Lecture Notes for the Course Numerical Methods for Time-Dependent Problems

Walter Zulehner
Institute for Computational Mathematics
Johannes Kepler University Linz

Summer Semester 2017

Contents

Chapter 1

Introduction

1.1 Examples

Chemical reactions

Brusselator: Six substances (species) A, B, D, E, X, Y undergo the following reactions:

$$\begin{aligned} A &\xrightarrow{k_1} X \\ B + X &\xrightarrow{k_2} Y + D \\ 2X + Y &\xrightarrow{k_3} 3X \\ X &\xrightarrow{k_4} E \end{aligned}$$

Here, k_i are the rate constants. Using the law of mass action (Massenwirkungsgesetz) we obtain the following differential equations for the concentrations c_A, c_B, c_D, c_E, c_X, c_Y:

$$\begin{aligned} c_A'(t) &= -k_1\, c_A(t), \\ c_B'(t) &= -k_2\, c_B(t) c_X(t), \\ c_D'(t) &= k_2\, c_B(t) c_X(t), \\ c_E'(t) &= k_4\, c_X(t), \\ c_X'(t) &= k_1\, c_A(t) - k_2\, c_B(t) c_X(t) + k_3\, c_X(t)^2 c_Y(t) - k_4\, c_X(t), \\ c_Y'(t) &= k_2\, c_B(t) c_X(t) - k_3\, c_X(t)^2 c_Y(t). \end{aligned}$$

Initial conditions: $c_A(0)$, $c_B(0)$, $c_D(0)$, $c_E(0)$, $c_X(0)$, $c_Y(0)$ are given

Mechanical systems

Elastic pendulum in 2D:

m: mass of the point mass. $x(t)$, $y(t)$: position of the point mass at time t.

Newton's second law:

$$m\,\ddot{x} = -k\,\left(\sqrt{x^2+y^2}-\ell\right)\frac{x}{\sqrt{x^2+y^2}},$$
$$m\,\ddot{y} = -mg - k\,\left(\sqrt{x^2+y^2}-\ell\right)\frac{y}{\sqrt{x^2+y^2}},$$

or, equivalently

$$m\,\ddot{x} = -2\,\lambda\,x,$$
$$m\,\ddot{y} = -mg - 2\,\lambda\,y$$

with

$$\lambda = \frac{k}{2}\,\frac{\sqrt{x^2+y^2}-\ell}{\sqrt{x^2+y^2}} = \frac{k}{2}\,\left(1-\frac{\ell}{\sqrt{x^2+y^2}}\right). \quad \text{i.e.} \quad x^2+y^2 = \frac{\ell^2}{\left(1-\frac{2\lambda}{k}\right)^2}.$$

In the limit case $k\to\infty$ (pendulum with fixed length) we obtain a system of differential algebraic equations (DAE):

$$m\,\ddot{x} = -2\,\lambda\,x,$$
$$m\,\ddot{y} = -mg - 2\,\lambda\,y,$$
$$0 = x^2+y^2-\ell^2.$$

Initial conditions: $x(0)$, $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$,

Restricted three body problem:

Two bodies of masses $1-\mu$ and μ at fixed positions $(-\mu,0)$ and $(1-\mu,0)$ in a rotating frame of reference. A third body at position $(y_1(t),y_2(t))$.

Newton's second law

$$y_1'' = y_1 + 2y_2' - \mu'\,\frac{y_1+\mu}{D_1^3} - \mu\,\frac{y_1-\mu'}{D_2^3}$$
$$y_2'' = y_2 - 2y_1' - \mu'\,\frac{y_2}{D_1^3} - \mu\,\frac{y_2}{D_2^3}$$

with

$$D_1 = \left[(y_1+\mu)^2+y_2^2\right]^{1/2}, \quad D_2 = \left[(y_1-\mu')^2+y_2^2\right]^{1/2}$$

and $\mu = 0.012277471$, $\mu' = 1-\mu$. Initial conditions: $y_1(0)$, $y_1'(0)$, $y_2(0)$, $y_2'(0)$ are given.

Partial differential equations

Initial boundary value problems

Heat equation

1D example:

$$u_t - a\,u_{xx} = f \quad x \in (0,1),\ t > 0$$

boundary conditions:

$$u(0,t) = u(1,t) = 0 \quad t > 0$$

initial conditions:

$$u(x,0) = u_0(x) \quad x \in [0,1]$$

Semi-discretization in space: method of lines. A finite element method (FEM) leads to

$$\begin{aligned} M_h \underline{u}_h'(t) + K_h \underline{u}_h(t) &= \underline{f}_h(t) \\ \underline{u}_h(0) &= \underline{u}_{0h} \end{aligned}$$

where M_h and K_h are the mass matrix and the stiffness matrix, respectively.

The problem is an example of a stiff problem.

MultiD: replace u_{xx} by Δu with

$$\Delta u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}.$$

Wave equation

1D example:

$$u_{tt} - c^2\,u_{xx} = f \quad x \in (0,1),\ t > 0$$

boundary conditions:

$$u(0,t) = u(1,t) = 0 \quad t > 0$$

initial conditions:

$$\begin{aligned} u(x,0) &= u_0(x) \quad x \in [0,1], \\ u_t(x,0) &= v_0(x) \quad x \in [0,1] \end{aligned}$$

Semi-discretization in space by a FEM:

$$\begin{aligned} M_h \underline{u}_h''(t) + K_h \underline{u}_h(t) &= \underline{f}_h(t), \\ \underline{u}_h(0) &= \underline{u}_{0h}, \\ \underline{u}_h'(0) &= \underline{v}_{0h}. \end{aligned}$$

This problem is also an example of a stiff problem.

Navier-Stokes equations

The velocity u and the pressure p of a Newtonian fluid satisfy the following system of PDEs:

$$\begin{aligned} u_t + (u \cdot \nabla)u - \nu\,\Delta u + \nabla p = f \quad & x \in \Omega,\ t > 0, \\ \nabla \cdot u = 0 \quad & x \in \Omega,\ t > 0 \end{aligned}$$

with appropriate boundary and initial conditions.

Semi-discretization in space leads to a DAE:

$$\begin{aligned} M_h \underline{u}_h'(t) + A_h(\underline{u}_h(t))\,\underline{u}_h(t) + B_h^T \underline{p}_h(t) = \underline{f}_h(t) \quad & t > 0, \\ B_h \underline{u}_h = \underline{g}_h \qquad & x \in \Omega,\ t > 0. \end{aligned}$$

1.2 Standard Forms

explicit ODEs:

Find $u : [0,T] \longrightarrow \mathbb{R}^N$ such that

$$\begin{aligned} u'(t) &= f(t, u(t)) \quad t \in (0,T), \\ u(0) &= u_0 \end{aligned}$$

with given right-hand side $f : D \times (0,T) \longrightarrow \mathbb{R}^N$, $D \subset \mathbb{R}^N$ and initial value $u_0 \in \mathbb{R}^N$.

Formulation as an operator equation:

$$\psi(u) = 0,$$

where $\psi\colon C^1(0,T) \cap C[0,T) \longrightarrow \mathbb{R} \times C(0,T)$ is given by

$$\psi(v) = \begin{pmatrix} v(t_0) - u_0 \\ v'(t) - f(t, v(t)),\ t \in (0,T) \end{pmatrix}.$$

Second (and higher) order initial value problems like

$$\begin{aligned} u''(t) &= f(t, u(t), u'(t)) \quad t \in (0,T), \\ u(0) &= u_0, \\ u'(0) &= v_0 \end{aligned}$$

can be transformed into this standard form: With

$$u_1(t) = u(t),\ u_2(t) = u'(t)$$

we have

$$u_1'(t) = u_2(t), \quad u_2'(t) = f(t, u_1(t), u_2(t))$$

with initial conditions:

$$u_1(0) = u_0, \quad u_2(0) = v_0.$$

Special cases:

- Linear ODEs with constant coefficients

$$u'(t) = Ju(t) + f(t)$$

- Right-hand side does not depend on u:

$$u'(t) = f(t)$$

  Then

$$u(t) = u(0) + \int_0^t u'(s)\ ds = u_0 + \int_0^t f(s)\ ds.$$

  (integration problem)

- autonomous ODEs:

$$u'(t) = f(u(t))$$

  Each ODE of the form

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

  can be transformed into an equivalent autonomous problem:

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}' = \begin{pmatrix} 1 \\ f(u_1(t), u_2(t)) \end{pmatrix}$$

  for

$$u_1(t) = t, \quad u_2(t) = u(t).$$

(fully) implicit ODEs:

$$F(t, u(t), u'(t)) = 0$$

semi-explicit DAEs:

$$u(t) = \begin{bmatrix} y(t) \\ z(t) \end{bmatrix}$$

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ 0 &= g(t, y(t), z(t)). \end{aligned}$$

Part I

Nonstiff Problems

Chapter 2

Runge-Kutta Methods

Initial value problem (IVP):

$$\begin{aligned} u'(t) &= f(t, u(t)), \quad t \in (t_0, T) \\ u(t_0) &= u_0 \end{aligned}$$

2.1 Explicit Runge-Kutta Methods

The approximation u_1 for $u(t_0 + \tau)$ is obtained from the exact relation

$$u(t_0 + \tau) = u(t_0) + \int_{t_0}^{t_0+\tau} f(t, u(t))\ dt$$

by using a quadrature rule

$$\int_{t_0}^{t_0+\tau} f(t, u(t))\ dt \approx \tau \sum_{i=1}^{s} b_i f(t_0 + c_i\tau, U_i),$$

where U_i denotes an approximation of $u(t_0 + c_i\tau)$.

Approximate values U_i for $u(t_0 + c_i\,\tau)$ are obtained from the exact relation

$$u(t_0 + c_i\,\tau) = u(t_0) + \int_{t_0}^{t_0+c_i\tau} f(t, u(t))\ dt$$

by again using quadrature rules

$$\int_{t_0}^{t_0+c_i\tau} f(t, u(t))\ dt \approx \tau \sum_{j=1}^{i-1} a_{ij} f(t_0 + c_j\tau, U_j).$$

This leads to the class of explicit s-stage Runge-Kutta methods (with $c_1 = 0$):

$$
\begin{aligned}
U_1 &= u_0 \\
U_2 &= u_0 + \tau\, a_{21}\, f(t_0, U_1) \\
U_3 &= u_0 + \tau \left[a_{31}\, f(t_0, U_1) + a_{32}\, f(t_0 + c_2\tau, U_2) \right] \\
&\vdots \\
U_s &= u_0 + \tau \left[a_{s1}\, f(t_0, U_1) + a_{s2}\, f(t_0 + c_2\tau, U_2) + \ldots + a_{s,s-1}\, f(t_0 + c_{s-1}\tau, U_{s-1}) \right] \\
\\
u_1 &= u_0 + \tau \left[b_1\, f(t_0, U_1) + b_2\, f(t_0 + c_2\tau, U_2) + \ldots + b_{s-1}\, f(t_0 + c_{s-1}\tau, U_{s-1}) + b_s\, f(t_0 + c_s\tau, U_s) \right]
\end{aligned}
$$

or in the form

$$
\begin{aligned}
U_1 &= u_0 \\
U_2 &= u_0 + \tau\, a_{21}\, U_1' \\
U_3 &= u_0 + \tau \left[a_{31}\, U_1' + a_{32}\, U_2' \right] \\
&\vdots \\
U_s &= u_0 + \tau \left[a_{s1}\, U_1' + a_{s2}\, U_2' + \ldots + a_{s,s-1}\, U_{s-1}' \right] \\
\\
u_1 &= u_0 + \tau \left[b_1\, U_1' + b_2\, U_2' + \ldots + b_{s-1}\, U_{s-1}' + b_s\, U_s' \right]
\end{aligned}
$$

with the abbreviation (this is not a differential equation):

$$U_i' = f(t_0 + c_\tau, U_i), \quad i = 1, \ldots, s.$$

Butcher tableau:

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

In short:

$$
\begin{array}{c|c}
c & A \\
\hline
 & b^T
\end{array}
\quad \text{with} \quad A \in \mathbb{R}^{s \times s} \text{ strictly lower triangular, } b, c \in \mathbb{R}^s
$$

Examples:

- explicit (forward) Euler method (left endpoint rule)

$$
\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}
\qquad \text{i.e.} \qquad
\begin{aligned}
U_1 &= u_0 \\
u_1 &= u_0 + \tau\, f(t_0, U_1)
\end{aligned}
\qquad \text{i.e.} \qquad
u_1 = u_0 + \tau\, f(t_0, u_0)
$$

- explicit midpoint method (Runge):

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \qquad \text{i.e.} \qquad \begin{aligned} U_1 &= u_0 \\ U_2 &= u_0 + \tfrac{1}{2}\tau f(t_0, U_1) \\ u_1 &= u_0 + \tau\, f(t_0, U_2) \end{aligned}$$

- explicit trapezoidal rule (Runge):

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \text{i.e.} \qquad \begin{aligned} U_1 &= u_0 \\ U_2 &= u_0 + \tau f(t_0, U_1) \\ u_1 &= u_0 + \tfrac{1}{2}\tau \left[f(t_0, U_1) + f(t_0 + \tau, U_2) \right] \end{aligned}$$

- "The" Runge-Kutta method:

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

i.e.

$$\begin{aligned} U_1 &= u_0 \\ U_2 &= u_0 + \tfrac{1}{2}\tau f(t_0, U_1) && \text{left endpoint rule} \\ U_3 &= u_0 + \tfrac{1}{2}\tau f\left(t_0 + \tfrac{1}{2}\tau, U_2\right) && \text{right endpoint rule} \\ U_4 &= u_0 + \tau\, f(t_0 + \tfrac{1}{2}\tau, U_3) && \text{midpoint rule} \end{aligned}$$

$$u_1 = u_0 + \tfrac{1}{6}\tau \left[f(t_0, U_1) + 2\, f\left(t_0 + \tfrac{1}{2}\tau, U_2\right) + 2\, f\left(t_0 + \tfrac{1}{2}\tau, U_3\right) + f(t_0 + \tau, U_4) \right]$$

2.2 Local Error and Order Conditions

An explicit s-stage Runge-Kutta method applied to

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

reads:

$$\begin{aligned} U_i &= u_0 + \tau \sum_{j=1}^{i-1} a_{ij}\, f(t_0 + c_j\, \tau, U_j), \quad i = 1, \ldots, s, \\ u_1 &= u_0 + \tau \sum_{i=1}^{s} b_i\, f(t_0 + c_i\, \tau, U_j). \end{aligned}$$

If applied to the equivalent autonomous system

$$\begin{pmatrix} t \\ u(t) \end{pmatrix}' = \begin{pmatrix} 1 \\ f(t, u(t)) \end{pmatrix}$$

we obtain

$$\begin{aligned} T_i &= t_0 + \tau \sum_{j=1}^{i-1} a_{ij}, \quad i = 1, \dots, s, \\ U_i &= u_0 + \tau \sum_{j=1}^{i-1} a_{ij} f\big(T_j, U_j\big), \quad i = 1, \dots, s, \\ t_1 &= t_0 + \tau \sum_{i=1}^{s} b_i \\ u_1 &= u_0 + \tau \sum_{i=1}^{s} b_i f\big(T_i, U_i\big) \end{aligned}$$

Hence, by eliminating T_i, we obtain

$$\begin{aligned} U_i &= u_0 + \tau \sum_{j=1}^{i-1} a_{ij} f\big(t_0 + \tau \textstyle\sum_{k=1}^{j-1} a_{jk}, U_j\big), \quad i = 1, \dots, s, \\ u_1 &= u_0 + \tau \sum_{i=1}^{s} b_i f\big(t_0 + \tau \textstyle\sum_{j=1}^{i-1} a_{ij}, U_j\big), \end{aligned}$$

which coincides with the original method provided

$$c_i = \sum_{j=1}^{i-1} a_{ij}.$$

So, under this simplifying condition we can restrict ourselves to autonomous systems

$$u'(t) = f(u(t)).$$

Definition 2.1. *1. The error after one step of a Runge-Kutta method $u(t_0 + \tau) - u_1$ is called the local error.*

2. A Runge-Kutta method is called consistent if

$$u(t_0 + \tau) - u_1 = o(\tau).$$

3. A Runge-Kutta method is called of (consistency) order $p \in \mathbb{N}$ if

$$u(t_0 + \tau) - u_1 = O(\tau^{p+1}).$$

The order of a Runge-Kutta method can be determined by a Taylor expansion, provided the involved functions are sufficiently smooth:

Taylor expansion of the exact solution:

$$u(t_0+\tau) = u_0 + \tau\, u'(t_0) + \frac{\tau^2}{2}\, u''(t_0) + \frac{\tau^3}{6}\, u'''(t_0) + \frac{\tau^4}{24}\, u''''(t_0) + \dots$$

For the derivatives of u we have

- $$u' = f(u)$$
- $$u'' = f'(u)[u'] = f'(u)[f(u)],$$

  where
  $$f'(u)[v] = \sum_{I=1}^{N} \frac{\partial f}{\partial u_I}(u) v_I.$$
- $$u''' = f''(u)[f(u), f(u)] + f'(u)\big[f'(u)[f(u)]\big],$$

  where
  $$f''(u)[v,w] = \sum_{I,J=1}^{N} \frac{\partial^2 f}{\partial u_I \partial u_J}(u) v_I w_J.$$
- $$\begin{aligned} u'''' = {} & f''''(u)[f(u), f(u), f(u)] + 3f'''(u)\big[f'(u)[f(u], f(u)\big] \\ & + f'(u)\big[f''(u)[f(u), f(u)]\big] + f'(u)\Big[f'(u)\big[f'(u)[f(u)]\big]\Big], \end{aligned}$$

  where
  $$f''''(u)[v,w,r] = \sum_{I,J,K=1}^{N} \frac{\partial^3 f}{\partial u_I \partial u_J \partial u_K}(u) v_I w_J r_K.$$

For the approximation u_1 given by

$$u_1 = u_1(\tau) = u_0 + \tau \sum_{i=1}^{s} b_i f(U_i) \quad \text{with} \quad U_i = U_i(\tau) = u_0 + \tau \sum_{j=1}^{i-1} a_{ij} f(U_j(\tau)),$$

the Taylor expansion reads

$$u_1 = u_0 + \tau\, u_1'(0) + \frac{\tau^2}{2}\, u_1''(0) + \frac{\tau^3}{6}\, u_1'''(0) + \frac{\tau^3}{24}\, u_1''''(0) + \dots$$

For computing the derivatives of u_1 we use:

Lemma 2.1 (Leibniz' formula)**.**

$$[\tau \cdot \phi(\tau)]^{(q)} \big|_{\tau=0} = q \cdot \phi^{(q-1)}(0).$$

Proof. By induction:
$q = 1$:

$$[\tau \cdot \phi(\tau)]' = \phi(\tau) + \underbrace{\tau \cdot \phi'(\tau)}_{= 0 \text{ for } \tau = 0}$$

$q \to q + 1$:

$$[\tau \cdot \phi(\tau)]^{(q+1)} = [\phi(\tau) + \tau \cdot \phi'(\tau)]^{(q)} = \phi^{(q)}(\tau) + [\tau \cdot \phi'(\tau)]^{(q)}$$

Therefore,

$$[\tau \cdot \phi(\tau)]^{(q+1)} \big|_{\tau=0} = \phi^{(q)}(0) + q \cdot \phi^{(q)}(0) = (q+1) \cdot \phi^{(q)}(0).$$

□

Then

- $$u_1'(0) = 1 \cdot \left\{ \sum_i b_i \right\} f(u_0)$$

  using
  $$U_i(0) = u_0.$$

- $$u_1''(0) = 2 \cdot \sum_i b_i \left(f(U_i) \right)'(0)$$

  Now
  $$\left(f(U_i) \right)' = f'(U_i)[U_i'].$$

  With
  $$U_i'(0) = 1 \cdot \left\{ \sum_j a_{ij} \right\} f(u_0)$$

  we obtain
  $$\left(f(U_i) \right)'(0) = f'(u_0) \left[\sum_j a_{ij} f(u_0) \right] = \left\{ \sum_j a_{ij} \right\} f'(u_0)[f(u_0)]$$

  and, therefore,
  $$u_1''(0) = 2 \left\{ \sum_{i,j} b_i \, a_{ij} \right\} f'(u_0)[f(u_0)].$$

- $$u_1'''(0) = 3 \sum_i b_i \big(f(U_j)\big)''(0)$$

  Now
  $$\big(f(U_i)\big)'' = f''(U_i)[U_i', U_i'] + f'(U_i)[U_i''].$$
  With
  $$U_i''(0) = 2 \left\{\sum_j a_{ij}\right\} \big(f(U_j)\big)'(0) = 2 \left\{\sum_{j,k} a_{ij}\, a_{jk}\right\} f'(u_0)[f(u_0)]$$
  we obtain
  $$\begin{aligned} u_1'''(0) &= 3 \sum_i b_i \left\{ f''(u_0)\Big[\sum_j a_{ij} f(u_0), \sum_k a_{ik} f(u_0)\Big] \right. \\ &\qquad \left. + f'(u_0)\Big[2 \sum_{jk} a_{ij}\, a_{jk} f'(u_0)[f(u_0)]\Big] \right\} \\ &= 3 \left\{\sum_{i,j,k} b_i a_{ij} a_{ik}\right\} f''(u_0)[f(u_0), f(u_0)] \\ &\qquad + 6 \left\{\sum_{i,j,k} b_i\, a_{ij}\, a_{jk}\right\} f'(u_0)\big[f'(u_0)[f(u_0)]\big] \end{aligned}$$

Therefore, we obtain the following expansion of the local error:

$$\begin{aligned} u(t_0+\tau) - u_1 = \tau &\Big(1 - \sum_i b_i\Big) f(u_0) \\ &+ \frac{\tau^2}{2}\Big(1 - 2\sum_{i,j} b_i\, a_{ij}\Big) f'(u_0)[f(u_0)] \\ &+ \frac{\tau^3}{6}\Big(1 - 3\sum_{i,j,k} b_i a_{ij} a_{ik}\Big) f''(u_0)[f(u_0), f(u_0)] \\ &+ \frac{\tau^3}{6}\Big(1 - 6\sum_{i,j,k} b_i\, a_{ij}\, a_{jk}\Big)\, f'(u_0)\big[f'(u_0)[f(u_0)]\big] + O(\tau^4) \end{aligned}$$

Theorem 2.1. *The Runge-Kutta method is of order 1 iff* (2.1) *holds, of order 2 iff* (2.1),

(2.2) *hold, of order 3 iff* (2.1), (2.2), (2.3), (2.4) *hold, where*

$$\sum_i b_i = 1, \tag{2.1}$$

$$2\sum_{i,j} b_i a_{ij} = 1, \tag{2.2}$$

$$3\sum_{i,j,k} b_i a_{ij} a_{ik} = 1, \tag{2.3}$$

$$6\sum_{i,j,k} b_i a_{ij} a_{jk} = 1. \tag{2.4}$$

If

$$c_i = \sum_{j=1}^{s} a_{ij} \quad \text{for } i = 1, \ldots, s,$$

these conditions can also be written in the form

$$\begin{aligned}
\sum_i b_i &= 1, \\
\sum_i b_i \big(\sum_j a_{ij} \big) = \sum_i b_i c_i &= \frac{1}{2}, \\
\sum_i b_i \Big(\sum_j a_{ij} \big(\sum_k a_{ik} \big) \Big) = \sum_i b_i \big(\sum_j a_{ij} \big) c_i = \sum_i b_i c_i^2 &= \frac{1}{3}, \\
\sum_j \Big(\sum_i b_i a_{ij} \big(\sum_k a_{jk} \big) \Big) = \sum_j \big(\sum_i b_i a_{ij} \big) c_j &= \frac{1}{6}.
\end{aligned}$$

Examples:

- explicit (forward) Euler method: $p = 1$
- explicit midpoint method: $p = 2$
- explicit trapezoidal rule (Runge): $p = 2$
- "The" Runge-Kutta method: $p = 4$

The different terms appearing for the derivative $u^{(q)}$ are called elementary differentials. The number of different elementary differential #(ElemDiff) grows with the order q of differentiation:

q	1	2	3	4	5	6	7	8	9	10
#(ElemDiff)	1	1	2	4	9	20	48	115	286	719

The number of conditions #(Cond) for obtaining a consistency order of at least p is given by the number of elementary differentials up to order p:

p	1	2	3	4	5	6	7	8	9	10
#(Cond)	1	2	4	8	17	37	85	200	486	1205

The total number #(Coeff) of coefficients a_{jk} and b_j of an explicit s-stage Runge-Kutta method is $s(s+1)/2$:

s	1	2	3	4	5	6	7	8	9	10
#(Coeff)	1	3	6	10	15	21	28	36	45	55

Observe that #(Coeff) grows slower with s than #(Cond) grows with p. Therefore, it is plausible that the minimum number $s_{\min}$ of stages necessary to obtain a method of order p is at least p. The next table shows the relation between p and $s_{\min}$:

p	1	2	3	4	5	6	7	8
$s_{\min}$	1	2	3	4	6	7	9	11

The vertical lines after $p = 4, 6, 7$ are called Butcher barriers. They illustrate the following results:

Theorem 2.2 (around 1963)**.** *For $p \geq 5$ no explicit Runge-Kutta method exists of order p with $s \leq p$ stages.*

Theorem 2.3 (Butcher, 1965)**.** *For $p \geq 7$ no explicit Runge-Kutta method exists of order p with $s \leq p + 1$ stages.*

Theorem 2.4 (Butcher, 1985)**.** *For $p \geq 8$ no explicit Runge-Kutta method exists of order p with $s \leq p + 2$ stages.*

A method by Hairer (1978) of order $p = 10$ (#(Cond) = 1205) with $s = 17$ (#(Coeff) = 153) stages made it into the Guinness Book of Records.

