Mathematics for chemical engineers

Drahoslava Janovská

4. Numerical solution of ordinary differential equations

Initial value problem

Outline

Introduction

Numerical solution of differential equations – necessity for engineering applications.

In this lecture we study numerical methods for solving a first order differential equation

$$y' = f(x, y)\,, \quad y(x_0) = y_0\,.$$

In particular

- Euler's method, which is really too crude to be of much use in practical applications. However, its simplicity allows for an introduction to the ideas required to understand the better methods.
- The Runge-Kutta method, perhaps the most widely used method for numerical solution of differential equations.

★ Solution of the initial value problem by a one step method

We solve the initial value (Cauchy) problem

$$y' = f(x, y), \quad y(x_0) = y_0 \,.$$

Uniqueness and existence of a solution:
If $f(x, y)$ is continuous in
$\Omega = \{(x, y),\ |x - x_0| \leq a,\ a > 0,\ |y - y_0| \leq b,\ b > 0\}$ and if we denote

$$M = \max_{(x,y)\in\Omega} |f(x, y)|\,, \quad \alpha = \min(a, \frac{b}{M})\,,$$

then there exist a solution of the equation $y' = f(x, y)$ defined in the interval $(x_0 - \alpha, x_0 + \alpha)$.

Moreover, if f is a Lipschitz function, i.e.

$$|f(x, y_1) - f(x, y_2)| \leq L\,|y_1 - y_2| \quad L > 0, \quad \forall (x, y_1), (x, y_2) \in \Omega\,,$$

the solution is unique.

Remark: L ... Lipschitz constant

Taylor expansion of the solution

We are looking for a function $y = y(t)$ such that

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y' = f(t, y), \quad y(0) = y_0\,, \quad 0 < t < T\,.$$

Let us suppose that we have already computed the solution $y_n := y(t_n)$ in time t_n and we want to find a solution in time t_{n+1}. Let $h := t_{n+1} - t_n$ be the corresponding time step.

Taylor expansion of y_{n+1} in the point y_n:

$$y_{n+1} = y_n + h \cdot \underbrace{y_n'}_{= f(t_n, y_n) \text{ from the differential equation}} + \frac{h^2}{2} y_n'' + \frac{h^3}{6} y_n''' + \ldots$$

And y_n''?

$$y'' = \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)' = \frac{\partial f}{\partial t} \cdot 1 + \frac{\partial f}{\partial y} \cdot y' = f_t + f_y \cdot f$$

$$y''' = \frac{\partial}{\partial t}(f_t + f\, f_y) + \frac{\partial}{\partial y}(f_t + f\, f_y)\, f = f_{tt} + 2f\, f_{yt} + f_t\, f_y + f\, f_y^2 + f^2\, f_{yy}\,,$$

... etc.

Error of the discretization

What is the difference between the solution obtained by a numerical method and the exact solution? In other words what is the global discretization error

$$e_i = y(x_i) - y_i \ ?$$

If we solve numerically a differential equation we compute each iteration with a so called local discretization error. The global discretization error is just the accumulation of the local discretization errors. The methods that manage to keep a small error with respect to the exact solution are called stable methods.

For a description of the rate of convergence we use the term order of the method. The order of the method is a natural number p such that for small h local discretization error is of order h^{p+1}.
For example

$$y_{n+1} = y_n + hy'_n + \underbrace{\mathcal{O}(h^2)}_{\text{local discretization error}} \quad \text{means that} \quad \lim_{h \to O_+} \frac{y_{n+1}}{h^2} = k \neq 0,$$

the method is of the order $p = 1$. The following Euler's method is the method of the first order.

Euler's method

Euler's method is based on the assumption that the tangent line to the integral curve at $(t_n, y(t_n))$ approximates the integral curve over the interval (t_n, t_{n+1}). Because of the linearization, we use only the first two terms of the Taylor expansion:

We choose $y(0) := y_0$ and construct a sequence

$$y_{n+1} = y_n + hf(t_n, y_n)\,, \quad n = 0, 1, 2, \dots .$$

Let us note that Euler's method is the first order method.

