CHAPTER 1. INITIAL VALUE PROBLEMS

Analysis of multistep methods

We follow the lines and definitions of one step methods, i.e. we define first the local residual and the global error increment, then consistency, look at zero-stability, and give a condition for convergence (Lax principle again).

The local error for multistep methods is defined again as

\[ e_n := y_n - u_n. \]  

(1.33)

This is the quantity of interest in practical computations and ideally, one would like to directly control it, i.e. to assure that it is limited by a given error bound TOL.

Unfortunately this is not so easy as it highly depends on the dynamic properties of the differential equation at hand and those are normally not known. Therefore one has to control local quantities, which can be estimated during the computational process.

**Definition 29** The local residual \( l(y, t, h) \) is defined as

\[
y(t+h) + \sum_{i=1}^{k} \alpha_{k-i} y(t-(i-1)h) - h \sum_{i=0}^{k} \beta_{k-i} f(t-(i-1)h, y(t-(i-1)h) = l(y, t, h) \]

with \( y \in C^1[t_{n+1-k}, t_{n+1}] \).

If \( y \) is sufficiently smooth, the local residual can be expressed in terms of powers of \( h \) by Taylor expansion.

**Example 30** The local residual of the two-step implicit Adams method is defined by

\[
l(y, t, h) = y(t+h) - y(t) - h \left( \frac{5}{12} \dot{y}(t+h) + \frac{8}{12} \dot{y}(t) - \frac{1}{12} \ddot{y}(t-h) \right). \]

Taylor expansion leads to

\[
l(y, t, h) = h \ddot{y}(t) + \frac{1}{2} h^2 \dddot{y}(t) + \frac{1}{6} h^3 \ddddot{y}(t) + \frac{1}{24} h^4 y^{(4)}(t) + \ldots
- \frac{5}{12} h \ddot{y}(t) - \frac{5}{12} h^2 \dddot{y}(t) - \frac{5}{24} h^3 y^{(3)}(t) - \frac{5}{72} h^4 y^{(4)}(t) + \ldots
- \frac{8}{12} h \dddot{y}(t)
+ \frac{1}{12} h \ddot{y}(t) - \frac{1}{12} h^2 \dddot{y}(t) + \frac{1}{24} h^3 y^{(3)}(t) - \frac{1}{72} h^4 y^{(4)}(t) + \ldots
= -\frac{1}{24} h^4 y^{(4)}(t) + \mathcal{O}(h^5)
= : CAM_4^4
\]
1.4. MULTISTEP METHODS

We define consistency of the method like in the one-step case, i.e. a method is called consistent if
\[ l(y, t, h) = C_{p+1} h^{p+1} y^{(p+1)}(t) + \mathcal{O}(h^{p+2}) \]
with \( p \geq 1 \) and \( p \) is called the order of consistency. The constant \( C_{p+1} \) is called the error constant, which is known for the particular method from Taylor expansion (see example above). The unknown quantity in this expression is the \( p + 1 \)st derivative of \( y \).

We will see later in this section how this quantity can be estimated, when \( y \) is the exact solution of the ODE. First we have to relate the local residual to the global error.

If we insert in (1.34) the exact solution \( y \) of the ODE, set \( t = t_n \), and subtract from this expression (1.29) we obtain the following expression
\[
y_{n+1} - u_{n+1} + \sum_{i=1}^{k} \alpha_{k-i} (y_{n+1-i} - u_{n+1-i})
- h \sum_{i=0}^{k} \beta_{k-i} (f(t_{n+1-i}, y_{n+1-i}) - f(t_{n+1-i}, u_{n+1-i}))
= l(y, t_n, h) \quad (1.35)
\]

By applying the mean value theorem we can simplify the expressions for \( f \) by
\[
f(t_{n+1-i}, y_{n+1-i}) - f(t_{n+1-i}, u_{n+1-i}) = f'_y(t_{n+1}, \xi_{n+1-i}) (y_{n+1-i} - u_{n+1-i}),
\]
where \( f'_y \) denotes the derivative of \( f \) with respect to its second argument.

By inserting the expression for the global error we finally get a recursion formula for the global error
\[
e_{n+1} = - \sum_{i=1}^{k} a_{-1}^i a_{k-i} e_{n+1-i} + a_{-1}^k l(y, t_n, h)
\quad (1.36)
\]
with \( a_i := \alpha_i - h \beta_i f'_y(t_{n+1-i}, \xi_{n+1-i}) \).