2.3 Implicit Runge-Kutta Methods

A general s-stage Runge-Kutta method applied to

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

reads:

$$U_i = u_0 + \tau \sum_{j=1}^{s} a_{ij}\, f(t_0 + c_j\, \tau, U_j), \quad i = 1, \ldots, s$$

$$u_1 = u_0 + \tau \sum_{i=1}^{s} b_i\, f(t_0 + c_i\, \tau, U_i)$$

with associated Butcher tableau

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} \qquad \text{or, in short,} \qquad \begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Definition 2.2. *An s-stage Runge-Kutta method*

1. *is called explicit, if A is a strictly lower triangular matrix,*
2. *is called implicit, otherwise.*

Usually one additionally assumes that $c_1 = 0$ for explicit Runge-Kutta methods.

Examples

- implicit (backward) Euler method (right endpoint rule)

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \quad \text{i.e.} \quad \begin{array}{l} U_1 = u_0 + \tau\, f(t_0 + \tau, U_1) \\ u_1 = u_0 + \tau\, f(t_0 + \tau, U_1) \end{array} \quad \text{i.e.} \quad u_1 = u_0 + \tau\, f(t_0 + \tau, u_1)$$

- implicit midpoint method (Gauss method):

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array} \quad \text{i.e.} \quad \begin{array}{l} U_1 = u_0 + \frac{1}{2}\tau f(t_0 + \frac{1}{2}\tau, U_1) \\ u_1 = u_0 + \tau\, f(t_0 + \frac{1}{2}\tau, U_1) \end{array}$$

- implicit trapezoidal rule:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad \text{i.e.} \quad u_1 = u_0 + \tfrac{1}{2}\tau \left[f(t_0, u_0) + f(t_0 + \tau, u_1) \right]$$

- θ-method:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1-\theta & \theta \\ \hline & 1-\theta & \theta \end{array} \quad \text{i.e.} \quad u_1 = u_0 + \tfrac{1}{2}\tau \left[(1-\theta)\, f(t_0, u_0) + \theta\, f(t_0 + \tau, u_1) \right]$$

  Special cases: $\theta = 0$: explicit Euler method, $\theta = \frac{1}{2}$: implicit trapezoidal rule, $\theta = 1$: implicit Euler method.

For an implicit Runge-Kutta method, the computation of the next approximation u_1 requires, in general, the solution of s equations (s systems of equations) in fixed point forms:

$$U = \Phi(U; t_0, u_0, \tau) \tag{2.5}$$

with $U = (U_i)_{i=1,\dots,s}$ and $\Phi(U) = \Phi(U; t_0, u_0, \tau) = (\Phi_i(U; t_0, u_0, \tau))_{i=1,\dots,s}$ given by

$$\Phi_i(U; t_0, u_0, \tau) = u_0 + \tau \sum_{j=1}^{s} a_{ij}\, f(t_0 + c_j\, \tau, U_j)$$

or, equivalently,

$$U' = \Psi(U'; t_0, u_0, \tau)$$

with $U' = (U'_j)_{j=1,\dots,s}$ and $\Psi(U') = \Psi(U'; t_0, u_0, \tau) = (\Psi_i(U'; t_0, u_0, \tau))_{i=1,\dots,s}$ given by

$$\Psi_i(U'; t_0, u_0, \tau) = f\Big(t_0 + c_i\, \tau, u_0 + \tau \sum_{j=1}^{s} a_{ij}\, U'_j\Big).$$

Theorem 2.5. *Let $D = \{(t, v) \in \mathbb{R} \times \mathbb{R}^N : t_0 \le t \le T, \|v - u_0\| \le b\}$. Assume that f is continuous on D with*

$$\|f(t, v)\| \le K \quad \textit{for all } (t, v) \in D$$

and f satisfies the Lipschitz condition

$$\|f(t, w) - f(t, v)\| \le L\, \|w - v\| \quad \textit{for all } (t, v), (t, w) \in D.$$

If $t_0 + c_i\, \tau \in [t_0, T]$ for $i = 1, \dots, s$, $\tau\, \|A\|_\infty\, K \le b$, and $\tau\, \|A\|_\infty\, L < 1$ with $\|A\|_\infty = \max_i \sum_j |a_{ij}|$, then there exists a unique solution to the fixed point equations (2.5) in D and the fixed point iteration converges to this solution for any initial guess in D.

Proof. The theorem follows from Banach's fixed point theorem applied to $\Phi : \big(\mathbb{R}^N\big)^s \longrightarrow \big(\mathbb{R}^N\big)^s$ with norm

$$\|U\| = \max_{i=1,\dots,s} \|U_i\| \quad \text{for} \quad U = (U_i)_{i=1,\dots,s} \in \big(\mathbb{R}^N\big)^s$$

Let

$$C = \{U = (U_i)_{i=1,\dots,s} \in \big(\mathbb{R}^N\big)^s : \|U_i - u_0\| \le b \text{ for } i = 1, \dots, s\}.$$

Then

$$\Phi(C) \subset C,$$

since

$$\|\Phi_i(U) - u_0\| = \tau \Big\| \sum_{j=1}^{s} a_{ij}\, f(t_0 + c_j\, \tau, U_j) \Big\| \le \tau \sum_{j=1}^{s} |a_{ij}|\, K \le \tau\, \|A\|_\infty\, K \le b,$$

and

$$\begin{aligned} \|\Phi_i(W) - \Phi_i(V)\| &= \tau \Big\| \sum_{j=1}^{s} a_{ij}\, \left[f(t_0 + c_j\, \tau, W_j) - f(t_0 + c_j\, \tau, V_j)\right] \Big\| \\ &\le \tau \sum_{j=1}^{s} |a_{ij}|\, L\, \|W_j - V_j\| \le \tau\, \|A\|_\infty\, L\, \|W - V\| = q\, \|W - V\| \end{aligned}$$

with $q = \tau\, \|A\|_\infty\, L < 1$. □

2.4 Order of Consistency of Runge-Kutta Methods

From a Taylor expansion of the local error

$$u(t_0 + \tau) - u_1$$

we obtain completely analogously as for explicit Runge-Kutta methods the same order conditions, see Theorem 2.1 for the cases $p = 1, 2, 3$.

Examples:

- θ-method: $p = 1$ for $\theta \neq \frac{1}{2}$, $p = 2$ for $\theta = \frac{1}{2}$.
- 1-stage method of order 2: implicit midpoint rule
- 2-stage method of order 4: method by Hammer-Hollingsworth

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

Theorem 2.6 (Butcher)**.** *If the conditions*

$$\sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k} \qquad k = 1, \ldots, p \qquad B(p)$$

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k} \qquad i = 1, \ldots, s,\ k = 1, \ldots, q \qquad C(q)$$

$$\sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{b_j}{k}(1 - c_j^k) \qquad j = 1, \ldots, s,\ k = 1, \ldots, r \qquad D(r)$$

are satisfied with $p \leq q + r + 1$, $p \leq 2q + 2$, then the method is of order p.

Discussion of condition $B(p)$

The coefficients c_i and b_i, $i = 1, \ldots, s$ determine a quadrature rule:

$$\int_{t_0}^{t_0+\tau} f(t)\ dt \approx \tau \sum_{i=1}^{s} b_i\, f(t_0 + c_i\, \tau).$$

The quadrature rule is exact for polynomials of degree $\leq p - 1$, if and only if $B(p)$ is satisfied: Without loss of generality: $t_0 = 0$, $\tau = 1$. For $f(t) = t^{k-1}$, $k = 1, \ldots, p - 1$ we obtain the conditions

$$\sum_{i=1}^{s} b_i\, c_i^{k-1} = \int_0^1 t^{k-1}\ dt = \frac{1}{k}.$$

Discussion of condition $C(q)$

The coefficients c_i and a_{ij}, $j = 1, \ldots, s$ determine a quadrature rule:

$$\int_{t_0}^{t_0+c_i\,\tau} f(t)\; dt \approx \tau \sum_{j=1}^{s} a_{ij}\, f(t_0 + c_j\,\tau).$$

The quadrature rule is exact for polynomials of degree $\leq q-1$, if and only if $C(q)$ is satisfied. q is called the stage order.

Discussion of condition $D(r)$

This condition allows to simplify order conditions like (2.4): If $D(1)$ is satisfied, we obtain

$$\sum_j \big(\sum_i b_i a_{ij}\big) c_j = \sum_j b_j (1 - c_j) c_j = \sum_j b_j c_j - \sum_j b_j c_j^2.$$

Gauß methods

For the definition we need the Legendre polynomials $P_n(x)$, which are polynomials of degree n that form a sequence of orthogonal polynomials with respect to the inner product

$$\int_{-1}^{1} f(x)\, g(x)\; dx.$$

Definition 2.3. *The s-stage Gauß method is given by:*

- *$c_1, \ldots, c_s$ are the roots of $P_s(2x-1)$.*
- *b_i are determined by condition $B(s)$.*
- *a_{ij} are determined by condition $C(s)$.*

Theorem 2.7. *The s-stage Gauß method satisfies $B(2s)$, $C(s)$, $D(s)$ and is of order $p = 2s$.*

Butcher's Radau and Lobatto methods

- Butcher's type I methods (Radau I): $c_1 = 0$ and $a_{1j} = 0$ for $j = 1, \ldots, s$.

  Then $U_1 = u_0$ and the fixed point problem $U = \Phi(U)$ reduces to a system of $(s-1)\,N$ equations for $U_2, \ldots, U_s$.

- Butcher's type II methods (Radau II): $c_s = 1$ and $a_{is} = 0$ for $i = 1, \ldots, s$.

  Then the fixed point problem $U = \Phi(U)$ reduces to a system of $(s-1)\,N$ equations for $U_1, \ldots, U_{s-1}$, while U_s is explicitly given in terms of $U_1, \ldots, U_{s-1}$.

- Butcher's type III methods (Lobatto III): $c_1 = 0$ and $c_s = 1$, $a_{1j} = 0$ for $j = 1, \ldots, s$, and $a_{is} = 0$ for $i = 1, \ldots, s$.

  Then the fixed point problem $U = \Phi(U)$ reduces to a system of $(s-2)\,N$ equations for $U_2, \ldots, U_{s-1}$, while $U_1 = u_0$ and U_s is explicitly given in terms of $U_1, \ldots, U_{s-1}$.

For definition we need the Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$, which are polynomials of degree n that form a sequence of orthogonal polynomials with respect to the inner product

$$\int_{-1}^{1} f(x)\, g(x)\, (1-x)^{\alpha}\, (1+x)^{\beta}\, dx.$$

Definition 2.4. *1. Butcher's s-stage type I method (Radau I) is given by*

- $c_1 = 0$, $c_2, \ldots, c_s$ *are the roots of* $P_{s-1}^{(0,1)}(2x-1)$.
- b_i *are determined by condition* $B(s)$.
- $a_{1j} = 0$ *for* $j = 1, \ldots, s$. *The remaining coefficients* a_{ij} *are determined by* $C(s)$ *(or, equivalently, by* $D(s-1)$*).*

2. Butcher's s-stage type II method (Radau II) is given by

- $c_s = 1$, $c_1, \ldots, c_{s-1}$ *are the roots of* $P_{s-1}^{(1,0)}(2x-1)$.
- b_i *are determined by condition* $B(s)$.
- $a_{is} = 0$ *for* $i = 1, \ldots, s$. *The remaining coefficients* a_{ij} *are determined by condition* $D(s)$ *(or, equivalently, by* $C(s-1)$*).*

3. Butcher's s-stage type III method (Lobatto II) is given by

- $c_1 = 0$, $c_s = 1$, $c_2, \ldots, c_{s-1}$ *are the roots of* $P_{s-2}^{(1,1)}(2x-1)$.
- b_i *are determined by condition* $B(s)$.
- $a_{1j} = 0$ *for* $j = 1, \ldots, s$, $a_{is} = 0$ *for* $i = 1, \ldots, s$. *The remaining coefficients* a_{ij} *are determined by condition* $C(s-1)$ *(or, equivalently, by* $D(s-1)$*).*

Theorem 2.8. *1. Butcher's s-stage type I method (Radau I) satisfies* $B(2s-1)$, $C(s)$, $D(s-1)$ *and has order* $p = 2s-1$.

2. Butcher's s-stage type II method (Radau II) satisfies $B(2s-1)$, $C(s-1)$, $D(s)$ *and has order* $p = 2s-1$.

3. Butcher's s-stage type III method (Lobatto III) satisfies $B(2s-1)$, $C(s-1)$, $D(s-1)$ *and has order* $p = 2s-2$.

Remark. *Two other frequently used classes of implicit Runge-Kutta methods:*

- *DIRK methods (diagonal implicit Runge-Kutta methods) with tableau*

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & \dots & 0 \\ c_2 & a_{21} & a_{22} & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ c_s & a_{11} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

  The fixed point equation $U = \Phi(U)$ reduces to s systems of N equations for each U_i.

- *the SDIRK methods (singly diagonal implicit Runge-Kutta methods) with tableau*

$$\begin{array}{c|cccc} c_1 & \gamma & 0 & \dots & 0 \\ c_2 & a_{21} & \gamma & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ c_s & a_{11} & a_{s2} & \dots & \gamma \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

  This further simplifies the computation of U.

Chapter 3

Convergence Analysis for One-Step Methods

Subdivision of $I = [t_0, T]$:

$$t_0 < t_1 < \ldots < t_m = T$$

Notations:

- $\tau_j = t_{j+1} - t_j$: step size ($t_{j+1} = t_j + \tau_j$)
- $\tau = (\tau_0, \tau_1, \ldots, \tau_{m-1})$.
- $|\tau| = \max\{\tau_j : j = 0, 1 \ldots, m-1\}$.
- $I_\tau = \{t_0, t_1, \ldots, t_m\}$.

A one-step method for solving the initial value problem:

$$\begin{aligned} u'(t) &= f(t, u(t)), \quad t \in (t_0, T), \\ u(t_0) &= u_0 \end{aligned} \tag{3.1}$$

is a method of the form

$$u_{j+1} = u_j + \tau_j \, \phi(t_j, u_j, \tau_j) \quad \text{for } j = 0, \ldots, m-1.$$

The function ϕ is called the increment function.

Example. *The Runge-Kutta methods are obviously one-step methods. Let $U(t_0, u_0, \tau)$ denote the (unique) solution of (2.5). Then the Runge-Kutta method can be written as a one-step method:*

$$u_1 = u_0 + \tau \, \phi(t_0, u_0, \tau)$$

with

$$\phi(t_0, u_0, \tau) = \sum_{i=1}^{s} b_i \, f(t_0 + c_i \, \tau, U_i(t_0, u_0, \tau)).$$

The approximations u_j determine a grid function $u_\tau : I_\tau \longrightarrow \mathbb{R}^N$, given by $u_\tau(t_j) = u_j$. The set of all grid functions on the subdivision τ is denoted by X_τ. Notation: For a grid function $v_\tau \in X_j$ the value $v_\tau(t_j)$ will also be denoted by v_j.

The global error (discretization error) $e_\tau \in X_\tau$ is the grid function, given by

$$e_\tau(t_j)(= e_j) = u(t_j) - u_\tau(t_j) = u(t_j) - u_j.$$

The following norm is introduced on X_τ:

$$\|v_\tau\|_{X_\tau} = \max_{j=0,1,\ldots,m} \|v_j\|.$$

Definition 3.1. *A one-step method is called convergent, if*

$$\|e_\tau\|_{X_\tau} \to 0 \quad \textit{for } |\tau| \to 0. \quad (\textit{in short: } \|e_\tau\|_{X_\tau} = o(|\tau|)).$$

If there is a constant $C \geq 0$ such that

$$\|e_\tau\|_{X_\tau} \leq C\,|\tau|^p \quad (\textit{in short: } \|e_\tau\|_{X_\tau} = O(|\tau|^p)),$$

then the one-step method is called convergent of order p.

The one-step method can be written as

$$\psi_\tau(u_\tau) = 0$$

with the mapping $\psi_\tau : X_\tau \longrightarrow X_\tau$, given by

$$\psi_\tau(v)(t_{j+1}) = \frac{1}{\tau_j}(v(t_{j+1}) - v(t_j)) - \phi(t_j, v(t_j), \tau_j) \quad \text{for } j = 0, \ldots, m-1$$

and $\psi_\tau(v_\tau)(t_0) = v(t_0) - u_0$.

Definition 3.2. *1. Let u be the exact solution of the initial value problem* (3.1)*. Then the grid function $\psi_\tau(u)$ is called the consistency error (approximation error, local truncation error).*

2. A one-step method is called consistent with the initial-value problem at u, if

$$\|\psi_\tau(u)\|_{Y_\tau} \to 0 \quad \textit{for } |\tau| \to 0 \quad (\textit{in short: } \|\psi_\tau(u)\|_{Y_\tau} = o(|\tau|)).$$

with

$$\|y_\tau\|_{Y_\tau} = \|y_0\| + \sum_{j=0}^{m-1} \tau_j \|y_{j+1}\|$$

3. If a constant $C_A \geq 0$ exists such that

$$\|\psi_\tau(u)\|_{Y_\tau} \leq C_A\,|\tau|^p \quad (\textit{in short: } \|\psi_\tau(u)\|_{Y_\tau} = O(|\tau|^p)),$$

then the one-step method is called consistent of order p.

The consistency error is directly related to the local error d_τ, given by

$$d_\tau(u)(t_{j+1})(= d_{j+1}(u)) = u(t_{j+1}) - \big[u(t_j)) + \tau_j\, \phi(t_j, u(t_j), \tau_j)\big]$$

and $d_\tau(u)(t_0) = d_0(u) = u(t_0) - u_0 = 0$. Obviously:

$$\psi_{j+1}(u) = \frac{1}{\tau_j} d_{j+1}(u)$$

and

$$\|\psi_\tau(u)\|_{Y_\tau} = \sum_{j=0}^{m-1} \|d_{j+1}(u)\|.$$

Now we show Theorem 2.6 in the special case $r = 0$.

Theorem 3.1. *Let the assumptions of Theorem 2.5 be satisfied and $u \in C^{p+1}[0,T]$. If $B(p)$ and $C(q)$ hold for some $p \le q+1$, then the Runge-Kutta method is consistent with the initial value problem at least of order p.*

Proof. We have

$$\begin{aligned}
u(t_0 + c_i\tau) - U_i &= u(t_0 + c_i\tau) - u_0 - \tau \sum_j a_{ij}\, f(U_j) \\
&= u(t_0 + c_i\tau) - u_0 - \tau \sum_j a_{ij}\, f(u(t_0 + c_j\tau)) + \tau \sum_j a_{ij} \big[f(u(t_0 + c_j\tau)) - f(U_j)\big] \\
&= \int_{t_0}^{t_0 + c_i\tau} u'(t)\; dt - \tau \sum_j a_{ij}\, u'(t_0 + c_j\tau) + \tau \sum_j a_{ij} \big[f(u(t_0 + c_j\tau)) - f(U_j)\big].
\end{aligned}$$

By using the Lipschitz condition it follows that

$$\max_i \|u(t_0 + c_i\tau) - U_i\| \le \frac{1}{1 - \tau\|A\|_\infty L} \max_i \|d_{1,i}\|$$

with

$$d_{1,i} = \int_{t_0}^{t_0 + c_i\tau} u'(t)\; dt - \tau \sum_j a_{ij}\, u'(t_0 + c_j\tau).$$

From $C(q)$ one easily obtains estimates of the form

$$\|d_{1,i}\| \le c_A\, \tau^q \int_{t_0}^{t_0+\tau} \|u^{(q+1)}(t)\|\; dt$$

by a Taylor expansion.

Now we consider the local error:

$$\begin{aligned} d_1 &= u(t_0+\tau) - u_1 = u(t_0+\tau) - u_0 - \tau \sum_i b_i\, f(U_i) \\ &= u(t_0+\tau) - u_0 - \tau \sum_i b_i\, f(u(t_0+c_i\tau)) + \tau \sum_i b_i \left[f(u(t_0+c_i\tau)) - f(U_i) \right] \\ &= \int_{t_0}^{t_0+\tau} u'(t)\; dt - \tau \sum_i b_i\, u'(t_0+c_i\tau) + \tau \sum_i b_i \left[f(u(t_0+c_i\tau)) - f(U_i) \right] \end{aligned}$$

Then

$$\|d_1\| \le \|d_{1,s+1}\| + \frac{\tau\, \|b\|_1 L}{1-\tau\|A\|_\infty L} \max_i \|d_{1,i}\|$$

with

$$d_{1.s+1} = \int_{t_0}^{t_0+\tau} u'(t)\; dt - \tau \sum_i b_i\, u'(t_0+c_i\tau),$$

for which we have analogously to above an estimate of the form

$$\|d_{1,s+1}\| \le c_b\, \tau^p \int_{t_0}^{t_0+\tau} \|u^{(p+1)}(t)\|\; dt.$$

This leads to the following estimate of the local error

$$\|d_1\| \le c_b\, \tau^p \int_{t_0}^{t_0+\tau} \|u^{(p+1)}(t)\|\; dt + \frac{c_A\, \|b\|_1 L}{1-\tau\|A\|_\infty L}\, \tau^{q+1} \int_{t_0}^{t_0+\tau} \|u^{(q+1)}(t)\|\; dt$$

If this estimate is applied to each sub-interval $[t_j, t_j+\tau_j]$ with $q+1=p$, one obtains

$$\|\psi_\tau(u)\|_{Y_\tau} = \sum_{j=0}^{m-1} \|d_{j+1}(u)\| \le c_b\, \tau^p \int_{t_0}^{T} \|u^{(p+1)}(t)\|\; dt + \frac{c_A\, \|b\|_1 L}{1-\tau\|A\|_\infty L}\, \tau^p \int_{t_0}^{T} \|u^{(p)}(t)\|\; dt,$$

which completes the proof. □

If we compare the definition of the consistency error, written in the form

$$\frac{1}{\tau_j}\big(u(t_{j+1}) - u(t_j)\big) - \phi(t_j, u(t_j), \tau_j) = \psi_\tau(u)(t_{j+1}) \quad \text{for } j = 0, \ldots, m-1,$$

with the one-step method

$$\frac{1}{\tau_j}\big(u_{j+1} - u_j\big) - \phi(t_j, u_j, \tau_j) = 0 \quad \text{for } j = 0, \ldots, m-1,$$

we see that the exact solutions at the grid points result from the same one-step method perturbed by the consistency error on the right-hand side.

