Lecture Notes

July 6, 2015

Abstract

These lecture notes collect some useful material for the course on Structure-Preserving integrators at the Atlantic Association for Research in the Mathematical Sciences - SUMMER SCHOOL Dalhousie University, Halifax, Nova Scotia (Canada), July 2015.

Contents

1 Background

In this section some useful notation on ordinary differential equations (ODEs) will be presented. We will also recall some classical existence and uniqueness results for solutions of ordinary differential equations.

A system of m first order ordinary differential equation is given by

$$y' = f(t, y) \tag{1}$$

or, written out, as

$$\begin{aligned} y_1' &= f_1(t, y_1, \cdots, y_m), \\ y_2' &= f_2(t, y_1, \cdots, y_m), \\ &\vdots \\ y_m' &= f_m(t, y_1, \cdots, y_m). \end{aligned}$$

This is an *initial value problem* (IVP) if the solution is given at some point t_0, thus

$$y_1(t_0) = y_{1,0},\ y(t_0) = y_{2,0},\ \cdots\ y_m(t_0) = y_{m,0}.$$

Example 1.1. *The following equation is an example of the Lotka-Volterra equation:*

$$\begin{aligned} y_1' &= y_1 - y_1 y_2, \\ y_2' &= y_1 y_2 - 2y_2. \end{aligned}$$

An ODE is called *autonomous* if f is not a function of t, but only of y. The Lotka-Volterra equation is an example of an autonomous ODE. A nonautonomous system can be made autonomous by a simple trick, just add the equation

$$y_{m+1}' = 1, \qquad y_{m+1}(t_0) = t_0,$$

and replace t with y_{m+1}. Also higher order ODE/IVPs

$$u^{(m)} = f(t, u, u', \cdots, u^{(m-1)}), \qquad u(t_0) = u_0,\ u'(t_0) = u_0',\ \cdots, u^{(m-1)}(t_0) = u_0^{(m-1)},$$

where $u^{(m)} = d^m u/dt^m$, can be written as a system of first order equations, again by a simple trick: Let

$$y_1 = u,\ y_2 = u',\ \cdots y_m = u^{(m-1)},$$

and we get the system

$$\begin{aligned} y_1' &= y_2, & y_1(t_0) &= u_0, \\ y_2' &= y_3, & y_2(t_0) &= u_0', \\ &\vdots & &\vdots \\ y_{m-1}' &= y_m, & y_{m-1}(t_0) &= u_0^{(m-2)}, \\ y_m' &= f(t, y_1, y_2, \cdots, y_m), & y_m(t_0) &= u_0^{(m-1)}. \end{aligned}$$

Example 1.2. *Van der Pol's equation is given by*

$$u'' + \mu(u^2 - 1)u' + u = 0.$$

Using $y_1 = u$ and $y_2 = u'$ this equation can be rewritten as

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1. \end{aligned}$$

This problem was first introduced by Van der Pol in 1926 in the study of an electronic oscillator.

1.1 Existence and uniqueness of solutions

We here present some existence and uniqueness results for solution of ODEs.

Definition 1.3. *A function $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ satisfies the* Lipschitz condition *with respect to y on a domain $(a, b) \times D$ where $D \subset \mathbb{R}^m$ if there exist a constant $L > 0$ and a norm $|\cdot|$ of $\mathbb{R}^m$ such that*

$$|f(t, y) - f(t, \tilde{y})| \leq L|y - \tilde{y}|, \qquad \text{for all} \quad t \in [a, b],\ y, \tilde{y} \in D.$$

The constant L is called the Lipschitz constant.

It is not hard to show that the function f satisfies the Lipschitz condition if $\partial f_i/\partial y_j,\ i, j = 1, \cdots, m$ are continuous and bounded on the domain.

Theorem 1.4. *Consider the initial value problem*

$$y' = f(t, y), \qquad y(t_0) = y_0. \tag{2}$$

If

1. *$f(t, y)$ is continuous in $[a, b] \times \mathbb{R}^m$,*
2. *$f(t, y)$ satisfies the Lipschitz condition with respect to y in $[a, b] \times \mathbb{R}^m$.*

with given initial values $t_0 \in (a, b)$ and $y_0 \in \mathbb{R}^m$, then (2) has one and only one solution in $y \in C^1([a, b] \times \mathbb{R}^m)$.

Proof. Let $t_0 = a$. The initial value problem (2) is equivalent to

$$y(t) = y_0 + \int_{t_0}^{t} f(x, y(x))\, dx, \quad t_0 \leq t \leq t \tag{3}$$

which is an integral equation in $C^0([a, b], \mathbb{R}^m)$. Divide $[a, b]$ in N intervals of size $h = \frac{a-b}{N}$, and such that

$$L\, h < 1.$$

Take $t_n = t_0 + nh$, $n = 1, 2, \ldots, N$. Consider the operator

$$\Phi_n : C^0([t_{n-1}, t_n], \mathbb{R}^m) \times \mathbb{R}^m \to C^0([t_{n-1}, t_n], \mathbb{R}^m)$$

defined by

$$\Phi_n(y,u) := u + \int_{t_{n-1}}^{t} f(x,y(x))\,dx, \quad t_{n-1} \le t \le t_n.$$

We will prove that $\forall u \in \mathbb{R}^m$ the following operator

$$\Phi_{n,u} : y \to \Phi_n(y,u)$$

on $C^0([t_{n-1},t_n],\mathbb{R}^m)$ is a contraction in the norm

$$\|y\|_\infty := \max_{t_{n-1}\le t\le t_n} |y(t)|,$$

where we denote by $|\cdot|$ the norm for which the Lipschitz condition for f is satisfied.

In fact $\forall y, z \in C^0([t_{n-1},t_n],\mathbb{R}^m)$ we have

$$\begin{aligned}
\|\Phi_n(u,y) - \Phi_n(u,z)\|_\infty &= \max_{t_{n-1}\le t\le t_n} \left| u + \int_{t_{n-1}}^{t} f(x,y(x))\,dx - u - \int_{t_{n-1}}^{t} f(x,z(x))\,dx \right| \\
&\le \max_{t_{n-1}\le t\le t_n} \int_{t_{n-1}}^{t} |f(x,y(x)) - f(x,z(x))|\,dx \\
&\le \max_{t_{n-1}\le t\le t_n} L \int_{t_{n-1}}^{t} |y(x) - z(x)|\,dx \\
&\le L\,h \|y - z\|_\infty.
\end{aligned}$$

Since $L\,h < 1$ and

$$\|\Phi_n(u,y) - \Phi_n(u,z)\|_\infty \le L\|y - z\|_\infty$$

$\Phi_{n,u}$ is a contraction in the norm $\|\cdot\|_\infty$. So

$$y = \Phi_1(y_0,y) \Longleftrightarrow y(t) = y_0 + \int_{t_0}^{t} f(x,y(x))\,dx, \quad t_0 \le t \le t_1$$

has a unique solution, and Φ_{1,y_0} is a contraction on $C^0([t_0,t_1],\mathbb{R}^m)$ with respect to the norm $\|\cdot\|_\infty$. [1]. Similarly

$$y = \Phi_2(y(t_1),y) \Longleftrightarrow y(t) = y(t_1) + \int_{t_1}^{t} f(x,y(x))\,dx, \quad t_1 \le t \le t_2$$

has a unique solution in $C^0([t_1,t_2],\mathbb{R}^m)$. Consider $y(t)$ on $[t_0,t_2]$ obtained by putting together the obtained solutions on $[t_0,t_1]$ and on $[t_1,t_2]$. We have

$$y(t) = y(t_1) + \int_{t_1}^{t} f(x,y(x))\,dx = y_0 + \int_{t_0}^{t_1} f(x,y(x))\,dx, + \int_{t_1}^{t} f(x,y(x))\,dx$$

i.e.

$$y(t) = y_0 + \int_{t_0}^{t} f(x,y(x))\,dx.$$

We can this way use induction to extend the result on the whole interval $[a,b]$. □

[1] $C^0([t_0,t_1],\mathbb{R}^m)$ with the norm $\|\cdot\|_\infty$ is a complete metric space.

Theorem: *Every contraction of a complete metric space has one and only one fixed point.*

1.2 Continuous dependence on initial data

The following result on continuous dependence on initial data is giving information about the stability of the problems with respect to perturbations of the initial conditions.

Theorem 1.5. *Assume $f(t,y)$ is continuous in t and y and Lipschitz on $[t_0,t_f]\times \mathrm{R}^m$ with respect to y, then the solution of (2) and the solution of*

$$z' = f(t,z), \qquad z(t_0)=z_0, \quad t_0 \le t \le t_f. \tag{4}$$

with $z_0 \neq y_0$ are such that

$$|y(t)-z(t)| \le e^{L(t-t_0)}|y_0-z_0|, \quad t_0 \le t \le t_f$$

and L the Lipschitz constant.

Proof. Consider

$$z(t) = z_0 + \int_{t_0}^{t} f(x,z(x))\,dx, \qquad t_0 \le t \le t_f.$$

Since

$$y(t) = y_0 + \int_{t_0}^{t} f(x,y(x))\,dx, \qquad t_0 \le t \le t_f,$$

then

$$\begin{aligned} |y(t)-z(t)| &\le |y_0-z_0| + \int_{t_0}^{t} |f(x,y(x))-f(x,z(x))|\,dx \\ &\le |y_0-z_0| + L\int_{t_0}^{t} |y(x)-z(x)|\,dx. \end{aligned}$$

Consider

$$\varphi(t) := \int_{t_0}^{t} |y(x)-z(x)|\,dx,$$

$\varphi(t_0)=0$ and

$$\varphi'(t) = |y(t)-z(t)|,$$

so

$$\varphi'(t) - L\varphi(t) \le |y_0-z_0|.$$

Let $\alpha(t) := \varphi'(t) - L\varphi(t)$ such that

$$\alpha(t) \le |y_0-z_0|.$$

Now consider the scalar differential equation

$$\varphi'(t) - L\varphi(t) = \alpha(t), \quad t_0 \le t \le t_f, \quad \varphi(t_0)=0.$$

Its unique solution is

$$\varphi(t) = e^{L(t-t_0)} \int_{t_0}^{t} e^{-L(x-t_0)}\alpha(x)\,dx.$$

Since $\alpha(t) \leq |y_0 - z_0|$ we have

$$\begin{aligned}\varphi(t) &\leq e^{L(t-t_0)}\,|y_0 - z_0| \int_{t_0}^{t} e^{-L(x-t_0)}\,dx = \\ &= e^{L(t-t_0)}\,|y_0 - z_0| \frac{1}{L}\left(1 - e^{-L(t-t_0)}\right) = \\ &= \frac{1}{L}|y_0 - z_0|\,\left(e^{L(t-t_0)} - 1\right)\end{aligned}$$

and so

$$|y(t) - z(t)| = \varphi'(t) \leq |y_0 - z_0| + L\varphi(t) = |y_0 - z_0|e^{L(t-t_0)}.$$

□

2 Numerical solution of ODEs.

In this section we develop some simple methods for the solution of initial value problems. In both cases, let us assume that we somehow have found solutions $y_l \approx y(t_l)$, for $l = 0, 1, \cdots, n$, and we want to find an approximation $y_{n+1} \approx y(t_{n+1})$ where $t_{n+1} = t_n + h$, where h is the stepsize. Basically, there are two different classes of methods in practical use.