Remark The step of the method h can be changed with a particular iteration (adaptive choice of step):

$$y_{n+1} = y_n + h_n f(t_n, y_n)\,.$$

★

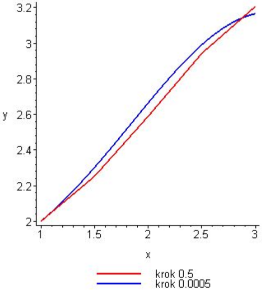
Example Solve the following initial value problem by Euler's method. Find the solution in time $t = 3$,

$$y' = 0,3\, y\, \sin(t)\,, \qquad y(1) = 2\,.$$

For simplicity, let us consider $n = 4$, i.e., $h = 0.5$.

$$y_{j+1} = y_j + h\, f(t_j, y_j), \quad \text{where} \quad f(t_j, y_j) = 0,3 y_j sin(t_j)\,.$$

t_j	step 0.5	step 0.005
1	2	2
1.5	2.252441295	2.30249902026881692
2	2.589461130	2.66460601831410714
2.5	2.942649681	2.99089235783755570
3	3.206813761	3.16533517440834976



★ Example

Example

Solve the following initial value problem by Euler's method. Use the integration step $h = 0.2$ and compute three iterations.

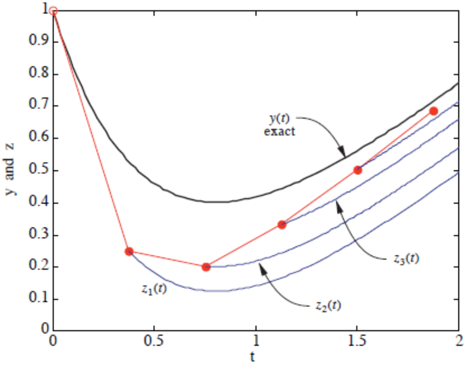
$$y' = t - 2y\,, \quad y(0) = 1\,.$$

Exact solution: $y(t) = \frac{1}{4}\left[2t - 1 + 5e^{-2t}\right]$.

j	t_j	$f(t_{j-1}, y_{j-1})$	Euler's method $y_j = y_{j-1} + hf(t_{j-1}, y_{j-1})$	exact sol. $y(t_j)$	error $y_j - y(t_j)$
0	0.0		initian cond. $= 1.0000$	0	0
1	0.2	$0 - 2 \cdot 1 = -2.000$	$1.0 + (0.2)(-2.0) = 0.6000$	0.6879	-0.0879
2	0.4	$0.2 - (2)(0.6) = -1.000$	$0.6 + (0.2)(-1.0) = 0.4000$	0.5117	-0.1117
3	0.6	$0.4 - (2)(0.4) = -0.400$	$0.4 + (0.2)(-0.4) = 0.3200$	0.4265	-0.1065

★

In the following figure, you may see a comparison of the exact solution and the numerical one by (●) Euler's method. Integral curves $z(t)$ start always from points of the numerical solution as from the new initial value condition for the given equation.



★ Stability of Euler's method

Let us solve the model problem

$$y' = \lambda y\,, \quad \lambda \text{ a constant} \tag{1}$$

Euler's method $\Longrightarrow$

$$y_{n+1} = y_n + \lambda h\, y_n\,, \quad \text{t.j.} \quad y_{n+1} = y_n(1 + \lambda\, h)\,,$$

i.e., we obtain

$$y_n = y_{n-1}(1 + \lambda\, h) = y_{n-2}(1 + \lambda\, h)^2 = \ \dots \ = y_0(1 + \lambda\, h)^n\,.$$

For $\lambda = \lambda_1 + \mathrm{i}\lambda_2$ ... imaginary, we have

$$y_n = y_0(\underbrace{1 + \lambda_1 h + \mathrm{i}\lambda_2 h}_{})^n = y_0\, \sigma^n\,.$$

σ ... so called amplification factor

The numerical solution is stable (i.e. it will remain limited for the growing (large) n, too), if $|\sigma| \le 1$.

