Fundamentals of Numerical Mathematics

Vít Dolejší
Charles University Prague
Faculty of Mathematics and Physics
Czech Republic
dolejsi@karlin.mff.cuni.cz

January 7, 2019

Preface

These lecture notes more or less cover the part of the lecture Fundamentals of Numerical Mathematic given by the author at the bachelor program at the Charles University in Prague, the Faculty of Mathematics and Physics. They should serve as a survey of the lecture without a mathematically rigorous derivation and without explaining all details. Most ideas are explained by some examples.

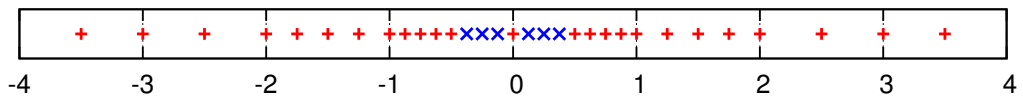
Contents

Chapter 1

Machine arithmetic (1/2 week)

1.1 Machine arithmetic

- $\mathbb{R}$ (system of real numbers) is infinite
- computers can contain only the finite number of real numbers, system $\mathbb{F}$
- computations in $\mathbb{R}$ has to use rounding



1.2 Machine representation of real numbers: system $\mathbb{F}$

The computers use mostly the binary system ($\beta = 2$):

$$x = \pm \left(d_0 + \frac{d_1}{2} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + \cdots + \frac{d_{t-1}}{2^{t-1}} \right) 2^e, \tag{1.1}$$

where $d_i \in \{0, 1\},\ i = 0, \ldots, t-1$ and $L \leq e \leq U$, e and $d_i,\ i = 0, \ldots, t-1$ are integers. The t-plet $(d_0 d_1 \ldots d_{t-1})$ is called the mantissa (or also significant), the number e is the exponent and β is base.

The numbers in $\mathbb{F}$ are not distributed equidistantly (only relatively equidistantly).

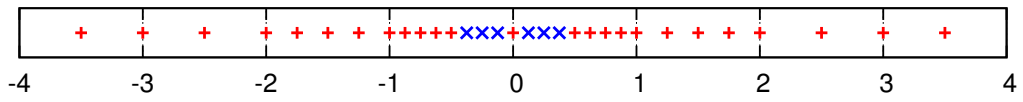
Example 1.1. *Let $\beta = 2$, $t = 3$, $L = -1$ and $U = 1$.*

$x.xx$	2^{-1}	2^0	2^1
0.00	0		
0.01	1/8		
0.10	2/8		
0.11	3/8		
1.00	4/8	4/4	4/2
1.01	5/8	5/4	5/2
1.10	6/8	6/4	6/2
1.11	7/8	7/4	7/2

$\Longleftrightarrow$

$x.xx$	2^{-1}	2^0	2^1
0.00	0		
0.01	0.125		
0.10	0.250		
0.11	0.375		
1.00	0.500	1.00	2.0
1.01	0.625	1.25	2.5
1.10	0.750	1.50	3.0
1.11	0.875	1.75	3.5

Then the numbers from $\mathbb{F} = \mathbb{F}(\beta, t, L, U)$ are plotted here:



1.3 Standard of IEEE and IEC

The standard of the Institute of Electrical and Electronics Engineers (IEEE) and International Electrotechnical Commission (IEC) from 1985:

precision	β	t	L	U	$\#\mathbb{F}$	UFL	OFL	ϵ_{mach}
single	2	24	-126	127	4.26E+09	2.8E-45	6.8E+38	5.96E-08
double	2	53	-1022	1023	1.84E+19	9.9E-304	3.6E+308	1.11E-16
extended	2	64	-16 382	16 383	6.04E+23			5.42E-20
quadruple	2	113	-16 382	16 383	3.40E+38			9.6.E-35

1.4 Under- and over-flow levels

There exists the maximal and the minimal positive numbers of $\mathbb{F}$ by

$$\text{OFL} := \max_{x \in \mathbb{F}} |x| = (1 - \beta^{-t})\beta^{U+1}, \tag{1.2}$$

$$\text{UFL} := \min_{x \in \mathbb{F}} |x| = \beta^{L-t+1}, \tag{1.3}$$

where OFL means the over-flow level and UFL means the under-flow level.

1.5 Rounding

Generally, $x \notin \mathbb{F}$ for $x \in \mathbb{R}$. Hence, we define $\widehat{x} \in \mathbb{F}$, $x \approx \widehat{x}$, e.g.,

$$\widehat{x} = \arg \min_{y \in \mathbb{F}} |x - y|. \tag{1.4}$$

We define the positive real number ϵ_{mach} by

$$\epsilon_{\text{mach}} := \max_{x \in \mathbb{R} \cap [\text{UFL}, \text{OFL}]} \left| \frac{\widehat{x} - x}{x} \right|. \tag{1.5}$$

The number ϵ_{mach} is called the machine accuracy or machine epsilon or simply the accuracy and it represents the maximal possible relative rounding error

Alternatively, ϵ_{mach} is the minimal positive number such that

$$1 + \epsilon_{\text{mach}} > 1 \qquad \text{(in the computer representation)}$$

We have

$$\epsilon_{\text{mach}} = \beta^{-t}.$$

Remark 1.2. *If $x \in \mathbb{R}$ such that $UFL \le |x| \le OFL$ then there exists $\delta \in \mathbb{R}$, $|\delta| \le \epsilon_{\text{mach}}$ such that $\widehat{x} = x(1 + \delta)$.*

1.6 Mathematical operations in the system $\mathbb{F}$

- The system $\mathbb{F}$ was introduce to approximate the real numbers $\mathbb{R}$.
- We need to deal with the usual mathematical operations (e.g., adding, subtracting, multiplication, division) within the system $\mathbb{F}$. We speak about the finite precision arithmetic.

1.7 Basic aspects of the finite precision arithmetic

- Let $*$ denote a mathematical operation on the real numbers $\mathbb{R}$, i.e., $x*y \in \mathbb{R}$ for any $x, y \in \mathbb{R}$. E.g., $* \in \{+, -, \times, /\}$.
- If $x, y \in \mathbb{F}$ then $x*y \notin \mathbb{F}$ in general,
- we have already introduced the embedding $\widehat{\ }: \mathbb{R} \to \mathbb{F}$ (rounding)
- $* : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, we define its analogue $\widehat{*} : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ by

$$x\widehat{*}y = \widehat{x*y} \tag{1.6}$$

- In virtue of Remark 1.2, we have $\widehat{x} = x(1+\rho)$, where $|\rho| \le \epsilon_{\text{mach}}$. Analogously,

$$x\widehat{*}y = (x*y)(1+\rho), \quad |\rho| \le \epsilon_{\text{mach}}. \tag{1.7}$$

Example 1.3. *Let $x, y, x \in \mathbb{R}$. We assume that $|x+y+z| \le OFL$, for simplicity. We want to compute $x+y+z$. In the finite precision arithmetic, we can evaluate only $x\widehat{+}y\widehat{+}z$. We investigate the corresponding rounding error. Then, using* (1.7)*, we have*

$$\begin{aligned}(x\widehat{+}y)\widehat{+}z &= (x+y)(1+\rho_1)\widehat{+}z = [(x+y)(1+\rho_1)+z](1+\rho_2)\\ &= x+y+z+(x+y)(\rho_1+\rho_2+\rho_1\rho_2)+z\rho_2,\end{aligned}$$

where $|\rho_1| \le \epsilon_{\text{mach}}$ and $|\rho_2| \le \epsilon_{\text{mach}}$. Using the different order of adding, we have

$$\begin{aligned}x\widehat{+}(y\widehat{+}z) &= x\widehat{+}(y+z)(1+\rho_3) = [x+(y+z)(1+\rho_3)](1+\rho_4)\\ &= x+y+z+x\rho_4+(y+z)(\rho_3+\rho_4+\rho_3\rho_4),\end{aligned}$$

where $|\rho_3| \le \epsilon_{\text{mach}}$ and $|\rho_4| \le \epsilon_{\text{mach}}$. From the above relations we deduce that the adding in the finite precision arithmetic is not associative. Similarly, we obtain the same conclusion for the multiplication.

Remark 1.4. *The adding (and similarly the multiplication) in the finite precision arithmetic is usually commutative, we can write*

$$x\widehat{+}y = \widehat{x+y} = \widehat{y+x} = y\widehat{+}x.$$

Example 1.5. *Let as consider the infinite row $\sum_{n=1}^{\infty} \frac{1}{n}$. Obviously, this row diverges (the sum is infinity). However, the evaluation in $\mathbb{F}$ leads to a finite limit number (approximately 15.40 in the single precision and 22.06 in the double precision – these values may depends on the used computer and the programming language translator).*

This follows from the fact that

$$\exists n_0 \in \mathbb{N}: \ \frac{1}{n_0} \leq \epsilon_{\text{mach}} \sum_{n=1}^{n_0-1} \frac{1}{n}.$$

Therefore, the terms $1/n_0$, $1/(n_0+1), \ldots$ *does not bring any increase of the sum.*

1.8 Cancellation

The subtraction of two similar numbers leads to a large loss of the accuracy. This effect is called the cancellation and it is illustrated in the following example.