1. *One-step methods.* Only y_n is used to find the approximation y_{n+1}. One-step methods usually require more than one function evaluation pr. step. They can all be put in a general abstract form
$$y_{n+1} = y_n + h\Phi(t_n, y_n; h).$$
2. *Linear multistep methods:* y_{n+1} is approximated from $y_{n-k+1}, \cdots, y_n$.

2.1 Some examples of one-step methods.

Assume that t_n, y_n is known. The exact solution $y(t_{n+1})$ with $t_{n+1} = t_n + h$ of (1) passing through this point is given by

$$y(t_n + h) = y_n + \int_{t_n}^{t_{n+1}} y'(\tau)d\tau = y_n + \int_{t_n}^{t_{n+1}} f(\tau, y(\tau))d\tau. \tag{5}$$

The idea is to find approximations to the last integral. The simplest idea is to use $f(\tau, y(\tau)) \approx f(t_n, y_n)$, in which case we get the Euler method:

$$y_{n+1} = y_n + hf(t_n, y_n).$$

The integral can also be approximated by the trapezoidal rule

$$\int_{t_n}^{t_{n+1}} f(\tau, y(\tau)) = \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y(t_{n+1})).$$

By replacing the unknown solution $y(t_{n+1})$ by y_{n+1} we get the *trapezoidal method*

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right).$$

Here y_{n+1} is available by solving a (usually) nonlinear system of equations. Such methods are called implicit. To avoid this extra difficulty, we could replace y_{n+1} on the right hand side by the approximation from Eulers method, thus

$$\begin{aligned}\tilde{y}_{n+1} &= y_n + hf(t_n, y_n);\\ y_{n+1} &= y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right).\end{aligned}$$

This method is called the *improved Euler method*. Similarly, we could have used the midpoint rule for the integral,

$$\int_{t_n}^{t_{n+1}} f(\tau, y(\tau)) = \left(f(t_n + \frac{h}{2}, y(t_n + \frac{h}{2})\right),$$

and replaced $y(t_n + \frac{h}{2})$ by one half Euler step. The result is the *modified Euler method*:

$$\begin{aligned}\tilde{y}_{n+\frac{1}{2}} &= y_n + \frac{h}{2}f(t_n, y_n),\\ y_{n+1} &= y_n + hf(t_n + \frac{h}{2}, \tilde{y}_{n+\frac{1}{2}}).\end{aligned}$$

Do we gain anything by constructing these methods?

Example 2.1. *The ODE/IVP*

$$y' = -2ty, \qquad 0 \le t \le 1, \qquad y(0) = 1.0$$

has as solution the function

$$y(t) = e^{-t^2}.$$

Let us solve the problem from Example 2.1 using improved/modified Euler with $h = 0.1$. For each step, also the global error $e_n = y(t_n) - y_n$ is computed. For comparison, also the result for the Euler method is included.

	Euler		improved Euler		modified Euler	
t_n	y_n	e_n	y_n	e_n	y_n	e_n
0.0	1.000000	0	1.000000	0	1.000000	0
0.1	1.000000	$-9.95 \cdot 10^{-3}$	0.990000	$4.98 \cdot 10^{-5}$	0.990000	$4.98 \cdot 10^{-5}$
0.2	0.980000	$-1.92 \cdot 10^{-2}$	0.960696	$9.34 \cdot 10^{-5}$	0.960597	$1.92 \cdot 10^{-4}$
0.3	0.940800	$-2.69 \cdot 10^{-2}$	0.913814	$1.17 \cdot 10^{-4}$	0.913528	$4.03 \cdot 10^{-4}$
0.4	0.884352	$-3.22 \cdot 10^{-2}$	0.852040	$1.04 \cdot 10^{-4}$	0.851499	$6.45 \cdot 10^{-4}$
0.5	0.813604	$-3.48 \cdot 10^{-2}$	0.778765	$3.60 \cdot 10^{-5}$	0.777930	$8.71 \cdot 10^{-4}$
0.6	0.732243	$-3.46 \cdot 10^{-2}$	0.697773	$-9.69 \cdot 10^{-5}$	0.696636	$1.04 \cdot 10^{-3}$
0.7	0.644374	$-3.17 \cdot 10^{-2}$	0.612924	$-2.98 \cdot 10^{-4}$	0.611507	$1.12 \cdot 10^{-3}$
0.8	0.554162	$-2.69 \cdot 10^{-2}$	0.527850	$-5.58 \cdot 10^{-4}$	0.526202	$1.09 \cdot 10^{-3}$
0.9	0.465496	$-2.06 \cdot 10^{-2}$	0.445717	$-8.59 \cdot 10^{-4}$	0.443904	$9.54 \cdot 10^{-4}$
1.0	0.381707	$-1.38 \cdot 10^{-2}$	0.369053	$-1.17 \cdot 10^{-3}$	0.367153	$7.27 \cdot 10^{-4}$

As we can see, there is a significant improvement in accuracy, compared with the Euler method.

3 Runge-Kutta methods

The Euler method, as well as the improved and modified Euler methods are all examples on *explicit Runge-Kutta methods* (ERK). Such schemes are given by

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + c_2 h, y_n + h a_{21} k_1), \\ k_3 &= f\big(t_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)\big), \\ &\vdots \\ k_s &= f\big(t_n + c_s h, y_n + h \sum_{j=1}^{s-1} a_{sj} k_j\big), \\ y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i k_i, \end{aligned} \tag{6}$$

where c_i, a_{ij} and b_i are coefficients defining the method. We always require $c_i = \sum_{j=1}^{s} a_{ij}$. Here, s is the number of *stages*, or the number of function evaluations needed for each step. The vectors k_i are called stage derivatives. The improved Euler method is then a two-stage RK-method, written as

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + h, y_n + h k_1), \\ y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2). \end{aligned}$$

Also implicit methods, like the trapezoidal rule,

$$y_{n+1} = y_n + \frac{h}{2}\big(f(t_n, y_n) + f(t_n + h, y_{n+1})\big)$$

can be written in a similar form,

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f\big(t_n + h, y_n + \frac{h}{2}(k_1 + k_2)\big), \\ y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2). \end{aligned}$$

But, contrary to what is the case for explicit methods, a nonlinear system of equations has to be solved to find k_2.

Definition 3.1. *An s-stage Runge-Kutta method is given by*

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

The method is defined by its coefficients, which is given in a Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}, \qquad \text{where } c_i = \sum_{j=1}^{s} a_{ij}, \quad i = 1, \cdots, s.$$

The method is explicit *if* $a_{ij} = 0$ *whenever* $j \geq i$*, otherwise* implicit*.*

Example 3.2. *The Butcher-tableaux for the methods presented so far are*

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Euler *improved Euler* *modified Euler* *trapezoidal rule*

When the method is explicit, the zeros on and above the diagonal is usually ignored. We conclude this section by presenting the maybe most popular among the RK-methods over times, *The* 4th order Runge-Kutta method (Kutta – 1901):

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1) \\ k_3 &= f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_2) \\ k_4 &= f(t_n + h, y_n + hk_3) \\ y_{n+1} &= y_n + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \qquad \text{or} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} . \tag{7}$$

4 Convergence of Runge-Kutta methods

Definition 4.1. *The* local truncation error d_{n+1} *is the error made in one step (from* t_n *to* t_{n+1}*) when starting at the exact value of the solution* $y(t_n)$ *at* t_n*.*

By using the Taylor expansion of $y(t_n + h)$ around t_n one easily shows that the local truncation error for Euler's method is

$$d_{n+1} = y(t_n + h) - y(t_n) - hf(t_n, y(t_n)) = \frac{1}{2}h^2 y''(\xi),$$

where $\xi \in (t_n, t_{n+1})$.

Definition 4.2. *The* global error *is the difference between the exact and the numerical solution at point* t_n*, thus* $e_n := y(t_n) - y_n$*. The integration method is said to be convergent on the interval* $[t_0, t_f]$ *if and only if*

$$\lim_{N \to \infty} \|e_N\| = 0, \quad \text{with } h := \frac{t_f - t_0}{N}.$$

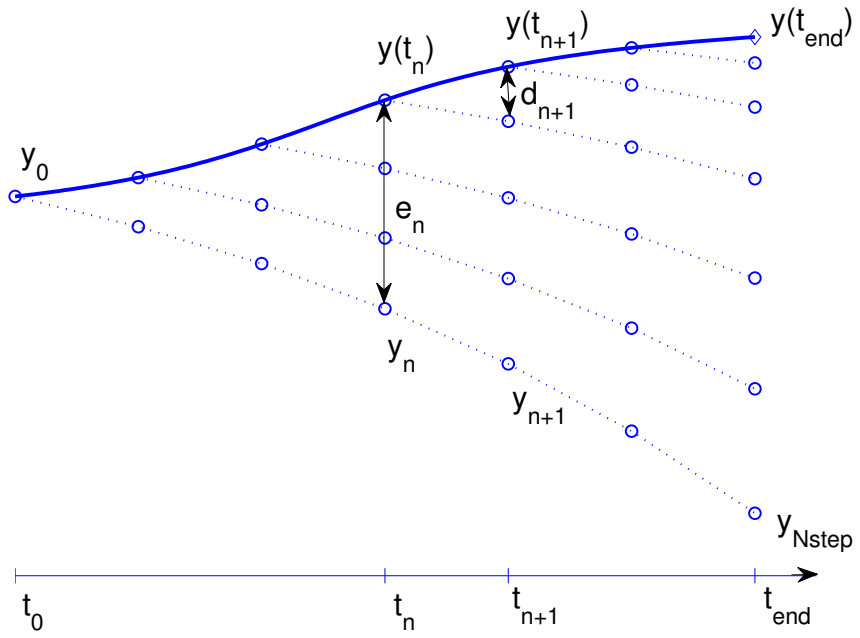


Figure 1: Propagation of the error for one step integration methods.

Exercise 4.3. *Assume Euler's method is applied to* (2) *and that* $f(t, y)$ *satisfies the Lipschitz condition with respect to* y *in* $[t_0, t_f] \times \mathbb{R}^m$*, with Lipschitz constant* $L > 0$*. Prove that Euler's method converges.*

The following theorem holds for general one step-methods

Theorem 4.4. *Let*

$$y' = f(t, y), \qquad y(t_0) = y_0, \qquad t_0 \le t \le t_{end}$$

be solved by a one-step method

$$y_{n+1} = y_n + h\Phi(t_n, y_n; h), \tag{8}$$

with step size $h = (t_f - t_0)/N$*. If*

1. *the increment function* Φ *is Lipschitz in* y*, and*
2. *the local truncation error* $d_{n+1} = \mathcal{O}(h^{p+1})$ *,*

then the method is of order p*, that is, the global error at* t_f *satisfies*

$$e_N = y(t_f) - y_N = \mathcal{O}(h^p).$$

A Runge-Kutta method is a one-step method with increment function $\Phi(t_n, y_n; h) = \sum_{i=1}^{s} b_i k_i$. It is possible to show that Φ is Lipschitz in y whenever f is Lipschitz and $h \le h_{max}$,

where h_{max} is some predefined maximal stepsize. What remains is the order of the local truncation error. To find it, we take the Taylor-expansions of the exact and the numerical solutions and compare them. The local truncation error is $\mathcal{O}(h^{p+1})$ if the two series match for all terms corresponding to h^q with $q \leq p$. In principle, this is trivial. In practise, it becomes extremely tedious (give it a try). Fortunately, it is possible to express the two series very elegantly by the use of *B-series* and *rooted trees*. Here, we present shortly the procedure without proofs.