The driving term in this recursion, \( \epsilon_n := a^{-1}_k l(y, t_n, h) \) is called the global error increment.

The recursion for the global error is a \( k \)-th order difference equation with non constant coefficients (they depend on \( n \)). This recursion can be written in first-order form, c.f. (1.26):
\[
\begin{pmatrix}
\epsilon_{n+1} \\
e_n \\
\vdots \\
\epsilon_{n+k-1} \\
e_{n+2k-1}
\end{pmatrix}
= \Phi_n(h)
\begin{pmatrix}
e_{n} \\
e_{n-1} \\
\vdots \\
e_{n+k-1} \\
e_{n+2k-1}
\end{pmatrix}
+ \begin{pmatrix}
\epsilon_{n} \\
0 \\
\vdots \\
0 \\
0
\end{pmatrix}
\quad (1.37)
\]
A method is convergent, if \( \max |e_i| \to 0 \) with \( h \to 0 \). In order to achieve convergence, we have to ask for

- **consistency**, i.e. the driving term \( \epsilon_n = \mathcal{O}(h^{p+1}) \),
- \( E_0 = \mathcal{O}(h^p) \), and
- **zero stability**, i.e. \( \Phi(0) \) satisfies the conditions of Theorem 26.

Note, that \( \Phi(0) \) depends only on the coefficients of the \( \rho \) polynomials and in particular not on the differential equation itself. Obviously, all Adams methods are zero stable. For BDF-k methods it can be shown, that they are zero stable up to \( k = 6 \).

### 1.4.1 Error Estimation for Multistep Methods

Numerical methods for solving ODEs are nowadays mostly applied with step size control, i.e. the step size is not kept constant during the integration process. Controlling the step size requires the following steps

- Estimating the global error increment in every step.
- Computing an optimal step size by comparing this estimate to a given tolerance bound.
- Adjusting the coefficients and data representation of the method according to a variable grid.

We will not discuss here the last point of this list as it requires many details and techniques which are beyond the scope of this course. Nevertheless it should be noted, that this is the point, which makes the implementation of multistep methods a rather sophisticated task, which is still more than 100 years after the development of the first multistep method a topic of research.

There are different ways to estimate the global error increment. We will consider here a technique, which is called Milne’s device. We assume, that we have computed a step with two different multistep methods of the **same** order starting from the same data. We will indicate the different methods by superscripts (1) and (2):

First, we will consider two methods of order \( p \), distinguished in notation by superscripts (1) and (2). From the definition of the global error increment \( \epsilon \) and the properties of the local residual error \( l(y, t_n, h) \) we get

\[
\epsilon_n^{(1)} - \epsilon_n^{(2)} = \left( a_k^{(1)} - a_k^{(2)} \right) C_{p+1}^{(1)} - a_k^{(2)} C_{p+1}^{(2)} h^{p+1} y^{(p+1)}(t_n) + \mathcal{O}(h^{p+2}).
\] (1.38)
Let $\bar{u}_n^{(i)}$ be the numerical solution of method $i$ after one step initialized with exact values, i.e.

$$
0 = \bar{u}_n^{(i)} - h\beta_k^{(i)} f(t_n, \bar{u}_n) + \sum_{i=1}^{k} \alpha_k^{(i)} y(t_{n-i}) - h \sum_{i=1}^{k} \beta_k^{(i)} f(t_{n-i}, y(t_{n-i})),
$$

then it is easy to show, that

$$
\epsilon_n^{(i)} := y(t_n) - \bar{u}_n^{(i)} + \mathcal{O}(h^{p+1}).
$$

Consequently,

$$
\epsilon_n^{(1)} - \epsilon_n^{(2)} := \bar{u}_n^{(2)} - \bar{u}_n^{(1)} + \mathcal{O}(h^{p+1})
$$

and finally

$$
h^{p+1} y^{(p+1)}(t_n) = \left( a_k^{(1)} C_{p+1}^{(1)} - a_k^{(2)} C_{p+1}^{(2)} \right)^{-1} \left( \bar{u}_n^{(2)} - \bar{u}_n^{(1)} \right) + \mathcal{O}(h^{p+2}).
$$