This leads to the more general question: How does the difference $v_\tau - u_\tau$ depend on the perturbation y_τ with $y_\tau(t_j) = y_j$, where the grid function v_τ is given by

$$\frac{1}{\tau_j}(v_{j+1} - v_j) - \phi(t_j, v_j, \tau_j) = y_{j+1}, \quad j = 0, 1, \ldots, m-1 \tag{3.2}$$

with initial value

$$v_0 = u_0 + y_0 \tag{3.3}$$

or, in short,

$$\psi_\tau(v_\tau) = y_\tau$$

Lemma 3.1. *If the increment function satisfies the Lipschitz condition*

$$\|\phi(t, w, \tau) - \phi(t, v, \tau)\| \leq \Lambda \|w - v\| \quad \textit{for all } t, v, w \textit{ and all } \tau,$$

then the following estimate is satisfied for (3.3), (3.2):

$$\|v_j - u_j\| \leq e^{\Lambda (t_j - t_0)} \|y_\tau\|_{Y_\tau} \quad \textit{for all } j = 0, 1, \ldots, m.$$

Proof. In a first step, we consider only the contribution of the perturbation y_0 to the difference $v_j - u_j$, which is $v_j^{(0)} - u_j$, where $v_\tau^{(0)}$ is given by the one-step method

$$v_{j+1}^{(0)} = v_j^{(0)} + \tau_j \phi(t_j, v_j^{(0)}, \tau_j), \quad j = 0, 1, \ldots, m-1$$

with

$$v_0^{(0)} = u_0 + y_0.$$

From

$$v_{j+1}^{(0)} - u_{j+1} = v_j^{(0)} - u_j + \tau_j [\phi(t_j, v_j^{(0)}, \tau_j) - \phi(t_j, u_j, \tau_j)]$$

and the Lipschitz condition it follows that

$$\|v_{j+1}^{(0)} - u_{j+1}\| \leq (1 + \Lambda \tau_j) \|v_j^{(0)} - u_j\| \leq e^{\Lambda \tau_j} \|v_j^{(0)} - u_j\|.$$

Hence

$$\|v_j^{(0)} - u_j\| \leq e^{\Lambda \tau_{j-1}} e^{\Lambda \tau_{j-2}} \cdots e^{\Lambda \tau_0} \|v_0^{(0)} - u_j\| = e^{\Lambda (t_j - t_0)} \|y_0\|.$$

Next we consider the contribution of the perturbation y_1 to the difference $v_j - u_j$, which is $v_j^{(1)} - v_j^{(0)}$, where $v_\tau^{(1)}$ is given by the one-step method

$$v_{j+1}^{(1)} = v_j^{(1)} + \tau_j \phi(t_j, v_j^{(1)}, \tau_j), \quad j = 1, \ldots, , m-1$$

with

$$v_1^{(1)} = v_1 = v_1^{(0)} + \tau_0 \, y_1.$$

It follows analogously

$$\|v_j^{(1)} - v_j^{(0)}\| \leq e^{\Lambda (t_j - t_1)} \|v_1^{(1)} - v_1^{(0)}\| = e^{\Lambda (t_j - t_1)} \tau_0 \|y_1\|.$$

In general, we obtain for the contribution of the perturbation y_i, $i = 1, \ldots, j$:

$$\|v_j^{(i)} - v_j^{(i-1)}\| \leq e^{\Lambda\,(t_j - t_i)} \|v_i^{(i)} - v_i^{(i-1)}\| = e^{\Lambda\,(t_j - t_i)} \tau_{i-1} \|y_i\|,$$

where $v_\tau^{(i)}$ is given by the one-step method

$$v_{j+1}^{(i)} = v_j^{(i)} + \tau_j \phi(t_j, v_j^{(i)}, \tau_j), \quad j = i, \ldots, m-1$$

with

$$v_i^{(i)} = v_i = v_i^{(i-1)} + \tau_{i-1}\, y_i.$$

Then, for the difference

$$v_j - u_j = (v_j^{(j)} - v_j^{(j-1)}) + (v_j^{(j-1)} - v_j^{(j-2)}) + \ldots (v_j^{(1)} - v_j^{(0)}) + (v_j^{(0)} - u_j)$$

we obtain the estimate

$$\begin{aligned}
\|v_j - u_j\| &\leq \|v_j^{(j)} - v_j^{(j-1)}\| + \|v_j^{(j-1)} - v_j^{(j-2)}\| + \ldots \|v_j^{(1)} - v_j^{(0)}\| + \|v_j^{(0)} - u_j\| \\
&\leq \left[e^{\Lambda(t_j - t_j)} \tau_{j-1} \|y_j\| + e^{\Lambda(t_j - t_{j-1})} \tau_{j-2} \|y_{j-1}\| + \ldots + e^{\Lambda(t_j - t_1)} \tau_0 \|y_1\| + e^{\Lambda(t_j - t_0)} \|y_0\| \right] \\
&\leq e^{\Lambda(t_j - t_0)} \|y_\tau\|_{Y_\tau}
\end{aligned}$$

□

Theorem 3.2. *If the increment function satisfies the Lipschitz condition*

$$\|\phi(t, w, \tau) - \phi(t, v, \tau)\| \leq \Lambda \|w - v\| \quad \textit{for all } t, v, w \textit{ and all } \tau,$$

then a constant $C \geq 0$ exists with

$$\|v_\tau - u_\tau\|_{X_\tau} \leq C \|\psi_\tau(v_\tau) - \psi_\tau(u_\tau)\|_{Y_\tau} \quad \textit{for all } v_\tau \in X_\tau \textit{ and all } \tau.$$

Proof. For $y_\tau = \psi_\tau(v_\tau)$ it follows from Lemma 3.1

$$\|v_\tau - u_\tau\|_{X_\tau} = \max_{j=0,1,\ldots,m} \|v_j - u_j\| \leq e^{\Lambda(T - t_0)} \|y_\tau\|_{Y_\tau}.$$

□

Definition 3.3. *A one-step method is called stable at u_τ if a constant $C_S \geq 0$ exists with*

$$\|v_\tau - u_\tau\|_{X_\tau} \leq C_S \|\psi_\tau(v_\tau) - \psi_\tau(u_\tau)\|_{X_\tau} \quad \textit{for all } v_\tau \in X_\tau \textit{ and all } \tau.$$

Theorem 3.3. *If a one-step method is consistent (of order p) at the exact solution u and stable at the approximate solution u_τ, then the method is convergent (of order p).*

Proof. The approximate solution u_τ satisfies

$$\psi_\tau(u_\tau) = 0.$$

From the stability it follows that

$$\|u - u_\tau\|_{X_\tau} \leq C_S \, \|\psi_\tau(u)\|_{Y_\tau}.$$

From the consistency it follows

$$\|\psi_\tau(u)\|_{Y_\tau} \to 0 \quad \text{for } |\tau| \to 0$$

and, therefore,

$$\|e_\tau\|_{X_\tau} \to 0 \quad \text{for } |\tau| \to 0.$$

From

$$\|\psi_\tau(u)\|_{Y_\tau} \leq C_A \, |\tau|^p$$

it follows

$$\|e_\tau\|_{X_\tau} \leq C_S \, C_A \, |\tau|^p.$$

□

Under the assumptions of Theorem 3.1 the stability also follows:

Theorem 3.4. *Let f be continuous and satisfies the Lipschitz condition*

$$\|f(t, w) - f(t, v)\| \leq L\|w - v\| \quad \textit{for all } t, v, w.$$

Then the Runge-Kutta method is stable.

Proof. Let $V = U(t, v, h)$ and $W = U(t, w, h)$. Then

$$\|\phi(t, w, \tau) - \phi(t, v, \tau)\| \leq \|b\|_1 \, L \, \|W - V\|.$$

and

$$\|W - V\| \leq \|w - v\| + \tau \, L \, \|A\|_\infty \, \|W - V\|,$$

hence

$$\|W - V\| \leq \frac{1}{1 - \tau \, L \, \|A\|_\infty} \|w - v\|.$$

This leads to

$$\|\phi(t, w, \tau) - \phi(t, v, \tau)\| \leq \frac{L \, \|b\|_1}{1 - \tau \, L \, \|A\|_\infty} \|w - v\|.$$

The rest follows from Theorem 3.2. □

Remark. *The results of this chapter can also be shown under the local Lipschitz condition*

$$\|f(t,v) - f(t,w)\| \leq L\|v - w\| \quad \textit{for all } (t,v),(t,w) \in \mathcal{U},$$

where $\mathcal{U} \subset I \times \mathbb{R}^N$ *is a neighborhood of the graph of* f*, given by* $\{(t, f(t,u(t))) : t \in I\}$*, and* $u(t)$ *denotes the exact solution of the initial value problem.*

For the local variant of Theorem 3.2 a local Lipschitz condition suffices:

$$\|\phi(t,v,\tau) - \phi(t,w,\tau)\| \leq \Lambda\|v - w\| \quad \textit{for all } (t,v),(t,w) \in \mathcal{U},\ \tau \leq \bar{\tau},$$

if it is additionally assumed that the method is consistent. The stability estimate can be shown for all $v_\tau \in X_\tau$ *with* $\|\psi_\tau(v_\tau)\|_{X_\tau} \leq \eta$*, if* η *is sufficiently small.*

Chapter 4

Practical Computation

The right choice of the step sizes is of great importance for the efficiency of a one-step method. The aim of a step size control is to achieve a prescribed tolerance of the local error. In the next section an approach is discussed how to approximately compute the local error. Subsequently, an automatic step size control is presented. Finally, the question is discussed how to efficiently calculate approximate solutions at prescribed points.

4.1 Embedded Runge-Kutta Methods

Consider a Runge-Kutta method of order p. Then, for the local error, we have:

$$u(t_0 + \tau) - u_1 = O(\tau^{p+1}),$$

The basic idea for estimating the local error is to use a second Runge-Kutta method of higher order q, leading to

$$u(t_0 + \tau) - \hat{u}_1 = O(\tau^{q+1}).$$

It follows that

$$u(t_0 + \tau) - u_1 = \hat{u}_1 - u_1 + O(\tau^{q+1}).$$

Since $q > p$, $\hat{u}_1 - u_1$ is a good approximation of the local error, which leads to the following measure of the local error:

$$err = max_{i=1,\ldots,n} \frac{|\hat{u}_{1,i} - u_{1,i}|}{d_i},$$

where d_i is an appropriate scaling factor. Typical values: $d_i = 1$ for absolute errors, $d_i = |\hat{u}_{2,i}|$ for component-wise relative errors.

In order to keep the extra computational costs low we assume that the two Runge-Kutta methods have the same coefficients c and A and differ only in the last row (b and $\hat{b}$). A pair of such methods (called embedded Runge-Kutta methods) are usually represented

by one tableau for the coefficients A, b, c with an extra row for $\hat{b}$. For an explicit method the tableau has the form:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \\ \hline & \hat{b}_1 & \hat{b}_2 & \dots & \hat{b}_{s-1} & \hat{b}_s \end{array}$$

The two approximate solutions are given by

$$u_1 = u_0 + \tau \left(b_1 k_1 + b_2 k_2 + \cdots + b_s k_s \right)$$

and

$$\hat{u}_1 = u_0 + \tau \left(\hat{b}_1 k_1 + \hat{b}_2 k_2 + \cdots + \hat{b}_s k_s \right).$$

Example. *We start with a general explicit 3-stage Runge-Kutta method:*

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \\ \hline & \hat{b}_1 & \hat{b}_2 & \hat{b}_s \end{array}$$

The conditions for order 2 for the first method are:

$$\begin{aligned} b_1 + b_2 + b_3 &= 1, \\ b_2 c_2 + b_3 c_3 &= \frac{1}{2}. \end{aligned}$$

The conditions for order 3 for the second method are:

$$\begin{aligned} \hat{b}_1 + \hat{b}_2 + \hat{b}_3 &= 1, \\ \hat{b}_2 c_2 + \hat{b}_3 c_3 &= \frac{1}{2}, \\ \hat{b}_2 c_2^2 + \hat{b}_3 c_3^2 &= \frac{1}{3}, \\ \hat{b}_3 a_{32} c_2 &= \frac{1}{6}. \end{aligned}$$

The choice

$$c_2 = 1, \quad c_3 = \frac{1}{2}, \quad b_3 = 0$$

leads to a so called Runge-Kutta-Fehlberg method abbreviated by RKF 2(3). The symbol 2(3) (in general p(q)) means that the basic method is of order 2 (p), the second method

used for estimating the local error is a method of order 3 (q). The tableau for RKF 2(3) is given by:

0			
1	1		
1/2	1/4	1/4	
	1/2	1/2	0
	1/6	1/6	4/6

The weights b_i correspond to the trapezoidal rule, the weights $\hat{b}_i$ to Simpson's rule.

Example. *Another important example of an embedded explicit Runge-Kutta method was constructed by Dormand and Prince, (in short: DOPRI (4)5), whose tableau is given by:*

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Observe that $a_{sj} = b_j$ which additionally reduces the computational work.

It is reasonable to continue the computation not with u_1 but with the more accurate approximation $\hat{u}_1$.

4.2 Step Size Control

We assume that an estimate *err* of the local error is available and that

$$err = C\,\tau^{p+1}.$$

The aim is to keep the local error within a given tolerance *tol*. This leads to an optimal step size τ_{new}, satisfying the relation

$$tol = C\,\tau_{new}^{p+1}.$$

From these two conditions the unknown constant C can be eliminated and one obtains:

$$\tau_{new} = \tau\,(tol/err)^{1/(p+1)}. \tag{4.1}$$

This motivates the following strategy for a step size selection:

1. One step with a given step size τ is computed together with the estimate err for the local error.

2. If $err \leq tol$, the step is accepted and the method is continued with the next step size τ_{new}, given by (4.1).

3. Otherwise, the step is rejected and the method is restarted with the new step size τ_{new}, given by (4.1).

In order to be sure that the new step size produces a local error below tol the optimal step size is reduced by a safety factor fac (e.g.: $fac = 0.8$). Additionally, it is reasonable to limit the change of the step size from τ to τ_{new} by factors $facmax$ for the maximal relative increase and $facmin$ for the maximal decrease, in order to prevent too dramatic changes of step sizes. With these modifications the new formula for the optimal step size becomes

$$\tau_{new} = \tau \cdot \min(facmax, \max(facmin, fac \cdot (tol/err)^{1/(p+1)})).$$

Additionally, it is advisable to set $facmax = 1$ right after a step rejection.

4.3 Dense Output

It is often required to compute the approximate solution on a set of prescribed points without interfering with the steps size control. This can be done by so called continuous Runge-Kutta methods: These methods contain a parameter $\theta \in (0, 1]$ and allow the computation of approximations for $u(t_0 + \theta\,\tau)$. For $\theta = 1$ the original Runge-Kutta method is obtained. For efficiency reasons we assume that the coefficients c and A are independent of θ. Only the coefficients of b are allowed to depend on θ. Approximate solutions at prescribed points can be computed without extra function evaluations and with no influence on the step size control.

Example. *An explicit 3-stage Runge-Kutta method for approximating $u(t_0 + \theta\,\tau)$ for all $\theta \in (0, 1]$ is of order 3 iff the following conditions are satisfied:*

$$\begin{aligned} b_1 + b_2 + b_3 &= \theta, \\ b_2 c_2 + b_3 c_3 &= \frac{\theta^2}{2}, \\ b_2 c_2^2 + b_3 c_3^2 &= \frac{\theta^3}{3}, \\ b_3 a_{32} c_2 &= \frac{\theta^3}{6}. \end{aligned}$$

This is not possible for coefficients c_2, c_3 and a_{32} independent of θ. Instead we require order 3 only for $\theta = 1$ and order 2, otherwise. This guarantees a global error of order 3,

also at intermediate points. For $c_2 = 1/2$ *and* $c_3 = 1$ *one obtains the following tableau of a continuous Runge-Kutta method:*

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\theta(1+\theta(-\frac{3}{2}+\frac{2}{3}\theta))$	$\theta^2(2-\frac{2}{3}\theta)$	$\theta^2(\frac{2}{3}-\frac{1}{2}\theta)$

Chapter 5

Multistep Methods

A method is called a multistep method (more precisely a k-step method) if the computation of the next approximate solution u_{j+1} is based on the last approximate solutions $u_j,\ u_{j-1}, \ldots, u_{j-k+1}$. If it is more convenient we will also use the notations u_{j+k} for the new approximate solution and $u_{j+k-1}, \ldots, u_j$ for the previously computed approximate solutions.

In order to perform a k-step method, a starting procedure has to be done first to compute $u_0, u_1, \ldots, u_{k-1}$. The starting procedure can be done, e.g., by using a one-step method (with small step sizes) or by a multistep method with a growing number of steps.

5.1 Classical Linear Multistep Methods

5.1.1 Explicit Adams Methods

We know that

$$u(t_{j+1}) = u(t_j) + \int_{t_j}^{t_{j+1}} f(t, u(t))\ dt \tag{5.1}$$

for a solution u of the ODE

$$u'(t) = f(t, u(t)).$$

The Runge-Kutta methods are based on quadrature rules whose nodes are typically inside the interval $[t_j, t_{j+1}]$ and, therefore, require function evaluations at additional points. If instead the grid points $t_j, t_{j-1}, \ldots, t_{j-k+1}$ are used as nodes for a quadrature rule, no additional function evaluations are required.

One possible strategy is to replace the function $f(t, u(t))$ in (5.1) by an interpolation polynomial. For simplicity we restrict ourselves to the case of equidistant step sizes: The interpolation polynomial of degree $k-1$ with the nodes

$$t_i, \quad i = j-k+1, \ldots, j-1, j,$$

and the values

$$f_i = f(t_i, u_i), \quad i = j-k+1, \ldots, j-1, j,$$

can be written in the following form (Newton's interpolation formula):

$$p(t) = p(t_j + s\,\tau) = \sum_{i=0}^{k-1} (-1)^i \binom{-s}{i} \nabla^i f_j,$$

with

$$\binom{-s}{0} = 1, \quad \binom{-s}{i} = \frac{(-s)(-s-1)\cdots(-s-i+1)}{i!} \quad \text{for } i \geq 1.$$

∇ denotes the backward difference:

$$\nabla f_j = f_j - f_{j-1},$$

whose powers ∇ are given by:

$$\nabla^0 = I, \quad \nabla^{i+1} = \nabla^i \nabla,$$

e.g.,

$$\nabla^2 f_j = \nabla f_j - \nabla f_{j-1} = f_j - 2f_{j-1} + f_{j-2}.$$

If $f(t, u(t))$ is replaced by $p(t)$ in (5.1), we obtain the following class of explicit multistep methods (explicit Adams methods, Adams-Bashforth methods):

$$u_{j+1} = u_j + \tau \sum_{i=0}^{k-1} \gamma_i \nabla^i f_j$$

with

$$\gamma_i = (-1)^i \int_0^1 \binom{-s}{i}\, ds.$$

The following table shows a few values of γ_i:

i	0	1	2	3	4	5	6	7	8
γ_i	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$	$\frac{5257}{17280}$	$\frac{1070017}{3628800}$

The first three Adams-Bashforth methods are:

$$\begin{aligned}
k = 1: &\quad u_{j+1} = u_j + \tau\, f_j, \\
k = 2: &\quad u_{j+1} = u_j + \tau \left[\frac{3}{2} f_j - \frac{1}{2} f_{j-1}\right], \\
k = 3: &\quad u_{j+1} = u_j + \tau \left[\frac{23}{12} f_j - \frac{16}{12} f_{j-1} + \frac{5}{12} f_{j-2}\right].
\end{aligned}$$

For $k = 1$ one obtains Euler's method.

5.1.2 Implicit Adams Methods

For this class of multistep methods, the node t_{j+1} is also used for the interpolation. Then the interpolation polynomial of degree k has the form:

$$p^*(t) = p^*(t_j + s\,\tau) = p(t_{j+1} + (s-1)\,\tau) = \sum_{i=0}^{k} (-1)^i \binom{-s+1}{i} \nabla^i f_{j+1}.$$

The corresponding quadrature is given by:

$$u_{j+1} = u_j + \tau \sum_{i=0}^{k} \gamma_i^* \nabla^i f_{j+1} \tag{5.2}$$

with

$$\gamma_i^* = (-1)^i \int_0^1 \binom{-s+1}{i}\, ds.$$

These methods are implicit and require the solution of a (in general nonlinear) system of equations, in order to compute u_{j+1}. For sufficiently small step sizes the solution u_{j+1} exists and the method is well-defined. An approximation for u_{j+1} can be obtained, e.g., by a fixed point iteration for (5.2) or by Newton's method. As an initial guess one can use u_j or the result of one step of the corresponding explicit Adams method. Often, it suffices to perform one step of the explicit Adams method (predictor) followed by one step on an iterative method for the implicit Adams method (corrector).

The following table contains a few values for γ_i^*:

i	0	1	2	3	4	5	6	7	8
γ_i	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$	$-\frac{275}{24192}$	$-\frac{33953}{3628800}$

The first three Adams-Moulton methods are:

$$\begin{aligned}
k = 0: &\quad u_{j+1} = u_j + \tau\, f_{j+1}, \\
k = 1: &\quad u_{j+1} = u_j + \tau \left[\frac{1}{2} f_{j+1} + \frac{1}{2} f_j\right], \\
k = 2: &\quad u_{j+1} = u_j + \tau \left[\frac{5}{12} f_{j+1} + \frac{8}{12} f_j - \frac{1}{12} f_{j-1}\right].
\end{aligned}$$

For $k = 0$ one obtains the implicit Euler method, for $k = 1$ the implicit trapezoidal rule.

5.1.3 BDF-Methods

This class of methods is based on numerical differentiation: The interpolation polynomial q of degree k with nodes

$$t_i, \quad i = j-k+1, \ldots, j, j+1,$$

and values

$$u_i \quad i = j-k+1, \ldots, j, j+1,$$

has the following form:

$$q(t) = q(t_j + s\,\tau) = \sum_{i=0}^{k} (-1)^i \binom{-s+1}{i} \nabla^i u_{j+1}.$$

The differential equation

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

at $t = t_{j+1}$ is replaced by

$$q'(t_{j+1}) = f_{j+1}.$$

This leads to a multistep method:

$$\sum_{i=0}^{k} \delta_i^* \nabla_i u_{j+1} = \tau\, f_{j+1}$$

with

$$\delta_i^* = (-1)^i \, \frac{d}{ds} \binom{-s+1}{i} \bigg|_{s=1}.$$

This method is called backward differencing formula (BDF-method).

The values δ_i^* can be easily calculated:

$$\delta_0^* = 0, \qquad \delta_i^* = \frac{1}{i} \quad \text{for } i \geq 1.$$

The first three BDF-methods are:

$$\begin{aligned}
k = 1: &\quad u_{j+1} - u_j = \tau\, f_{j+1} \\
k = 2: &\quad \frac{3}{2} u_{j+1} - 2u_j + \frac{1}{2} u_{j-1} = \tau\, f_{j+1} \\
k = 3: &\quad \frac{11}{6} u_{j+1} - 3u_j + \frac{3}{2} u_{j-1} - \frac{1}{3} u_{j-2} = \tau\, f_{j+1}
\end{aligned}$$

These methods are implicit.

5.2 Consistency of Linear Multistep Methods

All multistep methods discussed so far are of the following form:

$$\alpha_k u_{j+k} + \alpha_{k-1} u_{j+k-1} + \cdots + \alpha_0 u_j = \tau\,(\beta_k f_{j+k} + \beta_{k-1} f_{j+k-1} + \cdots + \beta_0 f_j). \tag{5.3}$$

A multistep method of this form is called a linear multistep method, more precisely, a linear k-step method.

The coefficient α_i and β_i are not uniquely determined by the method. They allow an additional scaling condition, e.g., $\alpha_k = 1$ or $\sum_{i=0}^{k} \beta_i = 1$.