4.1 Order conditions for Runge-Kutta methods: B-series and rooted trees

We assume that the equation is rewritten in autonomous form

$$y(t)' = f(y(t)), \qquad y(t_0) = y_0. \tag{9}$$

The Taylor expansion of the exact solution of (9) is given by

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{h}{2}y''(t_0) + \cdots + \frac{h^p}{p!}y^{(p)}(t_0) + \cdots . \tag{10}$$

From the ODE (9) and repeated use of the chain rule, we get $y' = f$, $y'' = f_y f$, $y''' = f_{yy}ff + f_y f_y f$, etc. Each higher derivative of y is split into several terms, denoted as *elementary differentials*. These can be represented by rooted trees. A node $\bullet$ represents f. A branch out from a bullet represent the derivative of f with respect to y. As the chain rule apply, this will always mean that we multiply by $y' = f$, represented by a new node on the end of the branch. We get the following table:

Elementary differentials	corresponding trees
$y' = f$	
$y'' = f_y f$	
$y''' = f_{yy}ff + f_y f_y f$	
$y^{iv} = f_{yyy}fff + f_{yy}f_y ff + f_{yy}ff_y f$	
$+f_{yy}ff_y f + f_y f_{yy}ff + f_y f_y f_y f$	

The elementary differentials corresponding to the trees  and  are equal, thus

$$y^{iv} = f_{yyy}fff + 3f_{yy}f_y ff + f_y f_{yy}ff + f_y f_y f_y f.$$

And we can go on like that. For each tree τ with p nodes we construct a set of total p new trees with $p+1$ nodes by adding one new node to an existing node in τ. This procedure might produce the same tree several times, and the total number of ways to construct a distinct tree is denoted by $\alpha(\tau)$. Let T be the set of all possible, distinct, rooted trees constructed this way, and let $\tau \in T$. A tree with p nodes corresponds to one of the terms in $y^{(p)}$, thus we call this *the order* of the tree and denote it $|\tau|$. The elementary differentials corresponding to a tree is denoted $F(\tau)(y)$.

Example 4.5.

For $\tau =$  *we have* $|\tau| = 4, \qquad F(\tau)(y) = f_y f_{yy} ff, \qquad \alpha(\tau) = 1.$

For $\tau =$  *we have* $|\tau| = 4, \qquad F(\tau)(y) = f_{yy}f_yff, \qquad \alpha(\tau) = 3.$

Here, f and its differentials are evaluated in y.

Putting this together: If $y(t)$ is the solution of (9), then

$$y^{(p)}(t_n) = \sum_{\substack{\tau \in T \\ |\tau| = p}} \alpha(\tau) F(\tau)(y(t_n)).$$

Insert this into (10), and we can write the exact solution as a B-series:

$$y(t_n + h) = y(t_n) + \sum_{\tau \in T} \frac{h^{|\tau|}}{|\tau|!}\, \alpha(\tau)\; F(\tau)(y(t_n)). \tag{11}$$

The numerical solution after one step can also be written as a B-series, but with some different coefficients

$$y_{n+1} = y_n + \sum_{\tau \in T} \frac{h^{|\tau|}}{|\tau|!}\, \gamma(\tau)\varphi(\tau)\; \alpha(\tau)\; F(\tau)(y_n). \tag{12}$$

where $\gamma(\tau)$ is an integer, and $\varphi(\tau)$ depends on the method coefficients, given in the Butcher tableau in Definition 3.1. Both can be found quite easily by the following procedure: Take a tree τ. Label the root with i, and all other non-terminal nodes by $j, k, l, \cdots$. The root correspond to b_i. A branch between a lower node j and an upper node k correspond to a_{jk}. A terminal node, connected to a node with label k corresponds to c_k. $\phi(\tau)$ is found by multiplying all these coefficients, and then take the sum over all the indices from 1 to s.

Example 4.6.

The tree $\tau =$  *can be labelled so that* $\varphi(\tau) = \sum_{i,j,k,l=1}^{s} b_i a_{ij} a_{jk} a_{il}.$

A tree τ can also be described by its subtrees. Let $\tau = [\tau_1, \tau_2, \cdots, \tau_l]$ be the tree composed by joining the root of the subtrees $\tau_1,\ \tau_2,\ \cdots,\ \tau_l$ to a joint new root. The term $\gamma(\tau)$ is defined recursively by

- $\gamma(\bullet) = 1$.
- $\gamma(\tau) = |\tau| \cdot \gamma(\tau_1) \cdots \gamma(\tau_l)$ for $\tau = [\tau_1, \tau_2, \cdots, \tau_l]$.

Example 4.7.

$\tau =$  $= [\bullet],$ $\qquad \gamma(\tau) = 2 \cdot 1 = 2$

$\tau =$  $= [\bullet, \bullet],$ $\qquad \gamma(\tau) = 3 \cdot 1 \cdot 1 = 3$

$\tau =$  $= [$  $]$ $\qquad \gamma(\tau) = 4 \cdot 3 = 12$

$\tau =$ $= [$  $,$$],$ $\qquad \gamma(\tau) = 7 \cdot 12 \cdot 2 = 168$

By comparing the two series (11) and (12) with $y(t_n) = y_n$ we can state the following theorem:

Theorem 4.8. *A Runge-Kutta method is of order p if and only if*

$$\varphi(\tau) = \frac{1}{\gamma(\tau)} \qquad \forall \tau \in T, \quad |\tau| \le p.$$

The order conditions up to order 4 are:

τ	$\lvert\tau\rvert$	$\varphi(\tau) = 1/\gamma(\tau)$
	1	$\sum b_i = 1$
	2	$\sum b_i c_i = 1/2$
	3	$\sum b_i c_i^2 = 1/3$
		$\sum b_i a_{ij} c_j = 1/6$
	4	$\sum b_i c_i^3 = 1/4$
		$\sum b_i c_i a_{ij} c_j = 1/8$
		$\sum b_i a_{ij} c_j^2 = 1/12$
		$\sum b_i a_{ij} a_{jk} c_k = 1/24$

5 Stiff equations and linear stability

Example 5.1. *Given the ODE*

$$y' = -1000y, \qquad y(0) = 1.$$

with exact solution

$$y(t) = e^{-1000t}.$$

Thus $y(t) \to 0$ as $t \to \infty$. The Euler method applied to this problem yields

$$y_{n+1} = y_n - 1000hy_n = (1 - 1000h)y_n.$$

so that $y_n = (1 - 1000h)^n$. This gives us two situations:

$$\textit{If} \quad |1 - 1000h| < 1 \quad \textit{then} \quad y_n \to 0 \ \textit{as} \ n \to \infty.$$

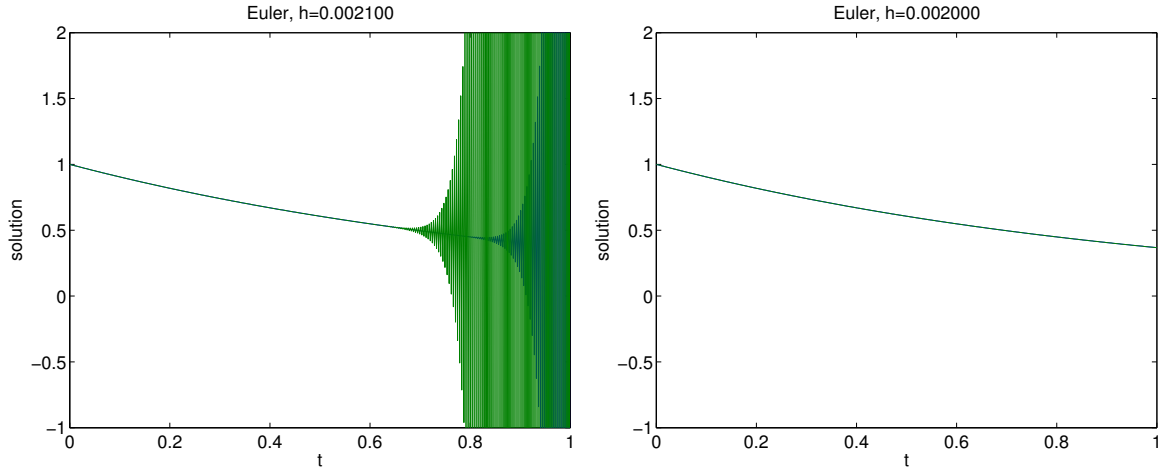
$$\textit{If} \quad |1 - 1000h| > 1 \quad \textit{then} \quad |y_n| \to \infty \ \textit{as} \ n \to \infty$$

Clearly, the second situation does not follow at all the behaviour of the exact solution. We have to choose a step size $h < 0.002$ to get a numerical solution decaying as the exact solution does for this problem.

Example 5.2. *Given*

$$y' = \begin{bmatrix} -2 & 1 \\ 998 & -999 \end{bmatrix} y, \qquad y(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

with exact solution $y_1(t) = y_2(t) = e^{-t}$. The matrix has eigenvalues -1 and -1000. The initial values are chosen so that the fast decaying mode is missing in the exact solution. This problem is solved by Eulers method, with two almost equal stepsizes, $h = 0.0021$ and $h = 0.002$. The difference is striking, the situation is similar to the case of Example 5.1.



Example 5.2 is a typical example of a stiff equation. The stepsize is restricted by a fast decaying component. Stiffness occurs in situations with fast decaying solutions (transients) in combination with slow components.

The *backward Euler* method is one way to overcome this problem:

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) \tag{13}$$

or, applied to the problem in Example 5.2

$$y_{n+1} = y_n + h\lambda y_{n+1}, \qquad \lambda = -1000$$

and

$$y_{n+1} = \frac{1}{1 - h\lambda} y_n.$$

Since $|1/(1 - h\lambda)| \leq 1$ whenever $\mathrm{Re}(\lambda) \leq 0$ there is no stepsize restriction caused by stability issues. In fact, $y_{n+1} \to 0$ as $\mathrm{Re}(h\lambda) \to -\infty$, so fast transients decay quickly, as they are supposed to. But the drawback is that to implement this method for a nonlinear ODE, a nonlinear system of equations has to be solved at each step.

5.1 Linear stability theory for Runge-Kutta methods.

Given the linear test equation

$$y' = \lambda y, \qquad \lambda \in \mathbb{C}. \tag{14}$$

Thus $\lambda = \alpha + i\beta$. The solution can be expressed by

$$y(t_n + h) = e^{\alpha h} e^{i\beta h} y(t_n).$$

Clearly, the solution is stable if $\alpha \leq 0$, that is $\lambda \in \mathbb{C}^-$. For the numerical solution we then require the stepsize h to be chosen so that

$$|y_{n+1}| \leq |y_n| \quad \text{whenever} \lambda \in \mathbb{C}^- \tag{15}$$

When a RK method is applied (14), we simply get

$$y_{n+1} = R(z) y_n, \qquad z = h\lambda$$

where R is a polynomial or a rational function. R is called *the stability function* of the RK method. The numerical solution is stable if $|R(z)| \leq 1$, otherwise it is unstable. This motivates the following definition of *the region of absolute stability* as

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

The condition (15) is satisfied for all $h > 0$ if

$$\mathbb{C}^- \subseteq \mathcal{D},$$

Methods satisfying this condition are called *A-stable*. The Backward Euler method (13) is an example of an A-stable method.

