

1. Machine computing
2. Nonlinear equations & optimization
3. Interpolation
- 4. Numerical integration**
5. Ordinary differential equations
6. Systems of linear equations
7. Least squares problems
8. Numerical approximation of eigenvalues
9. Literature

Goal:

$$f : [a, b] \rightarrow \mathbb{R}, \quad I(f) = \int_a^b f(x) \, dx = ?$$

Approach:

1. segmentation

$$a = x_0 < x_1 < \dots < x_{n-1} < x_n = b,$$

2. approximation of $f(x)$ on $[x_{k-1}, x_k]$ by a function (polynomial $p_k(x)$) which is easy to integrate and use

$$\begin{aligned} \int_a^b f(x) \, dx &= \sum_{k=1}^n \int_{x_{k-1}}^{x_k} f(x) \, dx \\ &\approx \sum_{k=1}^n \int_{x_{k-1}}^{x_k} p_k(x) \, dx. \end{aligned}$$

Definition 4.1

In general an approximation formula of the form

$$Q_m(f) = \sum_{k=0}^m a_k f(x_k) \approx \int_a^b f(x) \, dx$$

is called an quadrature rule.

Remark 1

A quadrature rule is a weighted sum of the function values, as

$$\text{for 1 segment: } Q_m(f) = \int_a^b p_m(x) \, dx = \sum_{k=0}^m f(x_k) \underbrace{\int_a^b l_k(x) \, dx}_{q_k \in \mathbb{R}, \text{ weights}}.$$

Definition 4.2

The summation over all segments results in the composite rectangle rule

$$R_n = h(f_0 + f_1 + \dots + f_{n-1}),$$

with $f_i = f(t_i)$, the composite midpoint rule

$$M_n = h(f(a + 0.5h) + f(t_1 + 0.5h) + \dots + f(b - 0.5h)),$$

and the composite trapezium rule

$$T_n = h(0.5f_0 + f_1 + \dots + f_{n-1} + 0.5f_n).$$

4. Numerical integration

4.1 Standard integration problem

4.2 Newton-Cotes formulas

4.3 Composite Newton-Cotes rules

4.4 Romberg's method

4.5 Gaussian quadrature - optimal choice of nodes

4.6 Two-dimensional integrals - cubature rules

Standard integration problem

Standardization:

general domain $x \in [a, b] \rightarrow t \in [-1, 1]$.

Definition 4.3

In the following we consider the standard integration problem

$$\int_{-1}^1 f(t) \, dt.$$

The transformation to a general interval is

$$[-1, 1] \rightarrow [a, b], \quad \varphi(t) = \frac{b+a}{2} + t \frac{b-a}{2}.$$

In

$$Q_n(f) = \sum_{k=0}^n a_k f(t_k)$$

we call $t_k \in [-1, 1]$ the nodes and a_k the weights.

4. Numerical integration

4.1 Standard integration problem

4.2 Newton-Cotes formulas

4.3 Composite Newton-Cotes rules

4.4 Romberg's method

4.5 Gaussian quadrature - optimal choice of nodes

4.6 Two-dimensional integrals - cubature rules

Idea:

- generalization of the composite integration rules,
- divide the interval $[-1, 1]$ into m equal parts by

$$t_k = -1 + kh, \quad k = 0, \dots, m, \quad \text{with} \quad h = \frac{2}{m},$$

- take all $m + 1$ nodes and replace the function $f(t)$ by its interpolation polynomial $p_m(t)$ of degree $\leq m$,
- for simplification we focus not on subintervals but on the whole $[-1, 1]$ and derive formulas

$$p_m(t) = \sum_{i=0}^m f(t_i) l_i(t), \quad l_i(t) = \prod_{j=0, j \neq i}^m \frac{t - t_j}{t_i - t_j}.$$

Definition 4.4

The degree of exactness of an quadrature rule $Q_m(f)$ is the maximal integer $q \geq 0$, for which holds

$$I(p(t)) = Q_m(p(t))$$

for all polynomials $p(t)$ of degree q .

Theorem 4.5

Let $t_0, \dots, t_m \in [-1, 1]$ be $m + 1$ pairwise different nodes. The quadrature rule

$$I(f) = \int_{-1}^1 g(t) \, dt \rightarrow Q_m(f) = \sum_{i=0}^m a_i g(t_i)$$

is exact for all polynomials of degree $\leq m \Leftrightarrow$ the weights are given by

$$a_i = \int_{-1}^1 \prod_{j=0, j \neq i}^m \frac{t - t_j}{t_i - t_j} \, dt.$$

(Integral of the Lagrange basis functions.)

Computation of the weights

Conditions:

- quadrature rule

$$\int_{-1}^1 g(t) \, dt \rightarrow \sum_{i=0}^m a_i g(t_i),$$

- exactness for all polynomials of degree $\leq m \Leftrightarrow$ it is exact for $t^0, t^1, t^2, \dots, t^m$,
- thus we demand

$$\int_{-1}^1 t^j \, dt = \sum_{i=0}^m a_i (t_i)^j, \quad j = 0, \dots, m,$$

- the weights a_i satisfy the linear system of equations

$$\sum_{i=0}^m t_i^j a_i = \begin{cases} \frac{2}{j+1} & \text{for even } j \\ 0 & \text{for odd } j \end{cases}$$

for all $j = 0, \dots, m$.

Computation of the weights

Example:

- set up a formula of second order using the nodes $t_0 = -1$, $t_1 = 0$ and $t_2 = 1$,
- formula has the form

$$\int_{-1}^1 g(x) \, dx \rightarrow a_0 g(t_0) + a_1 g(t_1) + a_2 g(t_2)$$

- system of linear equations

$$\begin{array}{lllll} g(t) = t^0 = 1 : & a_0 & + a_1 & + a_2 & = 2 \\ g(t) = t^1 : & -a_0 & & + a_2 & = 0 \\ g(t) = t^2 : & a_0 & & + a_2 & = \frac{2}{3} \end{array}$$

- solution is $a_0 = \frac{1}{3}$, $a_1 = \frac{4}{3}$, $a_2 = \frac{1}{3}$, (Simpson's rule)

$$\int_{-1}^1 g(t) \, dt \approx \frac{1}{3} g(-1) + \frac{4}{3} g(0) + \frac{1}{3} g(1).$$

Simpson's rule for an arbitrary interval:

$$\int_a^b f(x) \, dx \approx \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right).$$

Remark 2

In general the degree of exactness of Newton-Cotes formulas using $m + 1$ nodes is

m for odd m ,

$m + 1$ for even m .

