weights, $c_i \in [0, 1]$.

2.21 Stage derivative implementation

Alternatively, Runge–Kutta methods in *stage derivative form*:
$k_i := f(t_n + c_i h, X_i)$, :

$$
\begin{aligned}
k_1 &= f(t_n, x_n) \\
k_i &= f(t_n + c_i h, x_n + h \sum_{j=1}^{i-1} a_{ij} k_j) \qquad i = 2, \dots, s \\
x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i k_i.
\end{aligned}
$$

2.22 Butcher Tableau

Coefficients written compactly: *Butcher tableau*:

$$
\begin{array}{c|ccccc}
c_1 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
\quad \text{or} \quad
\begin{array}{c|c}
c & A \\
\hline
 & b^T
\end{array}
$$

with the $s \times s$ matrix $A = (a_{ij})$ and $a_{ij} = 0$ for $j \geq i$ in case of *explicit RK methods*.

2.23 Embedded RK methods

Error estimation by comparing two methods of different order

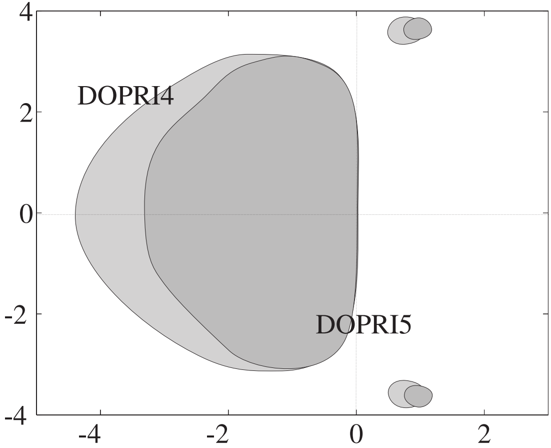
$$
\begin{array}{c|ccccc}
c_1 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \vdots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1^{\mathrm{p}} & b_2^{\mathrm{p}} & \cdots & b_{s-1}^{\mathrm{p}} & b_s^{\mathrm{p}} \\
\hline
 & b_1^{\mathrm{p+1}} & b_2^{\mathrm{p+1}} & \cdots & b_{s-1}^{\mathrm{p+1}} & b_s^{\mathrm{p+1}} \\
\hline
\end{array}
$$

where p indicates the order of the method and where

$$
\begin{aligned}
x_{n+1}^{(p)} &= x_n + h\sum_{i=1}^{s} b_i^{\mathrm{p}} k_i \\
x_{n+1}^{(p+1)} &= x_n + h\sum_{i=1}^{s} b_i^{\mathrm{p+1}} k_i.
\end{aligned}
$$

2.24 Stability region

Here a typical stability region for explicit RK methods



Stability regions for the Runge–Kutta pair DOPRI4 and DOPRI5.The methods are stable inside the gray areas.

2.25 Implicit RK methods

$$
\begin{aligned}
k_1 &= f(t_n, x_n) \\
k_i &= f(t_n + c_i h, x_n + h \sum_{j=1}^{s} a_{ij} k_j) \qquad i = 2, \ldots, s \\
x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i k_i.
\end{aligned}
$$

(Note, the upper bound in the sum!)

2.26 Butcher Tableau

Coefficients written compactly: *Butcher tableau*:

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

with the $s \times s$ matrix, generally full matrix $A = (a_{ij})$ in case of *implicit RK methods*.

2.27 Implicit RK methods - Demands/ wishes

Why implicit methods?

- Hope for bigger stability regions. A-stable methods?
- Hope for larger step sizes = faster method

Price?