Example 5.3. *A 2-stage ERK applied to* (14) *is given by:*

$$k_1 = \lambda y_n, \qquad k_2 = \lambda(y_n + ha_{21}\lambda y_n), \qquad y_{n+1} = y_n + h\lambda(b_1 + b_2)y_n + (h\lambda)^2 b_2 a_{21} y_n$$

If this method is of order 2, then $b_1 + b_2 = 1$ *and* $b_2 a_{21} = 1/2$*, so that*

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

The stability function of an s-stage ERKs is a polynomial of degree s. As a consequence, no ERKs can be A-stable! If the order of the method is s, then

$$R(z) = \sum_{i=0}^{s} \frac{z^i}{i!}.$$

See Figure 2 for plots of the stability regions. But it has been proved that ERK with $p = s$ only exist for $s \leq 4$. To get an order 5 ERK, 6 stages are needed.

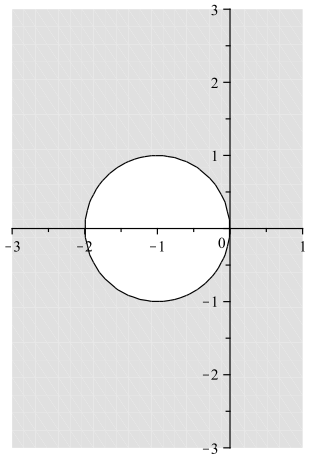
Example 5.4. *The trapezoidal rule applied to* (14) *gives*

$$y_{n+1} = y_n + \frac{h}{2}(\lambda y_n + \lambda y_{n+1}) \qquad \Rightarrow \qquad R(z) = \frac{1+z}{1-z}.$$

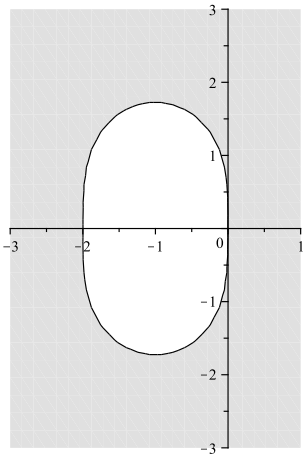
In this case $\mathcal{D} = \mathbb{C}^-$*, which is perfect.*

To summarise:

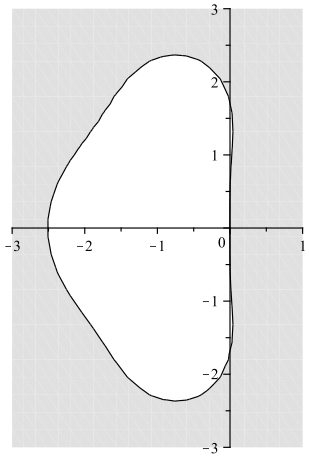
- For a given $\lambda \in \mathbb{C}^-$, choose a stepsize h so that $h\lambda \in \mathcal{D}$.
- If your problem is stiff, use an A-stable method.
- There are no A-stable explicit methods.



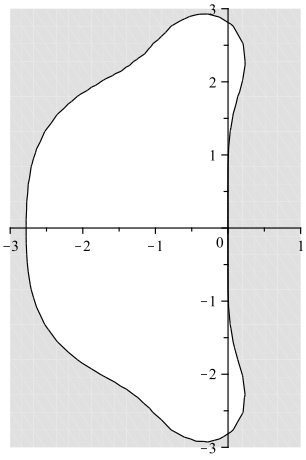
$p = s = 1$



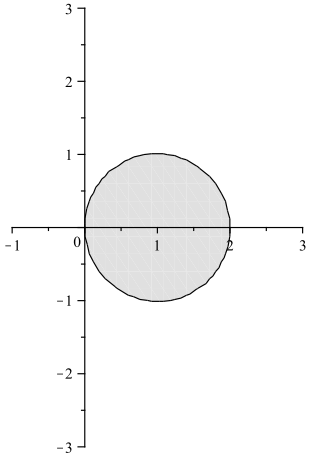
$p = s = 2$



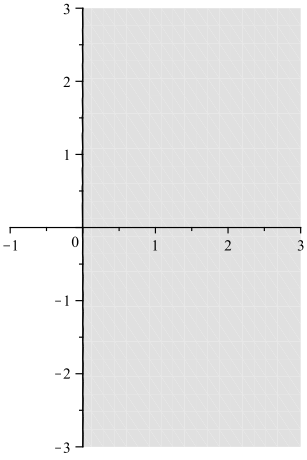
$p = s = 3$



$p = s = 4$



Backward Euler



Trapezoidal rule

Figure 2: Stability regions in $\mathbb{C}^-$: The first four are the stability regions for explicit RK methods of order $p = s$. The white regions are stable, the grey unstable.

6 Collocation.

Given an ordinary differential equation

$$y'(t) = f(t, y(t)), \qquad y(t_n) = y_n, \qquad t \in [t_n, t_n + h].$$

Recall also the defintion of a Runge–Kutta method applied to the ODE:

$$\begin{aligned} k_i &= f(t_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} k_j) \\ y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{16}$$

The idea of collocation methods for ODEs is as follows: Assume that s distinct points $c_1, c_2, \ldots, c_s \in [0, 1]$ are given. We will then search for a polynomial $u \in \mathbb{P}_s$ satisfying the ODE in the points $t_n + c_i h$, $i = 1, \cdots, s$. These are the *collocation conditions*, expressed as

$$u'(t_n + c_i h) = f(t_n + c_i h, u(t_n + c_i h)), \qquad i = 1, 2, \ldots, s, \tag{17}$$

The initial condition is $u(t_n) = y_n$. When finally $u(t)$ has been found, we will set $y_{n+1} = u(t_n + h)$.

Let k_i be some (so far unknown) approximation to $y'(t_n + c_i h)$, and let $u' = p \in \mathbb{P}_{s-1}$ be the interpolation polynomial satisfying $p(t_n + c_i h) = k_i$. For simplicity, introduce the change of variables $t = t_n + \tau h$, $\tau \in [0, 1]$. Then

$$p(t) = p(t_n + h\tau) = \sum_{j=1}^{s} k_j \ell_j(\tau), \qquad \ell_j(\tau) = \prod_{\substack{k=1 \\ k \neq j}}^{s} \frac{\tau - c_k}{c_j - c_k}.$$

The polynomial $u(t)$ is given by

$$u(\tilde{t}) = u(t_n) + \int_{t_n}^{\tilde{t}} p(t)dt = y_n + h \sum_{j=1}^{s} k_j \int_0^{\tilde{\tau}} \ell_j(\tau)d\tau, \qquad \tilde{t} \in [t_n, t_n + h] \qquad (\text{thus } \bar{\tau} \in [0, 1])$$

The collocation condition becomes

$$u'(t_n + c_i h) = f\left(t_n + c_i h, y_n + h \sum_{j=1}^{s} k_j \int_0^{c_i} \ell_j(\tau)d\tau.\right)$$

and in addition we get

$$u(t_n + h) = y_n + h \sum_{j=1}^{s} k_j \int_0^1 \ell_j(\tau)d\tau.$$

But $u'(t_n + c_i h) = k_i$, and $y_{n+1} = u(t_n + h)$ so this is exactly the Runge–Kutta method (16) with coefficients

$$a_{ij} = \int_0^{c_i} \ell_j(\tau)d\tau, \quad i, j = 1, \ldots, s, \qquad\qquad b_i = \int_0^1 \ell_i(\tau)d\tau, \quad i = 1, \ldots, s. \tag{18}$$

Runge–Kutta methods constructed this way are called *collocation methods* and they are of order at least s. But a clever choice of the c_i's will give better results.

Example 6.1. *Let $P_2(x) = x^2 - 1/3$ be the second order Legendre polynomial. Let c_1 and c_2 be the zeros of $P_2(2x-1)$ (using a change of variable to transfer the interval [-1,1] to [0,1]), that is*

$$c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \qquad c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}.$$

The corresponding cardinal functions becomes

$$\ell_1(\tau) = -\sqrt{3}\left(\tau - \frac{1}{2} - \frac{\sqrt{3}}{6}\right), \qquad \ell_2(\tau) = \sqrt{3}\left(\tau - \frac{1}{2} + \frac{\sqrt{3}}{6}\right).$$

From (18) *we get a Runge–Kutta method with the following Butcher tableau:*

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

This method has order 4 *(check it yourself).*

Collocation methods using the zeros of the Legendre-polynomials $P_s(2x-1)$ for the c_i's are called Gauss-Legendre methods, also known as Kuntzmann-Butcher methods. They are of order $2s$, they are also A-stable.

7 Linear multistep methods

A k-step linear multistep method (LMM) applied to the ODE

$$y' = f(t,y), \quad y(t_0) = y_0, \quad t_0 \le t \le t_{end}.$$

is given by

$$\sum_{l=0}^{k} \alpha_l y_{n+l} = h \sum_{l=0}^{k} \beta_l f_{n+l}, \tag{19}$$

where α_l, β_l are the coefficients of the method, $f_j = f(t_j, y_j)$ and $t_j = t_0 + jh$, $h = (t_{end} - t_0)/Nstep$. Usually we require

$$\alpha_k = 1 \quad \text{and} \quad |\alpha_0| + |\beta_0| \neq 0.$$

To get started with a k-step method, we also need starting values $y_l \approx y(t_l)$, $l = 0, 1, \cdots, k-1$. A method is explicit if $\beta_k = 0$, otherwise implicit. The *leapfrog method*

$$y_{n+2} - y_n = 2hf(t_{n+1}, y_{n+1}) \tag{20}$$

and the method given by

$$y_{n+2} - y_{n+1} = h\left(\frac{3}{2} f_{n+1} - \frac{1}{2} f_n\right) \tag{21}$$

are both examples of explicit 2-step methods.

Example 7.1. *Given the problem*

$$y' = -2ty, \qquad y(0) = 1$$

with exact solution $y(t) = e^{-t^2}$. *Let* $h = 0.1$, *and* $y_1 = e^{-h^2}$. *This problem is solved by* (21), *and the numerical solution and the error is given by*

t_n	y_n	$\lvert e_n \rvert$
0.0	1.000000	0.00
0.1	0.990050	0.00
0.2	0.960348	$4.41 \cdot 10^{-4}$
0.3	0.912628	$1.30 \cdot 10^{-3}$
0.4	0.849698	$2.45 \cdot 10^{-3}$
0.5	0.775113	$3.69 \cdot 10^{-3}$
0.6	0.692834	$4.84 \cdot 10^{-3}$
0.7	0.606880	$5.75 \cdot 10^{-3}$
0.8	0.521005	$6.29 \cdot 10^{-3}$
0.9	0.438445	$6.41 \cdot 10^{-3}$
1.0	0.361746	$6.13 \cdot 10^{-3}$

The corresponding MATLAB code is given in `lmm.m`.