★

Let in (1), $\lambda = \lambda_1 + \mathrm{i}\lambda_2, \quad \lambda_1 \le 0\,, \quad \sigma = 1 + \lambda_1 h + \mathrm{i}\lambda_2 h\,.$

Then the region of stability for Euler's method is part of the left half of the complex plane, in particular inside of the circle

$$|\sigma|^2 = (1 + \lambda_1 h)^2 + \lambda_2^2 h^2 = 1\,.$$

For any value of λh in the left half of the complex plane outside of this circle the numerical solution is blowing up, while the exact solution decreases. If we want to have a stable solution, we must reduce the h so, that λh would be inside the circle.

Implicit (backward) Euler's method

The y_{n+1} occurs in the equation implicitly:

$$y_{n+1} = y_n + h\, f(t_{n+1}, y_{n+1})\,.$$

Disadvantage: the method is computationally more demanding than explicit Euler's method.

Advantage: it is more stable, sometimes linearization of f can be exploited with advantage.

★ **Example** Use Euler's implicit method with step sizes $h = 0.5, h = 0.1$ to find approximate values of the solution of the initial value problem

$$y' = \frac{2t+1}{5t^4+1}\,, \quad y(2) = 4, \quad \text{at points} \quad t = 4, t = 5\,.$$

Present your results in tabular form. Compute the analytic solution by separation of variables and compare the exact values with results obtained by Euler's implicit method.

★ Implicit Euler's method – Example

Example Let us apply the implicit Euler's method to the model problem (1):

$$y_{n+1} = y_n + \lambda h y_{n+1} \quad \Longrightarrow \quad y_{n+1} = \frac{1}{1-\lambda h} y_n \,, \quad \text{i.e.,}$$

$$y_n = \frac{1}{1-\lambda h} y_{n-1} = \left(\frac{1}{1-\lambda h}\right)^2 y_{n-2} = \ldots = \left(\frac{1}{1-\lambda h}\right)^n y_0 \,.$$

We obtain:

$$y_n = \sigma^n y_0 \,, \qquad \sigma = \frac{1}{1-\lambda h} \,.$$

★ θ–methods

We can define the following one-parameter class of one-step methods, called $\theta-$methods:

For a given initial approximation y_0, we define y_{n+1} as a convex combination of $f(t_n, y_n)$ and $f(t_{n+1}, y_{n+1})$ (θ . . . parameter):

$$y_{n+1} = y_n + h[(1-\theta)f(t_n, y_n) + \theta f(t_{n+1}, y_{n+1})]\,, \; n = 0, 1, \ldots, N-1\,, \; \theta \in \langle 0, 1\rangle\,.$$

- $\theta = 0 \implies y_{n+1} = y_n + hf(t_n, y_n)$ . . . (explicit) Euler's method
- $\theta = 1 \implies y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ . . . implicit Euler's method
- $\theta = \frac{1}{2} \implies y_{n+1} = y_n + \frac{1}{2}h[f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$ . . . trapezoidal rule.

It can be shown that a $\theta-$method is explicit for $\theta = 0$ and it is implicit for $0 < \theta \leq 1$.

Runge - Kutta methods

Runge - Kutta methods (RK) – more precise than Euler's methods:

- explicit: The solution in time t_{n+1} is computed from the values $y_n, f(t_n, y_n)$ and from $f(t, y)$ enumerated at a point between points t_n and t_{n+1} $\Longrightarrow$ better accuracy because we use more information about the function f .
- implicit: They usually lead to the solution of nonlinear algebraic equations, but the amount of work involved is balanced by better numerical stability.