Example 1.6. *Let*

$$x = 123.456478, \quad y = 123.432191 \qquad \Longrightarrow \qquad x - y = 0.0024267 = 2.4267 \times 10^{-2}$$

We consider $\mathbb{F}$ *with* $\beta = 10$ *and* $t = 6$*. The representation of the numbers* x *and* y *in* $\mathbb{F}$ *reads*

$$x = 1.23456 \times 10^2, \quad y = 1.23432 \times 10^2$$

and their difference in $\mathbb{F}$ *is*

$$x \widehat{-} y = 2.40000 \times 10^{-2},$$

hence the result has only two decimal digits. Therefore, the relative rounding error of this computation is

$$\frac{(x \widehat{-} y) - (x - y)}{x - y} = \frac{2.4 \times 10^{-2} - 2.4267 \times 10^{-2}}{2.4267 \times 10^{-2}} = 0.011003$$

i.e., more than 10^{-2} *(using* $t = 6$*).*

Example 1.7. *Let us consider the quadratic equation*

$$ax^2 + bx + c = 0. \tag{1.8}$$

The roots are given either by

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1.9}$$

or by

$$x_{1,2} = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}} \tag{1.10}$$

Let $a = 0.05010$, $b = -98.78$ *and* $c = 5.015$*. The exact roots of* (1.8) *are*

$$x_1 = 1971.605916, \quad x_2 = 0.05077068387$$

Let us consider the system $\mathbb{F}$ *with* $\beta = 10$ *and* $t = 4$*. Then the roots evaluated in the finite precision arithmetic by formula* (1.9) *are*

$$x_1 = 1972, \quad x_2 = 0.0998$$

and by formula (1.10) *are*

$$x_1 = 1003, \quad x_2 = 0.05077.$$

Therefore, x_2 *given by* (1.9) *and* x_1 *given by* (1.10) *are completely wrong. The reason is the cancellation since* $\sqrt{b^2 - 4ac} = 98.77 = b$ *in the finite precision arithmetic.*

Example 1.8. *Let* $h > 0$*, we define the sequence*

$$\begin{aligned} y_0 &= 1, \\ y_{k+1} &= y_k + h\left(-100y_k + 100hk + 101\right), \quad k = 0, 1, \ldots. \end{aligned} \tag{1.11}$$

Let us put $h = 0.1$*, we can derive that*

$$\begin{aligned} y_1 &= 1.0 + 0.1 \cdot (-100 \cdot 1.0 + 100 \cdot 0.0 + 101) = 1.1 \\ y_2 &= 1.1 + 0.1 \cdot (-100 \cdot 1.1 + 100 \cdot 0.1 + 101) = 1.2 \\ y_3 &= 1.2 + 0.1 \cdot (-100 \cdot 1.2 + 100 \cdot 0.2 + 101) = 1.3 \\ y_4 &= 1.3 + 0.1 \cdot (-100 \cdot 1.3 + 100 \cdot 0.3 + 101) = 1.4 \\ &\vdots \end{aligned}$$

```
    y0 = 1.D+00
    h = 0.1
    k = 0
    write(*,  '(i5, 3es14.6)' ) 0, h, 0., y0

10  continue
    y1 = y0 + h*(-100* y0 + 100 * h*k + 101)
    k = k + 1

    write(*,  '(i5, 3es14.6)' ) k, h, h*k, y1
    y0 = y1

    if(t < 2.) goto 10
```

gives the output

```
    0  1.000000E-01  0.000000E+00  1.000000E+00
    1  1.000000E-01  1.000000E-01  1.100000E+00
    2  1.000000E-01  2.000000E-01  1.200000E+00
    3  1.000000E-01  3.000000E-01  1.299999E+00
    4  1.000000E-01  4.000000E-01  1.400007E+00
    5  1.000000E-01  5.000000E-01  1.499938E+00
    6  1.000000E-01  6.000000E-01  1.600556E+00
```

```
 7  1.000000E-01  7.000000E-01  1.694994E+00
 8  1.000000E-01  8.000000E-01  1.845049E+00
 9  1.000000E-01  9.000000E-01  1.494555E+00
10  1.000000E-01  1.000000E+00  5.649004E+00
11  1.000000E-01  1.100000E+00 -3.074104E+01
12  1.000000E-01  1.200000E+00  2.977694E+02
13  1.000000E-01  1.300000E+00 -2.657824E+03
14  1.000000E-01  1.400000E+00  2.394352E+04
15  1.000000E-01  1.500000E+00 -2.154676E+05
16  1.000000E-01  1.600000E+00  1.939234E+06
17  1.000000E-01  1.700000E+00 -1.745308E+07
18  1.000000E-01  1.800000E+00  1.570777E+08
19  1.000000E-01  1.900000E+00 -1.413700E+09
20  1.000000E-01  2.000000E+00  1.272330E+10
```

Show on the computer
code `./stab_euler 0.1` *in directory* `~/vyuka/ZNM/arithemtic`
It is caused by the instability of (1.11) *(for the time step* $h = 0.1$*) and the rounding errors. It is possible to prove that*

$$y_k = 1 + hk.$$

Inserting into (1.11)

$$\begin{aligned} y_{k+1} &= y_k + h\left(-100y_k + 100hk + 101\right) \\ &= 1 + hk + h\left(-100(1 + hk) + 100hk + 101\right) \\ &= 1 + hk + h\left(-100 + 101\right) = 1 + h(k + 1). \end{aligned}$$

Show on the computer
code `./stab_euler 0.02` in directory `~/vyuka/ZNM/arithemtic`

1.9 Costly disasters caused by rounding errors

- Intel Pentium flaw (chyba) 1994, new Pentium chip has a "bug in the floating point unit" million dollars to covers costs
- 1996, accident of Racket Ariane 5, 7 billion dollars, allocated memory for deviation was not enough
- 1991, Gulf War, the Patriot missile defence system failed due to roundoff error: approximation of 0.1 in 24 bits causes a rounding error which increases after 100 hours of operations (28 American soldiers died).

Chapter 2

Numerical mathematics for the mathematical analysis (1/2 week)

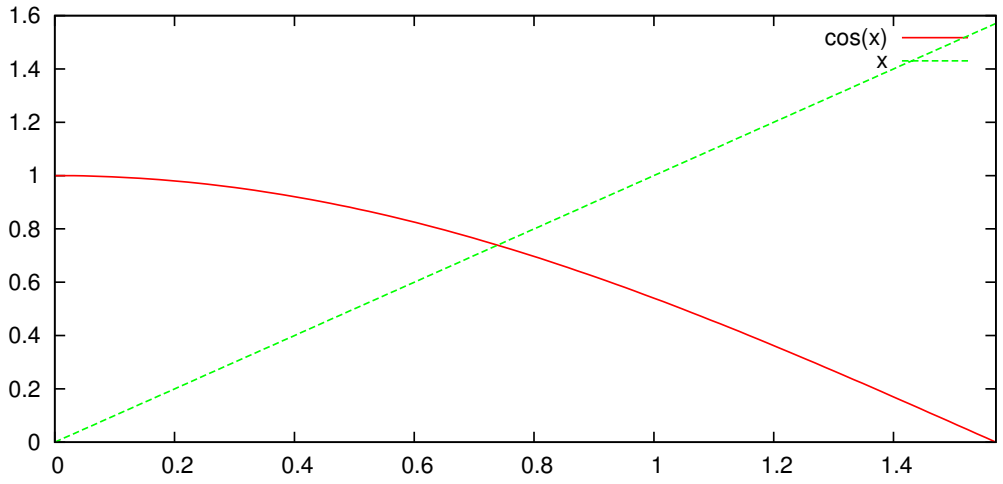
- Numerical mathematics for linear algebra (Z. Strakoš)
- Now: numerical solution of problems of mathematical analysis
- Numerical computations always suffers from errors
  - problems of linear algebra: dominate the rounding errors
  - problems of mathematical analysis: dominate the discretization errors
- Practically, we are able to solve only LINEAR problems (exception is, e.g., quadratic algebraic equation)
- We use often a linearization

2.1 Nonlinear algebraic equation

Let $f : [a, b] \to \mathbb{R}$ be a given continuous function such that $f(a)f(b) < 0$, thus $\exists \bar{x} \in [a, b]$ such that

$$f(\bar{x}) = 0. \tag{2.1}$$

Example 2.1. *Let* $f(x) = x - \cos(x)$, $a = 0$ *and* $b = \pi/2$, *there exists one* $\bar{x}$ *satisfying* (2.1)



- We can not evaluate $\bar{x}$ exactly,
- We approximate $\bar{x}$ solving (2.1) numerically.
- We use an iterative process, define $\{x_k\}$ such that $x_k \to x$.