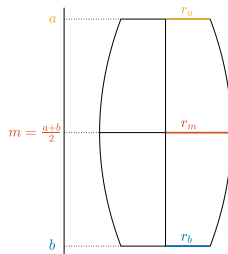
Application from 1615:

- already Kepler used Simpson's method without knowing it,
- computation of the volume of a barrel, thus approximation of

$$V = \pi \int_a^b (f(x))^2 dx,$$

- Kepler used

$$V \approx \pi \frac{h}{6} (r_a^2 + 4r_m^2 + r_b^2).$$



Weights:

- for the standard integration problem on $[-1, 1]$,
- using one single subinterval and the nodes

$$t_k = -1 + k \frac{2}{m}, \quad k = 0, \dots, m.$$

m	weights (a_k)	degree of exactness	error	name
0	2	0	$\frac{h}{2}f'(\xi)$	rectangle rule
1	1, 1	1	$\frac{h^2}{12}f''(\xi)$	trapezium rule
2	$\frac{1}{3}, \frac{4}{3}, \frac{1}{3}$	3	$\frac{h^4}{90}f^{(4)}(\xi)$	Simpson's rule
3	$\frac{1}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{4}$	3	$\frac{3h^4}{80}f^{(4)}(\xi)$	3/8 rule
4	$\frac{7}{45}, \frac{32}{45}, \frac{12}{45}, \frac{12}{45}, \frac{32}{45}, \frac{7}{45}$	5	$\frac{8h^6}{945}f^{(6)}(\xi)$	Milne's rule
$\vdots$	$\vdots$			

Remarks:

1. Don't use Newton-Cotes formulas of higher order, because some weights will become negative, which will cause cancellation of significant digits.
2. In general, the sequence

$$Q_m(f), \quad m = 1, 2, 3, \dots$$

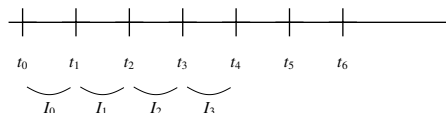
does not converge to $\int_a^b f(x) \, dx$.

4. Numerical integration

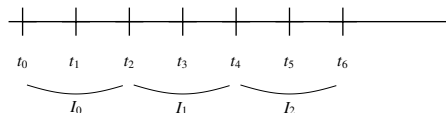
- 4.1 Standard integration problem
- 4.2 Newton-Cotes formulas
- 4.3 Composite Newton-Cotes rules**
- 4.4 Romberg's method
- 4.5 Gaussian quadrature - optimal choice of nodes
- 4.6 Two-dimensional integrals - cubature rules

Composite rule:

- use a discretization of the complete interval,
- apply the previous quadrature rules for the single segments
- $m = 0$: constant polynomial, rectangle rule,
- $m = 1$: polynomial of degree 1, trapezoid rule,



- $m = 2$: polynomial of degree 2, Simpson's rule,



Examples for the standard interval $[-1, 1]$:

1. composite rectangle rule

$$h(f(-1) + f(-1 + h) + f(-1 + 2h) + \dots + f(1 - h))$$

2. composite trapezion rule

$$\frac{h}{2}(f(-1) + 2f(-1 + h) + 2f(-1 + 2h) + \dots + 2f(1 - h) + f(1))$$

3. composite Simpson's rule for an even n

$$\frac{h}{6}(f(-1) + 4f(-1 + h) + 2f(-1 + 2h) + 4f(-1 + 3h) + \dots + 4f(1 - h) + f(1)).$$

Composite rules for a arbitrary interval

Procedure for an arbitrary interval:

- select a number n of nodes $x_i = a + ih$ with stepsize $h = \frac{b-a}{n}$,
- compute the function values $f_i = f(x_i)$
- select an integration rule

The composite rules are:

$$R_n = h(f_0 + f_1 + f_2 + \dots + f_{n-1}),$$

$$T_n = \frac{h}{2}(f_0 + 2f_1 + 2f_2 + \dots + 2f_{n-1} + f_n),$$

$$S_n = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \dots + 2f_{n-2} + 4f_{n-1} + f_n).$$

Example:

- consider the approximation of

$$I = \frac{1}{\ln(2)} \int_0^{\pi/2} \frac{x}{1 + \sin(x)} dx = 1.$$

- using different composite rules for different discretizations

$$x_i = \frac{\pi}{2n} i, \quad i = 0, \dots, n.$$

n	R_n	M_n	T_n	S_n
5	0.8162	1.0029	0.9942	0.999975
10	0.9095	1.0007	0.9985	
50	0.9821	1.0000	0.9999	1.0
100	0.9911	1.0000	1.0000	1.0
200	0.9955	1.0000	1.0000	1.0

Error terms for an arbitrary interval:

$$\begin{aligned}\left| \int_a^b f(x) \, dx - R_n \right| &\leq \frac{(b-a)h}{2} \max_{x \in [a,b]} |f'(x)|, \\ \left| \int_a^b f(x) \, dx - T_n \right| &\leq \frac{(b-a)h^2}{12} \max_{x \in [a,b]} |f''(x)|, \\ \left| \int_a^b f(x) \, dx - S_n \right| &\leq \frac{(b-a)h^4}{180} \max_{x \in [a,b]} |f^{(4)}(x)|.\end{aligned}$$

Remark 3

1. *The rectangle rule is exact only for constant polynomials, the trapezoid rule is exact for polynomials of degree 1 and Simpson's rule is exact for polynomials of degree 3.*
2. *The integration errors can be used to determine a proper number of nodes n so that a given error level is not exceeded.*

4. Numerical integration

4.1 Standard integration problem

4.2 Newton-Cotes formulas

4.3 Composite Newton-Cotes rules

4.4 Romberg's method

4.5 Gaussian quadrature - optimal choice of nodes

4.6 Two-dimensional integrals - cubature rules

Idea:

- multiple calculation with a quadrature rule with halved step size each time,
- use of previously calculated values for composite trapezoidal rule,

$$I(h) = T_{2n} = \frac{1}{2}T_n + \frac{h}{2} \sum_{i=1}^n f(a + \frac{2i-1}{2}h)$$

- extrapolation to improve the approximation

$$\lim_{h \rightarrow 0} I(h) = \int_a^b f(x) \, dx.$$

Romberg's method

Procedure:

- basic step size $\bar{h} = (b - a)/m$,
- initialization for $i = 0, 1, 2, \dots$,

$$I_{i,0} = T_{2^i m} \quad (\text{comp. trapezoid rule for } h = \bar{h}/2^i),$$

- extrapolation to $h = 0$ using $I_{i,0}$ yields Romberg's scheme

$$I_{i,j} = \frac{4^j I_{i,j-1} - I_{i-1,j-1}}{4^j - 1}, \quad j = 1, \dots, i,$$

- scheme for basic step size $\bar{h}$

$$\begin{array}{ccccccc} \bar{h} : & I_{0,0} & & & & & \\ & \searrow & & & & & \\ \frac{\bar{h}}{2} : & I_{1,0} & \rightarrow & I_{1,1} & & & \\ & \searrow & & \searrow & & & \\ \frac{\bar{h}}{4} : & I_{2,0} & \rightarrow & I_{2,1} & \rightarrow & I_{2,2} & \\ & \searrow & & \searrow & & \searrow & \\ \frac{\bar{h}}{8} : & I_{3,0} & \rightarrow & I_{3,1} & \rightarrow & I_{3,2} & \rightarrow I_{3,3}. \end{array}$$

Romberg's method

Example:

- approximation of

$$\int_1^{10} \ln(x) \, dx,$$

- using trapezoid rule and Romberg's scheme for extrapolation,
- run the iteration until

$$|T_{n,n-1} - T_{n,n}| \leq \varepsilon,$$

- the scheme (using 5 digits) reads

10.362							
12.852	13.682						
13.685	13.963	13.982					
13.934	14.017	14.021	14.022				
14.002	14.025	14.026	14.026	14.026			
14.020	14.026	14.026	14.026	14.026	14.026		
14.024	14.026	14.026	14.026	14.026	14.026	14.026	
14.025	14.026	14.026	14.026	14.026	14.026	14.026	14.026

- error for $\varepsilon = 10^{-5}$ is $6.3753 \cdot 10^{-6}$,
- error for $\varepsilon = 10^{-10}$ is $-3.8345 \cdot 10^{-10}$.

4. Numerical integration

- 4.1 Standard integration problem
- 4.2 Newton-Cotes formulas
- 4.3 Composite Newton-Cotes rules
- 4.4 Romberg's method
- 4.5 Gaussian quadrature - optimal choice of nodes**
- 4.6 Two-dimensional integrals - cubature rules

Example:

- consider a quadrature rule for $[-1, 1]$ with two nodes $Q(f) = a_0f(t_0) + a_1f(t_1)$,
- try to reach maximal degree of exactness (3 since 4 degrees of freedom), thus

$$Q(t^i) = I(t^i), \quad i = 0, \dots, 3,$$

- equations

$$Q(1) = a_0 + a_1 = 2,$$

$$Q(t) = a_0t_0 + a_1t_1 = 0,$$

$$Q(t^2) = a_0t_0^2 + a_1t_1^2 = \frac{2}{3},$$

$$Q(t^3) = a_0t_0^3 + a_1t_1^3 = 0,$$

- solution of the nonlinear system of equations is

$$t_{0,1} = \pm\sqrt{1/3}, \quad a_{0,1} = 1,$$

- quadrature rule (Gaussian quadrature for two nodes) reads

$$Q_1(f) = f(-\sqrt{1/3}) + f(\sqrt{1/3}).$$

Gaussian quadrature

General approach:

$$Q_m(f) = \sum_{i=0}^m a_i f(t_i) \quad (4)$$

- $2m + 2$ degrees of freedom, thus exactness for x^i for $i = 0, \dots, 2m + 1$.

Definition 4.6

With the initialization $p_0(x) = 1$ and $p_1(x) = x$ the Legendre polynomials are defined as

$$(n + 1)p_{n+1}(x) = (2n + 1)xp_n(x) - np_{n-1}(x).$$

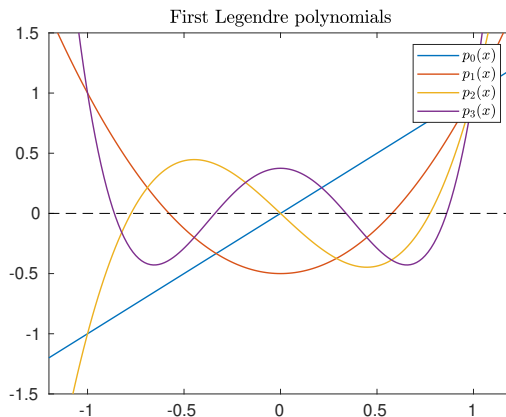
Examples:

$$p_2(x) = \frac{1}{2}(3x^2 - 1),$$

$$p_3(x) = \frac{1}{3}\left(\frac{5}{2}x(3x^2 - 1) - 2x\right) = \frac{1}{2}(5x^2 - 3)x,$$

$$p_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3).$$

Legendre polynomials for $[-1, 1]$



Theorem 4.7

1. *The nodes of the quadrature rule (4) are the zeros of the Legendre polynomial of degree $m + 1$.*
2. *All zeros are simple and in $[-1, 1]$.*
3. *The degrees of exactness of the Gaussian quadrature rules are $2m + 1$.*

Gaussian rule for $m = 2$ for $[-1, 1]$

$$Q_2(f) = \frac{1}{9} \left(5f\left(-\sqrt{\frac{3}{5}}\right) + 8f(0) + 5f\left(\sqrt{\frac{3}{5}}\right) \right).$$

Nodes and weights for $m = 3$

$$t_{0,3} = \pm 0.861136 \dots,$$

$$t_{1,2} = \pm 0.339981 \dots,$$

$$a_{0,3} = \pm 0.347854 \dots,$$

$$a_{1,2} = \pm 0.652145 \dots$$

Arbitrary integration domain:

- for the complete domain with $h = b - a$,
- degree of exactness 3

$$Q_1(f) = \frac{h}{2} \left(f\left(\frac{a+b}{2} - \sqrt{\frac{1}{3}} \frac{b-a}{2}\right) + f\left(\frac{a+b}{2} + \sqrt{\frac{1}{3}} \frac{b-a}{2}\right) \right),$$

- degree of exactness 5

$$Q_2(f) = \frac{h}{2} \left(\frac{5}{9} f\left(\frac{a+b}{2} - \sqrt{\frac{3}{5}} \frac{b-a}{2}\right) + \frac{8}{9} f\left(\frac{a+b}{2}\right) + \frac{5}{9} f\left(\frac{a+b}{2} + \sqrt{\frac{3}{5}} \frac{b-a}{2}\right) \right).$$

Example:

- consider (with the true value $I = 1$)

$$I = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-x^2) dx,$$

- use the coordinate transformation

$$x = \tan(t), \quad \frac{dx}{dt} = \frac{1}{\cos(t)^2} \quad dx = \frac{1}{\cos(t)^2} dt,$$

- approximation of

$$\frac{1}{\sqrt{\pi}} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \exp(-\tan(t)^2) \frac{1}{\cos(t)^2} dt,$$

- using composite Gaussian quadrature rules with $n \in \{5, 10\}$ subintervals

n	$Q_2(f)$	$Q_3(f)$
4	1.0289	0.99392
10	1.00088	0.99995

4. Numerical integration

- 4.1 Standard integration problem
- 4.2 Newton-Cotes formulas
- 4.3 Composite Newton-Cotes rules
- 4.4 Romberg's method
- 4.5 Gaussian quadrature - optimal choice of nodes
- 4.6 Two-dimensional integrals - cubature rules

Problem:

$$u : S \rightarrow \mathbb{R}, S \subset \mathbb{R}^2, \quad \iint_S u(x, y) \, d\sigma = ?$$

Applications:

- finite elements method to solve elliptic pde's numerically.

Different situations:

- quadrangles or triangles with (4 resp. 2) edges being parallel to the axes
→ simple
- arbitrary triangle/quadrangle
→ more complex, use a coordinate transformation and cubature rules,
- general domain S (curved boundary)
→ most complex.

Definition 4.8

The triangle with the vertices $(0, 0)$, $(1, 0)$ and $(0, 1)$ is called reference triangle T_0 .

Approximation of

$$I = \iint_{T_0} f(x, y) \, dy \, dx = \int_0^1 \int_0^{1-x} f(x, y) \, dy \, dx.$$

Example

1. The following cubature rule is exact for all polynomials up to degree 1

$$\frac{1}{6} (f(0, 0) + f(1, 0) + f(0, 1)) \approx I.$$

2. The following cubature rule is exact for all polynomials up to degree 2

$$\frac{1}{6} (f(0.5, 0) + f(0, 0.5) + f(0.5, 0.5)) \approx I.$$

1. Machine computing
2. Nonlinear equations & optimization
3. Interpolation
4. Numerical integration
- 5. Ordinary differential equations**
6. Systems of linear equations
7. Least squares problems
8. Numerical approximation of eigenvalues
9. Literature

Ordinary differential equations

Definition 5.1

An ordinary differential equation of first order (ode) has the form

$$y'(x) = f(x, y(x)), \quad a \leq x \leq b,$$

with $f : [a, b] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ (this includes also systems with $n > 1$). Typically f is called the right-hand side of the ode.

Definition 5.2

An initial value problem or Cauchy problem consists of an ordinary differential equation of first order and an initial value

$$\begin{aligned} y'(x) &= f(x, y(x)), & a \leq x \leq b, \\ y(a) &= y_0 \end{aligned}$$

with $f : [a, b] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Wanted: $y : [a, b] \rightarrow \mathbb{R}^n$.

5. Ordinary differential equations

5.1 Numerical solution & Euler's method

5.2 Runge-Kutta methods

5.3 Local truncation error and order of Runge-Kutta methods

5.4 Adaptive Runge-Kutta methods and step size control

5.5 Linear multistep methods - BDF methods

5.6 Stiff problems and A -stability

5.7 Implicit Runge-Kutta methods

5.8 Boundary value problem of second order

General procedure:

- discretisation of the interval $[a, b]$

$$a = x_0 < x_1 < x_2 < \dots < x_N = b,$$

- step sizes $h_i = x_{i+1} - x_i$,
- approximations $Y_i \approx y(x_i)$,
- initialization and iteration

$$Y_0 = y(a) = y_0,$$

$$Y_{i+1} = Y_i + h_i \Phi(x_i, Y_i, h_i), \quad i = 0, \dots, N-1.$$

Euler's method (only to explain the basic idea):

$$Y_{i+1} = Y_i + hf(x_i, Y_i), \quad i = 0, 1, 2, \dots$$

Crank-Nicolson (Trapezium rule):

$$Y_{i+1} = Y_i + \frac{h_i}{2} (f(x_i, Y_i) + f(x_{i+1}, Y_{i+1})).$$

Midpoint-rule (Gaussian integration):

$$Y_{i+1} = Y_i + hf(x_i + h_i/2, (Y_i + Y_{i+1})/2)).$$

Euler-Heun method:

$$Y_{i+1} = Y_i + \frac{h_i}{2} (f(x_i, Y_i) + f(x_i + h, Y_i + hf(x_i, Y_i))).$$

5. Ordinary differential equations

5.1 Numerical solution & Euler's method

5.2 Runge-Kutta methods

5.3 Local truncation error and order of Runge-Kutta methods

5.4 Adaptive Runge-Kutta methods and step size control

5.5 Linear multistep methods - BDF methods

5.6 Stiff problems and A -stability

5.7 Implicit Runge-Kutta methods

5.8 Boundary value problem of second order

Idea:

- simple iterations

$$Y_{i+1} = Y_i + h_i \Phi(x_i, Y_i, h_i)$$

without using further iterations directly,

- using s stages between x_i and x_{i+1} for high accuracy,
- evaluation of K_j using results of previous stages $K_1, \dots, K_{j-1}$,
- linear combination of evaluations to define the next iterate,
- simple analysis of the error and the order.