RK method of the second order

Let us solve again the equation $y' = f(t, y)$.
In the time step t_{n+1}, we obtain the solution from the equation

$$y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2 \,, \tag{2}$$

where

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + \alpha h, y_n + \beta k_1,) \,, \quad \alpha, \beta, \gamma_1, \gamma_2 \in \mathbb{R} \,. \end{aligned}$$

The constants $\alpha, \beta, \gamma_1, \gamma_2$ have to be determined in such a way that the method should have the highest order of accuracy possible. To determine the order of accuracy, we exploit the Taylor expansion of $y(t_{n+1})$

$$y_{n+1} = y_n + h \underbrace{y'_n}_{f(t_n, y_n)} + \frac{h^2}{2} \underbrace{y''_n}_{f_t + f\, f_y} + \cdots \quad \Longrightarrow$$

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2}(f_{t_n} + f_n f_{y_n}) + \cdots \tag{3}$$

and compare the coefficients in (2) and (3).

★

Taylor series for the function of two variables $k_2 = hf(t_n + \alpha h, y_n + \beta k_1) \quad \Longrightarrow$

$$k_2 = h\left(f(t_n, y_n) + \beta k_1 f_{y_n} + \alpha h f_{t_n} + \mathcal{O}(h^2)\right) .$$

Remark Symbol $\mathcal{O}$ (capital O)

$g(h) = \mathcal{O}(h^p) \quad \Longleftrightarrow \quad |g(h)| \le C \cdot h^p , \quad C$ is a constant independent of h.

Then $\quad y_{n+1} = y_n + (\gamma_1 + \gamma_2)hf_n + \gamma_2 \beta h^2 f_n f_{y_n} + \gamma_2 \alpha h^2 f_{t_n} + \cdots$

We compare the result with (3) and obtain three nonlinear equations for 4 unknowns:

$$\gamma_1 + \gamma_2 = 1 , \quad \gamma_2 \alpha = \frac{1}{2} , \quad \gamma_2 \beta = \frac{1}{2} .$$

Let $\alpha \in \mathbb{R}$ be a parameter. Then $\quad \gamma_2 = \dfrac{1}{2\alpha} , \quad \beta = \alpha , \quad \gamma_1 = (1 - \dfrac{1}{2\alpha}) .$

We obtain Runge - Kutta methods of the 2nd order:

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

We choose α and get the method. For example

$$\alpha = \frac{1}{2} \quad \Longrightarrow \quad \gamma_2 = 1\,, \quad \beta = \frac{1}{2}\,, \quad \gamma_1 = 0 \quad \Longrightarrow$$

$$y_{n+1} = y_n + k_2 = y_n + hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)\,.$$

Remark RK method of the 2nd order requires at each step twice quantification of function values.

RK methods of 4th order

For solving initial value problems, the RK methods of 4th order are mostly used

$$y_{n+1} = y_n + \frac{1}{6}k_1 + \frac{1}{3}(k_2 + k_3) + \frac{1}{6}k_4 \,,$$

where

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + \tfrac{h}{2}, y_n + \tfrac{1}{2}k_1) \\ k_3 &= hf(t_n + \tfrac{h}{2}, y_n + \tfrac{1}{2}k_2) \\ k_4 &= hf(t_n + h, y_n + k_3)\,. \end{aligned}$$

At each step, we need to compute function values 4 times.

Although laborious, the RK method of 4th order is stable and very accurate. It is easily programmable, because it requires no differentiation, only computation of function values.

★ Example

By Runge-Kutta method of 4th order solve the initial value problem

$$y' = t^2 - y\,, \quad y(0) = 1,$$

with step $h = 0.1$ on interval $\langle 0; 0.5 \rangle$.