- Large, $ns \times ns$ implicit equation system to solve at every step .
- Problematic starting values for Newton iteration (bad predictor)

2.28 Construction: Collocation approach

Definition:
The polynomial u of degree s defined by the conditions

$$\begin{aligned} u(t_n) &= x_n \\ \dot{u}(t_n + c_i h) &= f(t_n + c_i h, u(t_n + c_i h)) \end{aligned}$$

with distinct values $c_i \in [0,1], i = 1, \ldots, s$ is called a *collocation polynomial* of the ODE $\dot{x} = f(t,x), x(t_n) = x_n$. The c_i are called *collocation points*.

The idea of collocation methods is to approximate $x(t_{n+1})$ by $x_{n+1} := u(t_n + h)$.

Compare to BDF methods' definition

2.28 Collocation approach (Cont.)

$\dot{u}$ in Lagrange basis:

$$\dot{u}(t_n + \theta h) = \sum_{i=1}^{s} f(t_n + c_i h, u(t_n + c_i h))\ l_i(\theta)$$

with

$$l_i(\theta) = \prod_{\substack{j=1 \\ j \neq i}}^{s} \frac{\theta - c_j}{c_i - c_j}.$$

We get by integration

$$u(t_n + c_i h) = x_n + h \int_0^{c_i} \dot{u}(t_n + \theta h) \mathrm{d}\theta = x_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, u(t_n + c_j h))$$

with

$$a_{ij} = \int_0^{c_i} l_j(\theta) \mathrm{d}\theta.$$

2.29 Collocation approach (Cont.)

By setting $X_i := u(t_n + c_i h)$ we can express the collocation method as

$$
\begin{aligned}
X_i &= x_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, X_j) \quad i = 1, \ldots, s \\
x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i f(t_n + c_i h, X_i)
\end{aligned}
$$

with

$$b_i = \int_0^1 l_i(\theta) \mathrm{d}\theta.$$

Note: all $s^2 + s$ coefficients a_{ij}, b_i defined by giving the s collocation points c_i.

2.31 Collocation example: Gauß Method

c_i roots of *Legendre polynomials*.
For $s = 3$ tyhese are $c_1 = 1/2 - \sqrt{15}/10, c_2 = 1/2$, and $c_3 = 1/2 + \sqrt{15}/10$.
This gives the 3-stage *Gauß method*:

$$
\begin{array}{c|ccc}
\frac{1}{2} - \frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\
\frac{1}{2} & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\
\frac{1}{2} + \frac{\sqrt{15}}{10} & \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \\
\hline
 & \frac{5}{18} & \frac{4}{9} & \frac{5}{18}
\end{array}
$$

Order: $2s$ (maximal obtainable order for an s stage method)
Method is symmetric, no evaluation at end points

2.32 Collocation example: Radau Method

c_i roots of

$$p(t) = \frac{\mathrm{d}^{s-1}}{\mathrm{d}t^{s-1}} \left(t^{s-1}(t-1)^s \right)$$

gives Radau IIa methods.
For $s = 3$ this gives the RADAU5 method:

$$\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}$$

Order: $2s-1$ (maximal obtainable order for an s stage method)
Method is not symmetric, evaluation at the right interval end.

2.33 Nonlinear system: Jacobian

Let

$$J \approx \frac{\partial f}{\partial x}(t_m, x(t_m))$$

at some time point $t_m \leq t_n$ the $ns \times ns$ then the iteration matrix is

$$\begin{pmatrix} I - ha_{11}J & \dots & -ha_{1s}J \\ \vdots & & \vdots \\ -ha_{s1}J & \dots & I - ha_{ss}J \end{pmatrix}.$$

2.34 Kronecker Product Notation

Matrix Kronecker Product

$$A \otimes B := (a_{ij}B)_{i,j=1,\ldots,\dim(A)}$$

Here,

$$(I_{ns} - hA \otimes J).$$

With this the iterates for the stages are defined by

$$\begin{aligned} (I_{ns} - hA \otimes J)\,\Delta X^{(k)} &= -X^{(k)} + x_n + h(A \otimes I_n)F(X^{(k)}) \\ X^{(k+1)} &:= X^{(k)} + \Delta X^{(k)} \end{aligned}$$

with the vector of the stage iterates

$$X := \left(X_1^{\mathrm{T}}, \ldots, X_s^{\mathrm{T}}\right)^{\mathrm{T}} \in \mathbb{R}^{ns}$$

and

$$F(X) := \left(f(t_n + c_1 h, X_1)^{\mathrm{T}}, \ldots, f(t_n + c_s h, X_s)^{\mathrm{T}}\right)^{T}.$$