Thus by comparing $\bar{u}_n^{(1)}$ and $\bar{u}_n^{(2)}$ we obtain an estimate for $h^{p+1} y^{(p+1)}(t_n)$ and can use this to estimate $\epsilon_n^{(2)}$:

$$
\epsilon_n^{(2)} = a_k^{(2)} C_{p+1}^{(2)} \left( a_k^{(1)} C_{p+1}^{(1)} - a_k^{(2)} C_{p+1}^{(2)} \right)^{-1} \left( \bar{u}_n^{(2)} - \bar{u}_n^{(1)} \right).
$$

Let the method denoted by the superscript (1) be the explicit predictor method, then $a_k^{(1)} = 1$. We obtain then

$$
\epsilon_n^{(2)} = \left( \frac{C_{p+1}^{(1)}}{C_{p+1}^{(2)}} a_k^{(2)} - 1 \right)^{-1} \left( \bar{u}_n^{(2)} - \bar{u}_n^{(1)} \right)
$$

$$
= \frac{C_{p+1}^{(2)}}{a_k^{(1,2)} (C_{p+1}^{(1)} - C_{p+1}^{(2)})} \left( \bar{u}_n^{(2)} - \bar{u}_n^{(1)} \right)
$$

with

$$
a_k^{(1,2)} := \left( 1 - h\beta_k^{(2)} C_{p+1}^{(1)} - C_{p+1}^{(2)} f_y'(t_n, \xi_n) \right).
$$

For $h$ small compared to $f_y'$ we can also set approximately $a_k^{(1,2)} \approx 1$ and get a simpler error estimator:

$$
\epsilon_n^{(2)} = \frac{C_{p+1}^{(2)}}{C_{p+1}^{(1)} - C_{p+1}^{(2)}} \left( \bar{x}_n^{(2)} - \bar{x}_n^{(1)} \right) + \mathcal{O}(h^{p+2}).
$$

In practice the values $\bar{u}_n$ cannot be computed. When estimating the error they are replaced by $u_n$, so that the error estimation is based on the predictor - corrector difference.
To check the success of a step the estimated error is checked against a given tolerance
\[
|\epsilon_n| \leq TOL. \quad (1.42)
\]
The tolerance is usually a combination of a relative tolerance bound $RTOL$ and an absolute tolerance bound $ATOL$ by setting:
\[
TOL := RTOL|u(t)| + ATOL.
\]
If the components of the solution $u$ differ very much in magnitude $ATOL$ is given as a vector.
If the error criterion (1.42) is not met, the step is rejected and will be recomputed with a smaller step size. If on the other hand this criterion is fulfilled, it will be checked if it would have been fulfilled even with a larger step size. If so, and if the gain is significant, the larger step size is taken for the next step.
The step size $\bar{h}_n$ corresponding to $TOL$ is computed from the actual step size $h_n$ and the error estimate by
\[
\bar{h}_n = \left( \frac{TOL}{|\epsilon_n|} \right)^{\frac{1}{p+1}} h_n.
\]
The result is often multiplied with an additional safety factor.
We will see an example how this works in the homework.

1.5 Implicit Runge–Kutta Methods

In the basic course you met explicit Runge–Kutta methods and their implementation with variable step size. Here we introduce the concept of implicit Runge–Kutta methods of collocation type without aiming for completeness. The main reason for this is to show the concept of collocation methods. Runge-Kutta methods of the Gauss type have certain advantages when dealing with boundary value problems due to their symmetry with respect to time.

1.5.1 Collocation Methods

**Definition 31** The polynomial $u$ of degree $s$ defined by the conditions
\[
\begin{align*}
    u(t_n) &= x_n \quad (1.43a) \\
    \dot{u}(t_n + c_i h) &= f(t_n + c_i h, u(t_n + c_i h)) \quad (1.43b)
\end{align*}
\]
and given distinct values $c_i \in [0,1], i = 1, \ldots, s$ and a step size $h$ is called a collocation polynomial of the differential equation $y' = f(t, y), y(t_n) = y_n$. The $c_i$ are called collocation points.
1.5. IMPLICIT RUNGE–KUTTA METHODS