Solution The data $t_0 = 0$, $y_0 = 1$, $f(t, y) = t^2 - y$ are given, we will compute y_1, i.e., the approximation of the solution in $t_1 = 0.1$.

$$\begin{aligned}
k_1 &= f(0; 1) = 0^2 - 1 \\
k_2 &= f(0 + \frac{1}{2}0.1; 1 + \frac{1}{2}0.1(-1)) = f(0.05; 0.95) = -0.9475 \\
k_3 &= f(0 + \frac{1}{2}0.1; 1 + \frac{1}{2}0.1(-0.9475)) = f(0.05; 0.952625) = -0.950125 \\
k_4 &= f(0 + 0.1; 1 + 0.1(-0.950125)) = f(0.1; 0.9049875) = -0.8949875 \\
y_1 &= y_0 + \frac{1}{6}0.1(k_1 + 2k_2 + 2k_3 + k_4) \doteq 0.9051627.
\end{aligned}$$

For comparison: the exact solution of our problem is $y = -\mathrm{e}^{-t} + t^2 - 2t + 2$ and $y(0.1) = 0.9051626$.

Compute aproximation of the solution and exact solution in points 0.2; 0.3; 0.4 and 0.5.

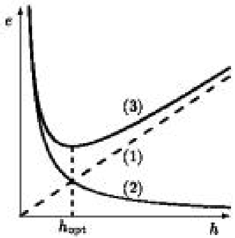
Computational error

Computational error

a) $y(t_n)$ ... exact solution in time t_n, y_n ... approximate solution in time t_n
$e_n = y_n - y(t_n)$ ... global error of the approximation

b) The computer works in finite arithmetic:
$r_n = \widetilde{y}_n - y_n$ ... rounding error
We want to compute f but in fact, we compute the numerical approximation $\widetilde{y}_n$.

For Euler's method:
(1) ... the global error of approximation e_n, E_n is directly proportional to the first power of h
(2) ... rounding error r_n is inversely proportional to the first power of h
(3) ... total error for Euler's method



About rounding errors, we can convince only if we repeat the calculation with the different pracission (double precission,...)

Richardson's extrapolation

Richardson's extrapolation is used to generate high-accuracy results while using loworder formulas.
Let us solve an initial value problem by a numerical method of the order p.
Let $y(x)$ be the exact solution of our problem. Let us choose two different steps $h = h_1$ and $h = h_2$ and let $y_1 = y(x, h_1)$ be the approximate value of the solution at the point x with the step h_1, $y_2 = y(x, h_2)$ with the step h_2. Then

$$y(x) \doteq y_1(x) + C \cdot h_1^p \qquad (4)$$

$$y(x) \doteq y_2(x) + C \cdot h_2^p, \qquad (5)$$

where C is a constant the same in both cases, independent on h.
From the equation (4) we substract the equation (5) and obtain

$$0 = y_2 - y_1 + C \cdot h_2^p - C \cdot h_1^p \quad \Rightarrow \quad C = \frac{y_2 - y_1}{h_1^p - h_2^p}\,.$$

We put this constant C into the equation (5):

$$y(x) \doteq y_2(x) + \frac{y_2 - y_1}{h_1^p - h_2^p} \cdot h_2^p \quad \Rightarrow \quad y(x) \doteq y_{12} = \frac{y_2 \left(\frac{h_1}{h_2}\right)^p - y_1}{\left(\frac{h_1}{h_2}\right)^p - 1}\,.$$

The approximation y_{12} is called Richardson's extrapolation of the solution y obtained from the values y_1 a y_2.

Linear multistep methods

One-step methods ... to find y_{n+1} we need information only from previous time level y_n

Multistep methods ... to find y_{n+1} we need information from more time levels (for example it is not sufficient to start from the initial condition)

Let us consider three consecutive time levels

$$t_{n-1}\,, \quad t_n = t_{n-1} + h\,, \quad t_{n+1} = t_{n-1} + 2h$$

and let us integrate a differential equation fom t_{n-1} to t_{n+1} using Simpson's rule:

$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 4f(x_{n-1}) + f(x_n)\right).$$

Remember also that

$$\int_{t_{n-1}}^{t_{n+1}} y'(t)\mathrm{d}t = y(t_{n+1}) - y(t_{n-1})\,.$$

So, we have

$$y(t_{n+1}) = y(t_{n-1}) + \int_{t_{n-1}}^{t_{n+1}} f(t, y(t))\mathrm{d}t \quad \approx$$

$$\approx y(t_{n-1}) + \frac{1}{3}h\left(f(t_{n-1}, y(t_{n-1})) + 4f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))\right)$$

Let $y_n \doteq y(t_n)$.