Definition 5.3 (Kutta 1901)

Let s be an integer (the number of stages) and $a_{ij}, b_i, c_l \in \mathbb{R}$. Then the method

$$K_1 = f(x_\ell, y_\ell),$$

$$K_2 = f(x_\ell + c_2h, y_\ell + ha_{21}K_1),$$

$$K_3 = f(x_\ell + c_3h, y_\ell + h(a_{31}K_1 + a_{32}K_2)),$$

$$\vdots$$

$$K_s = f(x_\ell + c_sh, y_\ell + h(a_{s1}K_1 + \dots + a_{s,s-1}K_{s-1}))$$

with

$$Y_{\ell+1} = Y_\ell + h(b_1K_1 + \dots + b_sK_s)$$

is called an s -stage explicit Runge-Kutta method. Further, $\Phi(x_\ell, Y_\ell, h)$ with

$$Y_{\ell+1} = Y_\ell + h\Phi(x_\ell, Y_\ell, h)$$

is called process function.

Runge-Kutta methods

Butcher tableau of a Runge-Kutta method:

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

Butcher-tableau for Euler's method

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

Butcher-tableau for Euler-Heun

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

The Runge-Kutta methods

Definition 5.4

The Butcher-tableau

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	1/6	1/3	1/3	1/6

with

$$K_1 = f(x_\ell, Y_\ell),$$

$$K_2 = f(x_\ell + \frac{1}{2}h, Y_\ell + \frac{h}{2}K_1),$$

$$K_3 = f(x_\ell + \frac{1}{2}h, Y_\ell + \frac{h}{2}K_2),$$

$$K_4 = f(x_\ell + h, Y_\ell + hK_3),$$

$$Y_{\ell+1} = Y_\ell + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

is called the classical Runge-Kutta method or RK 4.

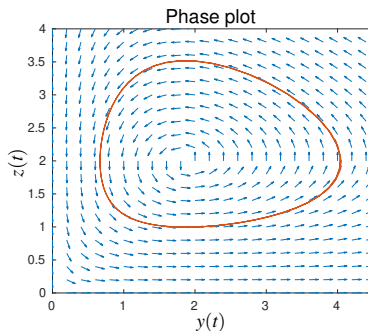
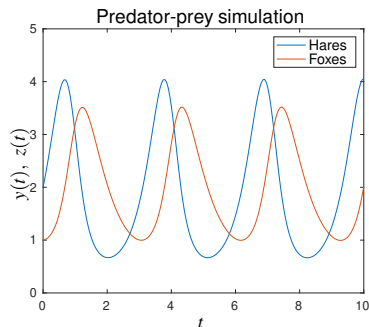
Predator-prey simulation

Consider the Predator-prey equation

$$\dot{y}(t) = ay(t) - by(t)z(t),$$

$$\dot{z}(t) = cy(t)z(t) - dz(t).$$

Simulation for $t \in [0, 10]$ as well as the initial values $y(0) = 2$, $z(0) = 1$ and the parameters $a = 3$, $b = 1.5$, $c = 0.8$ and $d = 1.5$.



5. Ordinary differential equations

5.1 Numerical solution & Euler's method

5.2 Runge-Kutta methods

5.3 Local truncation error and order of Runge-Kutta methods

5.4 Adaptive Runge-Kutta methods and step size control

5.5 Linear multistep methods - BDF methods

5.6 Stiff problems and A -stability

5.7 Implicit Runge-Kutta methods

5.8 Boundary value problem of second order

Evaluation of the local error in a single step.

Definition 5.5

Let $\Phi(x, y, h)$ be a *process function*. We call the term

$$\tau(x, h) = \frac{y(x+h) - y(x)}{h} - \Phi(x, y, h)$$

local truncation error of the Runge-Kutta method. If

$$\|\tau(x, h)\| = \mathcal{O}(h^p) \quad \text{for } h \rightarrow 0,$$

then we call p the order of the Runge-Kutta method.

Local truncation error for Euler's method ($\Phi(x, y, h) = f(x, y)$):

$$\begin{aligned}\tau(x, h) &= \frac{y(x+h) - y(x)}{h} - \Phi(x, y, h) \\ &= \frac{y(x) + hy'(x) + \mathcal{O}(h^2) - y(x)}{h} - f(x, y) \\ &= y'(x) + \mathcal{O}(h) - \underbrace{f(x, y)}_{y'(x)} = \mathcal{O}(h).\end{aligned}$$

→ order $p = 1$.

Taylor-series 1:

- the term

$$\frac{y(x+h) - y(x)}{h} = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \dots$$

is always the necessary,

- therefore we use

$$y'(x) = f(x, y),$$

$$y''(x) = (y'(x))' = \frac{df(x, y(x))}{dx} = f_x + f_y y' = f_x + f_y f,$$

$$y'''(x) = f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y f_x + f_y^2 f,$$

- so the transformed term using a Taylor-series reads

$$\frac{y(x+h) - y(x)}{h} = f + \frac{h}{2}(f_x + f_y f) + \frac{h^2}{6}(f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y f_x + f_y^2 f) + \dots,$$

- the other Taylor-series depends on the method.

For Euler-Heun:

- the iteration is

$$Y_{i+1} = Y_i + \frac{h_i}{2} (f(x_i, Y_i) + f(x_i + h, Y_i + hf(x_i, Y_i))),$$

- Taylor-series no. 1 from the last slide

$$\frac{y(x+h) - y(x)}{h} = f + \frac{h}{2}(f_x + f_y f) + \dots,$$

- Taylor-series no. 2

$$\Phi(x, y, h) = \frac{1}{2} (f(x, y) + f(x + h, y + hf(x, y)) + \dots) = \frac{1}{2} (f + f + hf_x + f_y hf + \dots),$$

- local truncation error

$$\begin{aligned}\tau(x, h) &= \frac{y(x+h) - y(x)}{h} - \Phi(x, y, h) \\ &= (f + \frac{h}{2}(f_x + f_y f')) - \frac{1}{2}(f + f + hf_x + f_y hf) + \mathcal{O}(h^2) = \mathcal{O}(h^2),\end{aligned}$$

- thus the method is of order 2 (at least).

Order of Runge-Kutta methods

Some orders of Runge-Kutta methods:

- Euler's method has order 1,
- Euler-Heun has order 2,
- Crank-Nicolson has order 2 (implicit),
- midpoint rule has order 2 (implicit),
- the Runge-Kutta method from definition 5.4 has order 4.