7.1 Consistency and order.

We define the *local discretization error* $\tau_{n+k}(h)$ by

$$h\tau_{n+k}(h) = \sum_{l=0}^{k} \left(\alpha_l y(t_{n+l}) - h\beta_l y'(t_{n+l}) \right). \tag{22}$$

You can think about the $h\tau_{n+k}$ as the defect obtained when plugging the exact solution into the difference equation (19). A method is *consistent* if $\tau_{n+k}(h) \underset{h\to 0}{\to} 0$. The term $h\tau_{n+k}(h)$ can be written as a power series in h

$$h\tau_{n+k}(h) = C_0 y(t_n) + C_1 h y'(t_n) + C_2 h^2 y''(t_n) + \cdots + C_q h^q y^{(q)}(t_n) + \cdots,$$

by expanding $y(t_n + lh)$ and $y'(t_n + lh)$ into their Taylor series around t_n,

$$y(t_n + lh) = y(t_n) + (lh)y'(t_n) + \frac{1}{2}(lh)^2 y''(t_n) + \cdots + \frac{(lh)^q}{q!} y^{(q)}(t_n) + \cdots$$

$$y'(t_n + lh) = y'(t_n) + (lh)y''(t_n) + \frac{1}{2}(lh)^2 y'''(t_n) + \cdots + \frac{(lh)^{q-1}}{q-1!} y^{(q)}(t_n) + \cdots$$

for sufficiently differentiable solutions $y(t)$. Insert this into (22), get the following expressions for C_q:

$$C_0 = \sum_{l=0}^{k} \alpha_l, \qquad C_q = \frac{1}{q!} \sum_{l=0}^{k} \left(l^q \alpha_l - q l^{q-1} \beta_l \right), \quad q = 1, 2, \cdots . \tag{23}$$

The method is consistent if $C_0 = C_1 = 0$. It is *of order* p if

$$C_0 = C_1 = \cdots = C_p = 0, \qquad C_{p+1} \neq 0.$$

The constant C_{p+1} is called the *error constant*.

Example 7.2. *The LMM* (21) *is defined by*

$$\alpha_0 = 0,\ \alpha_1 = -1,\ \alpha_2 = 1,\ \beta_0 = -\frac{1}{2},\ \beta_1 = \frac{3}{2},\ \beta_2 = 0,$$

thus

$$\begin{aligned} C_0 &= \alpha_0 + \alpha_1 + \alpha_2 = 0. \\ C_1 &= \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) = 0 \\ C_2 &= \frac{1}{2!} \left(\alpha_1 + 2^2 \alpha_2 - 2(\beta_1 + 2\beta_2) \right) = 0 \\ C_3 &= \frac{1}{3!} \left(\alpha_1 + 2^3 \alpha_2 - 3(\beta_1 + 2^2 \beta_2) \right) = \frac{5}{12}. \end{aligned}$$

The method is consistent and of order 2.

Example 7.3. *Is it possible to construct an explicit 2-step method of order 3? There are 4 free coefficients $\alpha_0, \alpha_1, \beta_0, \beta_1$, and 4 order conditions to be solved ($C_0 = C_1 = C_2 = C_3 = 0$). The solution is*

$$\alpha_0 = -5,\ \alpha_1 = 4,\ \beta_0 = 2,\ \beta_1 = 4.$$

Test this method on the ODE of Example 2.1. (Replace the method coefficients in `lmm.m`*.) The result is nothing but disastrous. Taking smaller steps only increases the problem.*

To see why, you have to know a bit about how to solve difference equations.

7.2 Linear difference equations

A linear difference equation with constant coefficients is given by

$$\sum_{l=0}^{k} \alpha_l y_{n+l} = \varphi_n, \qquad n = 0, 1, 2, \cdots . \tag{24}$$

The solution of this equation is a sequence $\{y_n\}$ of numbers (or vectors). Let $\{\tilde{y}_n\}$ be the general solution of the homogeneous problem

$$\sum_{l=0}^{k} \alpha_l y_{n+l} = 0. \tag{25}$$

Let ψ_n be one particular solution of (24). The general solution of (24) is then $\{y_n\}$ where $y_n = \tilde{y}_n + \psi_n$. To find a unique solution, we will need the starting values $y_0, y_1, \cdots, y_{k-1}$.

Let us try $\tilde{y}_n = r^n$ as a solution of the homogeneous equation (25). This is true if

$$\sum_{l=0}^{k} \alpha_l r^{n+l} = r^n \sum_{l=0}^{k} \alpha_l r^l = 0.$$

The polynomial $\rho(r) = \sum_{l=0}^{k} \alpha_l r^l$ is called *the characteristic polynomial*, and $\{r^n\}$ is a solution of (25) if r is a root of $\rho(r)$. The kth degree polynomial $\rho(r)$ has k roots altogether, $r_1, r_2, \cdots, r_k$, they can be distinct and real, they can be distinct and complex, in which case they appear in complex conjugate pairs, or they can be multiple. In the latter case, say $r_1 = r_2 = \cdots = r_\mu$ we get a set of linear independent solutions $\{r_1^n\}$, $\{nr_1^n\}$, $\cdots$, $\{n^{\mu-1}r_1^n\}$. Altogether we have found k linear independent solutions $\{\tilde{y}_{n,l}\}$ of the homogeneous equation, and the general solution is given by

$$y_n = \sum_{l=1}^{k} \kappa_l \tilde{y}_{n,l} + \psi_n.$$

The coefficients κ_l can be determined from the starting values.

Example 7.4. *Given*

$$y_{n+4} - 6y_{n+3} + 14y_{n+2} - 16y_{n+1} + 8y_n = n$$
$$y_0 = 1,\ y_1 = 2,\ y_2 = 3\ y_3 = 4.$$

The characteristic polynomial is given by

$$\rho(r) = r^4 - 6r^3 + 14r^2 - 16r + 8$$

with roots $r_1 = r_2 = 2$, $r_3 = 1 + i$, $r_4 = 1 - i$. As a particular solution we try $\psi_n = an + b$. Inserted into the difference equation we find this to be a solution if $a = 1$, $b = 2$. The general solution has the form

$$y_n = \kappa_1 2^n + \kappa_2 n 2^n + \kappa_3 (1+i)^n + \kappa_4 (1-i)^n + n + 2.$$

From the starting values we find that $\kappa_1 = -1$, $\kappa_2 = \frac{1}{4}$, $\kappa_3 = -i/4$ and $\kappa_4 = i/4$. So, the solution of the problem is

$$\begin{aligned} y_n &= 2^n \left(\frac{n}{4} - 1\right) - \frac{i(1+i)^n}{4} + \frac{i(1-i)^n}{4} + n + 2 \\ &= 2^n \left(\frac{n}{4} - 1\right) - 2^{\frac{n-2}{2}} \sin\left(\frac{n\pi}{4}\right) + n + 2. \end{aligned}$$

Example 7.5. *The homogeneous part of the difference equation of Example 7.3 is*

$$\rho(r) = r^2 + 4r - 5 = (r-1)(r+5).$$

One root is 5. Thus, one solution component is multiplied by a factor -5 for each step, independent of the stepsize. Which explain why this method fails.

7.3 Zero-stability and convergence

Let us start with the definition of convergence. As before, we consider the error at t_{end}, using $Nstep$ steps with constant stepsize $h = (t_{end} - t_0)/Nstep$.

Definition 7.6.

- *A linear multistep method* (19) *is* convergent *if, for all ODEs satisfying the conditions of Theorem 1.4 we get*

$$y_{Nstep} \underset{h\to 0}{\to} y(t_{end}), \qquad \textit{whenever} \qquad y_l \underset{h\to 0}{\to} y(t_0 + lh), \quad l = 0, 1, \cdots, k-1.$$

- *The method is* convergent of order p *if, for all ODEs with f sufficiently differentiable, there exists a positive h_0 such that for all $h < h_0$*

$$\|y(t_{end} - y_{Nstep}\| \le Kh^p \qquad \textit{whenever} \qquad \|y(t_0 + lh) - y_l\| \le K_0 h^p, \quad l = 0, 1, \cdots, k-1.$$

The first characteristic polynomial of an LMM (19) is

$$\rho(r) = \sum_{l=0}^{k} \alpha_l r^l,$$

with roots $r_1, r_2, \cdots, r_k$. From the section on difference equation, it follows that for the boundedness of the solution y_n we require:

1. $|r_i| \le 1$, for $i = 1, 2, \cdots, k$.
2. $|r_i| < 1$ if r_i is a multiple root.

A method satisfying these two conditions is called *zero-stable*.

We can now state (without proof) the following important result:

Theorem 7.7. *(Dahlquist)*

$$\textit{Convergence} \qquad \Leftrightarrow \qquad \textit{Zero-stability + Consistency.}$$

For a consistent method, $C_0 = \sum_{l=0}^{k} \alpha_l = 0$ so the characteristic polynomial $\rho(r)$ will always have one root $r_1 = 1$.

The zero-stability requirement puts a severe restriction on the maximum order of a convergent k-step method:

Theorem 7.8. *(The first Dahlquist-barrier) The order p of a zero-stable k-step method satisfies*

$$\begin{aligned} p &\le k+2 \quad && \textit{if } k \textit{ is even,} \\ p &\le k+1 \quad && \textit{if } k \textit{ is odd,} \\ p &\le k \quad && \textit{if } \beta_k \le 0. \end{aligned}$$

Notice that the last line include all explicit LMMs.

7.4 Adams-Bashforth-Moulton methods

The most famous linear multistep methods are constructed by the means of interpolation. For instance by the following strategy:

The solution of the ODE satisfy the integral equation

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t))dt. \tag{26}$$

Assume that we have found $f_i = f(t_i, y_i)$ for $i = n-k+1, \cdots, n$, with $t_i = t_0 + ih$. Construct the polynomial of degree $k-1$, satisfying

$$p_{k-1}(t_i) = f(t_i, y_i), \qquad i = n-k+1, \ldots, n.$$

The interpolation points are equidistributed (constant stepsize), so Newton's backward difference formula can be used in this case (see Exercise 2), that is

$$p_{k-1}(t) = p_{k-1}(t_n + sh) = f_n + \sum_{j=1}^{k-1} (-1)^j \binom{-s}{j} \nabla^j f_n$$

where

$$(-1)^j \binom{-s}{j} = \frac{s(s+1)\cdots(s+j-1)}{j!}$$

and

$$\nabla^0 f_n = f_n, \qquad \nabla^j f_n = \nabla^{j-1} f_n - \nabla^{j-1} f_{n-1}.$$

Using $y_{n+1} \approx y(t_{n+1})$. $y_n \approx y(t_n)$ and $p_{k-1}(t) \approx f(t, y(t))$ in (26) gives

$$\begin{aligned} y_{n+1} - y_n \int_{t_n}^{t_{n+1}} p_{k-1}(t)dt &= h \int_0^1 p_{k-1}(t_n + sh)ds \\ &= hf_n + h \sum_{j=1}^{k-1} \left((-1)^j \int_0^1 \binom{-s}{1} ds \right) \nabla^j f_n. \end{aligned} \tag{27}$$

This gives the *Adams-Bashforth methods*

$$y_{n+1} - y_n = h \sum_{j=0}^{k-1} \gamma_j \nabla^j f_n, \qquad \gamma_0 = 1, \quad \gamma_j = (-1)^j \int_0^1 \binom{-s}{j} ds.$$

Example 7.9. *We get*

$$\gamma_0 = 1, \quad \gamma_1 = \int_0^1 s ds = \frac{1}{2}, \quad \gamma_2 = \int_0^1 \frac{s(s+1)}{2} ds = \frac{5}{12}$$

and the first few methods becomes:

$$\begin{aligned} y_{n+1} - y_n &= hf_n \\ y_{n+1} - y_n &= h\left(\frac{3}{2} f_n - \frac{1}{2} f_{n-1} \right) \\ y_{n+1} - y_n &= h\left(\frac{23}{12} f_n - \frac{4}{3} f_{n-1} + \frac{5}{12} f_{n-1} \right) \end{aligned}$$