We assume that we are able to evaluate f and f' at any $x \in [a, b]$. We use the Newton method.

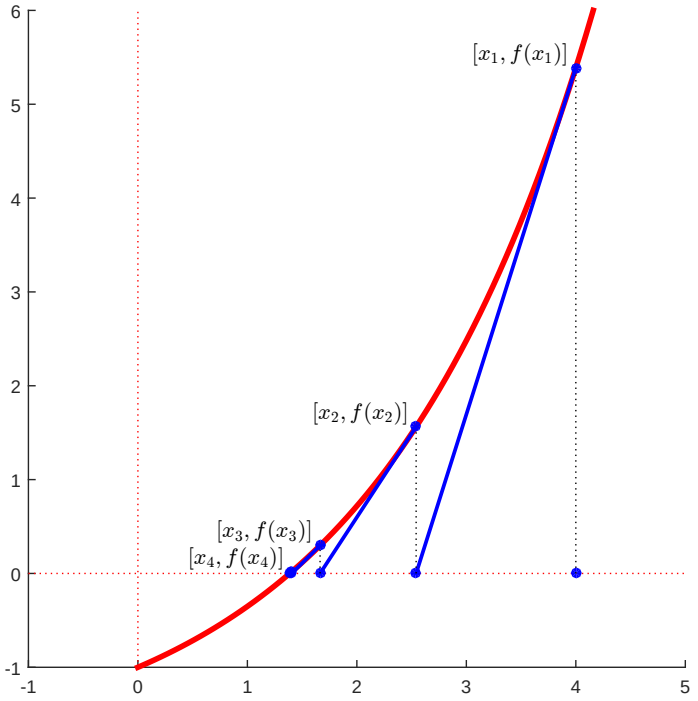
Let x_k be a given approximation, we replace f at x_k by a linear function (using the Taylor expansion):

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k) := \tilde{f}(x) \tag{2.2}$$

We seek x_{k+1} such that $\tilde{f}(x_{k+1}) = 0$, i.e.,

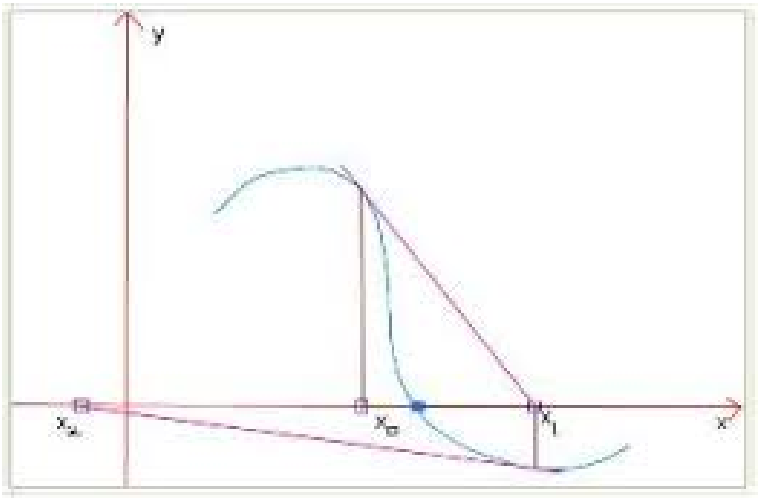
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

We put $k := k + 1$ and repeat the computation.



The difference $\bar{x} - x_k$ is called the discretization error. It arises due to the approximation in (2.2). All relations are in the exact arithmetic.

Newton method is very efficient but does not always converge.



2.1.1 System of nonlinear algebraic equations

The previous can be extended to a nonlinear algebraic system

$$\boldsymbol{f}(x) = 0, \qquad \text{where } \boldsymbol{f} = (f_1, \ldots, f_n)^{\mathrm{T}} : \mathbb{R}^n \to \mathbb{R}^n, \ x \in \mathbb{R}^n. \tag{2.3}$$

Newton method:

$$\begin{aligned} & x_{k+1} := x_k - (\mathbb{F}(x_k))^{-1} \boldsymbol{f}(x_k) \\ \Longleftrightarrow & \\ & x_{k+1} := x_k + \boldsymbol{d}_k, \quad \mathbb{F}(x_k)\boldsymbol{d}_k = \boldsymbol{f}(x_k) \end{aligned} \tag{2.4}$$

where $\mathbb{F}$ is the Jacobi matrix

$$\mathbb{F} = \{F_{ij}\}_{i,j=1}^{n}, \quad F_{ij} = \frac{\partial f_i}{\partial x_j}, \ i,j = 1,\ldots,n.$$

The numerical solution of the nonlinear algebraic system (2.3) was transformed to the numerical solution of a sequence of linear algebraic systems (2.4).

2.2 Numerical differentiation

Let $f : [a,b] \to \mathbb{R}$ be a given differentiable function, we want to evaluate

$$f'(\bar{x}), \quad \bar{x} \in [a,b].$$

In practice, f is an output of a code subroutine, hence we can evaluate f at any $x \in [a,b]$ but we can not evaluate f' analytically.

Definition of the derivative gives

$$f'(\bar{x}) = \lim_{h \to 0} \frac{f(\bar{x}+h) - f(\bar{x})}{h},$$

which we can use in the following approximation: Let $h > 0$ be given, then

$$f'(\bar{x}) \approx \frac{f(\bar{x}+h) - f(\bar{x})}{h} =: Df(\bar{x};h).$$

2.2.1 Discretization error

Discretization error of $f'(\bar{x}) - Df(\bar{x};h)$?

Let $f \in C^2(\mathbb{R})$. The Taylor expansion gives:

$$f(\bar{x}+h) - f(\bar{x}) = hf'(\bar{x}) + \frac{1}{2}h^2 f''(\bar{x}+\theta h), \quad \theta \in [0,1], \tag{2.5}$$

i.e.,

$$\frac{1}{h}\left(f(\bar{x}+h) - f(\bar{x})\right) = f'(\bar{x}) + \underbrace{\frac{1}{2}hf''(\bar{x}+\theta h)}_{\text{discretization error}}, \quad \theta \in [0,1].$$

If f'' is bounded then

$$Df(\bar{x};h) \to f'(\bar{x}) \text{ for } h \to 0.$$

The discretization error is $O(h)$, i.e., the first order method.

2.2.2 Rounding errors

However, in finite precision arithmetic:

we do not know $f(\bar{x})$ but $\hat{f}(\bar{x})$ $\quad |f(\bar{x}) - \hat{f}(\bar{x})| \le \epsilon_{\text{mach}} f(\bar{x})$

we do not know $f(\bar{x}+h)$ but $\hat{f}(\bar{x}+h)$ $\quad |f(\bar{x}+h) - \hat{f}(\bar{x}+h)| \le \epsilon_{\text{mach}} f(\bar{x}+h)$

(for simplicity let $\bar{x} = \widehat{\bar{x}},\ h = \widehat{h}, \dots$)

Then

$$\widehat{Df}(\bar{x};h) = \frac{\hat{f}(\bar{x}+h) - \hat{f}(\bar{x})}{h}$$

and the rounding error gives

$$Df(\bar{x};h) - \widehat{Df}(\bar{x};h) = \frac{f(\bar{x}+h) - f(\bar{x})}{h} - \frac{\hat{f}(\bar{x}+h) - \hat{f}(\bar{x})}{h}.$$