We obtain method

$$y_{n+1} = y_{n-1} + \frac{1}{3}h\left(f(t_{n-1}, y_{n-1}) + 4f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right) .$$

Let now a uniform partitioning with step h be given:

$$t_n, t_{n+1} = t_n + h, t_{n+2} = t_n + 2h, \ldots$$

The general linear $k-$steps method has the form:

$$\alpha_k y_{n+k} + \alpha_{k-1} y_{n+k-1} + \cdots + \alpha_0 y_n = h(\beta_k f_{n+k} + \beta_{k-1} f_{n+k-1} + \cdots + \beta_0 f_n)\,,$$

where constants $\alpha_j, \beta_j \in \mathbb{R},\ \alpha_k \neq 0$ and $\alpha_0^2 + \beta_0^2 > 0$; $f_n \doteq f(t_n, y(t_n))$.

If $\beta_k = 0$, then y_{n+k} can be computed explicitly from values $y_n, \ldots, y_{n+k-1}$ and from values of the function f in the previous time levels
$\Longrightarrow$ explicit $k-$step method

If $\beta_k \neq 0$, then y_{n+k} appears on both sides of the equation and as a consequence the method is implicit.

Remark linear – in the formula, only a linear combination of y_n and $f(t_n, y_n)$ occurs.

Examples

- Four-steps linear explicit Adams–Bashforth's method

$$y_{n+4} = y_{n+3} + \frac{1}{24}h\left(55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n\right)$$

- Four-steps linear implicit Adams–Moulton's method

$$y_{n+4} = y_{n+3} + \frac{1}{24}h\left(9f_{n+4} + 19f_{n+3} - 5f_{n+2} - 9f_{n+1}\right)$$

Remark Before we can apply a $k-$step method, we need to know k initial values $y_0, \ldots, y_{k-1}$, where y_0 is a given initial condition, $y_1, \ldots, y_{k-1}$ must be somehow calculated for example by Euler's method or RK method. In any case, the data contain numerical errors and it is important to know how these errors affect other approximations $y_n,\ n \geq k$, that are calculated by $k-$step method. Thus, we are interested in stability of numerical methods with respect to small perturbations of initial data.

★ Stability of $k-$steps methods

How to determine the stability?

Let $\{t_n\}$ be a uniform partitioning with step h.

A general linear $k-$step method has the form

$$\alpha_0 y_n+\alpha_1 y_{n+1}+\cdots+\alpha_k y_{n+k} = h\left(\beta_0 f(t_n, y_n) + \beta_1 f(t_{n+1}, y_{n+1}) + \cdots + \beta_k f(t_{n+k}, y_{n+k})\right),$$

where $\alpha_0, \ldots, \alpha_k$ a $\beta_0, \ldots, \beta_k$ are real constants, $\alpha_k \neq 0$, $\alpha_0^2 + \beta_0^2 > 0$.

Let us denote polynomials

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j = \alpha_0 + \alpha_1 z + \cdots + \alpha_k z^k$$ 1. characteristic polynomial

$$\sigma(z) = \sum_{j=0}^{k} \beta_j z^j = \beta_0 + \beta_1 z + \cdots + \beta_k z^k$$ 2. characteristic polynomial

Theorem The condition of stability

A linear multistep method is numerically stable for any differential equation $y' = f(t, y)$, where f is a Lipschitz function,

if and only if

the roots of the first characteristic polynomial $\rho(z)$ lie inside a closed unit circle, whereby the roots lying on the unit circle are simple.