Let (t, u) be given and let $u(s)$ be the exact solution of the differential equation with $u(t) = u$. The local error of a multistep method is the difference between exact solution and approximate solution:

$$u(t + k\,\tau) - u_\tau(t + k\,\tau),$$

where it is assumed that the starting procedure is exact, i.e.:

$$u(t), u(t + \tau), \ldots, u(t + (k - 1)\,\tau)$$

are the initial settings for the computation of $u_\tau(t + k\,\tau)$. The method is called consistent of order p, if

$$u(t + k\,\tau) - u_\tau(t + k\,\tau) = O(\tau^{p+1}).$$

Assume the scaling condition $\sum_{i=0}^{k} \beta_i = 1$. The consistency error is given by

$$\begin{aligned}\psi_\tau(u)(t + k\,\tau) &= \frac{1}{\tau} \sum_{i=0}^{k} \alpha_i u(t + i\,\tau) - \sum_{i=0}^{k} \beta_i f(t + i\,\tau, u(t + i\,\tau)) \\ &= \frac{1}{\tau} \sum_{i=0}^{k} \alpha_i u(t + i\,\tau) - \sum_{i=0}^{k} \beta_i u'(t + i\,\tau).\end{aligned}$$

We have the following connection between the local error and the consistency error:

Lemma 5.1. *Let f be a continuously differentiable function. Then*

$$u(t + k\,\tau) - u_\tau(t + k\,\tau) = \tau\ [\alpha_k I - \tau\,\beta_k J]^{-1}\ \psi_\tau(u)(t + k\,\tau)$$

with

$$J = \begin{pmatrix} \dfrac{\partial f_1}{\partial u}(t + k\,\tau, \nu_1) \\ \dfrac{\partial f_2}{\partial u}(t + k\,\tau, \nu_2) \\ \vdots \\ \dfrac{\partial f_n}{\partial u}(t + k\,\tau, \nu_n) \end{pmatrix}$$

and $\nu_i = u_\tau(t + k\,\tau) + \delta_i\,[u(t + k\,\tau) - u_\tau(t + k\,\tau)]$ for some $\delta_i \in [0, 1]$.

Proof. For one step of the method we obtain

$$\alpha_k u_\tau(t + k\,\tau) - \tau\,\beta_k f(t + k\,\tau, u_\tau(t + k\,\tau)) + \sum_{i=0}^{k-1} [\alpha_i u(t + i\,\tau) - \tau\beta_i f(t + i\,\tau, u(t + i\,\tau))] = 0.$$

By definition of the consistency it follows that

$$\begin{aligned}\tau\,\psi_\tau(u)(t+k\,\tau) &= \alpha_k\,u(t+k\,\tau) - \tau\,\beta_k\,f(t+k\,\tau,u(t+k\,\tau))\\ &\qquad - \alpha_k u_\tau(t+k\,\tau) + \tau\,\beta_k f(t+k\,\tau,u_\tau(t+k\,\tau))\\ &= \alpha_k[u(t+k\,\tau) - u_\tau(t+k\,\tau)]\\ &\qquad - \tau\,\beta_k[f(t+k\,\tau,u(t+k\,\tau)) - f(t+k\,\tau,u_\tau(t+k\,\tau))].\end{aligned}$$

From the mean value theorem it follows that

$$f(t+k\,\tau,u(t+k\,\tau)) - f(t+k\,\tau,u_\tau(t+k\,\tau)) = J[u(t+k\,\tau) - u_\tau(t+k\,\tau)],$$

which implies

$$\tau\,\psi_\tau(u)(t+k\,\tau) = [\alpha_k I - \tau\,\beta_k J](u(t+k\,\tau) - u_\tau(t+k\,\tau)).$$

□

For explicit methods, i.e., $\beta_k = 0$, the relation simplifies to $u(t+k\,\tau) - u_\tau(t+k\,\tau) = (\tau/\alpha_k)\,\psi_\tau(u)(t+k\,\tau)$.

Remark. *A multistep method can be written in the following form:*

$$\psi_\tau(u_\tau) = 0$$

with

$$\psi_\tau(v_\tau)(t+k\,\tau) = \frac{1}{\tau}\sum_{i=0}^{k}\alpha_i v_\tau(t+i\,\tau) - \sum_{i=0}^{k}\beta_i f(t+i\,\tau,v_\tau(t+i\,\tau))$$

and (assuming an ideal starting procedure) $\psi_\tau(v_\tau)(t_0+i\,\tau) = v_\tau(t_0+i\,\tau) - u(t+i\,\tau)$ *for* $i = 0,1,\ldots,k-1$. *As for one-step methods we call* $\psi_\tau(u)$ *the consistency error, where* u *is the exact solution.*

Next the so-called generating polynomials of the multistep method are introduced by

$$\begin{aligned}\rho(z) &= \alpha_k z^k + \alpha_{k-1}z^{k-1} + \cdots + \alpha_0,\\ \sigma(z) &= \beta_k z^k + \beta_{k-1}z^{k-1} + \cdots + \beta_0.\end{aligned}$$

We have

Theorem 5.1. *Let* f *be sufficiently smooth. A linear multistep method of the form* (5.3) *is consistent of order* p, *if*

$$\sum_{i=0}^{k}\alpha_i = 0 \quad \textit{and} \quad \sum_{i=0}^{k}\alpha_i i^q = q\sum_{i=0}^{k}\beta_i i^{q-1} \quad \textit{for } q = 1,2,\ldots,p.$$

Proof. We obtain by Taylor expansion:

$$\begin{aligned}\tau\,\psi_\tau(u)(t+k\,\tau) &= \sum_{i=0}^{k}\left[\alpha_i\sum_{q=0}^{p}\frac{i^q}{q!}u^{(q)}(t)\tau^q-\beta_i\tau\sum_{r=0}^{p-1}\frac{i^r}{r!}u^{(r+1)}(t)\tau^r\right]+O(\tau^{p+1})\\ &= \left[\sum_{i=0}^{k}\alpha_i\right]u(t)+\sum_{q=1}^{p}\frac{\tau^q}{q!}\left[\sum_{i=0}^{k}\alpha_i i^q-q\sum_{i=0}^{k}\beta_i i^{q-1}\right]u^{(q)}(t)+O(\tau^{p+1})\\ &= O(\tau^{p+1}).\end{aligned}$$

□

For the case $p=1$ the conditions can be written in the form:

$$\rho(1)=0,\quad \rho'(1)=\sigma(1).$$

Remark. *It is easy to show that the Adams-Bashforth methods and the BDF-methods are of order $p=k$, the Adams-Moulton methods are of order $p=k+1$. These classes of multistep methods are exact for the ODEs*

$$u'(t)=qt^{q-1}$$

with $q=0,1,\ldots,p$. Hence, with the exact solution $u(t)=t^q$, we obtain for the consistency error at $t=0$:

$$0=\tau\,\psi_\tau(u)(0+k\,\tau)=\begin{cases}\sum_{i=0}^{k}\alpha_i & \text{for } q=0,\\ \tau^q\left[\sum_{i=0}^{k}\alpha_i i^q-q\sum_{i=0}^{k}\beta_i i^{q-1}\right] & \text{for } q=1,\ldots,p,\end{cases}$$

which implies the corresponding order.

Remark. *The highest attainable order of a k-step method is $2k$.*

5.3 Stability of linear Multistep Methods

For Runge-Kutta methods a Lipschitz condition on f with respect to u is sufficient for stability. For multistep method the situation is more delicate.

Example. *The explicit 2-step method of maximum order 3 is given by:*

$$u_{j+2}+4u_{j+1}-5u_j=\tau\,(4f_{j+1}+2f_j).$$

If applied to the initial value problem

$$u'=u,\quad u(0)=1$$

with exact starting procedure $u_0=1$ and $u_1=e^\tau$, the method leads to completely useless results.

We start the discussion of stability for the trivial right-hand side $f = 0$, i.e., for the differential equation

$$u'(t) = 0.$$

Then the method is of the following form:

$$\alpha_k u_{j+k} + \alpha_{k-1} u_{j+k-1} + \cdots + \alpha_0 u_j = 0. \tag{5.4}$$

Theorem 5.2. *Let $\zeta_1, \zeta_2, \ldots, \zeta_l$ be the roots of ρ with multiplicity $m_1, m_2, \ldots, m_l$. Then the general solution of (5.4) is given by*

$$u_j = p_1(j)\,\zeta_1^j + p_2(j)\,\zeta_2^j + \cdots + p_l(j)\,\zeta_l^j,$$

where p_i are arbitrary polynomials of degree $\leq m_i - 1$.

Proof. The general solution is obtained as linear combination of the $m_1 + m_2 + \ldots + m_l = k$ particular solutions

$$u_j = \binom{j}{\mu} \zeta^j,$$

where ζ is a root of ρ with multiplicity m and $\mu \leq m - 1$. In order to verify that these sequences solve the recurrence relation (5.4), the identity

$$\binom{j+i}{\mu} = \sum_{\nu=0}^{\mu} \binom{j}{\mu-\nu} \binom{i}{\nu}$$

is used. Then we obtain

$$\begin{aligned}
\sum_{i=0}^{k} \alpha_i u_{j+i} &= \sum_{i=0}^{k} \alpha_i \binom{j+i}{\mu} \zeta^{j+i} = \zeta^j \sum_{\nu=0}^{\mu} \binom{j}{\mu-\nu} \sum_{i=0}^{k} \alpha_i \binom{i}{\nu} \zeta^i \\
&= \zeta^j \sum_{\nu=0}^{\mu} \binom{j}{\mu-\nu} \frac{\zeta^\nu}{\nu!} \underbrace{\sum_{i=0}^{k} \alpha_i\, i(i-1)\cdots(i-\nu+1)\zeta^{i-\nu}}_{= \rho^{(\nu)}(\zeta) = 0} = 0.
\end{aligned}$$

□

The k coefficients of the polynomials $p_1(j)$, $p_2(j)$, …, $p_l(j)$ are uniquely determined by prescribing the k values u_0, u_1, …, u_{k-1} of the starting phase. It immediately follows from the last theorem that the sequence $(u_j)_{j \in \mathbb{N}_0}$, generated by the linear multistep method, is bounded for arbitrary initial phase if and only if the roots ζ of ρ satisfy the following condition:

$$|\zeta| \leq 1 \quad \text{and} \quad |\zeta| = 1 \quad \text{only if } \zeta \text{ is simple.} \tag{5.5}$$

This leads to the following definition:

Definition 5.1. *The multistep method* (5.3) *is called 0–stable, if all roots* ζ *of* ρ *satisfy the condition* (5.5).

For the explicit and the implicit Adams methods we have:

$$\rho(z) = z^k - z^{k-1}.$$

0 is a root of multiplicity $(k-1)$, 1 is a simple root. Hence, the methods are 0–stable.

The analysis of the 0–stability of the BDF-methods is more difficult. It can be shown that these methods are 0-stable for $k \leq 6$ and not 0–stable for $k \geq 7$.

Theorem 5.3 (The first Dahlquist barrier)**.** *The order* p *of a 0–stable* k*-step method satisfies:*

$$p \leq \begin{cases} k+2 & \text{if } k \text{ is odd,} \\ k+1 & \text{if } k \text{ is even,} \\ k & \text{if } \beta_k/\alpha_k \leq 0. \end{cases}$$

5.4 Convergence of Linear Multistep Methods

A linear k-step method leads to a fixed point equation for u_{j+k}:

$$u_{j+k} = -\sum_{i=0}^{k-1} \alpha_i' u_{j+j} + \tau\, \beta_k'\, f(t_j + k\,\tau, u_{j+k}) + \tau \sum_{i=0}^{k-1} \beta_i', f_{j+i}, \tag{5.6}$$

where

$$\alpha_i' = \frac{\alpha_i}{\alpha_k}, \quad \beta_i' = \frac{\beta_i}{\alpha_k}.$$

Let f be continuous and satisfies the Lipschitz condition

$$\|f(t,w) - f(t,v)\| \leq L\,\|w - v\| \quad \text{for all } t, v, w.$$

Then there is a unique solution $u_{j+k} = \eta(t_j, u_{j+k-1}, u_{j+k-2}, \ldots, u_j, \tau)$ of (5.6) for arbitrary values t_j, $u_{j+k-1}, u_{j+k-2}, \ldots, u_j$ and sufficiently small τ. (For explicit k-step methods we have $\beta_k' = 0$, and $\eta(t_j, u_{j+k-1}, u_{j+k-2}, \ldots, u_j, \tau)$ is well-defined anyway.)

With

$$\begin{aligned} &\phi(t_j, u_{j+k-1}, u_{j+k-2}, \ldots, u_j, \tau) \\ &\qquad = \beta_k'\, f(t_j + k\,\tau, \eta(t_j, u_{j+k-1}, u_{j+k-2}, \ldots, u_j, \tau)) + \sum_{i=0}^{k-1} \beta_i'\, f_{j+i} \end{aligned}$$

the multistep method can be written in the form

$$u_{j+k} = -\sum_{i=0}^{k-1} \alpha_i'\, u_{j+i} + \tau\, \phi(t_j, u_{j+k-1}, u_{j+k-2}, \ldots, u_j, \tau). \tag{5.7}$$

The multistep method can also be written as a one-step method. Let

$$U_j = \begin{pmatrix} u_{j+k-1} \\ u_{j+k-2} \\ \vdots \\ u_j \end{pmatrix}, \quad A = \begin{pmatrix} -\alpha'_{k-1} & -\alpha'_{k-2} & \dots & -\alpha'_0 \\ 1 & 0 & \dots & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{pmatrix}, \quad e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Then (5.7) is equivalent to

$$U_{j+1} = (A \otimes I)\, U_j + \tau\, \Phi(t_j, U_j, \tau) \tag{5.8}$$

with

$$\Phi(t, U, \tau) = (e_1 \otimes I)\, \phi(t, U, \tau).$$

Here, $C \otimes D$ denotes the Kronecker product (tensor product) of two matrices:

$$C \otimes D = \begin{pmatrix} c_{11} D & c_{12} D & \cdots & c_{1n} D \\ c_{21} D & c_{22} D & \cdots & c_{2n} D \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} D & c_{m2} D & \cdots & c_{mn} D \end{pmatrix}.$$

Remark. *We have*

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

for arbitrary matrices A, B, C, D with suitable dimensions.

Let $u(t)$ be the exact solution of the initial value problem and set

$$U(t) = \begin{pmatrix} u(t + (k-1)\,\tau) \\ u(t + (k-2)\,\tau) \\ \vdots \\ u(t) \end{pmatrix}.$$

The approximate solution at $t+\tau$, which is obtained by (5.8) starting with $U(t)$ is denoted by $U_\tau(t+\tau)$. Then the first component of

$$U(t+\tau) - U_\tau(t+\tau)$$

is the local error of the multistep method, the other components all vanish. If the multistep method is consistent of order p it follows

$$\|U(t+\tau) - U_\tau(t+\tau)\| = O(\tau^{p+1}).$$

Lemma 5.2. *Assume that the multistep method is 0-stable. Then there is a vector norm on $(\mathbb{R}^N)^k$, such that*

$$\|A \otimes I\| \le 1$$

for the corresponding matrix norm.

Proof. The roots ζ of ρ are the eigenvalue of A with eigenvector $(\zeta^{k-1}, \zeta^{k-2}, \dots, \zeta, 1)^T$. Therefore, there is a transformation matrix T with

$$T^{-1}AT = \left(\begin{array}{ccc|ccccc} \zeta_1 & & & & & & & \\ & \ddots & & & & & & \\ & & \zeta_l & & & & & \\ \hline & & & \zeta_{l+1} & \delta_l & & & \\ & & & & \ddots & \ddots & & \\ & & & & & \ddots & \delta_{k-1} & \\ & & & & & & \zeta_k & \end{array}\right)$$

with $|\zeta_i| = 1$ for $i = 1, \dots, l$ and $|\zeta_i| < 1$, $\delta_i \in \{0, 1\}$ for $i = l+1, \dots, k$. Then

$$T_\varepsilon^{-1}AT_\varepsilon = \left(\begin{array}{ccc|ccccc} \zeta_1 & & & & & & & \\ & \ddots & & & & & & \\ & & \zeta_l & & & & & \\ \hline & & & \zeta_{l+1} & \varepsilon\,\delta_l & & & \\ & & & & \ddots & \ddots & & \\ & & & & & \ddots & \varepsilon\,\delta_{k-1} & \\ & & & & & & \zeta_k & \end{array}\right) = J$$

with $T_\varepsilon = TD_\varepsilon$ and the diagonal matrix $D_\epsilon = \operatorname{diag}(1, \varepsilon, \varepsilon^2, \dots, \varepsilon^{k-1})$. We choose $\varepsilon > 0$ sufficiently small, such that $|\varepsilon\delta_{i-1}| < 1 - |\zeta_i|$ for all $i = l+1, \dots, k$.

We define the following norm in $(\mathbb{R}^N)^k$: $\|x\| = \|(T_\varepsilon^{-1} \otimes I)x\|_\infty$. Then

$$\begin{aligned} \|(A \otimes I)x\| &= \|(T_\varepsilon^{-1} \otimes I)(A \otimes I)x\|_\infty = \|((T_\varepsilon^{-1}A) \otimes I)x\|_\infty \\ &= \|((JT_\varepsilon^{-1}) \otimes I)x\|_\infty = \|(J \otimes I)(T_\varepsilon^{-1} \otimes I)x\|_\infty \\ &\le \underbrace{\|(J \otimes I)\|_\infty}_{\le 1} \underbrace{\|(T_\varepsilon^{-1} \otimes I)x\|_\infty}_{= \|x\|} \end{aligned}$$

□

Remark. *A linear multistep method is 0–stable if and only if there exists a constant C such that*

$$\|A^j\| \le C \quad \textit{for all } j \in \mathbb{N}.$$

Theorem 5.4. *Assume that f satisfies the Lipschitz condition*

$$\|f(t, w) - f(t, v)\| \le L\,\|w - v\| \quad \textit{for all } t, v, w.$$

If the linear multistep method is consistent of order p and 0-stable, then the method is convergent of order p.

Proof. From the Lipschitz condition for f it easily follows a Lipschitz condition for Φ:

$$\|\Phi(t, W, \tau) - \Phi(t, V, \tau)\| \leq \Lambda \|W - V\| \quad \text{for all } t, V, W \text{ and for sufficiently small } \tau.$$

Together with Lemma 5.2 stability follows. The rest is obtained analogously to the convergence proof of one-step methods. □

Part II

Stiff Problems

Example. *Consider the problem*

$$u'(t) = -50\,u(t),$$
$$u(0) = 1.$$

Exact solution

$$u(t) = e^{-50\,t}.$$

The explicit Euler method produces reasonable approximation for $\tau = 1/25$ *but completely wrong approximations for the slightly larger step size* $\tau = 1/24$*. On the other hand, the implicit Euler method works fine for all step sizes.*

The Lipschitz constant for this problem is given by $L = 50$*. So we would expect a stability constant*

$$C_S = e^{L\,t},$$

which is of the order 10^{21}*. This does not explain why the explicit Euler method works for* $\tau = 1/25$ *nor why the implicit Euler method works.*

Chapter 6

One-Sided Lipschitz Conditions

Consider the differential equation

$$u'(t) = f(t, u(t)). \tag{6.1}$$

A Lipschitz condition

$$\|f(t, w) - f(t, v)\| \leq L \|w - v\|$$

implies the following stability estimation for two solutions v and w of (6.1):

$$\|w(t) - v(t)\| \leq e^{L(t-t_0)} \|w(t_0) - v(t_0)\| \quad \text{for all } t \geq t_0.$$

Consider the one-step method

$$u_{j+1} = u_j + \tau_j \, \phi(t_j, u_j, \tau_j). \tag{6.2}$$

A Lipschitz condition

$$\|\phi(t, w, \tau) - \phi(t, v, \tau)\| \leq \Lambda \|w - v\|$$

implies the following stability estimation for two sequences (v_j) and (w_j) produced by (6.2):

$$\|w_j - v_j\| \leq e^{\Lambda(t_j - t_0)} \|w_0 - v_0\| \quad \text{for all } j \geq t_0.$$

Observe that a linear multistep method can be written in the form

$$U_{j+1} = (A \otimes I) \, U_j + \tau \, \Phi(t_j, U_j, \tau_j). \tag{6.3}$$

A Lipschitz condition

$$\|\Phi(t, W, \tau) - \Phi(t, V, \tau)\| \leq \Lambda \|W - V\|$$

and 0-stability imply the following stability estimation for two sequences (V_j) and (W_j) produced by (6.3):

$$\|W_j - V_j\| \leq e^{\Lambda(t_j - t_0)} \|W_0 - V_0\| \quad \text{for all } j \geq t_0.$$

Let (v, w) denote an inner product in $\mathbb{R}^N$ with corresponding norm $\|v\| = \sqrt{(v, v)}$. For example: the Euclidean inner product is given by

$$(v, w)_{\ell^2} = v^T w.$$

One-sided Lipschitz condition: There exists a constant ν such that

$$(f(t, w) - f(t, v), w - v) \leq \nu \|w - v\|^2 \quad \text{for all } t, v, w.$$

Of course, a Lipschitz condition with constant L implies a one-sided Lipschitz constant with constant $\nu = L$. However, sometimes ν is considerably smaller, or even negative.

Example. *For $f(t, u) = -50\, u$ we have $L = 50$ but $\nu = -50$, since:*

$$((-50\, w) - (-50\, v)) \cdot (w - v) = -50\, (w - v)^2.$$

Definition 6.1. *A differential equation*

$$u' = f(t, u) \tag{6.4}$$

is dissipative if

$$(f(t, w) - f(t, v), w - v) \leq 0 \quad \textit{for all } t, v, w.$$

Example. *If the heat equation*

$$\begin{aligned} u_t(x, t) - a\, u_{xx}(x, t) &= f(x, t), \quad x \in (0, 1),\ t > 0, \\ u(0, t) = u(1, t) &= 0, \\ u(x, 0) &= u_0(x) \end{aligned}$$

is discretized with respect to x by a standard finite element method with a constant step size h, we obtain:

$$M_h\, \underline{u}_h'(t) + K_h\, \underline{u}_h(t) = \underline{f}_h(t)$$

with the mass matrix M_h and the stiffness matrix K_h, or in standard form

$$\underline{u}_h'(t) = f(t, \underline{u}_h(t))$$

with

$$f(t, \underline{u}_h) = J\, \underline{u}_h + M_h^{-1} \underline{f}_h(t) \quad \textit{with } J = -M_h^{-1} K_h.$$

M_h and K_h are symmetric and positive definite. Hence, the eigenvalues of $J = -M_h^{-1} K_h$ are real and negative ranging in modulus from $\lambda_{min}(M_h^{-1} K_h) = O(1)$ to $\lambda_{max}(M_h^{-1} K_h) = O(h^{-2})$. Moreover,

$$(f(t, \underline{w}_h) - f(t, \underline{v}_h), \underline{w}_h - \underline{v}_h)_{M_h} = -\,(K_h(\underline{w}_h - \underline{v}_h), \underline{w}_h - \underline{v}_h)_{\ell^2} \leq 0,$$

where $(\underline{w}_h, \underline{v}_h)_{M_h} = (M_h \underline{w}_h, \underline{v}_h)_{\ell^2}$. Therefore, $\nu = 0$, while $L = \|M_h^{-1} K_h\|_{M_h}$ $= \lambda_{max}(M_h^{-1} K_h) = O(h^{-2})$.