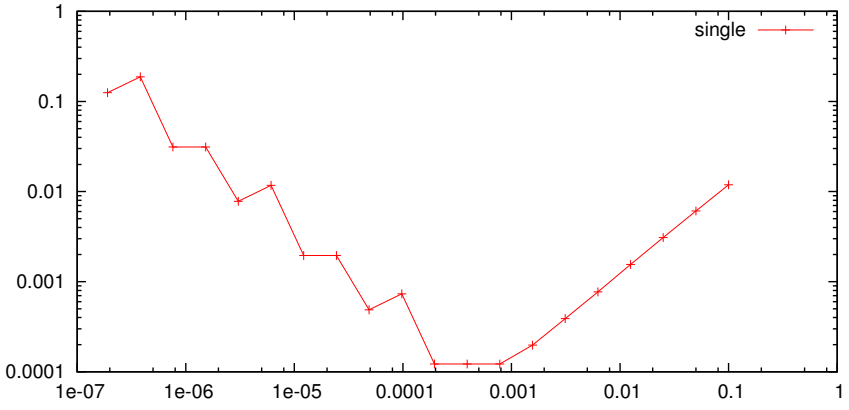
We estimate

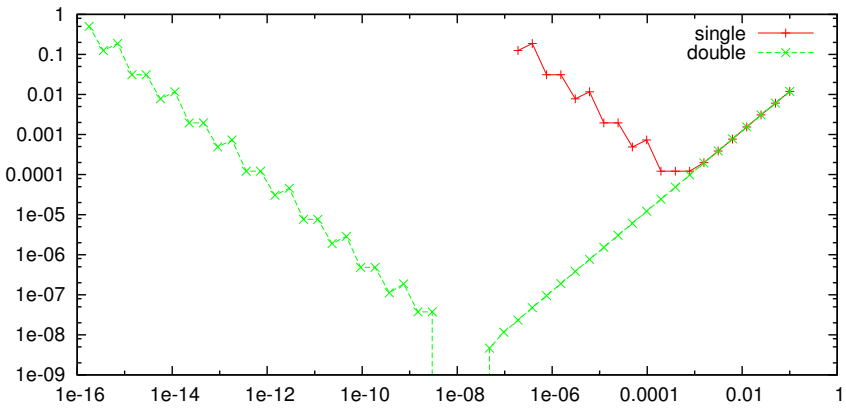
$$\begin{aligned}\left|Df(\bar{x};h) - \widehat{Df}(\bar{x};h)\right| &\le \left|\frac{f(\bar{x}+h) - f(\bar{x})}{h} - \frac{\hat{f}(\bar{x}+h) - \hat{f}(\bar{x})}{h}\right| \\ &\le \left|\frac{f(\bar{x}+h) - \hat{f}(\bar{x}+h)}{h}\right| + \left|\frac{f(\bar{x}) - \hat{f}(\bar{x})}{h}\right| \le 2\max(f(\bar{x}), f(\bar{x}+h))\frac{\epsilon_{\text{mach}}}{h}.\end{aligned}$$

Therefore, the total error (discretization + rounding)

$$\left|f'(\bar{x}) - \widehat{Df}(\bar{x};h)\right| \le \frac{1}{2}f''(\bar{x}+\theta(x-\bar{x}))h + 2\max(f(\bar{x}), f(\bar{x}+h))\frac{\epsilon_{\text{mach}}}{h}.$$

Example 2.2. *Example of computation f' for $f(x) = \sqrt{x}$ at $\bar{x} = 1$.*





2.2.3 Second order approximation

Similarly as in (2.5), we have

$$f(\bar{x}+h) - f(\bar{x}) = +hf'(\bar{x}) + \frac{1}{2}h^2 f''(\bar{x}) + \frac{1}{6}h^3 f'''(\bar{x}+\theta h), \quad \theta \in [0,1],$$
$$f(\bar{x}-h) - f(\bar{x}) = -hf'(\bar{x}) + \frac{1}{2}h^2 f''(\bar{x}) - \frac{1}{6}h^3 f'''(\bar{x}+\tilde{\theta} h), \quad \tilde{\theta} \in [0,1].$$

Subtracting we have

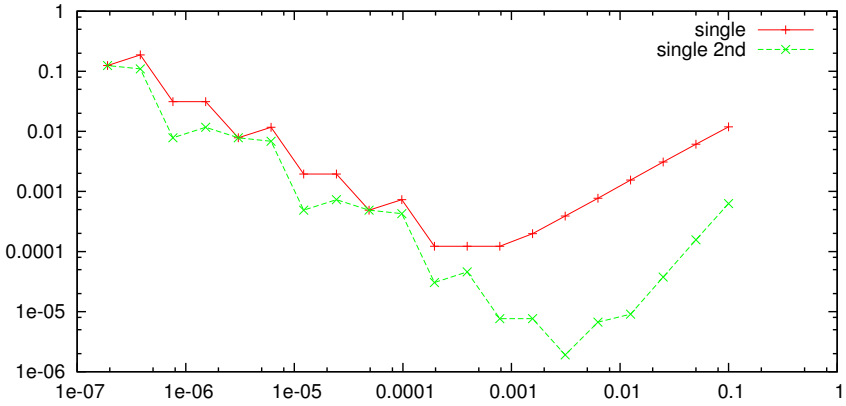
$$f(\bar{x}+h) - f(\bar{x}-h) = 2hf'(\bar{x}) + \frac{1}{6}h^3 \left(f'''(\bar{x}+\theta(x-\bar{x})) - f'''(\bar{x}+\tilde{\theta}(x-\bar{x})) \right)$$

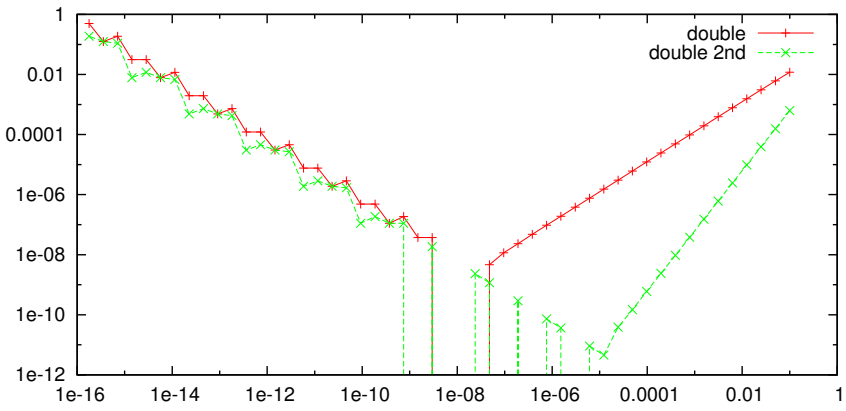
and thus

$$f'(\bar{x}) = \underbrace{\frac{f(\bar{x}+h) - f(\bar{x}-h)}{2h}}_{=:D^2 f(\bar{x})} + O(h^2),$$

which is the approximation of the second order.

Example 2.3. *Example of computation f' for $f(x) = \sqrt{x}$ at $\bar{x} = 1$.*





2.3 Numerical integration

Let $f : (a, b) \to \mathbb{R}$ be an integrable function, we want to evaluate

$$I(f) := \int_a^b f(x)\,\mathrm{d}x. \tag{2.6}$$

- Many integrals can not be evaluated analytically.
- Some approximation is necessary.

Idea: The definition of the Riemann integral.
Draw figure
Let $N \in \mathbb{N}$, $h = (b-a)/N$, $x_i = a + ih$, $i = 0, \ldots, N$ be a partition of $[a, b]$, then

$$\int_a^b f(x)\,\mathrm{d}x \approx h \sum_{i=1}^{N} \inf_{x \in (x_{i-1}, x_i)} f(x) =: M_h(f). \tag{2.7}$$

From the definition of the Riemann integral $M_h(f) \to I(f)$ if $h \to 0$.
However, the convergence $M_h(f) \to I(f)$ is slow, we can show that

$$|M_h(f) - I_h(f)| \approx O(h).$$

Draw figure More acurate is the trapezoidal rule

$$\int_a^b f(x)\,\mathrm{d}x \approx h \sum_{i=1}^{N} \frac{f(x_i) + f(x_{i-1})}{2} =: T_h(f), \tag{2.8}$$