Remark 4

The order of an explicit Runge-Kutta method is bounded by the number of stages as follows

s	1	2	3	4	5	6	7	8	9	$s \geq 9$
q	1	2	3	4	4	5	6	6	7	$q \leq s - 2$

The number of equations that we have to respect depend on the order as follows

s	1	2	3	4	5	6	7	8
q	1	2	4	8	17	37	85	200

Order of Runge-Kutta methods

Example:

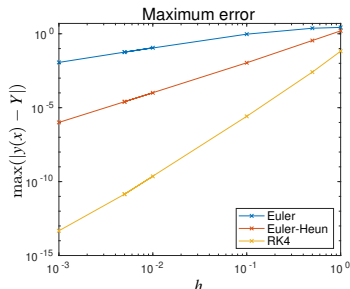
- consider the initial value problem

$$y'(x) = -\sin(x)y(x), \quad x \in [-10, 10],$$

$$y(-10) = e^{\cos(-10)}$$

- the analytical solution reads $y(x) = e^{\cos(x)}$,
- application of different numerical methods and different step sizes,
- calculation of the maximal error

$$\max_{x \in [-10, 10]} |y(x) - Y_h(x)|.$$



5. Ordinary differential equations

- 5.1 Numerical solution & Euler's method
- 5.2 Runge-Kutta methods
- 5.3 Local truncation error and order of Runge-Kutta methods
- 5.4 Adaptive Runge-Kutta methods and step size control**
- 5.5 Linear multistep methods - BDF methods
- 5.6 Stiff problems and A -stability
- 5.7 Implicit Runge-Kutta methods
- 5.8 Boundary value problem of second order

Accuracy and effort:

- accuracy of the approximations & required computation time depend on the method, the step size and the ode,
- small step sizes provide good approximations, but cause a high computational effort.

Adaptive Runge-Kutta methods:

- adjust the step size to the current behavior of the solution of the ode,
- estimate the suitable step size using a second method and compare the two approximations.

Procedure for a desired accuracy ε :

1. Compute approximations for the current iteration

$$Y = Y_i + h\Phi(x_i, Y_i, h_i),$$

$$\tilde{Y} = Y_i + h\tilde{\Phi}(x_i, Y_i, h_i)$$

using the process functions Φ and $\tilde{\Phi}$ with the orders p and $p + 1$,

2. Compute the value

$$\vartheta = \left(\frac{\varepsilon}{\|Y - \tilde{Y}\|} \right)^{\frac{1}{p+1}}.$$

3. Case differentiation

$\vartheta < 1$: Discard the step, set $h := \max(\vartheta, 0.5)h$, execute step 1 again.

$\vartheta \geq 1$: Accept the approximation and increase the step size for the next step

$$x_{i+1} = x_i + h, \quad Y_{i+1} = \tilde{Y}, \quad h := \min(2, \vartheta)h.$$

Embedded Runge-Kutta method DOPRI(4)5:

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

5. Ordinary differential equations

- 5.1 Numerical solution & Euler's method
- 5.2 Runge-Kutta methods
- 5.3 Local truncation error and order of Runge-Kutta methods
- 5.4 Adaptive Runge-Kutta methods and step size control
- 5.5 Linear multistep methods - BDF methods**
- 5.6 Stiff problems and A -stability
- 5.7 Implicit Runge-Kutta methods
- 5.8 Boundary value problem of second order

Idea of multistep methods:

- use previous iterates directly to compute the next approximation,
- iteration of the form

$$Y_{k+1} = Y_k + h_k \Phi(x_{k+1}, x_k, \dots, x_{k+1-m}, Y_{k+1}, Y_k, \dots, Y_{k+1-m}),$$

- m -step linear multistep method (with $a_0 \neq 0$)

$$\sum_{j=0}^m a_j Y_{k+1-j} = h \sum_{j=0(1)}^m b_j f(x_{k+1-j}, Y_{k+1-j}).$$

Remarks:

- more degrees of freedom allow (theoretically) a higher accuracy,
- stability of multistep methods is not for sure
→ use only well-known methods as Adams, Nyström or BDF.

BDF-formulas for equidistant h of order 1 to 5:

$$Y_{k+1} = Y_k + hf(x_{k+1}, Y_{k+1})$$

$$\frac{3}{2}Y_{k+1} = 2Y_k - \frac{1}{2}Y_{k-1} + hf(x_{k+1}, Y_{k+1})$$

$$\frac{11}{6}Y_{k+1} = 3Y_k - \frac{3}{2}Y_{k-1} + \frac{1}{3}Y_{k-2} + hf(x_{k+1}, Y_{k+1})$$

$$\frac{25}{12}Y_{k+1} = 4Y_k - 3Y_{k-1} + \frac{4}{3}Y_{k-2} - \frac{1}{4}Y_{k-3} + hf(x_{k+1}, Y_{k+1})$$

$$\frac{137}{60}Y_{k+1} = 5Y_k - 5Y_{k-1} + \frac{10}{3}Y_{k-2} - \frac{5}{4}Y_{k-3} + \frac{1}{5}Y_{k-4} + hf(x_{k+1}, Y_{k+1}).$$

Remarks:

- BDF formulas define implicit methods,
- up to $m = 6$ they are very effective for stiff problems.

Cyclic example:

- three-body problem,
- two bodies of masses $1 - \mu$ (Earth) and μ (Moon) move circularly in a plane,
- a third body (satellite), of negligible mass, also moves in the plane
- equations of motion for this body are

$$\ddot{y}_1 = y_1 + 2\dot{y}_2 - \mu' \frac{y_1 + \mu}{D_1} - \mu \frac{y_1 - \mu'}{D_2}$$

$$\ddot{y}_2 = y_2 - 2\dot{y}_1 - \mu' \frac{y_2}{D_1} - \mu \frac{y_2}{D_2}$$

$$D_1 = ((y_1 + \mu)^2 + y_2^2)^{3/2}, \quad D_2 = ((y_1 - \mu')^2 + y_2^2)^{3/2}$$

$$\mu = 0.012277472, \quad \mu' = 1 - \mu$$

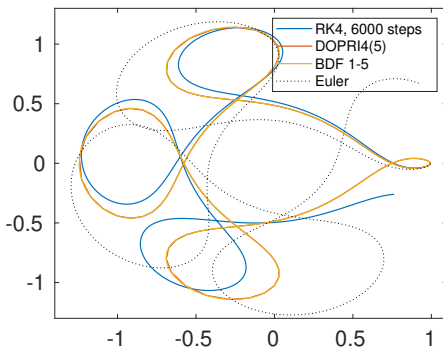
- interval $[t_0, t_1] = [0, 17.065216560157963]$, parameters

$$y_1(0) = 0.994, \quad \dot{y}_1(0) = y_2(0) = 0, \quad \dot{y}_2(0) = -2.0015851063790825.$$

Example

Applied methods:

- Euler's method using 100 000 steps,
- Runge-Kutta using 6 000 steps,
- DOPRI4(5) with step size control (finally uses 212 steps),
- BDF-1 and BDF-5 with step size control (ode15s in matlab, finally uses 468 steps).