Example. *If the wave equation*

$$\begin{aligned} u_{tt}(x,t) - c^2\, u_{xx}(x,t) &= f(x,t), \quad x \in (0,1),\ t > 0, \\ u(0,t) = u(1,t) &= 0, \\ u(x,0) &= u_0(x) \\ u_t(x,0) &= v_0(x) \end{aligned}$$

is discretized with respect to x by a standard finite element method with a constant step size h we obtain:

$$M_h \underline{u}_h''(t) + K_h\, \underline{u}_h(t) = \underline{f}_h(t),$$

or in standard form

$$\begin{bmatrix} \underline{u}_h'(t) \\ \underline{v}_h'(t) \end{bmatrix} = J \begin{bmatrix} \underline{u}_h(t) \\ \underline{v}_h(t) \end{bmatrix} + \begin{bmatrix} 0 \\ M_h^{-1} \underline{f}_h(t) \end{bmatrix} \quad \textit{with } J = \begin{bmatrix} 0 & I \\ -M_h^{-1} K_h & 0 \end{bmatrix}.$$

M_h and K_h is symmetric and positive definite. The eigenvalue value problem for J reads

$$\begin{bmatrix} 0 & I \\ -M_h^{-1} K_h & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix},$$

i.e.,

$$\begin{aligned} v &= \lambda\, u, \\ -M_h^{-1} K_h u &= \lambda\, v, \end{aligned}$$

or, equivalently,

$$M_h^{-1} K_h u = -\lambda^2\, u.$$

So, $\lambda = \pm\, i\sqrt{\mu}$, where μ is an eigenvalue of $M_h^{-1} K_h$. Hence, the eigenvalues of J are purely imaginary ranging in modulus from $O(1)$ to $O(h^{-1})$. Moreover, for the inner product, given by

$$\left(\begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \right) = (K_h u_1, u_2)_{\ell^2} + (M_h v_1, v_2)_{\ell^2}\,, = (u_1, u_2)_{K_h} + (v_1, v_2)_{M_h}\,,$$

we have

$$\begin{aligned} &\left(J \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{f}_h(t) \end{bmatrix} - J \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} - \begin{bmatrix} 0 \\ \underline{f}_h(t) \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} - \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \right) \\ &= \left(J \begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \end{bmatrix}, \begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \end{bmatrix} \right) = \left(\begin{bmatrix} v_2 - v_1 \\ -M_h^{-1} K_h (u_2 - u_1) \end{bmatrix}, \begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \end{bmatrix} \right) \\ &= (K_h(v_2 - v_1), u_2 - u_1)_{\ell^2} - (K_h(u_2 - u_1), v_2 - v_1)_{\ell^2} = 0. \end{aligned}$$

Therefore, $\nu = 0$, while $L = \|J\| = \rho(J) = O(h^{-1})$.

Lemma 6.1. *Let f be continuous and satisfy the one-sided Lipschitz condition*

$$(f(t,w) - f(t,v), w - v) \leq \nu \|w - v\|^2 \quad \textit{for all } t, v, w.$$

Then, for any two solutions $v(t)$ and $w(t)$ of (6.4), we have

$$\|w(t) - v(t)\| \leq e^{\nu (t-t_0)} \|w(t_0) - v(t_0)\| \quad \textit{for all } t \geq t_0.$$

Proof. For $m(t) = \|w(t) - v(t)\|^2 = (w(t) - v(t), w(t) - v(t))$ we have

$$\begin{aligned} m'(t) &= 2\,(w'(t) - v'(t), w(t) - v(t)) = 2\,(f(t,w(t)) - f(t,v(t)), w(t) - v(t)) \\ &\leq 2\nu \|w(t) - v(t)\|^2 = 2\nu\, m(t). \end{aligned}$$

Hence

$$(\ln m(t))' = \frac{m'(t)}{m(t)} \leq 2\nu.$$

By integrating we finally obtained

$$\ln m(t) - \ln m(t_0) \leq 2\nu(t - t_0),$$

i.e.:

$$m(t) \leq e^{2\nu (t-t_0)}\, m(t_0).$$

□

For dissipative problems we have

$$\|w(t) - v(t)\| \leq \|w(t_0) - v(t_0)\| \quad \text{for all } t \geq t_0.$$

Therefore, it is desirable to have an analogous property for one-step and multistep methods. This leads to the following definition:

Definition 6.2. *A one-step method is contractive if*

$$\|w_{j+1} - v_{j+1}\| \leq \|w_j - v_j\| \quad \textit{for all } v_j,\ w_j.$$

A multistep method is contractive if

$$\|W_{j+1} - V_{j+1}\| \leq \|W_j - V_j\| \quad \textit{for all } V_j,\ W_j.$$

Chapter 7

A-Stability

Linearization

Consider the differential equation

$$u'(t) = f(t, u(t)).$$

Let $\phi(t)$ be a given function, which is assumed to be a good approximation of the exact solution. Then

$$u(t) = \phi(t) + \bar{u}(t)$$

with a function $\bar{u}(t)$ with small values. Therefore,

$$\begin{aligned} u'(t) &= f(t, \phi(t) + \bar{u}(t)) = f(t, \phi(t)) + f_u(t, \phi(t))\, \bar{u}(t) + \dots \\ &= f(t, \phi(t)) + f_u(t, \phi(t))\, (u(t) - \phi(t)) + \dots \\ &= f_u(t, \phi(t))\, u(t) + f(t, \phi(t)) - f_u(t, \phi(t))\phi(t) + \dots \end{aligned}$$

Hence, the function $u(t)$ roughly satisfies a linear differential equation of the form

$$u'(t) = J(t)u(t) + f(t).$$

So, if a method behaves well for linear problems, we could hope for a equally good behavior for general differential equations.

Localization

Consider a linear differential equation

$$u'(t) = J(t)u(t) + f(t).$$

In a small neighborhood of some given value $\bar{t}$ we have

$$u'(t) = J(t)u(t) + f(t) = J(\bar{t})u(t) + f(t) + \dots$$

Hence, the function $u(t)$ roughly satisfies a linear differential equation with constant coefficients of the form

$$u'(t) = Ju(t) + f(t).$$

So, if a method behaves well for linear problems with constant coefficients, we could hope for a equally good behavior for general linear differential equations.

Diagonalization

Consider a linear differential equation with constant coefficients

$$u'(t) = Ju(t) + f(t).$$

Assume that the matrix J is diagonalizable, i.e.:

$$T^{-1}JT = D \quad \text{with } D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix}.$$

Then, u can be represented as a linear combination of the columns of T: $u = Tv$.

The function v satisfies the following differential equation:

$$v'(t) = T^{-1}u'(t) = T^{-1}Ju(t) + T^{-1}f(t) = DT^{-1}u(t) + T^{-1}f(t) = Dv(t) + T^{-1}f(t).$$

Hence, the components of v satisfy scalar linear differential equations of the form

$$u'(t) = \lambda\, u(t) + f(t).$$

So, if a method behaves well for such scalar linear problems, we could hope for a equally good behavior for general linear differential equations with constant coefficients.

Since the additive term $f(t)$ does not influence the stability properties, we will drop this term and consider in our further discussion the following simple model problem:

$$u'(t) = \lambda\, u(t) \tag{7.1}$$

with $\lambda \in \mathbb{C}$.

This requires to extend our considered class of problems: For given $u_0 \in \mathbb{C}^N$ and $f : \mathbb{R} \times \mathbb{C}^N \longrightarrow \mathbb{C}^N$, find $u : \mathbb{R} \longrightarrow \mathbb{C}^N$ such that

$$\begin{aligned} u'(t) &= f(t, u(t)), \quad t > 0 \\ u(0) &= u_0. \end{aligned} \tag{7.2}$$

Such an initial value problem in $\mathbb{C}^N$ can be written as an initial value problem in $\mathbb{R}^N \times \mathbb{R}^N$.

For this, observe first that an element $v \in \mathbb{C}^N$ can be uniquely written as $v = v_1 + i\, v_2$ with $v_1, v_2 \in \mathbb{R}^N$, which defines the associated element $V = (v_1, v_2)^T \in \mathbb{R}^N \times \mathbb{R}^N$. And, vice versa, each element $V = (v_1, v_2)^T \in \mathbb{R}^N \times \mathbb{R}^N$ defines an element $v = v_1 + i\, v_2 \in \mathbb{C}^N$.

For a given inner product (v, w) on $\mathbb{C}^N$, e.g., the Euclidean inner product

$$(v, w)_{\ell^2} = \bar{v}^T w,$$

we have

$$(v, w) = (v_1 + i\, v_2, w_1 + i\, w_2) = (v_1, w_1) + (v_2, w_2) + i\,[(v_1, w_2) - (v_2, w_1)].$$

And, vice versa, this line defines an inner product on $\mathbb{C}^N$ for a given inner product in $\mathbb{R}^N$.

Therefore, we have the following relation between the inner product in $\mathbb{C}^N$ and the associated inner product in $\mathbb{R}^N \times \mathbb{R}^N$:

$$(V, W) = (v_1, w_1) + (v_2, w_2) = \mathrm{Re}\,(v, w) \quad \text{with} \quad V = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, W = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

A differential equation in $\mathbb{C}^N$ of the form

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

can be rewritten as a differential equation in $\mathbb{R}^N \times \mathbb{R}^N$

$$U'(t) = F(t, U(t)),$$

where

$$U(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \text{ with } u_1(t) + i\, u_2(t) = u(t)$$

and

$$F(t, U) = \begin{pmatrix} f_1(t, u) \\ f_2(t, u) \end{pmatrix} \text{ with } U = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},\ u = u_1 + i\, u_2,\ f_1(t, u) + i\, f_2(t, u) = f(t, u).$$

The differential equation in $\mathbb{C}^N$ is called dissipative iff the associated differential equation in $\mathbb{R}^N \times \mathbb{R}^N$ is dissipative, i.e.,

$$(F(t, W) - F(t, V), W - V) \le 0 \quad \text{for all } t \in \mathbb{R},\ U, V \in \mathbb{R}^N,$$

which translates to the condition

$$\mathrm{Re}\,(f(t, w) - f(t, v), w - v) \le 0. \quad \text{for all } t \in \mathbb{R},\ u, v \in \mathbb{C}^N,$$

which is consistent with the original definition in the case of differential equations in $\mathbb{R}^N$.

Next we investigate under which condition model problem (7.1) is dissipative:

$$\mathrm{Re}\,(\lambda z, z) \le 0 \quad \text{for all } z \in \mathbb{C}$$

is equivalent to

$$\mathrm{Re}\ \bar{\lambda}|z|^2 \le 0 \quad \text{for all } z \in \mathbb{C},$$

i.e.

$$\mathrm{Re}\ \bar{\lambda} = \mathrm{Re}\ \lambda \le 0.$$

The exact solution of (7.1) is given by

$$u(t) = e^{\lambda\,(t - t_0)}\, u_0 = e^{\lambda_1\,(t - t_0)}\ (\cos(\lambda_2 (t - t_0)) + i\ \sin(\lambda_2\,(t - t_0)))\ u_0.$$

From this it easily follows that (7.1) is dissipative if and only if the solutions are bounded on $[t_0, \infty)$.

7.1 The Stability Function

If a Runge-Kutta method is applied to (7.1), then we obtain

$$U_i = u_0 + \tau\,\lambda\,\Big(\sum_{j=1}^{s} a_{ij}\,U_j\Big) \quad \text{and} \quad u_1 = u_0 + \tau\,\lambda\,\Big(\sum_{i=1}^{s} b_i\,U_i\Big),$$

i.e.:

$$U = u_0\,e + \tau\,\lambda\,AU \quad \text{and} \quad u_1 = u_0 + \tau\,\lambda\,b^T U \quad \text{with } e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Therefore,

$$U = u_0\,(I - \tau\,\lambda\,A)^{-1} e$$

and

$$u_1 = u_0 + u_0 \tau\,\lambda\,b^T (I - \tau\,\lambda\,A)^{-1} e = R(\tau\,\lambda)\,u_0,$$

where the complex function R is introduced in the following definition:

Definition 7.1. *The stability function $R(z)$ of a Runge-Kutta method is given by*

$$R(z) = 1 + z\,b^T (I - z\,A)^{-1} e.$$

Examples:

- explicit Euler method:

$$u_1 = u_0 + \tau\,\lambda\,u_0 = (1 + \tau\,\lambda)\,u_0 = R(\tau\,\lambda)\,u_0 \quad \text{with} \quad R(z) = 1 + z.$$

- implicit Euler method:

$$u_1 = u_0 + \tau\,\lambda\,u_1, \quad \text{i.e.} \quad (1 - \tau\,\lambda)\,u_1 = u_0.$$

  Hence

$$u_1 = \frac{1}{1 - \tau\,\lambda}\,u_0 = R(\tau\,\lambda)\,u_0 \quad \text{with} \quad R(z) = \frac{1}{1 - z}.$$

Lemma 7.1.

$$R(z) = \frac{P(z)}{Q(z)} \quad \textit{with} \quad P(z) = \det(I - z\,(A - eb^T)) \quad \textit{and} \quad Q(z) = \det(I - z\,A).$$

Proof. With $\tau\,\lambda$ replaced by z we have from above

$$\begin{aligned} (I - z\,A)\,U &= u_0\,e, \\ -zb^T U + u_1 &= u_0, \end{aligned}$$

i.e.:

$$\begin{bmatrix} I - z\,A & 0 \\ -zb^T & 1 \end{bmatrix} \begin{bmatrix} U \\ u_1 \end{bmatrix} = \begin{bmatrix} e \\ 1 \end{bmatrix} u_0.$$

By Cramer's rule it follows that

$$u_1 = \frac{\det \begin{bmatrix} I - z\,A & e \\ -zb^T & 1 \end{bmatrix}}{\det \begin{bmatrix} I - z\,A & 0 \\ -zb^T & 1 \end{bmatrix}} u_0.$$

Now

$$\det \begin{bmatrix} I - z\,A & 0 \\ -zb^T & 1 \end{bmatrix} = \det(I - zA)$$

and

$$\det \begin{bmatrix} I - z\,A & e \\ -zb^T & 1 \end{bmatrix} = \det \begin{bmatrix} I - z\,A + zeb^T & 0 \\ -zb^T & 1 \end{bmatrix} = \det(I - z\,(A - zeb^T)).$$

□

Consequences: $R(z)$ is a rational function with polynomials $P(z)$ and $Q(z)$ of degree $\leq s$. For explicit Runge-Kutta methods we have $Q(z) = 1$, so in this case $R(z)$ is a polynomial of degree s.

Next we investigate under which condition a Runge-Kutta method applied to (7.1) is contractive: A Runge-Kutta method is contractive if and only if

$$|w_1 - v_1| \leq |w_0 - v_0| \quad \text{for all } v_0, w_0 \in \mathbb{C} \quad \text{with } v_1 = R(\tau\,\lambda)\,v_0,\ w_1 = R(\tau\,\lambda)\,w_0,$$

i.e.

$$|R(\tau\,\lambda)|\,|w_0 - v_0| \leq |w_0 - v_0| \quad \text{for all } v_0, w_0 \in \mathbb{C},$$

which is equivalent to

$$|R(\tau\,\lambda)| \leq 1.$$

Example. *The differential equation*

$$u'(t) = -50\,u(t)$$

corresponds to the model problem (7.1) *with* $\lambda = -50$. *The explicit Euler method is contractive iff*

$$|1 - 50\,\tau| \leq 1 \quad \textit{i.e.: } \tau \leq \frac{1}{25}.$$

The implicit Euler method is contractive for all $\tau > 0$, *since*

$$\frac{1}{|1 + 50\,\tau|} \leq 1.$$

Definition 7.2. *The stability domain S of a Runge-Kutta method is given by*

$$S = \{z \in \mathbb{C} | R(z)| \leq 1\}.$$

Then we have:

Lemma 7.2. *A Runge-Kutta method applied to* (7.1) *is contractive, iff*

$$\tau\,\lambda \in S.$$

Examples:

- explicit Euler method:

$$S = \{z \in \mathbb{C} : |z + 1| \leq 1\} = \{z \in \mathbb{C} : |z - (-1)| \leq 1\}$$

  S is the disk with center -1 and radius 1.

- implicit Euler method:

$$S = \{z \in \mathbb{C} : \frac{1}{|z - 1|} \leq 1\} = \{z \in \mathbb{C} : |z - 1| \geq 1\}$$

  S is the set of all points in $\mathbb{C}$ which lie on or outside the disk with center 1 and radius 1.

Definition 7.3. *A Runge-Kutta method is called A-stable if*

$$\mathbb{C}^- \subset S$$

with $\mathbb{C}^- = \{z \in \mathbb{C} : \operatorname{Re} z \leq 0\}$.

Example. *The explicit Euler method is not A-stable, the implicit Euler method is A-stable.*

Lemma 7.3. *No explicit Runge-Kutta method is A-stable.*

Proof. The stability function is a polynomial of degree s. No such polynomial can be bounded (by 1) on $\mathbb{C}^-$. □

Lemma 7.4. *For dissipative problems of the form* (7.1) *an A-stable Runge-Kutta method is contractive for all* $\tau > 0$.

Proof. The problem (7.1) is dissipative iff $\operatorname{Re}\lambda \leq 0$, i.e. $\lambda \in \mathbb{C}^-$. Then $\tau\,\lambda \in \mathbb{C}^-$ for all $\tau > 0$. The Runge-Kutta method is A-stable, i.e. $\mathbb{C}^- \subset S$. Therefore,

$$\tau\,\lambda \in \mathbb{C}^- \subset S.$$

But that means that the method is contractive. □

For the exact solution of the model problem we have

$$u(t_1) = e^{\tau\,\lambda}\,u_0.$$

Observe that

$$u_1 = R(\tau\,\lambda)\,u_0$$

for the approximate solution. This shows that the approximate solution is the closer to the exact solution the better the stability function $R(z)$ approximates the exponential function e^z near $z = 0$. For λ very large and negative real (see the heat equation) $z = \tau\,\lambda$ is close to $-\infty$. For the exponential function we have $e^{-\infty} = 0$. The corresponding property of the Runge-Kutta method leads to the following definition:

Definition 7.4. *A Runge-Kutta method is called L-stable iff it is A-stable and*

$$\lim_{z\to\infty} R(z) = 0.$$

Examples:

- The implicit Euler method is L-stable since it is A-stable and:

$$R(z) = \frac{1}{1-z} \to 0 \quad \text{for } z \to \infty.$$

- For the implicit midpoint method applied to (7.1) we have:

$$U_1 = u_0 + \frac{1}{2}\,\tau\,\lambda\,U_1 \quad \Longrightarrow \quad U_1 = \frac{1}{1-\frac{1}{2}\,\tau\,\lambda}$$

$$u_1 = u_0 + \frac{1}{2}\,\tau\,\lambda\,U_1 = \left(1 + \frac{1}{1-\frac{1}{2}\,\tau\,\lambda}\right)\,u_0 = \frac{1+\frac{1}{2}\,\tau\,\lambda}{1-\frac{1}{2}\,\tau\,\lambda}\,u_0 = R(\tau\lambda)\,u_0$$

with

$$R(z) = \frac{1+\frac{1}{2}\,z}{1-\frac{1}{2}\,z} = \frac{2+z}{2-z}.$$

$S = \mathbb{C}^-$, so the method is A-stable.

$$R(\infty) = -1.$$

The method is not L-stable.

Lemma 7.5. *If an implicit Runge-Kutta method with non-singular matrix A satisfies one of the following two conditions:*

a) $a_{sj} = b_j$ *for* $j = 1, \ldots s$,

b) $a_{i1} = b_1$ *for* $i = 1, \ldots, s$,

then $R(\infty) = 0$.

Proof.

$$R(z) = 1 + zb^T(I - zA)^{-1}e = 1 + b^T\left(\frac{1}{z}I - A\right)^{-1}e.$$

Therefore

$$R(\infty) = 1 - b^T A^{-1}e.$$

a) $b^T = e_s^T A$. Therefore

$$1 - b^T A^{-1}e = 1 - e_s^T e = 1 - 1 = 0.$$

b) $Ae_1 = b_1\, e$. Therefore

$$1 - b^T A^{-1}e = 1 - b^T \frac{1}{b_1}e_1 = 1 - 1 = 0.$$

□

7.2 Padé Approximation of the Exponential Function

We already discussed that

$$R(z) \approx e^z \quad \text{near } z = 0.$$

Theorem 7.1. *If a Runge-Kutta method is of order p, then*

$$e^z - R(z) = O(z^{p+1}) \quad \textit{for } z \to 0.$$

Proof. The local error for the model problem (7.1) is given by:

$$u(t_0 + \tau) - u_1 = \left[e^{\tau\lambda} - R(\tau\lambda)\right]\, u_0.$$

If the method is of order p, we have:

$$u(t_0 + \tau) - u_1 = O(\tau^{p+1}).$$

Therefore, for $\lambda = z/|z|$, $\tau = |z|$, and $u_0 = 1$, it follows that

$$e^z - R(z) = O(z^{p+1}).$$

□

Theorem 7.2. *If an explicit Runge-Kutta method is of order p, then*

$$R(z) = 1 + z + \frac{1}{2!}\, z^2 + \ldots + \frac{1}{p!}\, z^p + O(z^{p+1}).$$

Proof. $R(z)$ is a polynomial with

$$e^z - R(z) = O(z^{p+1}).$$

On the other hand, by a Taylor expansion of e^z we have

$$e^z - \left[1 + z + \frac{1}{2!}\, z^2 + \ldots + \frac{1}{p!}\, z^p\right] = O(z^{p+1}).$$

The statement follows by subtracting these two identities . □

Among all polynomials of degree $\leq p$, the Taylor-polynomial

$$T_p(z) = 1 + z + \frac{1}{2!}\, z^2 + \ldots + \frac{1}{p!}\, z^p$$

of e^z is the unique polynomial approximation of e^z around $z = 0$ with

$$e^z - T_p(z) = O(|z|^{p+1}).$$

A similar result is available for rational approximations to e^z:

Theorem 7.3. *Let $j, k \in \mathbb{N}_0$. Among all rational functions with degrees k and j of the numerator and denominator, respectively, the (k, j)-Padé approximation to e^z, given by*

$$R_{kj}(z) = \frac{P_{kj}(z)}{Q_{kj}(z)},$$

where

$$\begin{aligned} P_{kj}(z) &= \sum_{\ell=0}^{k} \frac{(j+k-\ell)!}{(j+k)!}\, \frac{k!}{(k-\ell)!}\, \frac{z^\ell}{\ell!} \\ &= 1 + \frac{k}{j+k}\, z + \frac{k(k-1)}{(j+k)(j+k-1)}\, \frac{z^2}{2!} + \ldots + \frac{k(k-1)\ldots 1}{(j+k)\ldots(j+1)}\, \frac{z^k}{k!} \end{aligned}$$

and

$$\begin{aligned} Q_{kj}(z) = P_{jk}(-z) &= \sum_{\ell=0}^{j} (-1)^\ell\, \frac{(k+j-\ell)!}{(k+j)!}\, \frac{j!}{(j-\ell)!}\, \frac{z^\ell}{\ell!} \\ &= 1 - \frac{j}{k+j}\, z + \frac{j(j-1)}{(k+j)(k+j-1)}\, \frac{z^2}{2!} + \ldots + (-1)^j \frac{j(j-1)\ldots 1}{(k+j)\ldots(k+1)}\, \frac{z^j}{j!}, \end{aligned}$$

is the unique rational approximation to e^z with

$$e^z - R_{kj}(z) = O(z^{j+k+1}).$$

Moreover

$$e^z - R_{kj}(z) = C_{kj}\, z^{j+k+1} + O(z^{j+k+1}) \quad \text{with } C_{kj} = (-1)^{j+k}\, \frac{j!k!}{(j+k)!(j+k+1)!}.$$

For a proof, see [10].

The (k,k)-Padé approximations are called diagonal Padé approximations.

Theorem 7.4. *Assume that $R_{kj}(z)$ is the stability function of a Runge-Kutta method. Then the method is A-stable if and only if $k \leq j \leq k+2$.*

For a proof, see [10].

Gauß methods

Theorem 7.5. *The s-stage Gauß method is of order $2s$. Its stability function is $R_{s,s}(z)$ and the method is A-stable.*

Proof. The stability function is of the form $R(z) = P(z)/Q(z)$ with $\deg P \leq s$ and $\deg Q \leq s$, the method is of order $p = 2s$. Therefore

$$e^z - R(z) = O(z^{2s+1}).$$

From the uniqueness result of Theorem 7.3 it follows that $R(z) = R_{s,s}(z)$, the rest follows from Theorem 7.4. □

Radau methods

Radau I methods: $c_1 = 0$.

Butcher: The coefficients a_{ij} are determined by $C(s)$, which implies:

$$a_{1j} = 0 \quad \text{for } j = 1, \ldots, s.$$

Ehle: The coefficients a_{ij} are determined by $D(s)$, which implies:

$$a_{i1} = b_1 \quad \text{for } i = 1, \ldots, s.$$

These methods are called Radau I A methods. They are also of order $p = 2s - 1$.