Remark Function f is Lipschitz on the domain $J \times D \quad \Longleftrightarrow$

$$\exists\, L > 0 \;:\quad |f(t,y) - f(t,z)| \leq L|y - z| \quad \forall (t,y),(t,z) \in J \times D\,.$$

Remark We haven't study the error of the approximation, i.e. the accuracy of $k-$step methods.

Examples

1. Adams–Bashforth method

$$y_{n+4} = y_{n+3} + \frac{1}{24} h\,(55 f_{n+3} - 59 f_{n+2} + 37 f_{n+1} - 9 f_n)$$

$$y_{n+4} - y_{n+3} = \frac{1}{24} h\,(55 f_{n+3} - 59 f_{n+2} + 37 f_{n+1} - 9 f_n)$$

For the first characteristic polynomial $\rho(z)$ we have

$$\rho(z) = z^4 - z^3 = z^3(z-1) = 0 \quad \Longrightarrow$$

$z = 0$ is a triple zero inside the unit circle
$z = 1$ lies on the unit circle, single root
$\Longrightarrow$ the method is numerically stable .

2. Three steps method of 6th order

$$11 y_{n+3} + 27 y_{n+2} - 27 y_{n+1} - 11 y_n = 3h\,(f_{n+3} + 9 f_{n+2} + 9 f_{n+1} + f_n)$$

$$\rho(z) = 11 z^3 + 27 z^2 - 27 z - 11 = 0 \quad \text{(reciprocal equation)}\,.$$

Zeros: $z_1 = 1,\ z_2 \doteq -0{,}3189,\ z_3 \doteq -3{,}1356 \quad \Longrightarrow \quad |z_3| > 1 \Longrightarrow$
this method is not numerically stable .

★

3. Determine all values $b \in \mathbb{R}$, for which is the linear $k-$steps method

$$y_{n+3} + (2b-3)(y_{n+2} - y_{n+1}) - y_n = hb(f_{n+2} + f_{n+1})$$

numerically stable.

Solution

$$\rho(z) = z^3 + (2b-3)(z^2 - z) - 1 = 0\,.$$

Because $\rho(1) = 0$ then $z = 1$ is a single zero of $\rho(z)$,

$$z^3 + (2b-3)(z^2 - z) - 1 = (z-1) \cdot \underbrace{(z^2 + z + 1 + z(2b-3)}_{:= \rho_1(z) = z^2 + z(2b-2) + 1} = 0$$

Now we are looking for zeros of ρ_1. Let us try $\rho_1(1) = 2b \implies b \neq 0$, otherwise $z = 1$ wold not be a simple zero and the method would not be stable. From the same reason, because $\rho(-1) = -2b + 4$ then $b \neq 2$.

★

So, where are the zeros of ρ_1? We denote them z_1, z_2. Then

$$(z-z_1)(z-z_2) = z^2 + z(2b-2)+1 \implies -(z_1+z_2)z + z_1 z_2 = z(2b-2)+1\,,$$

i.e., $z_1 z_2 = 1$. But $z_1 \neq \pm 1$, $z_2 \neq \pm 1$, i.e., both z_1, z_2 are imaginary.

$$D = 4(b-1)^2 - 4 < 0 \quad \Longleftrightarrow \quad b \in (0,2)\,.$$

Conclusion

If $b \in (0,2)$ then the zeros of $\rho(z)$ are

$$z_1 = 1,\ z_{2,3} = 1 - b \pm \mathbf{i}\sqrt{1-(b-1)^2}\,,\ z_2 \neq z_3,\ |z_{2,3}| < 1\,,$$

i.e. all zeros of $\rho(z)$ lie for $b \in (0,2)$ in the closed unit circle $\implies$

the method is numerically stable $\Longleftrightarrow$ $b \in (0,2)$.

★

Theorem A necessary condition (but not sufficient) for convergence of a multistep method is the numerical stability of the method.