5. Ordinary differential equations

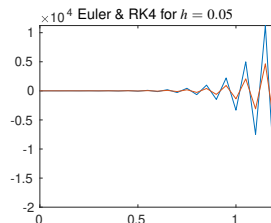
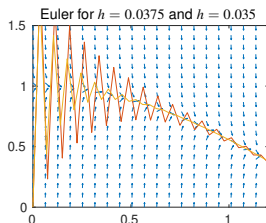
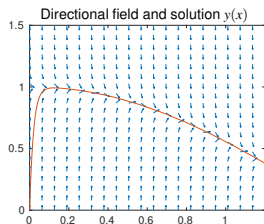
- 5.1 Numerical solution & Euler's method
- 5.2 Runge-Kutta methods
- 5.3 Local truncation error and order of Runge-Kutta methods
- 5.4 Adaptive Runge-Kutta methods and step size control
- 5.5 Linear multistep methods - BDF methods
- 5.6 Stiff problems and A -stability**
- 5.7 Implicit Runge-Kutta methods
- 5.8 Boundary value problem of second order

Curtiss, Hirschfelder 1952:

- initial value problem

$$y'(x) = -50(y(x) - \cos(x)), \quad y(0) = 0,$$

- solution by Euler's method and RK 4 using different step sizes.



~> Analysis of the behavior.

Model problem:

$$y' = \lambda y, \quad y(0) = y_0$$

with the solution

$$y(x) = y_0 e^{\lambda x}.$$

Boundedness:

$$\operatorname{Re}(\lambda) \leq 0 \quad \Rightarrow \quad |y(x)| = |y_0| \underbrace{\left(e^{(\operatorname{Re}(\lambda))x} (\cos(x \operatorname{Im}(\lambda)) + \mathbf{i} \sin(x \operatorname{Im}(\lambda))) \right)}_{\leq 1} \leq |y_0|.$$

Definition 5.6

A numerical method is called A-stable if its application to the model problem with $\operatorname{Re}(\lambda) \leq 0$ for any step size $h > 0$ always yields a sequence of iterates Y_k , $k = 1, 2, \dots$ which are bounded by $|y_0|$.

Remark

Any explicit s -step Kutta method applied to the model problem can be written in the form

$$Y_{i+1} = p(h\lambda)Y_i(*)$$

with a polynomial $p(\mu)$ of degree s .

Definition 5.7

The polynomial $p(\mu)$ from $(*)$ is called stability function.

Stability function for Euler's methods:

$$p(\mu) = 1 + \mu.$$

Definition 5.8

Let $p(\mu)$ be the stability function of the Runge-Kutta method. For constant step size h is

$$S = \{\mu \in \mathbb{C} \mid |p(\mu)| \leq 1\}$$

the stability domain of the Runge-Kutta method.

Conclusion:

1. For constant h the solutions remain bounded if and only if $h\lambda \in S$.
2. If the stability domain contains the left half plane

$$\{\mu \in \mathbb{C} \mid \operatorname{Re}(\mu) \leq 0\},$$

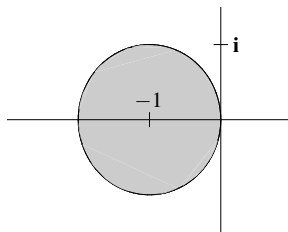
then the method is according to definition 5.6 A-stable.

Euler (explicit):

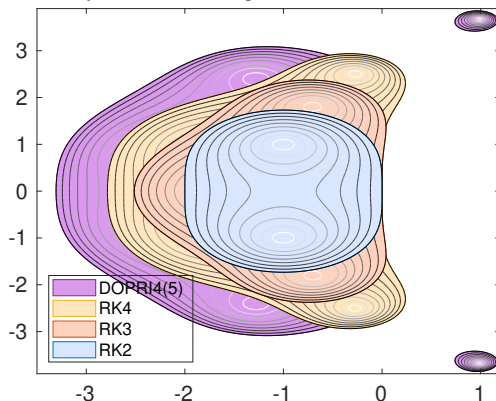
- the stability function is $p(\mu) = 1 + \mu$,
- the stability domain is

$$S = \{\mu \in \mathbb{C} \mid |p(\mu)| \leq 1\}$$

and thus a circle with radius 1 and center -1 .



Stability domains Runge Kutta 2-4 & DOPRI4(5)



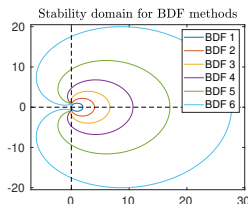
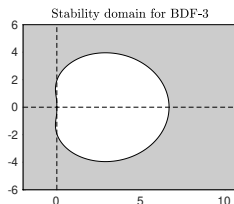
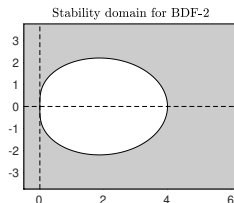
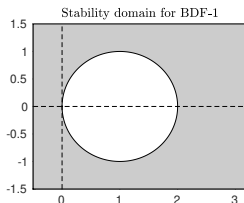
Theorem 5.9

No explicit Runge-Kutta procedure is A-stable.

Stability domains for BDF

Remarks:

- calculation of the stability domains for multistep methods is quite complex,
- only BDF-1 and BDF-2 are A -stable, but BDF-3 to BDF-6 have large and unbounded(!) stability domains, well suited for solving stiff problems.