Radau II methods: $c_s = 1$.

Butcher: The coefficients a_{ij} are determined by $D(s)$, which implies:

$$a_{is} = 0 \quad \text{for } i = 1, \ldots, s.$$

Ehle: The coefficients a_{ij} are determined by $C(s)$, which implies:

$$a_{sj} = b_j \quad \text{for all } j = 1, \ldots, s.$$

These methods are called Radau II A methods. They are also of order $p = 2s - 1$.

Theorem 7.6. *The s-stage Radau IA method and the s-stage Radau IIA method are of order $2s-1$. Their stability function is $R_{s-1,s}(z)$ and the methods are A-stable and also L-stable.*

Proof. The stability function is of the form $R(z) = P(z)/Q(z)$ with $\deg P \leq s$ and $\deg Q \leq s$, the method is of order $p = 2s-1$. Therefore

$$e^z - R(z) = O(z^{2s}).$$

Observe that $P(z) = \det(I - z\,(A - eb^T))$. For the Radau IA method we have

$$(A - eb^T)e_1 = b_1 e - b_1 e = 0.$$

So the first column of $A - eb^T$ vanishes, which implies $\det P(z) \leq s-1$. For the Radau IIA method we have

$$e_s^T(A - eb^T) = b^T - b^T = 0.$$

So the last row of $A - eb^T$ vanishes, which also implies $\det P(z) \leq s-1$. From the uniqueness result of Theorem 7.3 it follows that $R(z) = R_{s-1,s}(z)$, the rest follows from Theorem 7.4. □

Lobatto methods

$c_1 = 0$ and $c_s = 1$.

Butcher: The coefficients a_{ij} are determined by $a_{1j} = 0$ for $j = 1, \ldots, s$, $a_{is} = 0$ for $i = 1, \ldots, s$ and $C(s-1)$

Ehle introduced 3 classes of Lobatto methods:

- Lobatto IIIA: The coefficients a_{ij} are determined by $C(s)$.
- Lobatto IIIB: The coefficients a_{ij} are determined by $D(s)$.
- Lobatto IIIC: The coefficients a_{ij} are determined by $a_{i1} = b_1$ for $i = 1, \ldots, s$ and $C(s-1)$.

All these methods are also of order $p = 2s-2$.

Theorem 7.7. *The s-stage Lobatto IIIA, IIIB, and IIIC methods are of order $2s-2$. The stability function of the Lobatto IIIA and IIIB methods is $R_{s-1,s-1}(z)$, the stability function of the Lobatto IIIC method is $R_{s-2,s}(z)$. All these methods are A-stable, the Lobatto IIIC method is also L-stable.*

7.3 Linear Systems of ODEs with Constant Coefficients

In this section we consider linear systems

$$\begin{aligned} u'(t) &= Ju(t) + f(t), \\ u(0) &= u_0 \end{aligned} \tag{7.3}$$

with a constant matrix $J \in \mathbb{R}^{N\times N}$.

Observe that a scalar product (v, w) in $\mathbb{R}^N$ has a unique extension to $\mathbb{C}^N$: For $v = v_1 + iv_2$ and $w = w_1 + iw_2$ with $v_1, v_2, w_1, w_2 \in \mathbb{R}^N$ this extension is given by

$$(v, w) = (v_1, v_2) + (w_1, w_2) + i\left[(v_1, w_2) - (v_2, w_1)\right].$$

By definition, (7.3) is dissipative iff

$$(Jv, v) \le 0 \quad \text{for all } v \in \mathbb{R}^N,$$

which is equivalent to

$$\operatorname{Re}(Jv, v) \le 0 \quad \text{for all } v \in \mathbb{C}^N.$$

Case: J is normal

Definition 7.5. *A matrix $J \in \mathbb{C}^{N\times N}$ is normal iff $J^*J = JJ^*$, where J^* is the adjoint of J, given by*

$$(J^*v, w) = (v, Jw) \quad \textit{for all } v, w \in \mathbb{C}^N.$$

Lemma 7.6. *A matrix $J \in \mathbb{C}^{N\times N}$ is normal iff there exists an orthonormal basis of $\mathbb{C}^N$ consisting of eigenvectors of J:*

$$(e_i, e_j) = \delta_{ij} \quad \textit{for } i, j = 1, \dots, N \quad \textit{with} \quad Je_i = \lambda_i e_i,\ e_i \neq 0.$$

Lemma 7.7. *If $J \in \mathbb{C}^{N\times N}$ is normal, then*

$$\|J\| = \rho(J),$$

where $\rho(J)$ denotes the spectral radius of J.

Proof. Each vector $v \in \mathbb{C}^N$ can be written in the form

$$v = \sum_{i=1}^{N} \alpha_i e_i.$$

Then

$$Jv = \sum_{i=1}^{N} \alpha_i \lambda_i e_i.$$

Since $(e_i)_{i=1,\dots,N}$ is an orthonormal basis, we have

$$\|v\|^2 = \sum_{i=1}^{N} |\alpha_i|^2 \quad \text{and} \quad \|Jv\|^2 = \sum_{i=1}^{N} |\lambda_i|^2 \, |\alpha_i|^2,$$

which implies

$$\|J\|^2 = \sup_{0 \neq v \in \mathbb{C}^N} \frac{\|Jv\|^2}{\|v\|^2} = \sup_{0 \neq v \in \mathbb{C}^N} \frac{\sum_{i=1}^{N} |\lambda_i|^2 \, |\alpha_i|^2}{\sum_{i=1}^{N} |\alpha_i|^2} = \max_{i=1,\dots,N} |\lambda_i|^2.$$

□

Lemma 7.8. *If $J \in \mathbb{C}^{N \times N}$ is normal, then* (7.3) *is dissipative iff*

$$\operatorname{Re} \lambda \leq 0 \quad \textit{for all } \lambda \in \sigma(J),$$

where $\sigma(J)$ denotes the set of all eigenvalues of J.

Proof. With $v = \sum_{i=1}^{N} \alpha_i, e_i$ condition

$$\operatorname{Re}(Jv, v) \leq 0 \quad \text{for all } v \in \mathbb{C}^N,$$

becomes

$$\sum_{i=1}^{N} (\operatorname{Re} \overline{\lambda}_i) \, |\alpha_i|^2 \leq 0 \quad \text{for all } \alpha_i \in \mathbb{C}, \ i = 1, \dots, N,$$

which is equivalent to

$$\operatorname{Re} \lambda_i \leq 0 \quad \text{for all } i = 1, \dots, N.$$

□

If a Runge-Kutta method is applied to (7.3) with $f(t) \equiv 0$, we obtain

$$U_i = u_0 + \tau \sum_{j=1}^{s} a_{ij} \, J \, U_j,$$
$$u_1 = u_0 + \tau \sum_{i=1}^{s} b_i \, J \, U_i.$$

i.e.,

$$\begin{bmatrix} I - a_{11}\tau J & -a_{12}\tau J & \cdots & -a_{1s}\tau J & 0 \\ -a_{21}\tau J & I - a_{22}\tau J & \cdots & -a_{2s}\tau J & 0 \\ \vdots & & \ddots & & \\ -a_{s1}\tau J & -a_{s2}\tau J & \cdots & I - a_{ss}\tau J & 0 \\ -b_1\tau J & -b_2\tau J & \cdots & -b_s\tau J & I \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_s \\ u_1 \end{bmatrix} = \begin{bmatrix} I \\ I \\ \vdots \\ I \\ I \end{bmatrix} u_0,$$

or. in short, with $Z = \tau\, J$,

$$\begin{bmatrix} I \otimes I - A \otimes Z & 0 \\ -b^T \otimes Z & I \end{bmatrix} \begin{bmatrix} U \\ u_1 \end{bmatrix} = \begin{bmatrix} e \otimes I \\ I \end{bmatrix} u_0.$$

Observe that all entries of the block matrix are matrices of the form $\alpha\, I + \beta\, Z$, which all commute. Then, by Cramer's rule it follows that

$$Q(Z)\, u_1 = P(Z)\, u_0$$

with

$$P(Z) = \mathrm{Det} \begin{bmatrix} I \otimes I - A \otimes Z & e \otimes I \\ -b^T \otimes Z & I \end{bmatrix} = \mathrm{Det}(I \otimes I - (A - eb^T) \otimes Z) \in \mathbb{C}^{N \times N}$$

and

$$Q(Z) = \mathrm{Det} \begin{bmatrix} I \otimes I - A \otimes Z & 0 \\ -b^T \otimes Z & I \end{bmatrix} = \mathrm{Det}(I \otimes I - A \otimes Z) \in \mathbb{C}^{N \times N}.$$

Here Det denotes the matrix-valued version of the standard determinant for block matrices with commuting blocks of the same size, and $P(Z)$, $Q(Z) \in \mathbb{C}$ are matrix polynomials built from the corresponding polynomials $P(z)$, $Q(z)$ appearing in the stability function $R(z) = P(z)/Q(z)$ of the Runge-Kutta methods by replacing the monomials z^i by Z^i.

If we introduce the matrix version of the rational function $R(z)$ by

$$R(Z) = Q(Z)^{-1} P(Z),$$

we have

$$u_1 = R(\tau J)\, u_0.$$

Then the Runge-Kutta applied to (7.3) is contractive iff

$$\|u_1\| \le \|u_0\| \quad \text{for all } u_0 \in \mathbb{R}^N,$$

i.e.:

$$\|R(\tau J) u_0\| \le \|u_0\| \quad \text{for all } u_0 \in \mathbb{R}^N,$$

i.e.:

$$\|R(\tau J)\| \le 1.$$

Case: J is normal

If J is normal, then $R(\tau J)$ is also normal. Therefore,

$$\|R(\tau J)\| = \max\{|R(\tau\lambda)| : \lambda \in \sigma(J)\},$$

which directly implies

Lemma 7.9. *If J is normal, the Runge-Kutta method applied to* (7.3) *is contractive iff*

$$|R(\tau\lambda)| \leq 1 \quad \textit{for all } \lambda \in \sigma(J),$$

which is equivalent to

$$\tau\lambda \in S \quad \textit{for all } \lambda \in \sigma(J).$$

Example. *The semi-discretized heat equation is given by*

$$M_h \underline{u}_h'(t) + K_h \underline{u}_h(t) = \underline{f}_h(t),$$

where M_h and K_h are symmetric and positive definite matrix with eigenvalues

$$0 < O(1) = \mu_1 \leq \mu_2 \leq \ldots \leq \mu_N = O(h^{-2}).$$

This problem is of the form (7.3) *with $J = M_h^{-1}K_h$. So the eigenvalues λ of J are given by*

$$\lambda = -\mu \quad \textit{with } \mu \in \sigma(M_h^{-1}K_h).$$

The system is dissipative.

The explicit Euler method is contractive iff

$$\tau\lambda \in S \quad \textit{for all } \lambda \in \sigma(J),$$

which is equivalent to

$$\tau\lambda_N = -\tau\mu_N \geq -2$$

i.e.

$$\tau \leq \frac{2}{\mu_N} = O(h^2).$$

The implicit Euler method is A-stable. Therefore

$$\tau\,\lambda_i \in \mathbb{C}^- \subset S,$$

which shows that the method is contractive for all $\tau > 0$.

More generally, we have, of course, for normal matrices J:

Theorem 7.8. *If* (7.3) *is dissipative and if the Runge-Kutta method is A-stable, then the method is contractive for all $\tau > 0$.*

Observe that surprisingly the last theorem also holds without restriction on J.

Example. *The semi-discretized wave equation is given by*

$$M_h\,\underline{u}_h''(t) + K_h\,\underline{u}_h(t) = \underline{f}_h(t),$$

where M_h, K_h are symmetric and positive definite matrix with eigenvalues

$$0 < O(1) = \mu_1 \leq \mu_2 \leq \ldots \leq \mu_N = O(h^{-2}).$$

This problem can be written in the form

$$\begin{bmatrix} \underline{u}_h'(t) \\ \underline{v}_h'(t) \end{bmatrix} = J \begin{bmatrix} \underline{u}_h(t) \\ \underline{v}_h(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{f}_h(t) \end{bmatrix} \quad \textit{with } J = \begin{bmatrix} 0 & I \\ -M_h^{-1} K_h & 0 \end{bmatrix}.$$

The system is dissipative, since

$$\left(J \begin{bmatrix} u \\ v \end{bmatrix}, \begin{bmatrix} u \\ v \end{bmatrix} \right) = 0 \quad \textit{for all } u, v \in \mathbb{R}^N,$$

see Chapter 6. With

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} + \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}$$

it follows that

$$\begin{aligned} 0 &= \left(J \begin{bmatrix} u \\ v \end{bmatrix}, \begin{bmatrix} u \\ v \end{bmatrix} \right) \\ &= \underbrace{\left(J \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \right)}_{=0} + \left(J \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}, \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \right) + \left(J \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \right) + \underbrace{\left(J \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \right)}_{=0}, \end{aligned}$$

i.e.: $J^* = -J$, so J is normal. The eigenvalues λ of J are given by

$$\lambda = \pm i \sqrt{\mu} \quad \textit{with } \mu \in \sigma(M_h^{-1} K_h),$$

Observe that the eigenvalue of maximal modulus of J is of the order $O(h^{-1})$.

The explicit Euler method is never contractive, since

$$\tau\lambda \notin S \quad \textit{for all } \tau > 0,$$

or, equivalently,

$$\|R(\tau J)\| = \rho(R(\tau J)) > 1 \quad \textit{for all } \tau > 0.$$

The implicit Euler method is always contractive, see the last theorem.

If we apply a partitioned Runge-Kutta method consisting of the explicit Euler method for the first line and the implicit Euler method for the second, we obtain

$$\begin{aligned} \underline{u}_{h,1} &= \underline{u}_{h,0} + \tau\, \underline{v}_{h,0} \\ \underline{v}_{h,1} &= \underline{v}_{h,0} + \tau\, M_h^{-1} \left[\underline{f}_h(t_1) - K_h\, \underline{u}_{h,1} \right] \end{aligned}$$

Therefore,

$$\begin{bmatrix} I & 0 \\ \tau M_h^{-1} K_h & I \end{bmatrix} \begin{bmatrix} \underline{u}_{h,1} \\ \underline{v}_{h,1} \end{bmatrix} = \begin{bmatrix} I & \tau\, I \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{u}_{h,0} \\ \underline{v}_{h,0} \end{bmatrix} + \tau \begin{bmatrix} 0 \\ M_h^{-1} \underline{f}_h(t_1) \end{bmatrix},$$

i.e.:

$$\begin{bmatrix} \underline{u}_{h,1} \\ \underline{v}_{h,1} \end{bmatrix} = \begin{bmatrix} I & \tau\, I \\ -\tau\, M_h^{-1} K_h & I - \tau^2\, M_h^{-1} K_h \end{bmatrix} \begin{bmatrix} \underline{u}_{h,0} \\ \underline{v}_{h,0} \end{bmatrix} + \tau \begin{bmatrix} 0 \\ M_h^{-1} \underline{f}_h(t_1) \end{bmatrix}.$$

For the stability analysis it is sufficient to consider the case $\underline{f}_h(t) \equiv 0$, for which we obtain

$$\begin{bmatrix} \underline{u}_{h,1} \\ \underline{v}_{h,1} \end{bmatrix} = R_h \begin{bmatrix} \underline{u}_{h,0} \\ \underline{v}_{h,0} \end{bmatrix} \quad \text{with} \quad R_h = \begin{bmatrix} I & \tau\, I \\ -\tau\, M_h^{-1} K_h & I - \tau^2\, M_h^{-1} K_h \end{bmatrix}.$$

Let $(e_i)_{i=1,\dots,N}$ be a basis of $\mathbb{C}^N$ with $M_h^{-1} K_h e_i = \mu_i\, e_i$ and $(e_i, e_j)_{M_h} = \delta_{ij}$. Assume that

$$u = \sum_{i=1}^{N} \alpha_i\, e_i, \quad v = \sum_{i=1}^{N} \beta_i\, e_i, \quad \text{i.e.} \quad \begin{bmatrix} u \\ v \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} \alpha_i\, e_i \\ \beta_i\, e_i \end{bmatrix}.$$

Then

$$\left\| \begin{bmatrix} u \\ v \end{bmatrix} \right\|^2 = (K_h u, u)_{\ell^2} + (M_h v, v)_{\ell^2} = \sum_{i=1}^{N} \mu_i\, |\alpha_i|^2 + |\beta_i|^2 = \sum_{i=1}^{N} \left\| \begin{bmatrix} \sqrt{\mu_i}\, \alpha_i \\ \beta_i \end{bmatrix} \right\|_{\ell^2}^2$$

and

$$R_h \begin{bmatrix} u \\ v \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} 1 & \tau \\ -\tau\mu_i & 1 - \tau^2 \mu_i \end{bmatrix} \begin{bmatrix} \alpha_i\, e_i \\ \beta_i\, e_i \end{bmatrix}.$$

Therefore

$$\begin{aligned} \left\| R_h \begin{bmatrix} u \\ v \end{bmatrix} \right\|^2 &= \sum_{i=1}^{N} \left\| \begin{bmatrix} \sqrt{\mu_i} & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \tau \\ -\tau\mu_i & 1 - \tau^2 \mu_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \right\|_{\ell^2}^2 \\ &= \sum_{i=1}^{N} \left\| \begin{bmatrix} 1 & \tau\sqrt{\mu_i} \\ -\tau\sqrt{\mu_i} & 1 - \tau^2 \mu_i \end{bmatrix} \begin{bmatrix} \sqrt{\mu_i}\, \alpha_i \\ \beta_i \end{bmatrix} \right\|_{\ell^2}^2 = \sum_{i=1}^{N} \left\| G(\tau\sqrt{\mu_i}) \begin{bmatrix} \sqrt{\mu_i}\, \alpha_i \\ \beta_i \end{bmatrix} \right\|_{\ell^2}^2 \end{aligned}$$

with

$$G(z) = \begin{bmatrix} 1 & z \\ -z & 1 - z^2 \end{bmatrix},$$

which implies

$$\|R_h\| = \max_{i=1,\dots,N} \|G(\tau\sqrt{\mu_i})\|_{\ell^2}.$$

Unfortunately,

$$\|G(\tau\sqrt{\mu_i})\|_{\ell^2} > 1 \quad \text{for } z \neq 0.$$

Therefore, we cannot expect that the method is contractive with respect to the chosen norm. However, for stability with a stability constant independent of h, it is suffices to show

$$\|(R_h)^j\| \leq C \quad \text{for all } j \in \mathbb{N}$$

with a constant C independent of h, if we consider only the case of constant size τ. Analogously to above one can show that

$$\|(R_h)^j\| = \max_{i=1,\dots,N} \|G(\tau\sqrt{\mu_i})^j\|_{\ell^2} \quad \text{for all } j \in \mathbb{N}.$$

For $0 < z < 2$ the two eigenvalues of $G(z)$ build a complex conjugate pair $\lambda(z)$ and $\overline{\lambda(z)}$ with

$$\lambda(z) = 1 - \frac{z^2}{2} + i\, z\sqrt{1 - \frac{z^2}{2}} \quad \textit{and} \quad |\lambda(z)| = |\overline{\lambda(z)}| = 1.$$

The corresponding eigenvectors $e(z)$ and $\overline{e(z)}$ (of length 1) are given by

$$e = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -\frac{z}{2} + i\sqrt{1 - \frac{z^2}{4}} \end{bmatrix}.$$

So

$$S(z)^{-1} G(z) S(z) = \begin{bmatrix} \lambda(z) & 0 \\ 0 & \overline{\lambda(z)} \end{bmatrix} \quad \textit{with} \quad S(z) = \begin{bmatrix} e(z) & \overline{e(z)} \end{bmatrix}.$$

Therefore

$$G(z)^j = S(z) \begin{bmatrix} \lambda(z)^j & 0 \\ 0 & \overline{\lambda(z)}^j \end{bmatrix} S(z)^{-1}.$$

Hence

$$\|G(z)^j\|_{\ell^2} \le \|S(z)\|_{\ell^2} \|S(z)^{-1}\|_{\ell^2}.$$

Now

$$\|S(z)\|_{\ell^2} \|S(z)^{-1}\|_{\ell^2} = \sqrt{\frac{\lambda_{max}(S(z)^H S(z))}{\lambda_{min}(S(z)^H S(z))}}.$$

With

$$S(z)^H S(z) = \begin{bmatrix} \overline{e(z)}^T \\ e(z)^T \end{bmatrix} \begin{bmatrix} e(z) & \overline{e(z)} \end{bmatrix} = \begin{bmatrix} 1 & \frac{z^2}{4} + i\frac{z}{2}\sqrt{1 - \frac{z^2}{4}} \\ \frac{z^2}{4} - i\frac{z}{2}\sqrt{1 - \frac{z^2}{4}} & 1 \end{bmatrix}$$

we obtain

$$\|S(z)\|_{\ell^2} \|S(z)^{-1}\|_{\ell^2} = \sqrt{\frac{1 + \frac{z}{2}}{1 - \frac{z}{2}}}.$$

So summarizing, for any chosen $c \in (0, 2)$, we have: If

$$\tau\sqrt{\mu_i} \le c \quad \textit{for all } i = 1, \ldots, N,$$

it follows that

$$\|R_h^j\| \le \sqrt{\frac{1 + \frac{c}{2}}{1 - \frac{c}{2}}},$$

which shows stability of the method with a stability constant independent of h if

$$\tau \le \frac{c}{\sqrt{\mu_N}} = O(h).$$

Remark. *We have shown that there exists a constant C such that*

$$\|G(z)^j\|_{\ell^2} \le C \quad \textit{for all } z \in (0, c] \quad \textit{and all } j \in \mathbb{N}.$$

The set $\{G(z) : 0 < z \le c\}$ is then called a stable family. The analysis we used is part of a more powerful theory based on the famous Kreiss Matrix Theorem, see [11].

7.4 Multistep Methods for Stiff Problems

If a linear k-step method

$$\alpha_k\, u_{j+k} + \alpha_{k-1}\, u_{j+k-1} + \ldots + \alpha_0\, u_j = \tau\, [\beta_k\, f_{j+k} + \beta_{k-1}\, f_{j+k-1} + \ldots + \beta_0\, f_j]$$

is applied to the model problem

$$u' = \lambda u,$$

we obtain

$$(\alpha_k - \mu\,\beta_k)\, u_{j+k} + (\alpha_{k-1} - \mu\,\beta_{k-1})\, u_{j+k-1} + \ldots + (\alpha_0 - \mu\,\beta_0)\, u_j = 0$$

with $\mu = \tau\,\lambda$.

Definition 7.6. *The set*

$$\begin{aligned} S = \{\mu \in \mathbb{C} :\ & \textit{all roots } \zeta(\mu) \textit{ of } \rho(z) - \mu\sigma(z) \textit{ satisfy} \\ & \textit{either } |\zeta(\mu)| < 1 \textit{ or } (|\zeta(\mu)| = 1 \textit{ and } \zeta(\mu) \textit{ is a simple root})\} \end{aligned}$$

is the stability domain of the linear k-step method.

Remark. *A multistep method is 0-stable iff $0 \in S$.*

Definition 7.7. *A linear k-step method is called A-stable if $\mathbb{C}^- \subset S$.*

Theorem 7.9 (The second Dahlquist barrier)**.** *An A-stable linear multistep method must be of order $p \leq 2$. The implicit trapezoidal rule is that A-stable method of this class, which has the smallest principal error term.*

Because of the second Dahlquist barrier weaker concepts of stability were introduced, e.g.:

Definition 7.8. *A method is called $A(\alpha)$-stable if $\mathbb{C}_\alpha \subset S$ with*

$$\mathbb{C}_\alpha = \{z \in \mathbb{C} : |\arg(-z)| < \alpha, z \neq 0\}.$$

There are higher-order $A(\alpha)$-stable methods.

Part III

Differential-Algebraic Problems

Chapter 8

Index and Classification of DAEs

8.1 Linear DAEs with Constant Coefficients

We consider initial value problems of the form

$$\begin{aligned} Bu'(t) + Au(t) &= f(t), \\ u(0) &= u_0 \end{aligned} \tag{8.1}$$

with $A, B \in \mathbb{R}^{N\times N}$. Special case explicit ODEs: $B = I$.