where

$$|T_h(f) - I_h(f)| \approx O(h^2).$$

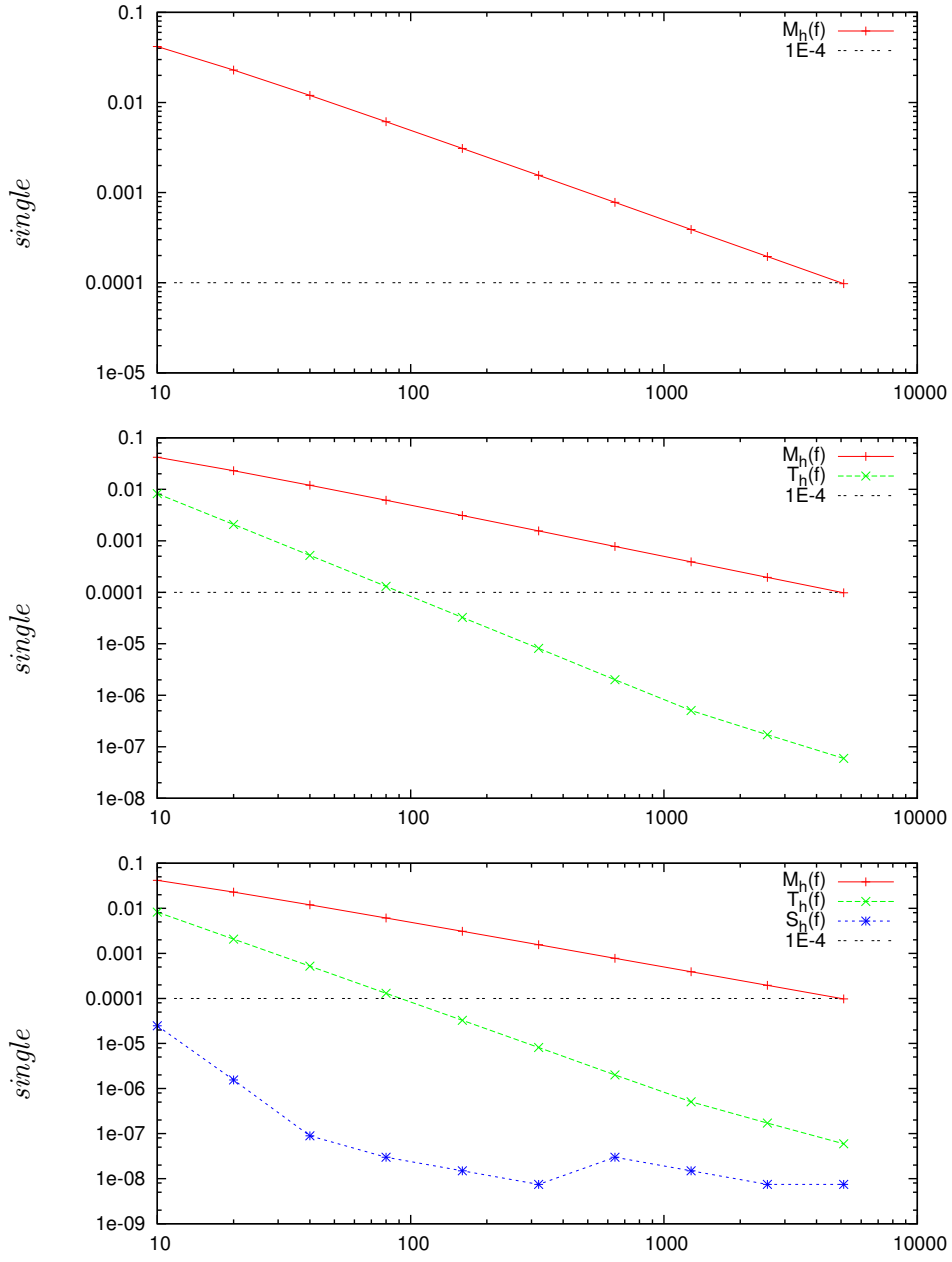
or the Simpson rule

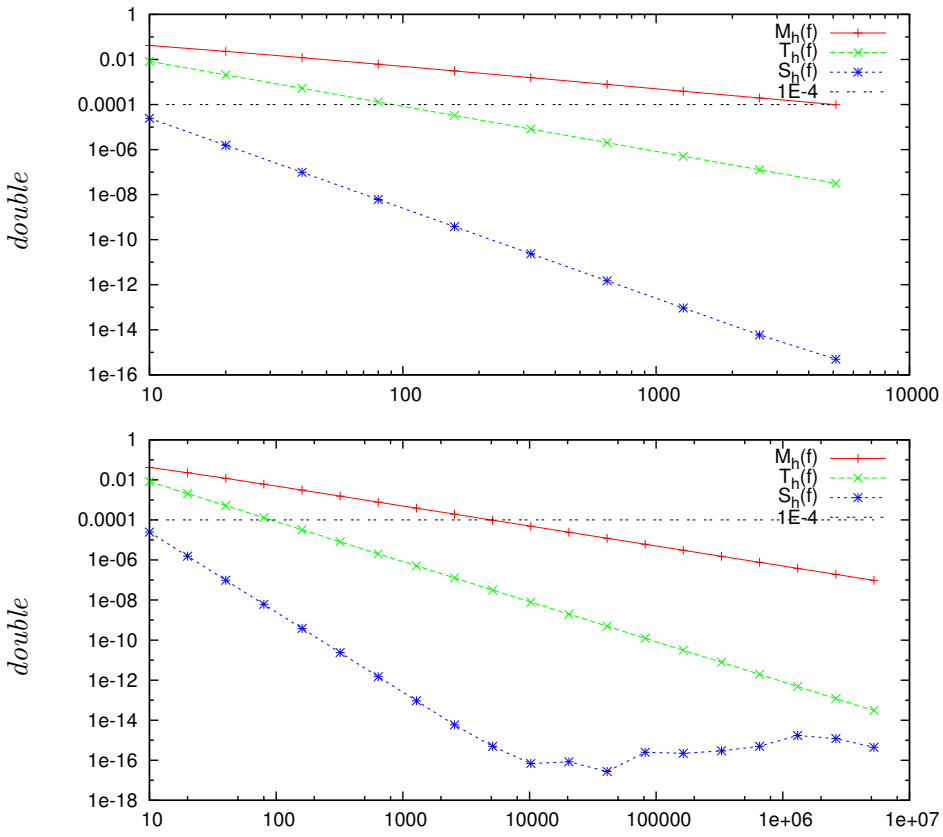
$$\int_a^b f(x)\,\mathrm{d}x \approx h \sum_{i=1}^{N} \frac{f(x_i) + 4f((x_i + x_{i-1})/2) + f(x_{i-1})}{6} =: S_h(f), \tag{2.9}$$

where

$$|S_h(f) - I_h(f)| \approx O(h^4).$$

Example 2.4. *Example of computation* $\int_0^a \sqrt{x}\,\mathrm{d}x$. **Show on the computer**





Efficiency of the method:
How many mathematical operations are necessary for achieving the given tolerance.

Very often we replace "number of mathematical operations" by the "number of degrees of freedom".

Chapter 3

Solution of nonlinear algebraic equations (1 week)

3.1 Solution of a single nonlinear equation

Let $f : [a, b] \to \mathbb{R}$ be a continuous function. We seek $\bar{x} \in [a, b]$ such that

$$f(\bar{x}) = 0. \tag{3.1}$$

Such $\bar{x}$ may not exist, generally.

Example 3.1. *Let $f(x) := x - cos(x)$ and $[a, b] = [0, \pi]$, there exists one solution of (3.1).*

Example 3.2. *The cannon fire on an enemy d meters away. The initial velocity v_0 is known. Set the correct angle of attack α.* **Draw figure** *Let $y(t)$ denote the height of the projectile. The gravity is the only one force, hence*

$$y''(t) = -g, \qquad g \approx 9.81m\, s^{-2}.$$

Integrating

$$y'(t) = -gt + c_1, \qquad \text{where } c_1 = v_0 \sin(\alpha).$$

Second integrating

$$y(t) = -\frac{1}{2}gt^2 + v_0 \sin(\alpha)t + c_2, \qquad \text{where } c_2 = 0 \text{ since } y(0) = 0.$$

Thus projectile hits the ground, when

$$0 = -\frac{1}{2}gt^2 + v_0 \sin(\alpha)t \quad \Longrightarrow \quad t = 0, \quad t = \frac{2v_0 \sin(\alpha)}{g}.$$

In horizontal direction the velocity is $v_0 \cos(\alpha)$, hence

$$d = v_0 \cos(\alpha)t = \frac{2v_0^2 \sin(\alpha)\cos(\alpha)}{g}.$$

We know d and seek α. It is a nonlinear equation for α, which can be solved numerically.

- *Analytical solution is* $\alpha = \frac{1}{2} arcsin \frac{dg}{v_0^2}$, *but how is arcsin evaluated?*
- *Model of shutting is an approximation, an approximate solution is enough.*
- *The solution may not exists: namely if* $d > \frac{v_0^2}{g}$ *then solution does not exists.*
- *Including the air resistance:*

$$y''(t) = -g - ky', \qquad k > 0 \text{ is the coefficient of the resistance.}$$

Analytical solution

$$y(t) = -\frac{1}{k} e^{-kt} v_0 \sin(\alpha) - \frac{g}{k} \left(t + \frac{1}{k} e^{-kt} \right) + \frac{1}{k} v_0 \sin(\alpha) + \frac{g}{k^2}. \tag{3.2}$$

Find t *such that* $y(t) = 0$ *is impossible analytically.*

Example 3.3. Inverse problem: *Previous example, but* k *is unknown. We experimentally found the pair* (α, d) *and seek* k *from* (3.2). *Can not be solved analytically.*

Example 3.4. Computer graphics: *Object defined by* $x^4 + y^4 \leq 1$ *and a line behind* $y = x + 0.5$ *We need the intersection nodes, i.e., equation* $x^4 + (x + 0.5)^4 = 1$. *It has to be solved numericaly.*

Let us go back to (3.1).