The idea of collocation methods is to approximate $y(t_{n+1})$ by $u_{n+1} := u(t_n + h)$. $\dot{u}$ can be expressed using the Lagrange formulation of polynomials (see basic course)

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

with

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

We get by integration

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

with

$$a_{ij} = \int_0^{c_i} l_j(\theta) d\theta. \quad (1.44)$$

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

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j) \quad i = 1, \ldots, s \quad (1.45a)$$

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(t_n + c_i h, Y_i) \quad (1.45b)$$

with

$$b_i = \int_0^{1} l_i(\theta) d\theta. \quad (1.46)$$

A method defined by (1.45) is called an implicit Runge–Kutta method. The stages $Y_i$ are defined here implicitly.

Like explicit Runge–Kutta methods, the implicit methods can be represented by a Butcher tableau:

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

with $c = (c_1, \ldots, c_s)$, $b = (b_1, \ldots, b_s)$, and $A = (a_{ij})_{i,j=1,\ldots,s}$.

Collocation methods can be related to quadrature formulas by applying them to the special case: $\dot{y}(t) = f(t), y(t_n) = y_n$ or, equivalently, to

$$y(t_{n+1}) = y_n + \int_{t_n}^{t_{n+1}} f(\theta) d\theta.$$


We define \( I_{n+1}^n(f) := y_n + \int_{t_n}^{t_{n+1}} f(\theta) d\theta \) and \( \tilde{I}_{n+1}^n(f) := y_{n+1} \) with \( y_{n+1} \) given by (1.45b). The collocation method is then constructed in such a way that for a \( k \) as large as possible the following requirement is met:

\[
I_{n+1}^n(\pi) = \tilde{I}_{n+1}^n(\pi) \quad \text{for all polynomials } \pi \text{ up to degree } k.
\]

This requirement defines the collocation points \( c_i \) and by (1.44) and (1.46) also the other Runge–Kutta coefficients \( a_{ij} \) and \( b_i \). The maximal degree \( k \) can be achieved by the so-called \textit{Gauß points}, where \( k = 2s - 1 \). The \( c_i \) are the roots of the \textit{Legendre polynomials}. For \( s = 3 \) these coefficients are \( c_1 = 1/2 - \sqrt{15}/10 \), \( c_2 = 1/2 \), and \( c_3 = 1/2 + \sqrt{15}/10 \). This gives the 3-stage \textit{Gauß integration method}:

\[
\begin{array}{cccc}
\frac{1}{2} - \frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\
\frac{1}{2} & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\
\frac{1}{2} + \frac{\sqrt{15}}{10} & \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \\
\end{array}
\]

By selecting the collocation points as roots of the polynomial

\[
\frac{d^{s-1}}{dt^{s-1}} \left( t^{s-1}(t - 1)^s \right)
\]

one obtains the class of Radau IIa methods. A widely used implementation of a three stage Radau method is the code RADAU5 by Hairer [HW96]. Its coefficients are

\[
\begin{array}{cccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\end{array}
\]

(1.47)
Bibliography


Index

A-stable, 10
absolute stability, 10
Adams–Moulton methods, 17
autonomous, 4
backward difference formulas, 17
backward difference operator, 16
BDF, 17
Cauchy problems, 1
characteristic polynomial, 13
collocation, 22, 23
point, 22
polynomial, 23
consistent, 6
difference equation, 11
homogene, 11
error constant, 19
evolution of the differential equation, 4
Explicit Euler Method, 5
flow of the differential equation, 4
forward shift operator, 16
fundamental solutions, 13
Gauß
integration method, 24
points, 24
generating functions, 17
global error, 6
global error increment, 6, 19
Gronwall’s Lemma, 3
Heun’s method, 5
Implicit Euler Method, 5
implicit Runge–Kutta method, 23
increment function, 5
initial value problem, 1
Lax-Richtmyer principle, 9
Legendre polynomials, 24
linear multistep method, 15
linear test equation, 10
Lipschitz continuous, 2
Milne’s device, 20
multistep method, 15
one step method, 5
order of consistency, 6, 19
region of absolute stability, 10
Stability
of an initial value problem, 3
of one step methods, 6
zero stable, 7
stability
absolute, 10
stability bound, 10
state variable, 1
step size, 5
Trapezoidal Rule, 5
zero stable, 7