For the homogeneous case $f(t) \equiv 0$, the ansatz $u(t) = e^{\lambda t}u_0$ leads to

$$[A + \lambda B]u_0 = 0. \tag{8.2}$$

So λ must be an eigenvalue of the generalized eigenvalue problem (8.2) and u_0 must be an associated eigenvector.

Lemma 8.1. *If* $\det[A + \lambda B] \equiv 0$*, then*

$$\begin{aligned} Bu'(t) + Au(t) &= 0, \\ u(0) &= 0 \end{aligned} \tag{8.3}$$

has a non-trivial solution.

Proof. Let $\lambda_1, \lambda_2, \ldots, \lambda_{N+1}$ be $N + 1$ different eigenvalues of (8.2) with associated eigenvectors $u_1, u_2 \ldots, u_{N+1}$. Then

$$\sum_{i=1}^{N+1} \alpha_i \, u_i = 0 \quad \text{for some } \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{N+1}) \neq 0.$$

It is easy to see that

$$u(t) = \sum_{i=1}^{N+1} \alpha_i \, e^{\lambda_i \, t} \, u_i$$

solves (8.3). Assume that $u(t) \equiv 0$. It follows that

$$u^{(k)}(0) = \sum_{i=1}^{N+1} \lambda_i^k \, \alpha_i \, u_i = 0 \quad \text{for all } k \geq 0.$$

In particular, we have

$$(V \otimes I) \begin{pmatrix} \alpha_1 \, u_1 \\ \alpha_2 \, u_2 \\ \vdots \\ \alpha_{N+1} \, u_{N+1} \end{pmatrix} = 0 \quad \text{with} \quad V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_{N+1} \\ \vdots & & \vdots & \vdots \\ \lambda_1^N & \lambda_2^N & \cdots & \lambda_{N+1}^N \end{pmatrix}.$$

Since the Vandermonde matrix V is non-singular, it follows that $\alpha_i \, u_i = 0$ and, therefore, $\alpha_i = 0$ for all $i = 1, 2, \ldots, N+1$, which contradicts the assumption $\alpha \neq 0$. □

These considerations motivate the following concepts:

Definition 8.1. *1. The expression $A + \lambda B$ as a function in $\lambda \in \mathbb{C}$ is called a matrix pencil (German: Matrixbüschel) .*

2. A matrix pencil is called regular if $\det[A + \lambda B] \not\equiv 0$.

Let P and Q be non-singular matrices. By multiplying with P and using the transformation $u(t) = Qv(t)$ we obtain

$$PBQv'(t) + PAQv(t) = Pf(t).$$

Example. *Explicit ODEs: $B = I$.*
Jordan decomposition:

$$A = QJQ^{-1} \quad \textit{with the Jordan canonical form } J = \operatorname{diag}(J_1, \ldots, J_k),$$

where the $\mu_i \times \mu_i$-matrices J_i (Jordan blocks) are of the form

$$J_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{pmatrix}.$$

The numbers λ_i are the eigenvalues of A and the corresponding columns of Q are the (generalized) eigenvectors.

With $P = Q^{-1}$ we obtain:

$$PAQ = J, \quad PIQ = I.$$

and

$$v'(t) + Jv(t) = g(t)$$

with $v(t) = Q^{-1}u(t)$ *and* $g(t) = Pf(t)$.

The system of differential equations consists of systems of the form

$$\begin{aligned} w_1'(t) + \lambda\, w_1(t) + w_2(t) &= k_1(t), \\ &\vdots \\ w_{\mu-1}'(t) + \lambda\, w_{\mu-1}(t) + w_\mu(t) &= k_{\mu-1}(t), \\ w_\mu'(t) + \lambda\, w_\mu(t) &= k_\mu(t), \end{aligned}$$

which have a unique solution for arbitrary initial values $w(0)$*, given by*

$$\begin{aligned} w_\mu(t) &= w_\mu(0)e^{-\lambda t} + \int_0^t k_\mu(s)e^{\lambda(s-t)}\, ds, \\ w_{\mu-1}(t) &= w_{\mu-1}(0)e^{-\lambda t} + \int_0^t [k_{\mu-1}(s) - w_\mu(s)]e^{\lambda(s-t)}\, ds, , \\ &= [w_{\mu-1}(0) - w_\mu(0)\, t]\, e^{-\lambda t} + \int_0^t [k_{\mu-1}(s) - k_\mu(s)\, s]\, e^{\lambda(s-t)}\, ds, \\ &\vdots \\ w_1(t) &= \left[w_1(0) - w_2(0)\, t + w_3(0)\frac{t^2}{2} - \ldots + (-1)^{\mu-1}w_\mu(0)\frac{t^{\mu-1}}{(\mu-1)!}\right] e^{-\lambda t} \\ &\quad + \int_0^t \left[k_1(s) - k_2(s)\, s + k_3(s)\frac{s^2}{2} - \ldots + (-1)^{\mu-1}k_\mu(s)\frac{s^{\mu-1}}{(\mu-1)!}\right] e^{\lambda(s-t)}\, ds. \end{aligned}$$

Therefore, the following stability estimation results:

$$\|w(t)\| \le C\, \left[\|w(0)\| + \int_0^t \|k(s)\|\, ds\right]$$

with a constant $C > 0$ *for all* $t \in [0, T]$.

In terms of the original quantities: There exists a unique solution for arbitrary initial values $u(0)$ *and*

$$\|u(t)\| \le C\, \left[\|u(0)\| + \int_0^t \|f(s)\|\, ds\right]$$

with a constant $C > 0$ *for all* $t \in [0, T]$.

Theorem 8.1 (Weierstraß, Kronecker)**.** *Let* $A + \lambda B$ *be a regular matrix pencil. Then there exist matrices* P *and* Q *such that*

$$PAQ = \begin{pmatrix} J & 0 \\ 0 & I \end{pmatrix}, \quad PBQ = \begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix},$$

where J is a Jordan canonical form, $N = \operatorname{diag}(N_1, \ldots, N_m)$ with $\nu_i \times \nu_i$-matrices N_i, given by

$$N_i = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ & & & 0 \end{pmatrix}.$$

Proof. Let $c \in \mathbb{C}$ such that $A + c\,B$ is non-singular. From

$$A + \lambda\, B = A + c\, B + (\lambda - c)\, B$$

one obtains the matrix pencil

$$I + (\lambda - c)\,(A + c\,B)^{-1} B$$

by multiplying with $(A + c\,B)^{-1}$.

Let $\operatorname{diag}(J_1, J_2)$ be the Jordan canonical form of $(A + c\,B)^{-1}B$, where J_1 contains all diagonal blocks with non-vanishing eigenvalues, and J_2 contains the diagonal blocks for the eigenvalue 0. The corresponding similarity transformation leads to

$$\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} + (\lambda - c) \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix} = \begin{pmatrix} I - c\,J_1 & 0 \\ 0 & I - c\,J_2 \end{pmatrix} + \lambda \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix}.$$

Observe that J_1 and $I - c\,J_2$ are non-singular.

By multiplying with $\operatorname{diag}(J_1^{-1}, (I - c\,J_2)^{-1})$ one obtains

$$\begin{pmatrix} J_1^{-1}(I - c\,J_1) & 0 \\ 0 & I \end{pmatrix} + \lambda \begin{pmatrix} I & 0 \\ 0 & (I - c\,J_2)^{-1} J_2 \end{pmatrix}.$$

Let J be the Jordan canonical form of $J_1^{-1}(I - c\,J_1)$ and N the Jordan canonical form of $(I - c\,J_2)^{-1}J_2$. Then the corresponding similarity transformation leads to

$$\begin{pmatrix} J & 0 \\ 0 & I \end{pmatrix} + \lambda \begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix}.$$

Since all eigenvalues of $(I - c\,J_2)^{-1}J_2$ are 0, the special form of N follows. □

For

$$Q^{-1}u(t) = v(t) = \begin{pmatrix} y(t) \\ z(t) \end{pmatrix} \quad \text{and} \quad Pf(t) = \begin{pmatrix} g(t) \\ h(t) \end{pmatrix}$$

we obtain

$$y'(t) + Jy(t) = g(t), \tag{8.4}$$
$$Nz'(t) + z(t) = h(t). \tag{8.5}$$

The system (8.4) is an explicit ODE and possesses a unique solution for arbitrary initial values $y(0)$, see above.

The system (8.5) consists of m systems of the form

$$\begin{aligned} w_2'(t) + w_1(t) &= k_1(t), \\ &\vdots \\ w_\nu'(t) + w_{\nu-1}(t) &= k_{\nu-1}(t), \\ w_\nu(t) &= k_\nu(t). \end{aligned}$$

Observe that $w_1(t)$ appears only algebraically, while the derivatives of all other components appear in the system. With $\tilde{w}(t) = (w_2(t), \ldots, w_\nu(t))^T$ the system can be written in the following way (semi-explicit form):

$$\begin{aligned} \tilde{w}'(t) &= \tilde{f}(\tilde{w}(t), w_1(t)), \\ 0 &= \tilde{g}(\tilde{w}(t), w_1(t)). \end{aligned} \tag{8.6}$$

This form immediately reveals that the initial values w_0 in the initial condition

$$w(0) = w_0$$

must necessarily satisfy one algebraic condition (8.6) for $t = 0$, otherwise no solution exists.

However, this algebraic condition is not always sufficient for the existence of a solution, as the following more detailed analysis shows:

The system from above is equivalent to

$$\begin{aligned} w_\nu(t) &= k_\nu(t), \\ w_{\nu-1}(t) &= k_{\nu-1}(t) - k_\nu'(t), \\ &\vdots \\ w_1(t) &= k_1(t) - k_2'(t) + k_3''(t) - \ldots + (-1)^{\nu-1} k_\nu^{(\nu-1)}(t). \end{aligned}$$

So, it is uniquely solvable (without any prescribed initial values for $w(0)$). For $t = 0$ we have

$$\begin{aligned} w_\nu(0) &= k_\nu(0), \\ w_{\nu-1}(0) &= k_{\nu-1}(0) - k_\nu'(0), \\ &\vdots \\ w_1(0) &= k_1(0) - k_2'(0) + k_3''(0) - \ldots + (-1)^{\nu-1} k_\nu^{(\nu-1)}(0). \end{aligned}$$

So, in addition to (8.6), further $\nu - 1$ conditions must be satisfied for w_0, otherwise no solution exists. These additional conditions are called hidden constraints.

An immediate consequence of the representation of the solution from above is an estimation of the form

$$\|w(t)\| \leq C \left[\|k(t)\| + \|k'(t)\| + \ldots + \|k^{(\nu-1)}(t)\| \right].$$

In summary, in terms of the original quantities

$$\|u(t)\| \leq C \left[\|u(0)\| + \int_0^t \|f(s)\| \, ds \right.$$
$$\left. + \max_{0 \leq s \leq t} \|f(s)\| + \max_{s \in [0,t]} \|f'(s)\| + \ldots + \max_{s \in [0,t]} \|f^{(\nu-1)}(s)\| \right].$$

The highest derivative is of order $\nu - 1$ with $\nu = \max\{\nu_i : 1 \leq i \leq m\}$.

Definition 8.2. *The index ν of a linear system of DAEs with constant coefficients is given by*

$$\nu = \max_{1 \leq i \leq m} \nu_i.$$

Equivalent definition of ν: The matrix N is nilpotent with index ν, i.e.,

$$N^{\nu-1} \neq 0 \quad \text{and} \quad N^{\nu} = 0.$$

Special cases:

$\nu = 0$: explicit ODEs. There is a unique solution for arbitrary initial values $u(0)$. The solution can be estimated by the initial data and the L^1-norm of the right hand side.

$\nu = 1$: This is equivalent to $N = 0$. The transformed problem consists of an explicit ODE and a purely algebraic problem. The initial value u_0 must satisfy this algebraic constraint. For estimating the solution we additionally need the L^∞-norm of the right hand side.

$\nu > 1$: higher index DAEs. There are hidden constraints. For estimating the solution we additionally need the L^∞-norm of derivatives of the right hand side.

8.2 Differentiation Index and Perturbation Index

General implicit ODE:

$$F(t, u(t), u'(t)) = 0 \tag{8.7}$$

Definition 8.3. *The implicit ODE (8.7) has differentiation index ν_d if $m = \nu_d$ is the smallest integer such that the system*

$$\begin{aligned} F(t, u(t), u'(t)) &= 0, \\ \frac{d}{dt} F(t, u(t), u'(t)) &= 0, \\ &\vdots \\ \frac{d^m}{dt^m} F(t, u(t), u'(t)) &= 0, \end{aligned}$$

or, in short,

$$G(t, u, u', w) = 0 \quad \text{with } w = (u'', \ldots, u^{(m+1)}),$$

allows to extract an explicit ODE

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

by purely algebraic manipulations.

- Consider the linear system of DAEs with constant coefficients

$$\begin{aligned} w_2'(t) + w_1(t) &= k_1(t), \\ w_3'(t) + w_2(t) &= k_2(t), \\ &\vdots \\ w_\nu'(t) + w_{\nu-1}(t) &= k_{\nu-1}(t), \\ w_\nu(t) &= k_\nu(t). \end{aligned}$$

  If the first equation is differentiated once, the second twice, and so on, we obtain

$$\begin{aligned} w_2''(t) + w_1'(t) &= k_1'(t), \\ w_3'''(t) + w_2''(t) &= k_2''(t), \\ &\vdots \\ w_\nu^{(\nu)}(t) + w_{\nu-1}^{(\nu-1)}(t) &= k_{\nu-1}^{(\nu-1)}(t), \\ w_\nu^{(\nu)}(t) &= k_\nu^{(\nu)}(t). \end{aligned}$$

  Hence

$$w_1'(t) = k_1'(t) - k_2''(t) + k_3'''(t) - \ldots + (-1)^{\nu-1} k_\nu^{(\nu)}(t).$$

  Therefore: $\nu_d = \nu$.

- Consider implicit ODEs

$$F(t, u(t), u'(t)) = 0$$

  with non-singular $F_{u'}(t, u, u')$: By the implicit function theorem it follows:

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

  therefore $\nu_d = 0$.

- Consider the semi-explicit DAE

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ 0 &= g(t, y(t), z(t)) \end{aligned} \tag{8.8}$$

with non-singular matrix $g_z(t, y, z)$. If the second equation is differentiated once, we obtain

$$g_z(t, y(t), z(t))z'(t) + g_y(t, y(t), z(t))y'(t) + g_t(t, y(t), z(t)) = 0,$$

Hence

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ z'(t) &= -g_z(t, y(t), z(t))^{-1}[g_y(t, y(t), z(t))f(t, y(t), z(t)) + g_t(t, y(t), z(t))]. \end{aligned}$$

Therefore: $\nu_d = 1$. The original DAE is called a Hessenberg index-1 system.

- Consider a semi-explicit DAE of the form

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ 0 &= g(t, y(t)) \end{aligned} \tag{8.9}$$

with non-singular matrix $g_y(t, y)\, f_z(t, y, z)$. If the second equation is differentiated once, we obtain

$$g_y(t, y(t))y'(t) + g_t(t, y(t)) = 0$$

and, therefore, the new (hidden) constraint

$$g_y(t, y(t))f(t, y(t), z(t)) + g_t(t, y(t)) = 0.$$

If this constraint is differentiated once, we obtain an explicit system of ODEs, since $g_y(t, y(t))f_z(t, y(t), z(t))$ is non-singular. Therefore: $\nu_d = 2$. It a called a Hessenberg index-2 system.

- Similarly one can show that the semi-explicit DAE

$$\begin{aligned} x'(t) &= f(t, x(t), y(t), z(t)), \\ y'(t) &= g(t, x(t), y(t)), \\ 0 &= h(t, y(t)) \end{aligned} \tag{8.10}$$

with non-singular matrix $h_y(t, y)g_x(t, x, y)f_z(t, x, y, z)$: $\nu_d = 3$. It a called a Hessenberg index-3 system.

Example. *Constrained mechanical systems*

$$\begin{aligned} M(q)\ddot{q} &= f(q, \dot{q}) - G(q)^T\lambda, \\ 0 &= g(q). \end{aligned}$$

with $G(q) = g_q(q)$. We assume that the matrix $M(q)$ is symmetric and positive definite and $G(q)$ has full rank equal to its number of rows.

- *The system can be written as a system of first order*

$$\begin{aligned} M(q)\dot{u} &= f(q,u) - G(q)^T\lambda, \\ \dot{q} &= u, \\ 0 &= g(q). \end{aligned}$$

If the first equation is multiplied by $M(q)^{-1}$, the system is of the form (8.10) *with*

$$x = u, \quad y = q, \quad z = \lambda$$

and

$$f(t,x,y,z) = M(y)^{-1}\left(f(y,x) - G(y)^T z\right), \quad g(t,x,y) = x, \quad h(t,y) = g(y).$$

Then

$$h_y(t,y)g_x(x,y)f_z(x,y,z) = -G(q)M(q)^{-1}G(q)^T \equiv S(q).$$

Since $S(q)$ is non-singular, it is a Hessenberg index-3 DAE.

- *If the constraint is differentiated once, we obtain*

$$\begin{aligned} M(q)\dot{u} &= f(q,u) - G(q)^T\lambda, \\ \dot{q} &= u, \\ 0 &= G(q)u. \end{aligned}$$

If the first equation is multiplied by $M(q)^{-1}$, the system is of the form (8.9) *with*

$$y = \begin{bmatrix} u \\ q \end{bmatrix}, \quad z = \lambda$$

and

$$f(t,y,z) = \begin{bmatrix} M(y_2)^{-1}\left(f(y_2,y_1) - G(y_2)^T z\right) \\ y_1 \end{bmatrix}, \quad g(t,y) = G(y_2)y_1.$$

Then

$$g_y(y)\, f_z(y,z) = \begin{bmatrix} G(q) & * \end{bmatrix} \begin{bmatrix} -M(q)^{-1}G(q)^T \\ 0 \end{bmatrix} = -G(q)M(q)^{-1}G(q)^T = S(q).$$

Since $S(q)$ is non-singular, it is a Hessenberg index-2 DAE.

- *If the constraint is differentiated twice, we obtain*

$$\begin{aligned} M(q)\dot{u} &= f(q,u) - G(q)^T\lambda, \\ \dot{q} &= u, \\ 0 &= G(q)\dot{u} + g_{qq}(q)(u,u). \end{aligned}$$

If the first equation is multiplied by $M(q)^{-1}$, and if $\dot{u}$ is eliminated from the third equation, the system is of the form (8.8) *with*

$$y = \begin{bmatrix} u \\ q \end{bmatrix}, \quad z = \lambda,$$

and

$$f(t,y,z) = \begin{bmatrix} M(y_2)^{-1}\left(f(y_2,y_1) - G(y_2)^T z\right) \\ y_1 \end{bmatrix},$$
$$g(t,y,z) = G(y_2)M(y_2)^{-1}\left(f(y_2,y_1) - G(y_2)^T z\right) + g_{qq}(y_2)(y_1,y_1).$$

Then

$$g_z(y,z) = -G(q)M(q)^{-1}G(q)^T = S(q).$$

Since $S(q)$ is non-singular, it is a Hessenberg index-1 DAE.

Consider the general implicit ODEs:

$$F(t,u(t),u'(t)) = 0. \tag{8.11}$$

Definition 8.4. *The implicit ODE (8.11) has perturbation index ν_p with respect to a solution $u(t)$, $t \in [0,T]$ if $m = \nu_p$ is the smallest integer such that, for all functions $\hat{u}(t)$ with*

$$F(t,\hat{u}(t),\hat{u}'(t)) = \delta(t),$$

there exists a constant $C > 0$ with

$$\|\hat{u}(t) - u(t)\| \le C \Bigg[\|\hat{u}(0) - u(0)\| + \int_0^t \|\delta(s)\|\, ds + \max_{0\le s\le t} \|\delta(s)\| + \max_{s\in[0,t]} \|\delta'(s)\| + \ldots + \max_{s\in[0,t]} \|\delta^{(m-1)}(s)\| \Bigg]$$

for all $t \in [0,T]$ and all sufficiently small perturbations δ.

For

- linear systems of DAEs with constant coefficients,
- DAEs in Hessenberg form.

it follows that

$$\nu_p = \nu_d.$$

See also Gear [3] for a more detailed discussion of various definitions of the index of a DAE.

Chapter 9

Numerical Methods for Implicit ODEs

9.1 Runge-Kutta Methods

Runge-Kutta methods, so far considered only for explicit ODEs, can also be applied to implicit ODEs

$$F(t, u(t), u'(t)) = 0.$$

Approximations U_i to the solution $u(t_0 + c_i\,\tau)$ are introduced by

$$U_i = u_0 + \tau \sum_{j=1}^{s} a_{ij} U'_j, \tag{9.1}$$

where U'_j denotes approximations of $u'(t_0 + c_j\,\tau)$. U_i and U'_i are related by the algebraic equations

$$F(t_0 + c_i\tau, U_i, U'_i) = 0 \quad \text{for } i = 1, \ldots, s. \tag{9.2}$$

The approximation u_1 is then given by

$$u_1 = u_0 + \tau \sum_{i=1}^{s} b_i U'_i.$$

For explicit ODEs $u'(t) = f(t, u(t))$, (9.2) simplifies to $U'_i = f(t_0 + c_i\tau, U_i)$, $i = 1, \ldots, s$.

In each step the algebraic system (9.1), (9.2) must be solved, which can be reduced to a system in U'_i, $i = 1, \ldots, s$ only:

$$F(t_0 + c_i\tau, u_0 + \tau \sum_{j=1}^{s} a_{ij} U'_j, U'_i) = 0 \quad \text{for } i = 1, \ldots, s.$$

9.1.1 Application to Linear DAEs with Constant Coefficients

For the linear DAE with constant coefficients

$$Nz'(t) + z(t) = h(t)$$

with $N^\nu = 0$, $N^{\nu-1} \neq 0$ for $\nu \geq 1$ a Runge-Kutta method reads

$$Z_i = z_0 + \tau \sum_{j=1}^{s} a_{ij} Z'_j,$$

with

$$NZ'_i + Z_i = h(t_0 + c_i\tau), \quad i = 1, \ldots, s,$$

and

$$z_1 = z_0 + \tau \sum_{i=1}^{s} b_i Z'_i.$$

In each step the algebraic system

$$NZ'_i + z_0 + \tau \sum_{j=1}^{s} a_{ij} Z'_j = h(t_0 + c_i\tau), \quad i = 1, \ldots, s,$$

must be solved, i.e.,

$$\left[\begin{pmatrix} N & 0 & \cdots & 0 \\ 0 & N & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N \end{pmatrix} + \tau \begin{pmatrix} a_{11} I & a_{12} I & \cdots & a_{1s} I \\ a_{21} I & a_{22} I & \cdots & a_{2s} I \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} I & a_{s2} I & \cdots & a_{ss} I \end{pmatrix} \right] \begin{pmatrix} Z'_1 \\ Z'_2 \\ \vdots \\ Z'_s \end{pmatrix} = \begin{pmatrix} h(t_0 + c_1\tau) - z_0 \\ h(t_0 + c_2\tau) - z_0 \\ \vdots \\ h(t_0 + c_s\tau) - z_0 \end{pmatrix}$$

or, in short,

$$[I \otimes N + \tau\, A \otimes I] \begin{pmatrix} Z'_1 \\ Z'_2 \\ \vdots \\ Z'_s \end{pmatrix} = \begin{pmatrix} h(t_0 + c_1\tau) - z_0 \\ h(t_0 + c_2\tau) - z_0 \\ \vdots \\ h(t_0 + c_s\tau) - z_0 \end{pmatrix}.$$

In order to obtain a well-defined method, the matrix $I \otimes N + \tau\, A \otimes I$ must be non-singular.

Lemma 9.1. *The matrix $I \otimes N + \tau\, A \otimes I$ is non-singular if and only if A is non-singular.*

Proof. If A is singular, then there is a vector $y \neq 0$ with $Ay = 0$. Since N is singular, there is a vector $z \neq 0$ with $Nz = 0$. Then: $(I \otimes N + \tau\, A \otimes I)(y \otimes z) = y \otimes Nz + \tau\, Ay \otimes z = 0$.