2.35 Simplifications by similarity transformation

Rule for Kronecker products

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

gives the transformed problem

$$\left(h^{-1}A^{-1} \otimes I_n - I_s \otimes J\right) \Delta X^{(k)} = (h^{-1}A^{-1} \otimes I_n)(x_n \otimes \mathbb{1} - X^{(k)}) + F(X^{(k)}).$$

Now, the constant, method dependent matrix A is separated from the state and problem dependent matrix J.

2.36 Simplifications by similarity transformation (Cont.)

A^{-1} can be transformed by means of a similarity transformation into a simpler form, i.e. a block diagonal form Λ

$$T^{-1}A^{-1}T =: \Lambda.$$

Coordinate transformation $Z := (T^{-1} \otimes I_n)X$, $z_n := (T^{-1} \otimes I_n)x_n$ simplifies the iteration to

$$\left(h^{-1}\Lambda \otimes I_n - I_s \otimes J\right)\Delta Z^{(k)} =$$
$$(h^{-1}\Lambda \otimes I_n)(z_n - Z^{(k)}) + (T^{-1} \otimes I_n)F((T \otimes I_n)Z^{(k)}).$$

2.37 Example: Radau5 Newton iteration

A^{-1} eigenvalues $\gamma = 3.6378$ and $\alpha \pm \beta = 2.6811 \pm 3.0504\,\mathrm{i}$.

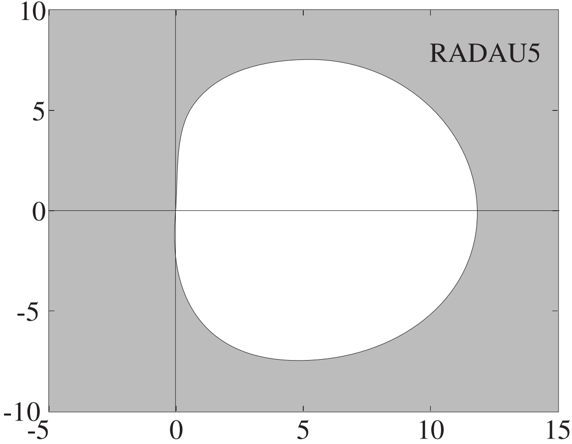
Thus A^{-1} can be transformed into

$$\Lambda = \begin{pmatrix} \gamma & 0 & 0 \\ 0 & \alpha & -\beta \\ 0 & \beta & \alpha \end{pmatrix}.$$

The linear system then takes the form

$$\begin{pmatrix} h^{-1}\gamma I_n - J & 0 & 0 \\ 0 & h^{-1}\alpha I_n - J & -h^{-1}\beta I_n \\ 0 & h^{-1}\beta I_n & h^{-1}\alpha I_n - J \end{pmatrix} \begin{pmatrix} \Delta Z_1 \\ \Delta Z_2 \\ \Delta Z_3 \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}$$

2.38 Stability region of Radau5



2.39 Stage derivative implementation for DAEs

Alternatively, Runge–Kutta methods in *stage derivative form*:
$F(t_n + c_i h, X_i, k_i) = 0$, :

$$\begin{aligned} 0 &= F(t_n + c_i h, x_n + h \sum_{j=1}^{i-1} a_{ij} k_j, k_i) \qquad i = 1, \ldots, s \\ x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i k_i. \end{aligned}$$

(see Slide 2.21)