3.1.1 Bisection method

Let $f : [a, b] \to \mathbb{R}$ be a continuous function such that $f(a)f(b) < 0$, hence there exists at least one solution.

Example 3.5. *Let us play a game. Select an integer number between 1 and 1 000 000. I have at most 20 question of type:*

"Is your selected number bigger (or smaller) than number a?"

What strategy is optimal? If I take always half of the interval, since $2^{20} = 1\,048\,576$, *I will always win.*

The *bisection method* is based on the same idea.

Draw figure

Algorithm 1 Bisection method

let $f : [a, b] \to \mathbb{R}$ such that $f(a)f(b) < 0$ be given
let $\delta > 0$ be the given accuracy
while $|b - a| > 2\delta$ **do**
  $c := (a + b)/2$
  if $f(c) = 0$ **then**
    $x^\star := c$ is the (exact) solution; end
  else if $f(a)f(c) < 0$ **then**
    $b := c$
  else
    $a := c$
  end if
end while
$x^\star := c$ is the approximate solution; end

The rate of the convergence

How fast converge the approximate solution $x^\star$ to $\bar{x}$?

At each step, the size of the interval is reduced by 2, hence

$$\frac{|b-a|}{2^k} \leq 2\delta \quad \Leftrightarrow \quad 2^{k+1} \geq \frac{|b-a|}{\delta} \quad \Leftrightarrow \quad k \geq \log_2 \frac{|b-a|}{\delta} - 1. \tag{3.3}$$

Hint for exercise: How many time steps are necessary to obtain the given accuracy?

It is a linearly convergent method (first order method), the error is reduced by factor 2 at each time step.

3.1.2 Method regula falsi

Modification of the secant method, c is not $(a+b)/2$, but the intersection of the line between $[a, f(a)]$ and $[b, f(b)]$ with $y = 0$. It can be faster than the bisection method.

Draw figure

- Usually, the problem is to find the interval $[a, b]$, where $f(a)f(b) < 0$.
- Equation like $x^2 = 0$ can not be solved. **Draw figure**

3.1.3 Newton method

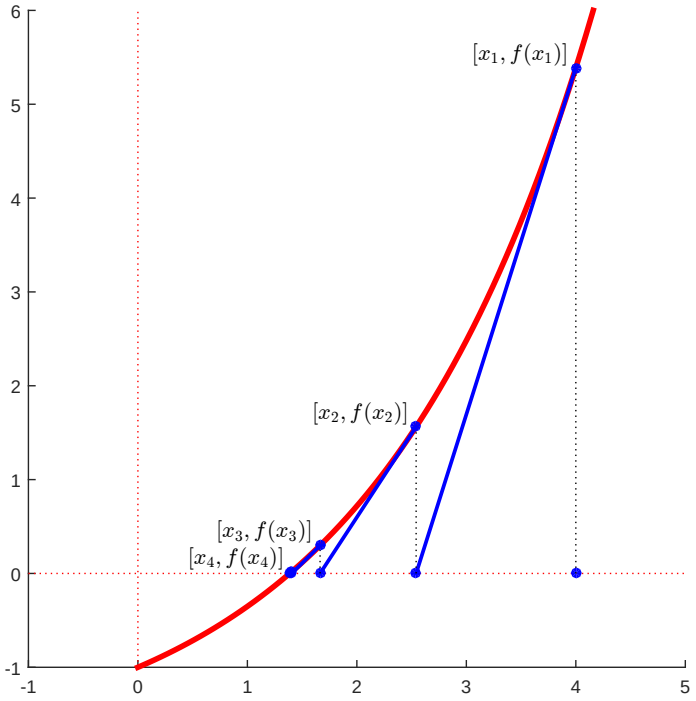
Let x_k be a given approximation, we replace f at x_k by a linear function (using the Taylor expansion):

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k) := \tilde{f}(x) \tag{3.4}$$

We seek x_{k+1} such that $\tilde{f}(x_{k+1}) = 0$, i.e.,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

We put $k := k + 1$ and repeat the computation.



Sometimes, the method is called (by engineers) the Newton-Raphson method.

Theorem 3.6. *If $f \in C^2(\mathbb{R})$, if x_0 is sufficiently close to the solution of (3.1) $\bar{x}$ and if $f'(\bar{x}) \neq 0$, then the Newton method converges to $\bar{x}$ and the asymptotic rate of the convergence is quadratic, i.e., $\exists C > 0$ such that*

$$\lim_{k\to\infty} \frac{|x_{k+1} - \bar{x}|}{|x_k - \bar{x}|^2} = C.$$

Remark 3.7. *Comments to the theorem:*

- *"sufficiently close" will be explained in the proof*
- *quadratic convergence is very fast: if $|x_k - \bar{x}| \approx 10^{-1}$, then $|x_{k+1} - \bar{x}| \approx 10^{-2}$, $|x_{k+2} - \bar{x}| \approx 10^{-4}$, $|x_{k+3} - \bar{x}| \approx 10^{-8}$, etc.*
- *however, the quadratic convergence is only asymptotical, can be slower, depends on x_0.*

Proof. Taylor expansion

$$f(\bar{x}) = f(x_k) + f'(x_k)(\bar{x} - x_k) + \frac{1}{2} f''(\xi_k)(\bar{x} - x_k)^2, \qquad \xi_k \text{ is between } x_k \text{ and } \bar{x}$$

$$\Leftrightarrow \quad \bar{x} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{1}{2}\frac{f''(\xi_k)}{f'(x_k)}(\bar{x} - x_k)^2. \tag{3.5}$$

The Newton method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{3.6}$$

Subtraction (3.6) from (3.5), we have

$$x_{k+1} - \bar{x} = \frac{f''(\xi_k)}{2f'(x_k)}(\bar{x} - x_k)^2. \tag{3.7}$$

Now, f'' is continuous, $f'(\bar{x}) \neq 0$ then

$$C_* := \left| \frac{f''(\bar{x})}{2f'(\bar{x})} \right| < \infty.$$

Let $C > C_*$ be any constant, then there exists $\delta > 0$ such that

$$\left| \frac{f''(\xi)}{2f'(x)} \right| \leq C \qquad \forall x, \xi \in (\bar{x} - \delta, \bar{x} + \delta) =: U.$$

Let

$$x_0 \in U \qquad \text{and} \qquad |x_0 - \bar{x}| < \frac{1}{C}. \tag{3.8}$$

Then (3.7) gives

$$|x_1 - \bar{x}| \leq C|x_0 - \bar{x}|^2 < |x_0 - \bar{x}|.$$

Hence, $x_1 \in U$ and $|x_1 - \bar{x}| < \frac{1}{C}$. By the induction we find that

$$|x_k - \bar{x}| < |x_0 - \bar{x}| \quad \text{and } x_k \in U \qquad \forall k \in \mathbb{N}.$$

Moreover, let $k \in \mathbb{N}$, then (using (3.7))

$$\begin{aligned}
|x_k - \bar{x}| \leq & C|x_{k-1} - \bar{x}|^2 \\
\leq & (C|x_{k-1} - \bar{x}|)\ |x_{k-1} - \bar{x}| \\
\leq & (C|x_{k-1} - \bar{x}|)\ (C|x_{k-2} - \bar{x}|)\ |x_{k-2} - \bar{x}| \\
& \vdots \\
\leq & (C|x_{k-1} - \bar{x}|)\ \ldots\ C|(x_1 - \bar{x}|)\ |x_0 - \bar{x}| \\
\leq & (C|x_0 - \bar{x}|)^k\ |x_0 - \bar{x}|.
\end{aligned}$$

Since $C|x_0 - \bar{x}| < 1$ then $|x_k - \bar{x}| \to 0$, i.e., $x_k \to \bar{x}$ as $k \to \infty$. The Newton method converges.

Moreover, since $x_k \to \bar{x}$ and consequently $\xi_k \to \bar{x}$, we have from (3.7) that

$$\frac{x_{k+1} - \bar{x}}{(\bar{x} - x_k)^2} = \frac{f''(\xi_k)}{2f'(x_k)} \to C_* \quad \text{ for } k \to \infty.$$

□

Remark 3.8.

- *The assumption x_0 is sufficiently close to the solution means* (3.8)*, it is difficult to verify since we can not evaluate C_*.*

- *There exist many other theorems with different assumptions, which are either too restrictive or too difficult to verify.*
- *It is possible to combine the Newton method, e.g., with the bisection method.*

Example 3.9. *Compute* $\sqrt{2}$. *We use the Newton method to solve* $x^2 - 2 = 0$. *Then* $f(x) = x^2 - 2$, $f'(x) = 2x$ *and*

$$x_{k+1} = x_k - \frac{x_k^2 - 2}{2x_k} = \frac{x_k^2 + 2}{2x_k}, \quad k = 1, 2, \ldots$$

Let $x_0 = 1$. *Then we obtain*

k	x_k	x_k - sqrt{2}
0	1.0000000000000000E+00	-4.1421356237309515E-01
1	1.5000000000000000E+00	8.5786437626904855E-02
2	1.4166666666666667E+00	2.4531042935715952E-03
3	1.4142156862745099E+00	2.1239014147411694E-06
4	1.4142135623746899E+00	1.5947243525715749E-12
5	1.4142135623730951E+00	0.0000000000000000E+00
6	1.4142135623730949E+00	-2.2204460492503131E-16

Let $x_0 = 0$, *then the method fails.*
Let $x_0 = -1$, *then* $x_k \to -\sqrt{2}$.