Assume that A is non-singular. Because of

$$I \otimes N + \tau\, A \otimes I = [\tau\, A \otimes I] \left[I \otimes I + \frac{1}{\tau} A^{-1} \otimes N \right]$$

it follows that

$$\begin{aligned}
[I \otimes N + \tau\, A \otimes I]^{-1} &= \left[I \otimes I + \frac{1}{\tau}\, A^{-1} \otimes N \right]^{-1} [\tau\, A \otimes I]^{-1} \\
&= \sum_{k=0}^{\nu-1} \frac{(-1)^k}{\tau^k} \left[A^{-1} \otimes N \right]^k \frac{1}{\tau} \left[A^{-1} \otimes I \right] \\
&= \sum_{k=0}^{\nu-1} \frac{(-1)^k}{\tau^{k+1}} \left[A^{-(k+1)} \otimes N^k \right].
\end{aligned}$$

□

The requirement that A is non-singular, excludes the class of explicit Runge-Kutta methods.

Remark. *1. The Gauß methods, the Radau IA methods, the Radau IIA methods, and the Lobatto IIIC methods have a non-singular coefficient matrix $A = (a_{ij})$.*

2. For the Lobatto IIIA methods, the first row of $A = (a_{ij})$ vanishes. Nevertheless, it can be shown that these methods are also suitable for DAEs, since A is of the form

$$A = \left(\begin{array}{c|c} 0 & 0 \\ \hline \underline{a} & \underline{A} \end{array} \right)$$

with a non-singular matrix $\underline{A}$, if properly modified: Set Z_1' equal to Z_s' from the previous step and determine $(Z_2', \cdots, Z_s')$ from the reduced system obtained by ignoring the first equation. This approach requires an initial value for Z_1' at $t = 0$, e.g. $Z_1' = z'(0)$.

3. For the Lobatto IIIB methods, the last column of $A = (a_{ij})$ vanishes, these methods are not appropriate methods for DAEs.

9.1.2 Application to Semi-Explicit DAEs

The Runge-Kutta method applied to the semi-explicit DAE

$$y'(t) = f(t, y(t), z(t)), \tag{9.3}$$
$$0 = g(t, y(t), z(t)), \tag{9.4}$$

reads

$$Y_i = y_0 + \tau \sum_{j=1}^{s} a_{ij} Y_j', \quad Z_i = z_0 + \tau \sum_{j=1}^{s} a_{ij} Z_j'. \tag{9.5}$$

with

$$Y_i' = f(t_0 + c_i\tau, Y_i, Z_i), \quad 0 = g(t_0 + c_i\tau, Y_i, Z_i) \tag{9.6}$$

and

$$y_1 = y_0 + \tau \sum_{i=1}^{s} b_i Y_i', \tag{9.7}$$

$$z_1 = z_0 + \tau \sum_{i=1}^{s} b_i Z_i'. \tag{9.8}$$

Practical Implementation

In each step the algebraic systems (9.5) and (9.6) must be solved. Assume that A is non-singular. Then the quantities Z_i' can be eliminated. From

$$Z_i = z_0 + \tau \sum_{j=1}^{s} a_{ij} Z_j'$$

one obtains

$$Z = e \otimes z_0 + \tau \, [A \otimes I] Z' \quad \text{with} \quad Z = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_s \end{pmatrix}, \quad Z' = \begin{pmatrix} Z_1' \\ Z_2' \\ \vdots \\ Z_s' \end{pmatrix}.$$

The first equation implies:

$$Z' = \frac{1}{\tau}[A \otimes I]^{-1} \, (Z - e \otimes z_0) = \frac{1}{\tau}[A^{-1} \otimes I] \, (Z - e \otimes z_0) = \frac{1}{\tau}[A^{-1} \otimes I] Z - \frac{1}{\tau}[A^{-1} e \otimes z_0].$$

Hence

$$\begin{aligned} z_1 &= z_0 + \tau \sum_{i=1}^{s} b_i Z_i' = z_0 + \tau \, [b^T \otimes I] Z' = z_0 + \tau \, [b^T \otimes I] Z' \\ &= z_0 + [b^T \otimes I] \left([A^{-1} \otimes I] Z - [A^{-1} e \otimes z_0] \right) \\ &= (1 - b^T A^{-1} e) z_0 + [b^T A^{-1} \otimes I] Z \\ &= R(\infty) \, z_0 + [b^T A^{-1} \otimes I] \, Z \\ &= R(\infty) \, z_0 + \sum_{i=1}^{s} \widetilde{b}_i \, Z_i \quad \text{with} \quad (\widetilde{b}_1, \ldots, \widetilde{b}_s) = b^T A^{-1}. \end{aligned}$$

The quantities Y_i' can be eliminated as well. Since $Y_i' = f(t_0 + c_i \tau, Y_i, Z_i)$ it follows

$$y_1 = y_0 + \tau \sum_{i=1}^{s} b_i \, f(t_0 + c_j \tau, Y_j, Z_j).$$

So it remains to solve the following system of equations for Y_i, Z_i, $i = 1, \ldots, s$:

$$Y_i - y_0 - \tau \sum_{j=1}^{s} a_{ij} f(t_0 + c_j\tau, Y_j, Z_j), = 0 \tag{9.9}$$

$$g(t_0 + c_i\tau, Y_i, Z_i) = 0. \tag{9.10}$$

This system is typically solved by the simplified Newton method with the following approximation of the Jacobian:

$$J = \begin{pmatrix} I - \tau\, a_{11} f_y & -\tau a_{12} f_y & \cdots & -\tau a_{1s} f_y & -\tau\, a_{11} f_z & -\tau a_{12} f_z & \cdots & -\tau a_{1s} f_z \\ -\tau\, a_{21} f_y & I - \tau a_{22} f_y & \cdots & -\tau a_{2s} f_y & -\tau\, a_{21} f_z & -\tau a_{22} f_z & \cdots & -\tau a_{2s} f_z \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\tau\, a_{s1} f_y & -\tau a_{s2} f_y & \cdots & I - \tau a_{ss} f_y & -\tau\, a_{s1} f_z & -\tau a_{s2} f_z & \cdots & -\tau a_{ss} f_z \\ g_y & 0 & \cdots & 0 & g_z & 0 & \cdots & 0 \\ 0 & g_y & \cdots & 0 & 0 & g_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_y & 0 & 0 & \cdots & g_z \end{pmatrix}$$

with $f_y = f_y(t_0, y_0, z_0)$, $f_z = f_z(t_0, y_0, z_0)$, $g_y = g_y(t_0, y_0, z_0)$, $g_z = g_z(t_0, y_0, z_0)$. J is obtained from the Jacobian at $Y_i = y_0$ and $Z_i = z_0$, where all arguments $t_0 + c_i\,\tau$ are replaced by t_0. In short,

$$J = \begin{pmatrix} I \otimes I - \tau\, A \otimes f_y & -\tau\, A \otimes f_z \\ I \otimes g_y & I \otimes g_z \end{pmatrix}.$$

If A is non-singular, then the Jacobian can be transformed in the following way:

$$\begin{pmatrix} (\tau\, A)^{-1} \otimes I & \\ 0 & I \otimes I \end{pmatrix} J = \begin{pmatrix} (\tau\, A)^{-1} \otimes I - I \otimes f_y & -I \otimes f_z \\ I \otimes g_y & I \otimes g_z \end{pmatrix}$$

If A^{-1} is further transformed to a simple matrix (e.g., diagonal matrix, Jordan normal form) by

$$T^{-1} A^{-1} T = \Lambda,$$

then the Jacobian can be further transformed to

$$\begin{pmatrix} T^{-1}(\tau\, A)^{-1} \otimes I & \\ 0 & I \otimes I \end{pmatrix} J \begin{pmatrix} T \otimes I & \\ 0 & I \otimes I \end{pmatrix} = \begin{pmatrix} \tau^{-1}\, \Lambda \otimes I - I \otimes f_y & -I \otimes f_z \\ I \otimes g_y & I \otimes g_z \end{pmatrix} = hat J.$$

If, e.g., $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_s)$, then

$$\hat{J} = \begin{pmatrix} \tau^{-1}\lambda_1\, I - f_y & 0 & \cdots & 0 & -f_z & 0 & \cdots & 0 \\ 0 & \tau^{-1}\lambda_2\, I - f_y & \cdots & 0 & 0 & -f_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tau^{-1}\lambda_s\, I - f_y & 0 & 0 & \cdots & -f_z \\ g_y & 0 & \cdots & 0 & g_z & 0 & \cdots & 0 \\ 0 & g_y & \cdots & 0 & 0 & g_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_y & 0 & 0 & \cdots & g_z \end{pmatrix},$$

which (after reordering) consists of sub-matrices of the form

$$\begin{pmatrix} \tau^{-1}\,\lambda_i\,I - f_y & -f_z \\ g_y & g_z \end{pmatrix}.$$

In this case the original linear system of dimension $s\,N$ is reduced to s linear systems of dimension N. The computational costs of a direct solver like Gaussian elimination reduces from $(4/3)(sN)^3 = (4/3)s^3N^3$ to $s(4/3)(N)^3 = (4/3)sN^3$ elementary operations.

The ε-embedding method

Consider the singular perturbation problem

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ \varepsilon\, z'(t) &= g(t, y(t), z(t)), \end{aligned}$$

for $\varepsilon \neq 0$. This is an explicit ODE, for which the Runge-Kutta methods reads

$$Y_i = y_0 + \tau \sum_{j=1}^{s} a_{ij} Y'_j, \quad Z_i = z_0 + \tau \sum_{j=1}^{s} a_{ij} Z'_j.$$

with

$$Y'_i = f(t_0 + c_i\tau, Y_i, Z_i), \quad \varepsilon\, Z'_i = g(t_0 + c_i\tau, Y_i, Z_i)$$

and

$$y_1 = y_0 + \tau \sum_{i=1}^{s} b_i Y'_i, \quad z_1 = z_0 + \tau \sum_{i=1}^{s} b_i Z'_i.$$

The limit case $\varepsilon = 0$ leads to (9.5), (9.6), (9.7), (9.8).

The State Space Form

Observe that the intermediate approximations Y_i and Z_i satisfy the algebraic constraint

$$0 = g(t_0 + c_i\,\tau, Y_i, Z_i),$$

while the new approximate solutions (y_1, z_1) need not necessarily satisfy

$$0 = g(t_1, y_1, z_1) \quad \text{with } t_1 = t_0 + \tau, \tag{9.11}$$

despite the fact that:

$$0 = g(t_0, y_0, z_0).$$

As an alternative one could replace the computation of z_1 according to (9.8) by solving (9.11) for given $t_1 = t_0 + \tau$ and y_1.

The solvability of (9.11) with respect to z_1 is guaranteed (at least for sufficiently small step sizes τ) by the implicit function theorem, if the Jacobian $g_z(t_0, y_0, z_0)$ is non-singular, i.e., for Hessenberg index-1 systems. In this case we have a representation

$$z = G(t, y)$$

for the solution z of the equation

$$0 = g(t, y, z).$$

Then the semi-explicit DAE (9.3), (9.4) can be reduced to the so-called state space form

$$y'(t) = f(t, y(t), G(t, y(t))), \tag{9.12}$$

which is an explicit ODE.

The application of a Runge-Kutta method to (9.12) reads

$$Y_i = y_0 + \tau \sum_{j=1}^{s} a_{ij}\, Y_j' \quad \text{with} \quad Y_i' = f(t_0 + c_i\,\tau, Y_i, G(t_0 + c_i\,\tau, Y_i))$$

and

$$y_1 = y_0 + \tau \sum_{i=1}^{s} b_i\, Y_i'.$$

The advantage of this approach is that it is not restricted to implicit Runge-Kutta methods and the analysis is covered by the analysis for explicit ODEs.

It is easy to see that this approach leads to identical approximations Y_i, Y_i', Z_i, y_1, and z_1 as given by (9.5), (9.6), (9.7), and (9.11) with

$$Z_i = G(t_0 + c_i\,\tau, Y_i, \quad z_1 = G(t_0 + \tau, y_1).$$

As before the practical implementation requires to solve the system (9.9), (9.10), and, additionally, (9.11).

The derivation of a method via the ε-embedding is often called the direct approach and the derived methods are referred to direct methods. Sometimes the derivation of a method via the state space form is called the indirect approach and the derived methods are referred to as state space methods.

Definition 9.1. *A Runge-Kutta method is called stiffly accurate if and only if*

$$c_s = 1, \quad b_j = a_{sj}, \quad j = 1, 2, \ldots, s, \tag{9.13}$$

If a Runge-Kutta method is stiffly accurate, then $y_1 = Y_s$ and $z_1 = Z_s$. Therefore, the new approximate solutions satisfy the constraint (9.11), since the intermediate values Y_s and Z_s satisfy the algebraic constraint. So, for Hessenberg index-1 systems, the direct and the indirect approach coincide. The result that the algebraic constraint is satisfied for a stiffly accurate Runge-Kutta method is not restricted to Hessenberg index-1 systems.

Remark. *The Radau IIA methods, the Lobatto IIIA methods, and the Lobatto IIIC methods satisfy* (9.13).

9.2 BDF-Methods

A BDF-methods applied to implicit ODE

$$F(t, u(t), u'(t)) = 0$$

reads

$$F\left(t_{j+k}, u_{j+k}, \frac{1}{\tau}\sum_{i=1}^{k}\frac{1}{i}\nabla^i u_{j+k}\right) = 0.$$

9.2.1 Application to Linear DAEs with Constant Coefficients

In particular, we obtain for the linear DAE with constant coefficients

$$Nz'(t) + z(t) = h(t)$$

with $N^\nu = 0$, $N^{\nu-1} \neq 0$ for $\nu \geq 1$:

$$N\frac{1}{\tau}\sum_{i=1}^{k}\frac{1}{i}\nabla^i z_{j+k} + z_{j+k} = h(t_{j+k}).$$

Hence

$$\left(I + N\frac{1}{\tau}\sum_{i=1}^{k}\frac{1}{i}\nabla^i\right) z_{j+k} = h(t_{j+k}).$$

Therefore

$$z_{j+k} = \left(I + N\frac{1}{\tau}\sum_{i=1}^{k}\frac{1}{i}\nabla^i\right)^{-1} h(t_{j+k}) = \sum_{\ell=0}^{\nu-1}(-1)^\ell N^\ell \left(\frac{1}{\tau}\sum_{i=1}^{k}\frac{1}{i}\nabla^i\right)^\ell h(t_{j+k})$$

for $j + k \geq (\nu - 1)\, k$.

This shows that the method is well-defined and that z_{j+k} depends only on the values of the right-hand side $h(t)$ at t_{j+k} and at the previous $(\nu - 1)\, k$ grid points.

For the exact solution we have a similar representation:

$$z(t) = \left(I + N\frac{d}{dt}\right)^{-1} h(t) = \left(\sum_{\ell=0}^{\nu-1}(-1)^\ell N^\ell \frac{d^\ell}{dt^\ell}\right) h(t) = \sum_{\ell=0}^{\nu-1}(-1)^\ell N^\ell h^{(\ell)}(t).$$

9.2.2 Application to Semi-Explicit DAEs

If a BDF-method is applied to

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ 0 &= g(t, y(t), z(t)), \end{aligned}$$

then

$$\sum_{i=1}^{k} \frac{1}{i} \nabla^i y_{j+k} = \tau \, f(t_{j+k}, y_{j+k}, z_{j+k}),$$
$$0 = g(t_{j+k}, y_{j+k}, z_{j+k}).$$

As for Runge-Kutta methods this method coincides with the BDF-method applied to the singular perturbation problem in the limit case $\varepsilon = 0$.

It is easy to see that the BDF-method applied to the state space form (9.12) leads to identical approximations with

$$z_{j+k} = G(t_{j+k}, y_{j+k}).$$

Chapter 10

Hessenberg Index-1 DAEs

For a Hessenberg index-1 system

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ 0 &= g(t, y(t), z(t)) \end{aligned}$$

we have that

$$g_z(t, y, z) \quad \text{is non-singular in a neighborhood of the solution.}$$

Then, by the implicit function theorem, we have locally:

$$z(t) = G(t, y(t))$$

and obtain the reduced problem (state space form):

$$y'(t) = f(t, y(t), G(t, y(t)). \tag{10.1}$$

10.1 The Nonlinear System

For both a Runge-Kutta method (9.5), (9.6), (9.7), (9.8) as well as a state space method (9.5), (9.6), (9.7), (9.11) the following (in general, nonlinear) system of equations must be solved:

$$\begin{aligned} Y_i - y_0 - \tau \sum_{j=1}^{s} a_{ij} f(t_0 + c_j\tau, Y_j, Z_j) = 0, \\ g(t_0 + c_i\tau, Y_i, Z_i) = 0. \end{aligned}$$

As we have seen, the discussion of the Jacobian of this system can be reduced to the discussion of matrices of the form

$$\begin{pmatrix} \lambda_i\, I - \tau\, f_y & -\tau\, f_z \\ g_y & g_z \end{pmatrix}.$$

For the BDF-methods the following (in general, nonlinear) system of equations must be solved:

$$\sum_{i=1}^{k} \frac{1}{i} \nabla^i y_{j+k} = \tau \, f(y_{j+k}, z_{j+k}), \tag{10.2}$$

$$0 = g(y_{j+k}). \tag{10.3}$$

The Jacobian of this system is of the same form as above. It is easy to see that the Jacobian is non-singular for $\tau = 0$ for each of these classes of methods. Then the implicit function theorem ensures the existence of a locally unique solution of the nonlinear system, provided y_0 and z_0 are sufficiently close to $y(t_0)$ and $z(t_0)$, respectively. Moreover, it can be shown that the simplified Newton method converges to this solution.

10.2 Summary of Convergence Results

Any method appropriate for explicit ODEs can be applied to (10.1). As a consequence, the following estimate for the global error holds

$$\|y_j - y(t_j)\| = O(\tau^p),$$

where p is the order of the method if applied to and explicit ODEs, and

$$\|z_j - z(t_j)\| = \|G(t_j, y_j) - G(t_j, y(t_j))\| = O(\|y_j - y(t_j)\|) = O(\tau^p),$$

if G is sufficiently smooth. This applies to Runge-Kutta methods of order p as well as to the k-step BFD-method, which is of order $p = k$, if the initial values satisfy

$$\|y_j - y(t_j)\| = O(\tau^k), \qquad \text{for all } j = 0, \ldots, k-1.$$

So, for state space methods applied to Hessenberg index-1 DAEs the convergence order of both $y(t)$ and of $z(t)$ is the same as for explicit ODEs.

The situation is more involved for the algebraic variable $z(t)$. Some of the known results are summarized in Table 10.1, where the second column contains the number of stages, the third column the convergence order of $u(t)$ for explicit ODEs, and the last column the convergence order of $y(t)$ and $z(t)$ for Hessenberg index 1- DAEs.

method	stages	explicit ODE	Hessenberg index-1 DAE		
		$u(t)$		$y(t)$	$z(t)$
Gauß	s	$2s$	s even:	$2s$	s
			s odd:	$2s$	$s+1$
Radau IA	$s-1$	$2s-1$		$2s-1$	s
Radau IIA	s	$2s-1$		$2s-1$	$2s-1$
Lobatto IIIA	s	$2s-2$		$2s-2$	$2s-2$
Lobatto IIIC	$s-1$	$2s-2$		$2s-2$	$2s-2$

Table 10.1: Convergence order Runge-Kutta methods for Hessenberg index-1 DAEs

Since the direct approach of a Runge-Kutta methods and the application of the same Runge-Kutta method for the state space form lead to the same approximations for $y(t)$, it follows that the convergence order of $y(t)$ is the same as for explicit ODEs. There is no order reduction for $z(t)$ for RadauIIA, Lobatto IIIA, and Lobatto IIC. These methods are stiffly accurate and, therefore, produce the same approximations for $z(t)$ as if applied to the state space form.

Chapter 11

Hessenberg Index-2 DAEs

For a Hessenberg index-2 DAE

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)), \\ 0 &= g(t, y(t)) \end{aligned}$$

we have that

$$g_y(t, y) f_z(t, y, z) \quad \text{is non-singular in a neighborhood of the solution,}$$

and that the following hidden algebraic constraint must hold:

$$0 = g_y(t, y(t))\, f(t, y(t), z(t)).$$

11.1 The Nonlinear System

The discussion of the solvability of the (in general, nonlinear) systems of equation for a Runge-Kutta method and for a BDF-method is more complicated. The existence of a locally unique solution and the convergence of the simplified Newton method to this solution can be shown.

11.2 Summary of Convergence Results

For BDF-methods it can be shown that

$$\|y_i - y(t_i)\| = O(\tau^k), \quad \|z_i - z(t_i)\| = O(\tau^k) \qquad \text{for all } i = k, k+1, \ldots.$$

if the initial values satisfy

$$\|y_i - y(t_i)\| = O(\tau^{k+1}) \qquad \text{for all } i = 0, \ldots, k-1.$$

Some of the known results for Runge-Kutta methods are summarized in Table 11.1.

method	stages	explicit ODE	Hessenberg index-1 DAE		
		$u(t)$		$y(t)$	$z(t)$
Gauß	s	$2s$	s odd:	$s+1$	$s-1$
			s even:	s	$s-2$
Radau IA	$s-1$	$2s-1$		s	$s-1$
Radau IIA	s	$2s-1$		$2s-1$	s
Lobatto IIIA	s	$2s-2$	s odd:	$2s-2$	$s-1$
			s even:	$2s-2$	s
Lobatto IIIC	$s-1$	$2s-2$		$2s-2$	$s-1$

Table 11.1: Convergence order Runge-Kutta methods for Hessenberg index-2 DAEs

See [10] and [8] for more details on the last two chapters.

Bibliography

[1] Uri M. Ascher and Linda R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations.* Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics, 1998.

[2] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations.* Classics in Applied Mathematics. 14. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics, 1996.

[3] Stephen L. Campbell and C.William Gear. The index of general nonlinear DAEs. *Numer. Math.*, 72(2):173–196, 1995.

[4] Peter Deuflhard and Folkmar Bornemann. *Numerische Mathematik. 2: Gewöhnliche Differentialgleichungen.* de Gruyter Lehrbuch. Berlin: de Gruyter, 2002.

[5] Griepentrog, Eberhard and März, Roswitha. *Differential-algebraic equations and their numerical treatment.* Teubner-Texte zur Mathematik, Bd. 88. Leipzig: BSB B. G. Teubner Verlagsgesellschaft, 1986.

[6] R.D. Grigorieff. *Numerik gewöhnlicher Differentialgleichungen. Band 1: Einschrittverfahren.* Teubner Studienbücher. Stuttgart: B. G. Teubner, 1972.

[7] Rolf Dieter Grigorieff and Hans Joachim Pfeiffer. *Numerik gewöhnlicher Differentialgleichungen. Band 2: Mehrschrittverfahren.* Teubner-Studienbücher: Mathematik. Stuttgart: B.G. Teubner, 1977.

[8] Ernst Hairer, Christian Lubich, and Michel Roche. The numerical solution of differential-algebraic systems by Runge-Kutta methods. Lecture Notes in Mathematics, 1409. Berlin etc.: Springer-Verlag. vii, 139 p. DM 25.00 (1989)., 1989.

[9] Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving ordinary differential equations. I: Nonstiff problems. 2nd rev. ed.* Springer Series in Computational Mathematics. 8. Berlin: Springer, 1993.

[10] Ernst Hairer and Gerhard Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems. 2nd rev. ed.* Springer Series in Computational Mathematics. 14. Berlin: Springer, 1996.

[11] H.-O. Kreiss. Über die Stabilitätsdefinition für Differenzengleichungen, die partielle Differentialgleichungen approximieren. *BIT*, 2:153–181, 1962.

[12] März, Roswitha. Numerical methods for differential algebraic equations. *Acta Numerica*, pages 141–198, 1992.

[13] L.F. Shampine and M.K. Gordon. *Computer solution of ordinary differential equations. The initial value problem.* San Francisco: W. H. Freeman & Comp., 1975.