The linear $k-$steps method with the characteristic polynomial $\rho(z) = z^k - z^{k-1} \dots$ are so called Adams methods

– explicit ... Adams–Bashforth methods
– implicit ... Adams–Moulton methods

The linear $k-$steps method with the characteristic polynomial $\rho(z) = z^k - z^{k-2}$

– explicit ... Nyström method
– implicit ... Milne–Simpson method .

Remark It is possible to study also so called absolute stability (A–stability) of linear multistep methods. We will not study it here.

"Stiff"systems

"Stiff"equations are differential equations for which the numerical method is numerically unstable, if the step is not extremely small.
In the equation there are terms that cause quick change of the solution.
These equations are for example of the type

$$y' = ky + f(t), \quad \text{where} \quad k \in \mathbb{C}, \ |k| \text{ large}$$

or systems

$$\mathbf{y}' = \mathbf{K}\mathbf{y} + \mathbf{f}(t)\,,$$

where **K** has one of the eigenvalues $\lambda \in \mathbb{C}$ such that $|\lambda|$ is large in comparison with $\mathbf{f}(t)$ or $\Re\lambda_i < 0,\ 1 \le i \le n$, but

$$\max_{1\le i\le n} |\Re\lambda_i| >> \min_{1\le i\le n} |\Re\lambda_i|\,.$$

As a measure of the stiffness of the given system the folowing number R may serve:

$$R = \frac{\max|\Re\lambda_i|}{\min|\Re\lambda_i|}\,, \qquad \lambda_i \text{ is the eigenvalue of the Jacobi matrix of the given system.}$$

So far no generally accepted definition of the "stiffness"exists.

★

Stiff equations generally can be predicted from the physical problem from which the equation is derived and, with care, the error can be kept under control.

The system of initial-value problems

$$\begin{aligned} u_1' &= 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \quad u_1(0) = \frac{4}{3}, \\ u_2' &= -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t, \quad u_2(0) = \frac{2}{3}, \end{aligned}$$

has the unique solution

$$u_1(t) = 2\mathrm{e}^{-3t} - \mathrm{e}^{-39t} + \frac{1}{3}\cos t, \quad u_2(t) = -\mathrm{e}^{-3t} + 2\mathrm{e}^{-39t} - \frac{1}{3}\cos t.$$

The transient term e^{-39t} in the solution causes this system to be stiff. Apply Runge-Kutta fourth-order method for systems with the stepsize $h = 0.005$ and with $h = 0.1$ and compare results with the values of the exact solution.

Predictor-corrector method

The combination of an explicit method to predict and an implicit to improve the prediction is called a predictor-corrector method.
Let AB be the explicit $k-$steps Adams–Bashforth method of the 2nd order, and AM be the implicit $k-$steps Adams–Moulton method of the 2nd order.

The idea:
Predictor – in our case explicit AB method . We consider its result as an intermediate result

$$\tilde{y}_{n+2} = y_{n+1} + \frac{h}{2}\left(3f(t_{n+1}, y_{n+1}) - f(t_n, y_n)\right) .$$

Now, we "correct" this approximation by making use of the implicit AM method where we insert the intermediate result $\tilde{y}_{n+2}$ to the right hand side. We obtain

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

Recommended literature

- Burden R. L., Faires J. D.: Numerical Analysis (ninth edition). Brooks/Cole Cengage Learning, 2011, ISBN-13: 978-0-538-73351-9
- Gear C. W.: Numerical Initial Value Problems in Ordinary Differential Equations (Automatic Computation). Prentice Hall (September 1971)
- Kubíček M., Dubcová M., Janovská D.: Numerical Methods and Algorithms, http://old.vscht.cz/mat/Ang/NM-Ang/NM-Ang.pdf
- Rasmuson A., Andersson B., Olsson L., Andersson R.: Mathematical Modeling in Chemical Engineering. Cambridge University Press, 2014.
- Recktenwald G.: Numerical Integration of Ordinary Differential Equations for Initial Value Problems. Portland State University 2007, `gerry@me.pdx.edu`.