Hint for exercise: On computers: practical examples of convergence and divergence of the Newton method

Hint for exercise: Theoretical: Newton method for the case $f'(\bar{x}) =??$, see [GC12, Theorem 4.3.2] .

3.1.4 Quasi-Newton methods

- The evaluation of f' may be expensive.

Hence, we use

$$x_{k+1} = x_k - \frac{f(x_k)}{g_k} \qquad \text{where} \quad g_k \approx f'(x_k).$$

Constant slope method

$$g_k := f'(x_0)$$

At most linear convergence.

Secant method

$$g_k := \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}, \qquad k = 1, 2, \ldots,$$

Draw figure

Then

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, \qquad k = 1, 2, \ldots.$$

We need x_0 and x_1 to start the secant method.
It can be proven that

- the secant method is convergent
- the order of convergence is $\frac{1+\sqrt{5}}{2}$.

3.1.5 Fixed point method

The problem: let $\varphi(x) : \mathbb{R} \to \mathbb{R}$, we seek $x \in \mathbb{R}$ satisfying

$$x = \varphi(x).$$

If $\bar{x} = \varphi(\bar{x})$, then $\bar{x}$ is called the fixed point of φ. The method given by

$$x_{k+1} = \varphi(x_k), \qquad k = 1, 2, \ldots. \tag{3.9}$$

is called the fixed point iteration.
Some equivalence with $f(x) = 0$, e.g.,

$$\varphi(x) := x + f(x) = x, \qquad \varphi(x) := x - f(x) = x, \qquad \varphi(x) := x + \lambda f(x) = x, \ \lambda \neq 0.$$

Example 3.10. *The Newton method can be considered as a fixed point iteration with*

$$\varphi(x) = x - \frac{f(x)}{f'(x)}$$

There are different types of convergence (Figure from [GC12]):

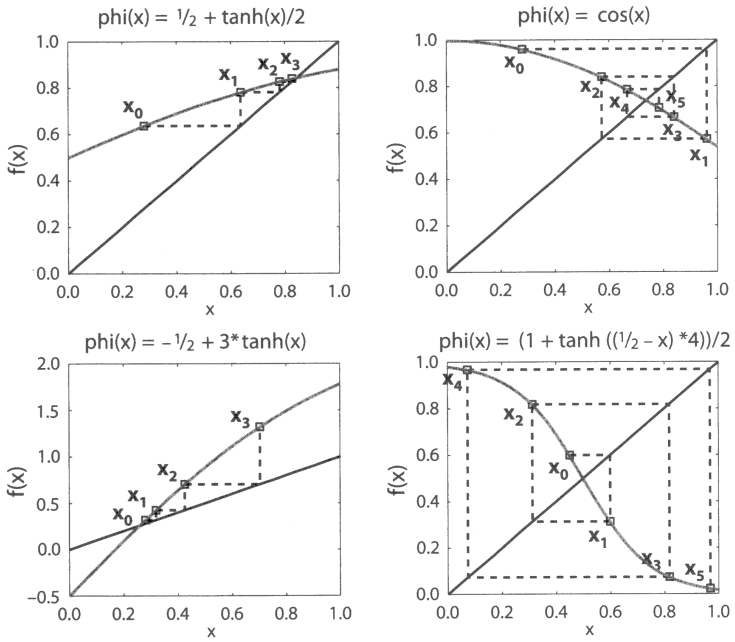


Figure 4.11. Fixed point iteration. The iteration may display monotonic convergence (*upper left*), oscillatory convergence (*upper right*), monotonic divergence (*lower left*), or oscillatory divergence (*lower right*).

Not all choices of φ are suitable:

Example 3.11. *Let* $f(x) = x^3 + 6x^2 - 8 = 0$. *We have* $f(1) = -1 < 0$ *and* $f(2) = 24 > 0$, *there is a root in* $[1, 2]$. *Then*

- $\varphi_1(x) = x^3 + 6x^2 + x - 8$,
- $\varphi_2(x) = \sqrt{\frac{8}{x+6}}$,
- $\varphi_3(x) = \sqrt{\frac{8-x^3}{6}}$.

Let $x_0 = 1.5$, *then the method with* φ_1 *does not converge and the method with* φ_3 *converges slower than the method with* φ_2.

Hint for exercise: Test on the computer.

Theorem 3.12. *Let* $\varphi \in C^1(\mathbb{R})$, $\varphi(\bar{x}) = \bar{x}$ *and let* I *be an interval,* $\bar{x} \in I$ *such that*

$$|\varphi'(x)| < 1 \qquad x \in I$$

and φ *maps* I *into* I, *i.e.,* $\varphi(I) \subset I$. *If* $x_0 \in I$, *then the fixed point iteration converges to* $\bar{x}$.

Proof. Taylor at $\bar{x}$ gives

$$\begin{aligned} x_{k+1} = \varphi(x_k) =& \varphi(\bar{x}) + (x_k - \bar{x})\varphi'(\xi_k) \qquad \xi_k \in (\bar{x}, x_k) \\ =& \bar{x} + (x_k - \bar{x})\varphi'(\xi_k). \end{aligned}$$

Hence

$$x_{k+1} - \bar{x} = (x_k - \bar{x})\varphi'(\xi_k) \qquad \Leftrightarrow \qquad e_{k+1} = e_k\varphi'(\xi_k),$$

where $e_k = x_k - \bar{x}$, $k = 0, 1, \ldots$. Thus

$$|e_{k+1}| = |e_k||\varphi'(\xi_k)|.$$

If $|\varphi'(x)| < 1$ for all $x \in I$, then the error decreases at each step at least by the factor $\max_{x\in I} |\varphi'(x)|$. Moreover, asymptotically

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|} = |\varphi'(\bar{x})|.$$

□

Hint for exercise: Using this theorem explain the convergence or divergence of methods from example (3.11).

The $\varphi'(\bar{x})$ may not exist. Favourable property is the contraction.

Definition 3.13. *We say that φ is a contraction on M, if there exists $L \in (0, 1)$ such that*

$$|\varphi(x) - \varphi(y)| \le L|x - y| \qquad \forall x, y \in M. \tag{3.10}$$

Theorem 3.14. *If φ is a contraction on $\mathbb{R}$, then φ has a unique fixed point $\bar{x}$ and $x_{k+1} = \varphi(x_k)$ converges to $\bar{x}$ for any $x_0 \in \mathbb{R}$).*

Proof. We show that $\{x_k\}$ is a Cauchy sequence. Let $k > j$ then

$$|x_k - x_j| \le |x_k - x_{k-1}| + |x_{k-1} - x_{k-2}| + \cdots + |x_{j+1} - x_j|.$$

Moreover,

$$\begin{aligned} &|x_m - x_{m-1}| = |\varphi(x_{m-1}) - \varphi(x_{m-2})| \le L|x_{m-1} - x_{m-2}| \\ \Longrightarrow &|x_m - x_{m-1}| \le L^{m-1}|x_1 - x_0|. \end{aligned}$$

Then

$$|x_k - x_j| \le (L^{k-1} + L^{k-2} + \cdots + L^j)|x_1 - x_0| = L^j \frac{1 - L^{k-j}}{1 - L}|x_1 - x_0|.$$

If $k \ge N$ and $j \ge N$ then

$$|x_k - x_j| \le L^N \frac{1}{1 - L}|x_1 - x_0| \to 0 \text{ for } N \to \infty,$$

hence $\{x_k\}$ is a Cauchy sequence and it converges to some $\bar{x}$.
Further, we prove that the limit value $\bar{x}$ is a fixed point of φ. If φ is a contraction then φ is a continuous function. Therefore,

$$\varphi(\bar{x}) = \varphi(\lim_{k\to\infty} x_k) = \lim_{k\to\infty} \varphi(x_k) = \lim_{k\to\infty} x_{k+1} = \bar{x},$$