5. Ordinary differential equations

- 5.1 Numerical solution & Euler's method
- 5.2 Runge-Kutta methods
- 5.3 Local truncation error and order of Runge-Kutta methods
- 5.4 Adaptive Runge-Kutta methods and step size control
- 5.5 Linear multistep methods - BDF methods
- 5.6 Stiff problems and A -stability
- 5.7 Implicit Runge-Kutta methods**
- 5.8 Boundary value problem of second order

Example (Implicit Euler method)

- *iteration*

$$Y_{i+1} = Y_i + hf(x_{i+1}, Y_{i+1}),$$

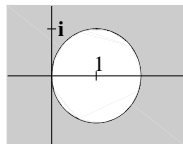
- *application for the model problem $y' = \lambda y$*

$$Y_{i+1} = Y_i + h\lambda Y_{i+1} \quad \Leftrightarrow \quad (1 - h\lambda)Y_{i+1} = Y_i \quad \Leftrightarrow \quad Y_{i+1} = \frac{1}{1 - h\lambda}Y_i,$$

- *boundedness of the iterates*

$$\left| \frac{1}{1 - h\lambda} \right| \leq 1 \quad \Leftrightarrow \quad 1 \leq |1 - h\lambda|,$$

- *satisfied with $h > 0$ as well as $\operatorname{Re}(\lambda) \leq 0 \Rightarrow A\text{-stable}$.*



Definition 5.10

Let $a_{jl}, b_j \in \mathbb{R}$ for $j, l = 1, \dots, s$ and $c_j = \sum_{l=1}^s a_{jl}$. Let the quantities $y_1, \dots, y_s$ be solutions of the (i. a. non-linear) system of equations

$$y_j = Y_i + h \sum_{l=1}^s a_{jl} f(x + c_l h, y_l)$$

for $j = 1, \dots, s$.

Then the new iterate using the s -step implicit Runge-Kutta method reads

$$Y_{i+1} = Y_i + h_i \sum_{j=1}^s b_j f(x + c_j h, y_j).$$

Butcher tableau:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

Remarks:

- Many, but not all, implicit Runge-Kutta methods are A -stable.
- This advantage comes with the price of solving a nonlinear system of equations.

Some A-stable methods:

1. The implicit Euler method has the Butcher-tableau

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}.$$

2. The Butcher-tableau

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

defines the Crank-Nicolson-Verfahren as

$$Y_{i+1} = Y_i + \frac{h}{2} (f(x_i, Y_i) + f(x_{i+1}, Y_{i+1})).$$

More A-stable methods:

3. The Radau-IA method with $s = 2$ stages has the Butcher-tableau

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

and is of order 3.

4. The Radau-IIA method with $s = 2$ stages has the Butcher-tableau

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

and is also of order 3.

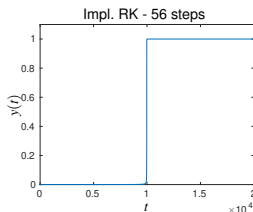
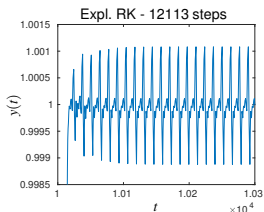
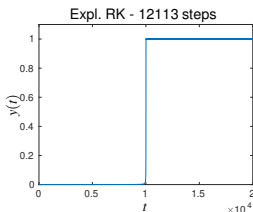
Example:

- simplified simulation of flame propagation in gases,
- initial value problem with $\varepsilon > 0$

$$y'(x) = y(x)^2 - y(x)^3, \quad 0 \leq x \leq \frac{2}{\varepsilon},$$

$$y(0) = \varepsilon.$$

- numerical results for $\varepsilon = 10^{-4}$



5. Ordinary differential equations

- 5.1 Numerical solution & Euler's method
- 5.2 Runge-Kutta methods
- 5.3 Local truncation error and order of Runge-Kutta methods
- 5.4 Adaptive Runge-Kutta methods and step size control
- 5.5 Linear multistep methods - BDF methods
- 5.6 Stiff problems and A -stability
- 5.7 Implicit Runge-Kutta methods
- 5.8 Boundary value problem of second order

Scalar linear boundary value problem of second order: (for $\lambda \geq 0$)

$$\begin{aligned} -y''(x) + \lambda y(x) &= f(x), & a \leq x \leq b, \\ y(a) &= \alpha, & y(b) = \beta \end{aligned}$$

Segmentation:

$$a = x_0 < x_1 < \dots < x_N < x_{N+1} = b, \quad x_j = a + jh, \quad h = \frac{b-a}{N+1}.$$

Boundary points: (x_0, α) and (x_{N+1}, β) .

Inner knots: $x_1, \dots, x_N$, to these knots we compute approximations $Y_i \approx y(x_i)$.

Finite differences:

$$-y''(x_i) \approx \frac{-Y_{i-1} + 2Y_i - Y_{i+1}}{h^2}.$$

Equations:

$$\begin{aligned} i = 1 : & \quad \frac{-\alpha + 2Y_1 - Y_2}{h^2} + \lambda Y_1 = f(x_1), \\ 2 \leq i \leq N - 1 : & \quad \frac{-Y_{i-1} + 2Y_i - Y_{i+1}}{h^2} + \lambda Y_i = f(x_i), \\ i = N : & \quad \frac{-Y_{N-1} + 2Y_N - \beta}{h^2} + \lambda Y_N = f(x_N). \end{aligned}$$

System of linear equations:

$$\frac{1}{h^2} \begin{pmatrix} 2 + \lambda h^2 & -1 & & & \\ -1 & 2 + \lambda h^2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 + \lambda h^2 & -1 \\ & & & -1 & 2 + \lambda h^2 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} \alpha/h^2 + f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ \beta/h^2 + f(x_N) \end{pmatrix}.$$

Connections:

- Sec. 6: solution of this system of equations for $\alpha^2 + \beta^2 > 0$, the matrix is for $\lambda > 0$ strictly diagonally dominant and iterative methods converge, see theorem 6.11,
- Sec. 8 and some problems from the tutorials: Calculation/approximation of the Eigenvalues and vectors for $\alpha = \beta = 0$ and $u'' = -\lambda u$.