A k-step Adams-Bashforth method is explicit, has order k (which is the optimal order for explicit methods) and it is zero-stable. In addition, the error constant $C_{p+1} = \gamma_k$. Implicit Adams methods are constructed similarly, but in this case we include the (unknown) point (t_{n+1}, f_{n+1}) into the set of interpolation points. So the polynomial

$$p_k^*(t) = p_k^*(t_n + sh) = f_{n+1} + \sum_{j=1}^{k} (-1)^j \binom{-s+1}{j} \nabla^j f_{n+1}$$

interpolates the points (t_i, f_i), $i = n-k+1, \ldots, n+1$. Using this, we get the *Adams-Moulton* methods

$$y_{n+1} - y_n = h \sum_{j=0}^{k} \gamma_j^* \nabla^j f_{n+1}, \qquad \gamma_0^* = 1, \quad \gamma_j^* = (-1)^j \int_0^1 \binom{-s+1}{j} ds.$$

Example 7.10. *We get*

$$\gamma_0^* = 1, \quad \gamma_1^* = \int_0^1 (s-1) ds = -\frac{1}{2}, \quad \gamma_2^* = \int_0^1 \frac{(s-1)s}{2} ds = -\frac{1}{12}$$

and the first methods becomes

$$y_{n+1} - y_n = h f_{n+1} \qquad \textit{(Backward Euler)}$$

$$y_{n+1} - y_n = h \left(\frac{1}{2} f_{n+1} + \frac{1}{2} f_n \right) \qquad \textit{(Trapezoidal method)}$$

$$y_{n+1} - y_n = h \left(\frac{5}{12} f_{n+1} + \frac{2}{3} f_n - \frac{1}{12} f_{n-1} \right).$$

A k-step Adams-Moulton method is implicit, of order $k+1$ and is zero-stable. The error constant $C_{p+1} = \gamma_{k+1}^*$. Despite the fact that the Adams-Moulton methods are implicit, they have some advantages compared to their explicit counterparts: They are of one order higher, the error constants are much smaller, and the linear stability properties (when the methods are applied to the linear test problem $y' = \lambda y$) are much better.

k	0	1	2	3	4	5	6
γ_k	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$
γ_k^*	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$

Table 1: The γ's for the Adams methods.

7.5 Predictor-corrector methods

A predictor-corrector (PC) pair is a pair of one explicit (predictor) and one implicit (corrector) methods. The nonlinear equations from the application of the implicit method are solved by a fixed number of fixed point iterations, using the solution by the explicit method as starting values for the iterations.

Example 7.11. *We may construct a PC method from a second order Adams-Bashforth scheme and the trapezoidal rule as follows:*

$$
\begin{aligned}
&y_{n+1}^{[0]} = y_n + \frac{h}{2}(3f_n - f_{n-1}) && \textit{(P : Predictor)}\\
&\textit{for } l = 0, 1, \ldots, m\\
&\quad f_{n+1}^{[l]} = f(t_{n+1}, y_{n+1}^{[l]}) && \textit{(E : Evaluation)}\\
&\quad y_{n+1}^{[l+1]} = y_n + \frac{h}{2}(f_{n+1}^{[l]} + f_n) && \textit{(C : Corrector)}\\
&\textit{end}\\
&y_{n+1} = y_{n+1}^{[m]}\\
&f_{n+1} = f(t_{n+1}, y_{n+1}). && \textit{(E : Evaluation)}
\end{aligned}
$$

Such schemes are commonly referred as P(EC)^{m}E schemes.

The predictor and the corrector is often by the same order, in which case only one or two iterations are needed.

Error estimation in predictor-corrector methods.

The local discretization error of some LMM is given by

$$h\tau_{n+1} = \sum_{l=0}^{k}(\alpha_l y(t_{n-k+1+l} - h\beta_l y'(t_{n-k+1+l})) = h^{p+1}C_{p+1}y^{(p+1)}(t_{n-k+1}) + \mathcal{O}(h^{p+2}).$$

But we can do the Taylor expansions of y and y' around t_n rather than t_{n-k+1}. This will not alter the principal error term, but the terms hidden in the expression $\mathcal{O}(h^{p+2})$ will change. As a consequence, we get

$$h\tau_{n+1} = h^{p+1}C_{p+1}y^{(p+1)}(t_n) + \mathcal{O}(h^{p+2}).$$

Assume that $y_i = y(t_i)$ for $i = n-k+1, \ldots, n$, and $\alpha_k = 1$. Then

$$h\tau_{n+1} = y(t_{n+1}) - y_{n+1} + \mathcal{O}(h^{p+2}) = h^{p+1}C_{p+1}y^{(p+1)}(t_n) + \mathcal{O}(h^{p+2}).$$

Assume that we have chosen a predictor-corrector pair, using methods of the same order p. Then

$$
\begin{aligned}
&(P) && y(t_{n+1}) - y_{n+1}^{[0]} \approx h^{p+1}C_{p+1}^{[0]}y^{(p+1)}(t_n),\\
&(C) && y(t_{n+1}) - y_{n+1} \approx h^{p+1}C_{p+1}y^{(p+1)}(t_n),
\end{aligned}
$$

and

$$y_{n+1} - y_{n+1}^{[0]} \approx h^{p+1}(C_{p+1}^{[0]} - C_{p+1})y^{(p+1)}(t_n).$$

From this we get the following local error estimate for the corrector, called *Milne's device*:

$$y(t_{n+1}) - y_{n+1} \approx \frac{C_{p+1}}{C_{p+1]}^{[0]} - C_{p+1}}(y_{n+1} - y_{n+1}^{[0]}).$$

Example 7.12. *Consider the PC-scheme of Example 7.11. In this case*

$$C_{p+1}^{[0]} = \frac{5}{12}, \qquad C_{p+1} = -\frac{1}{12}, \qquad \textit{so} \qquad \frac{C_{p+1}}{C_{p+1]}^{[0]} - C_{p+1}} = -\frac{1}{6}.$$

Apply the scheme to the linear test problem

$$y' = -y, \qquad y(0) = 1,$$

using $y_0 = 1$, $y_1 = e^{-h}$ *and* $h = 0.1$. *One step of the PC-method gives*

l	$y_2^{[l]}$	$\lvert y_2 - y_2^{[l]}\rvert$	$\lvert y(0.2) - y_2^{[l]}\rvert$	$\frac{1}{6}\lvert y_2^{[l]} - y_2^{[0]}\rvert$
0	0.819112	$4.49 \cdot 10^{-4}$	$3.81 \cdot 10^{-4}$	
1	0.818640	$2.25 \cdot 10^{-5}$	$9.08 \cdot 10^{-5}$	$7.86 \cdot 10^{-5}$
2	0.818664	$1.12 \cdot 10^{-6}$	$6.72 \cdot 10^{-5}$	$7.47 \cdot 10^{-5}$
3	0.818662	$5.62 \cdot 10^{-8}$	$6.84 \cdot 10^{-5}$	$7.49 \cdot 10^{-5}$

After 1-2 iterations, the iteration error is much smaller than the local error, and we also observe that Milne's device gives a reasonable approximation to the error.

Remark Predictor-corrector methods are not suited for stiff problems. You can see this by e.g. using the trapezoidal rule on $y' = \lambda y$. The trapezoidal rule has excellent stability properties. But the iteration scheme

$$y_{n+1}^{[l+1]} = y_n + \frac{h}{2}\lambda(y_{n+1}^{[l]} + y_n)$$

will only converge if $|h\lambda/2| < 1$.

8 Orthogonal polynomials.

The aim of this section is to construct "optimal" quadrature formulas. To be more specific, given the integral

$$I_w(f) = \int_a^b w(x)f(x)dx \tag{28}$$

in which $w(x)$ is a fixed, positive function. We want to approximate this using a quadrature formula on the form

$$Q_w(f) = \sum_{i=0}^{n} A_i f(x_i).$$

Such a formula can be constructed as follows: Choose $n+1$ distinct nodes, $x_0, x_1, \cdots, x_n$ in the interval $[a, b]$. Construct the interpolation polynomial

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x), \qquad \ell_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

An approximation to the integral is then given by

$$Q_w(f) = \int_a^b w(x)p_{n-1}(x)dx = \sum_{i=0}^{m} A_i f(x_i), \qquad A_i = \int_a^b w(x)\ell_i(x)dx. \tag{29}$$

The quadrature formula is of precision m if

$$I_w(p) = Q_w(p), \qquad \text{for all } p \in \mathbb{P}_m.$$

From the construction, these quadrature formulas is of precision at least n. The question is how too choose the nodes x_i, $i = 0, \ldots, n$ giving m as large as possible. The key concept here is *orthogonal polynomials*.

Orthogonal polynomials.

Given two functions $f, g \in C[a, b]$. We define an inner product of these two functions by

$$< f, g >= \int_a^b w(x)f(x)g(x)dx, \qquad w(x) > 0. \tag{30}$$

Thus the definition of the inner product depends on the integration interval $[a, b]$ and a given *weight function* $w(x)$. If $f, g, h \in C[a, b]$ and $\alpha \in \mathbb{R}$ then

$$\begin{aligned} \langle f, g\rangle_w &= \langle g, f\rangle_w \\ \langle f + g, h\rangle_w &= \langle f, h\rangle_w + \langle g, h\rangle_w \\ \langle \alpha f, g\rangle_w &= \alpha\langle f, g\rangle_w \\ \langle f, f\rangle_w &\geq 0, \qquad \text{and} \qquad \langle f, f\rangle_w = 0 \Leftrightarrow f \equiv 0. \end{aligned}$$

From an inner product, we can also define a norm on $C[a, b]$ by

$$\|f\|_w^2 = \langle f, f\rangle_w.$$

For the inner product (36) we also have

$$\langle xf, g\rangle_w = \int_a^b w(x)xf(x)g(x)dx = \langle f, xg\rangle_w. \tag{31}$$

Our aim is now to create an orthogonal basis for $\mathbb{P}$, that is, create a sequence of polynomials $\phi_k(x)$ of degree k (no more, no less) for $k = 0, 1, 2, 3, \ldots$ such that

$$\langle \phi_i, \phi_j\rangle_w = 0 \quad \text{for all} \quad i \neq j.$$

If we can make such a sequence, then

$$\mathbb{P}_{n-1} = \text{span}\{\phi_0, \phi_1, \cdots, \phi_{n-1}\} \qquad \text{and} \qquad \langle \phi_n, p\rangle_w = 0 \quad \text{for all} \qquad p \in \mathbb{P}_{n-1}.$$

Let us now find the sequence of orthogonal polynomials. This is done by a Gram-Schmidt process:

Let $\phi_0 = 1$. Let $\phi_1 = x - B_1$ where B_1 is given by the orthogonality condition:

$$0 = \langle \phi_1, \phi_0\rangle_w = \langle x, 1\rangle_w - B_1\langle 1, 1\rangle_w \qquad \Rightarrow \qquad B_1 = \frac{\langle x, 1\rangle_w}{\|1\|_w^2}.$$

Let us now assume that we have found ϕ_j, $j = 0, 1, \ldots, k-1$. Then, let

$$\phi_k = x\phi_{k-1} - \sum_{j=0}^{k-1} \alpha_j \phi_j.$$

Clearly, ϕ_k is a polynomial of degree k, and α_j can be chosen so that $\langle \phi_k, \phi_i \rangle_w = 0$, $i = 0, 1, \ldots, k-1$, or

$$\langle \phi_k, \phi_i \rangle_w = \langle x\phi_{k-1}, \phi_i \rangle_w - \sum_{j=0}^{k-1} \alpha_j \langle \phi_i, \phi_j \rangle_w = \langle x\phi_{k-1}, \phi_i \rangle_w - \alpha_i \langle \phi_i, \phi_i \rangle_w = 0, \qquad i = 0, 1, \cdots, k-1.$$

So $\alpha_i = \langle x\phi_{k-1}, \phi_i \rangle_w / \langle \phi_i, \phi_i \rangle_w$. But we can do even better. Since ϕ_{k-1} is orthogonal to all polynomials of degree $k-2$ or less, we get

$$\langle x\phi_{k-1}, \phi_i \rangle_w = \langle \phi_{k-1}, x\phi_i \rangle_w = 0 \quad \text{for} \quad i + 1 < k - 1.$$

So, we are left only with α_{k-1} and α_{k-2}. The following theorem concludes the argument:

Theorem 8.1. *The sequence of orthogonal polynomials can be defined as follows:*

$$\begin{aligned} &\phi_0(x) = 1, \qquad \phi_1(x) = x - B_1 \\ &\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k \phi_{k-2}(x), \qquad k \geq 2 \end{aligned}$$

with

$$B_k = \frac{\langle x\phi_{k-1}, \phi_{k-1} \rangle_w}{\|\phi_{k-1}\|_w^2}, \qquad C_k = \frac{\langle x\phi_{k-1}, \phi_{k-2} \rangle_w}{\|\phi_{k-2}\|_w^2} = \frac{\|\phi_{k-1}\|_w^2}{\|\phi_{k-2}\|_w^2}$$

The last simplification of C_k is given by:

$$\begin{aligned} \langle x\phi_{k-1}, \phi_{k-2} \rangle_w &= \langle \phi_{k-1}, x\phi_{k-2} \rangle_w \\ \phi_{k-1} &= x\phi_{k-2} - B_{k-1}\phi_{k-2} - C_{k-1}\phi_{k-3}. \end{aligned}$$

Solve the second with respect to $x\phi_{k-2}$, replace it into the right hand side of the first expression, and use the orthogonality conditions.

Example 8.2. *For the inner product*

$$\langle f, g \rangle = \int_{-1}^{1} f(x) g(x) dx$$

we get

$$\begin{array}{lllll} \phi_0 = 1, & \langle x\phi_0, \phi_0 \rangle = 0, & \langle \phi_0, \phi_0 \rangle = 2, & B_1 = 0, & \\ \phi_1 = x, & \langle x\phi_1, \phi_1 \rangle = 0, & \langle \phi_1, \phi_1 \rangle = \frac{2}{3}, & B_2 = 0, & C_2 = \frac{1}{3} \\ \phi_2 = x^2 - \frac{1}{3} & \langle x\phi_2, \phi_2 \rangle = 0, & \langle \phi_2, \phi_2 \rangle = \frac{8}{45}, & B_3 = 0, & C_3 = \frac{4}{15} \\ \phi_3 = x^3 - \frac{3}{5}x, & \textit{etc.} & & & \end{array}$$

These are the well known Legendre polynomials.

Example 8.3. *Let $w(x) = 1/\sqrt{1-x^2}$, and $[a,b] = [-1,1]$. We then get the sequence of polynomials:*

$$\begin{array}{llllll}
\phi_0 = 1, & \langle x\phi_0, \phi_0\rangle_w = 0, & \langle \phi_0, \phi_0\rangle_w = \pi, & B_1 = 0, & \\
\phi_1 = x, & \langle x\phi_1, \phi_1\rangle_w = 0, & \langle \phi_1, \phi_1\rangle_w = \frac{\pi}{2}, & B_2 = 0, & C_2 = \frac{1}{2} \\
\phi_2 = x^2 - \frac{1}{2} & \langle x\phi_2, \phi_2\rangle_w = 0, & \langle \phi_2, \phi_2\rangle_w = \frac{\pi}{2}, & B_3 = 0, & C_3 = \frac{1}{4} \\
\phi_3 = x^3 - \frac{3}{4}x, & \textit{etc.} & & &
\end{array}$$

These are nothing but the monic Chebyshev polynomials $\tilde{T}_k$.

The following theorem will become useful:

Theorem 8.4. *Let $f \in C[a,b]$, $f \not\equiv 0$ satisfying $\langle f, p\rangle_w = 0$ for all $p \in P_{k-1}$. Then f changes signs at least k times on (a,b).*

Proof. By contradiction. Suppose that f changes sign only $r < k$ times, at the points $t_1 < t_2 < \cdots < t_r$. Then f will not change sign on each of the subintervals:

$$(a,t_1), (t_1,t_2), \cdots, (t_{r-1},t_r), (t_r,b).$$

Let $p(x) = \prod_{i=1}^{r}(x-t_i) \in \mathbb{P}_r \subseteq \mathbb{P}_{k-1}$. Then $p(x)$ has the same sign properties as $f(x)$, and $f(x)p(x)$ does not change sign on the interval. Since $w > 0$ we get

$$\int_a^b w(x)f(x)p(x) \neq 0$$

which contradicts the assumption of the theorem. □

Corollary 8.5. *The orthogonal polynomial ϕ_k has exactly k distinct zeros in (a,b).*

9 Bernoulli polynomials and the Euler-Maclaurin formula.

Let the integral

$$I(f) = \int_a^b f(x)dx$$

be approximated by the Trapezoidal rule

$$T(h) = h\left(\frac{1}{2}f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}f(x_n)\right)$$

where $x_i = a + ih$, $i = 0, \cdots, n$ with $h = (b-a)/n$. The aim of this note is to prove that the error can be written as

$$I(f) - T(h) = \sum_{k=1}^{m-1} \frac{b_{2k}}{(2k)!}h^{2k}(f^{(2k-1)}(a) - f^{(2k-1)}(b)) - \frac{b_{2m}}{(2m)!}h^{2m}(b-a)h^{2m}f^{(2m)}(\eta) \quad (32)$$

where $\eta \in (a,b)$, and b_k are the Bernoulli numbers. These will be defined later. Obviously, the formula requires $f \in C^{2m}[a,b]$. The formula proves that the error can be written as an even power series of h, which is the theoretical fundament for the development of the Romberg integration algorithm.

We will first define the well known Bernoulli polynomials which will then be used to prove Euler-Maclaurins formula. This again will be used to prove (32).

Bernoulli polynomials.

For our purpose, it is convenient to define the Bernoulli polynomials by the following reccurence relation:

$$B_0(t) = 1,$$
$$B_k'(t) = kB_{k-1}(t) \quad \text{and} \quad \int_0^1 B_k(t) = 0, \qquad k \geq 1.$$

The first few polynomials becomes

$$B_1(t) = t - \frac{1}{2}$$
$$B_2(t) = t^2 - t + 1/6$$
$$B_3(t) = t^3 - \frac{3}{2}t^2 + \frac{1}{2}t.$$

We will need the following properties of these polynomials:

$$B_k(0) = B_k(1), \qquad k \geq 2, \tag{33a}$$
$$B_k(1-t) = (-1)^k B_k(t), \tag{33b}$$
$$B_k(t) - B_k(0) \qquad \text{has no zeros in (0,1) if } k \text{ is even.} \tag{33c}$$

The *Bernoulli numbers* is given by $b_k = B_k(0)$. As a consequence of (33a) and (33b) we get $b_k = 0$ for k odd and $k \geq 3$. The Bernoulli numbers can also be found by the generating function

$$\frac{t}{e^t - 1} = \sum_{k=0}^{\infty} \frac{b_k}{k!} t^k.$$

Euler-Maclaurins formula

By repeated use of integration by parts and the Bernoulli polynomials, we get

$$\begin{aligned}
\int_0^1 f(t)dt &= \int_0^1 f(t)B_0(t)dt \\
&= f(t)B_1(t)\Big|_0^1 - \int_0^1 B_1(t)f'(t)dt \\
&= \frac{1}{2}[f(1) + f(0)] - \frac{1}{2}B_2(t)f'(t)\Big|_0^1 + \frac{1}{2}\int_0^1 B_2(t)f''(t)dt \\
&= \frac{1}{2}[f(1) + f(0)] - \sum_{j=2}^{\bar{m}} \frac{(-1)^j}{j!} B_j(t)f^{(j-1)}(t)\Big|_0^1 + (-1)^{\bar{m}} \frac{1}{\bar{m}!}\int_0^1 B_n(t)f^{(n)}(t)dt.
\end{aligned}$$

Now, since $B_{2k+1}(0) = B_{2k+1} = 0$ and $B_{2k}(0) = B_{2k}(1) = b_{2k}$ this can be written as

$$\begin{aligned}
\int_0^1 f(t)dt = \frac{1}{2}[f(1) + f(0)] &- \sum_{k=1}^{m-1} \frac{b_{2k}}{(2k)!}\left[f^{(2k-1)}(1) - f^{(2k-1)}(0)\right] \\
&- \frac{b_{2m}}{(2m)!}\left[f^{(2m-1)}(1) - f^{(2m-1)}(0)\right] + \frac{1}{(2m)!}\int_0^1 B_{2m}(t)f^{(2m)}(t)dt.
\end{aligned}$$

Using property (33c) and the mean value theorem for integrals, the last two terms becomes

$$-\frac{1}{(2m)!}\int_0^1 \big[b_{2m} - B_{2m}(t)\big] f^{(2m)}(t)dt = -\frac{b_{2m}}{(2m)!} f^{(2m)}(\eta)$$

As a result, we get the *Euler-Maclaurin formula*:

$$\int_0^1 f(t)dt = \frac{1}{2}[f(1)+f(0)] - \sum_{k=1}^{m-1} \frac{b_{2k}}{(2k)!}\big[f^{(2k-1)}(1) - f^{(2k-1)}(0)\big] - \frac{b_{2m}}{(2m)!} f^{(2m)}(\eta).$$

Error of the Trapezoidal rule.

Applying the Euler-Maclaurin formula to a function $g(t) = f(x_i + ht)$, using the change of variable $t = (x - x_i)/h$ gives

$$\int_{x_i}^{x_{i+1}} f(x)dx = \frac{h}{2}[f(x_i)+f(x_{i+1})] - \sum_{k=1}^{m-1} \frac{b_{2k}}{(2k)!} h^{2k}\big[f^{(2k-1)}(x_{i+1}) - f^{(2k-1)}(x_i)\big] - \frac{b_{2m}}{(2m)!} h^{2m+1} f^{(2m)}(\eta_i)$$

The expression (32) is finally obtained by summing over all the intervals $[x_i, x_{i+1}]$.

10 Gauss quadrature and orthogonal polynomials.

The aim of this section is to construct "optimal" quadrature formulas. To be more specific, given the integral

$$I_w(f) = \int_a^b w(x) f(x) dx \tag{34}$$

in which $w(x)$ is a fixed, positive function. We want to approximate this using a quadrature formula on the form

$$Q_w(f) = \sum_{i=1}^{n} A_i f(x_i).$$

Such a formula can be constructed as follows: Choose n distinct nodes, $x_1, x_2, \cdots, x_n$ in the interval $[a, b]$. Construct the interpolation polynomial

$$p_{n-1}(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x), \qquad \ell_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

An approximation to the integral is then given by

$$Q_w(f) = \int_a^b w(x) p_{n-1}(x) dx = \sum_{i=1}^{m} A_i f(x_i), \qquad A_i = \int_a^b w(x)\ell_i(x)dx. \tag{35}$$

The quadrature formula is of precision m if

$$I_w(p) = Q_w(p), \qquad \text{for all } p \in \mathbb{P}_m.$$

From the construction, these quadrature formulas is of precision at least $n-1$. The question is how too choose the nodes x_i, $i = 1, \ldots, n$ giving m as large as possible. The key concept here is *orthogonal polynomials*.

Orthogonal polynomials.

Given two functions $f, g \in C[a,b]$. We define an inner product of these two functions by

$$< f, g >= \int_a^b w(x) f(x) g(x) dx, \qquad w(x) > 0. \tag{36}$$

Thus the definition of the inner product depends on the integration interval $[a,b]$ and a given *weight function* $w(x)$. If $f, g, h \in C[a,b]$ and $\alpha \in \mathbb{R}$ then

$$\begin{aligned} \langle f, g\rangle_w &= \langle g, f\rangle_w \\ \langle f+g, h\rangle_w &= \langle f, h\rangle_w + \langle g, h\rangle_w \\ \langle \alpha f, g\rangle_w &= \alpha \langle f, g\rangle_w \\ \langle f, f\rangle_w &\geq 0, \qquad \text{and} \qquad \langle f, f\rangle_w = 0 \Leftrightarrow f \equiv 0. \end{aligned}$$

From an inner product, we can also define a norm on $C[a,b]$ by

$$\|f\|_w^2 = \langle f, f\rangle_w.$$

For the inner product (36) we also have

$$\langle xf, g\rangle_w = \int_a^b w(x) x f(x) g(x) dx = \langle f, xg\rangle_w. \tag{37}$$

Our aim is now to create an orthogonal basis for $\mathbb{P}$, that is, create a sequence of polynomials $\phi_k(x)$ of degree k (no more, no less) for $k = 0, 1, 2, 3, \ldots$ such that

$$\langle \phi_i, \phi_j\rangle_w = 0 \quad \text{for all} \quad i \neq j.$$

If we can make such a sequence, then

$$\mathbb{P}_{n-1} = \text{span}\{\phi_0, \phi_1, \cdots, \phi_{n-1}\} \qquad \text{and} \qquad \langle \phi_n, p\rangle_w = 0 \quad \text{for all} \qquad p \in \mathbb{P}_{n-1}.$$

Let us now find the sequence of orthogonal polynomials. This is done by a Gram-Schmidt process:

Let $\phi_0 = 1$. Let $\phi_1 = x - B_1\phi_0$ where B_1 is given by the orthogonality condition:

$$0 = \langle \phi_1, \phi_0\rangle_w = \langle x\phi_0, \phi_0\rangle_w - B_1\langle \phi_0, \phi_0\rangle_w \qquad \Rightarrow \qquad B_1 = \frac{\langle x\phi_0, \phi_0\rangle_w}{\|\phi_0\|_w^2}.$$

Let us now assume that we have found ϕ_j, $j = 0, 1, \cdots, \phi_k$. Then, let

$$\phi_k = x\phi_{k-1} - \sum_{j=0}^{k-1} \alpha_j \phi_j.$$

Clearly, ϕ_k is a polynomial of degree k, and α_j can be chosen so that $\langle \phi_k, \phi_i\rangle_w = 0$, $i = 0, 1, \ldots, k-1$, or

$$\langle \phi_k, \phi_i\rangle_w = \langle x\phi_{k-1}, \phi_i\rangle_w - \sum_{j=0}^{k-1} \alpha_j \langle \phi_i, \phi_j\rangle_w = \langle x\phi_{k-1}, \phi_i\rangle_w - \alpha_i \langle \phi_i, \phi_i\rangle_w = 0, \qquad i = 0, 1, \cdots, k-1.$$

So $\alpha_i = \langle x\phi k-1, \phi_i\rangle_w / \langle \phi_i, \phi_i\rangle_w$. But we can do even better. Since ϕ_{k-1} is orthogonal to all polynomials of degree $k-2$ or less, we get

$$\langle x\phi_{k-1}, \phi_i\rangle_w = \langle \phi_{k-1}, x\phi_i\rangle_w = 0 \quad \text{for} \quad i+1 < k-1.$$

So, we are left only with α_{k-1} and α_{k-2}. The following theorem concludes the argument:

Theorem 10.1. *The sequence of orthogonal polynomials can be defined as follows:*

$$\begin{aligned} &\phi_0(x) = 1, \qquad \phi_1(x) = x - B_1 \\ &\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x), \qquad k \geq 2 \end{aligned}$$

with

$$B_k = \frac{\langle x\phi_{k-1}, \phi_{k-1}\rangle_w}{\|\phi_{k-1}\|_w^2}, \qquad C_k = \frac{\langle x\phi_{k-1}, \phi_{k-2}\rangle_w}{\|\phi_{k-2}\|_w^2} = \frac{\|\phi_{k-1}\|_w^2}{\|\phi_{k-2}\|_w^2}$$

The last simplification of C_k is given by:

$$\begin{aligned} \langle x\phi_{k-1}, \phi_{k-2}\rangle_w &= \langle \phi_{k-1}, x\phi_{k-2}\rangle_w \\ \phi_{k-1} &= x\phi_{k-2} - B_{k-1}\phi_{k-2} - C_{k-1}\phi_{k-3}. \end{aligned}$$

Solve the second with respect to $x\phi_{k-2}$, replace it into the right hand side of the first expression, and use the orthogonality conditions.

For the inner product

$$\langle f, g\rangle = \int_{-1}^{1} f(x)g(x)dx$$

we get the well known *Legendre polynomials*,

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= x^2 - \frac{1}{3} \\ P_3(x) &= x^3 - \frac{3}{5}x \\ &\vdots \end{aligned}$$

see Burden and Faires, sec. 8.2.

Example 10.2. *Let* $w(x) = 1/\sqrt{1-x^2}$, *and* $[a, b] = [-1, 1]$. *We then get the sequence of polynomials:*

$$\begin{array}{lllll} \phi_0 = 1, & \langle x\phi_0, \phi_0\rangle_w = 0, & \langle \phi_0, \phi_0\rangle_w = \pi, & B_1 = 0, & \\ \phi_1 = x, & \langle x\phi_1, \phi_1\rangle_w = 0, & \langle \phi_1, \phi_1\rangle_w = \frac{\pi}{2}, & B_2 = 0, & C_2 = \frac{1}{2} \\ \phi_2 = x^2 - \frac{1}{2} & \langle x\phi_2, \phi_2\rangle_w = 0, & \langle \phi_2, \phi_2\rangle_w = \frac{\pi}{2}, & B_3 = 0, & C_3 = \frac{1}{4} \\ \phi_3 = x^3 - \frac{3}{4}x, & \textit{etc.} & & & \end{array}$$

These are nothing but the monic Chebyshev polynomials $\tilde{T}_k$ *(see Burden and Faires, sec. 8.2).*

The following theorem will become useful:

Theorem 10.3. *Let $f \in C[a,b]$, $f \not\equiv 0$ satisfying $\langle f, p\rangle_w = 0$ for all $p \in P_{k-1}$. Then f changes signs at least k times on (a,b).*

Proof. By contradiction. Suppose that f changes sign only $r < k$ times, at the points $t_1 < t_2 < \cdots < t_r$. Then f will not change sign on each of the subintervals:

$$(a,t_1),(t_1,t_2),\cdots,(t_{r-1},t_r),(t_r,b).$$

Let $p(x) = \prod_{i=1}^{r}(x-t_i) \in \mathbb{P}_r \subseteq \mathbb{P}_{k-1}$. Then $p(x)$ has the same sign properties as $f(x)$, and $f(x)p(x)$ does Lt change sign on the interval. Since $w > 0$ we get

$$\int_a^b w(x)f(x)p(x) \neq 0$$

which contradicts the assumption of the theorem. □

Corollary 10.4. *The orthogonal polynomial ϕ_k has exactly k distinct zeros in (a,b).*

Gaussian quadrature

The main result of this section is the following result:

Theorem 10.5. *Let ϕ_n be the orthogonal polynomial of degree n based on the inner product* (36)*. Let $x_1, x_2, \cdots, x_n$ be the distinct zeros of ϕ_n. Using these as nodes, the quadrature formula constructed by* (35) *is of precision $2n-1$.*

Proof. Let $p \in \mathbb{P}_{2n-1}$. By polynomial division with ϕ_n we get

$$p = q\phi_n + r, \qquad q, r \in \mathbb{P}_{n-1}.$$

By orthogonality, we get

$$I_w(p) = \int_a^b w(x)p(x)dx = \int_a^b w(x)q(x)\phi_n(x)dx + \int_a^b w(x)r(x)dx = \int_a^b w(x)r(x)dx = I_w(r)$$

And, since the nodes are zeros of ϕ_n we get

$$Q_w(p) = \sum_{i=1}^{n} A_i p(x_i) = \sum_{i=1}^{n} A_i q(x_i)\phi_n(x_i) + \sum_{i=1}^{n} A_i r(x_i) = Q_w(r).$$

But by construction $Q_w(f)$ has precision at least $n-1$, so $Q_w(r) = I_w(r)$. □

For Gaussian quadratures it is possible to prove that, for $f \in C^{2n}[a,b]$ we get

$$I_w(f) = Q_w(f) + E \tag{38}$$

with

$$E = \frac{f^{(2n)}(\nu)}{(2n)!}\int_a^b w(x)\omega(x)^2 dx, \qquad \omega(x) = \prod_{i=1}^{n}(x-x_i), \qquad \nu \in (a,b).$$

Example 10.6. *Let $w(x) = 1$ and $[a, b] = [-1, 1]$, giving the Legendre polynomials as orthogonal polynomials. Choose e.g. $n = 3$, so that*

$$P_3(x) = x^3 - \frac{3}{5}x.$$

The zeros is $x_1 = -\sqrt{3/5}$, $x_2 = 0$ and $x_3 = \sqrt{3/5}$. From (35) we get $A_1 = A_3 = 5/9$, $A_2 = 8/9$, so

$$Q(f) = \frac{1}{9}\big(5f(x_1) + 8f(x_0) + 5f(x_2)\big). \tag{39}$$

The error of this formula becomes

$$E = \frac{f^{(6)}(\nu)}{15750}.$$

a) *Let us apply (39) to the integral*

$$\int_{-1}^{1} e^x dx = 2.350402387.$$

This gives

$$\frac{1}{9}\big(5e^{-\sqrt{3/5}} + 8e^0 + 5e^{\sqrt{3/5}}\big) = 2.350336929$$

giving an error on $6.54 \cdot 10^{-5}$.

b) *Using the same formula on*

$$\int_{-1}^{1} \frac{e^x}{\sqrt{1-x^2}} dx = 3.977463261$$

gives $Q(f) = 3.199641132$ which is a rather crude approximation. The reason is that $f^{(2n)}(x) \to \infty$ for $x \to \pm 1$ in this example.

Example 10.7. *Let $w(x) = 1/\sqrt{(1-x^2)}$ and $[a, b] = [-1, 1]$, giving the Chebyshev polynomials as orthogonal basis. Choose $n = 3$, so that*

$$\tilde{T}_3(x) = x^3 - \frac{3}{4}x,$$

with zeros $x_1 = -\sqrt{3}/2$, $x_2 = 0$ and $x_3 = \sqrt{3}/2$. The corresponding weights becomes $A_1 = A_2 = A_3 = \pi/3$, and

$$Q_w(f) = \frac{\pi}{3}\big(f(_1) + f(x_2) + f(x_3)\big).$$

The formula applied to the integral $\int_{-1}^{1} e^x/\sqrt{1-x^2}dx$ is

$$Q_w(e^x) = \frac{\pi}{3}\big(e^{-\sqrt{3}/2} + e^0 + e^{\sqrt{3}/2}\big) = 3.97732196$$

which is an significant improvement from the result of Example 9.6.b).