Hence the limit of the Cauchy sequence is the fixed point.
The uniqueness: let $\bar{x}$, $\bar{y}$ be two different fixed points, then

$$|\bar{x} - \bar{y}| = |\varphi(\bar{x}) - \varphi(\bar{y})| \le L|\bar{x} - \bar{y}|,$$

which is a contradiction since $L < 1$, thus $\bar{x} = \bar{y}$. □

3.2 System of nonlinear algebraic equations

We consider a nonlinear algebraic system

$$\boldsymbol{f}(x) = 0, \qquad \text{where } \boldsymbol{f} = (f_1, \ldots, f_n)^{\mathrm{T}} : \mathbb{R}^n \to \mathbb{R}^n, \ x \in \mathbb{R}^n. \tag{3.11}$$

3.2.1 Newton method

A direct generalization gives the Newton method

$$\begin{aligned} & x_{k+1} := x_k - (\mathbb{F}(x_k))^{-1} \boldsymbol{f}(x_k) \\ \iff & \\ & x_{k+1} := x_k + \boldsymbol{d}_k, \quad \mathbb{F}(x_k)\boldsymbol{d}_k = -\boldsymbol{f}(x_k), \end{aligned} \tag{3.12}$$

where $\mathbb{F}$ is a Jacobi matrix

$$\mathbb{F} = \{F_{ij}\}_{i,j=1}^n, \quad F_{ij} = \frac{\partial f_i}{\partial x_j}, \ i, j = 1, \ldots, n.$$

The numerical solution of the nonlinear algebraic system (3.11) was transformed to the numerical solution of a sequence of linear algebraic systems (3.12).

Remark 3.15. *Several comments*

- *Relation* (3.12) *has (analytically) equivalent form*

  $$\mathbb{F}(x_k)x_{k+1} = (\mathbb{F}(x_k))x_k - \boldsymbol{f}(x_k).$$

  However, from the numerical point of view (3.12) *is more stable.*
  Hint for exercise: *Explain why.*

- *The linear algebraic system can be solved directly or iteratively. Then, it is not necessary to solve the linear system as exactly as possible, suitable stopping criteria.*

- *The matrix* $\mathbb{F}$ *has* n^2 *entries, their evaluation may be complicated and/or time consuming. A simplification is possible.*

3.2.2 Fixed point method

The problem: let $\varphi(x) : \mathbb{R}^n \to \mathbb{R}^n$, we seek $x \in \mathbb{R}^n$ satisfying

$$x = \varphi(x).$$

If $\bar{x} = \varphi(\bar{x})$, then $\bar{x}$ is called the fixed point of φ. The method given by

$$x_{k+1} = \varphi(x_k), \qquad k = 1, 2, \ldots. \tag{3.13}$$

is called the fixed point iteration.

Definition 3.16. *We say that* φ *is a contraction on* $M \subset R^n$ *if there exists* $L < 1$ *such that*

$$\|\varphi(x) - \varphi(y)\| \le L\|x - y\| \qquad \forall x, y \in M. \tag{3.14}$$

The modified fixed point theorem is also valid.

Example 3.17. *Let us consider the problem* $F(x) = 0$, $F : \mathbb{R}^n \to \mathbb{R}^n$. *We introduce the method*

$$x_{k+1} = x_k + \delta F(x_k), \qquad k = 1, 2, \ldots,$$

where $\delta \in (0, 1)$ *is called the damping parameter. A suitable choice of* δ *can ensure the convergence of the method. E.g., we set* δ *such that the mapping* $x \to x + \delta F(x)$ *is a contraction.*

Hint for exercise: Scalar examples, where fixed point method converges with $\delta < 1$ but not with $\delta = 1$, theoretically as well as on the computer.

Chapter 4

Interpolation (1 week)

4.1 Motivation

- A general function $f : [a, b] \to \mathbb{R}$ is described by an infinite number of values $f(x),\ x \in [a, b]$.
- For practical computations it is advantageous to approximate f by a finite number of values. The use in engineering, animations, etc.
- Usually $f \approx \varphi$, φ is a polynomial approximation.
- Application: numerical quadrature, $\int_a^b f \,\mathrm{d}x \approx \int_a^b \varphi \,\mathrm{d}x$, the second integral can be evaluated exactly.

4.2 Polynomial approximation

Problem 4.1. *Let $f : [a, b] \to \mathbb{R}$ be a given function.*
Let $a \le x_0 < x_1 < \cdots < x_n \le b$ be a partition.
We seek a function $\varphi : [a, b] \to \mathbb{R}$ such that

- $\varphi(x_i) = f(x_i),\ i = 0, \ldots, n$
- $\varphi(x) \approx f(x) \quad \forall x \in [a, b]$
- *φ is a "nice function" (e.g., polynomial)*

We denote

$$y_i := f(x_i), \qquad i = 0, \ldots, n.$$

We say that φ interpolates f in $x_0, \ldots, x_n$.
Let $P^n(a, b)$ denote the set of polynomial functions of degree at most n over $[a, b]$.
The first idea: let x_i, y_i, $i = 0, \ldots, n$ be given, we seek $\varphi \in P^n(a, b)$ such that

$$\varphi(x_i) = y_i, \quad i = 0, \ldots, n. \tag{4.1}$$

A function from $P^n(a,b)$ has $n+1$ coefficients, we have $n+1$ conditions. Hence, we seek $c_j,\ j=0,\dots,n$ such that

$$y_i = \sum_{j=0}^{n} c_j (x_i)^j, \quad i = 0,\dots,n.$$

It is equivalent to

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}}_{=:\mathbb{V}} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{4.2}$$

This matrix is called the Vandermonde matrix.

Has this system a unique solution? Yes, if the Vandermonde matrix is regular. It is possible to prove that

$$\det \mathbb{V} = \prod_{0 \le i < j \le n} (x_i - x_j).$$

Thus if $x_i,\ i = 0,\dots,n$ are distinct, then there exists a unique solution of (4.2) and the interpolation polynomial has the form

$$\varphi(x) = \sum_{j=0}^{n} c_j x^j.$$

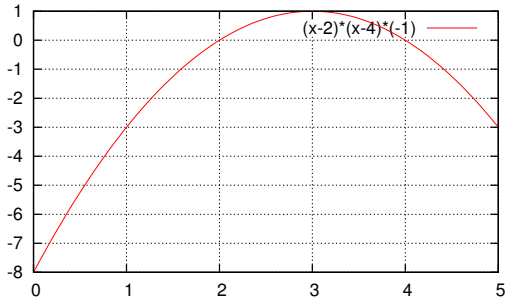
However, $\mathbb{V}$ from (4.2) is ill-conditioned and it is not suitable for practical computations.
Hint for exercise: Show a possible example .

4.2.1 The Lagrange form of the interpolation

We define

$$\varphi_i(x) := \prod_{j \ne i} \frac{x - x_j}{x_i - x_j}, \qquad i = 0,\dots,n.$$

Obviously, $\varphi_i \in P^n$ and $\varphi_i(x_j) = \delta_{ij}$, where δ_{ij} is the Kronecker symbol.



Then

$$\varphi(x) = \sum_{i=0}^{n} y_i \varphi_i(x) = \sum_{i=0}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

is the solution of our problem, since it is a polynomial of degree n and satisfies (4.1). It is called the Lagrange form of the interpolation polynomials.

- This is an equivalent formulation form the point of view of mathematical analysis, but not from the point of view numerical mathematics.
- If we solve the problem (4.2) exactly, then we obtain both polynomials more or less identical. However, in practice we are not able to solve (4.2) exactly.

Hint for exercise: Show a possible example.

4.2.2 The error of the polynomial interpolation

Theorem 4.2. *Let $f \in C^{n+1}([a,b])$ and $x_i \in [a,b],\ i = 0, \ldots, n$. Let $\varphi(x)$ be the polynomial of degree n that interpolates f in $x_i,\ i = 0, \ldots, n$. Then, for each $x \in [a,b]$, we have*

$$f(x) - \varphi(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{j=0}^{n} (x - x_j), \quad \xi_x \in [a,b]. \tag{4.3}$$

Proof. Obviously, (4.3) is valid for $x = x_i,\ i = 0, \ldots, n$.
Let $x \in [a,b],\ x \neq x_i,\ i = 0. \ldots, n$. Let q be the polynomial of degree $n+1$ which interpolates f in $x, x_0, \ldots, x_n$. Then

$$q(t) = \varphi(t) + \lambda \prod_{j=0}^{n} (t - x_j), \qquad \lambda := \frac{f(x) - \varphi(x)}{\prod_{j=0}^{n} (x - x_j)}. \tag{4.4}$$

Let $\phi(t) := f(t) - q(t)$. Then $\phi(t)$ vanishes at $n+2$ nodes $x, x_0, \ldots, x_n$,
Rolle's theorem implies that $\phi(t)'$ vanishes at $n+1$ nodes between successive pairs.
Rolle's theorem implies that $\phi(t)''$ vanishes at n nodes
$\vdots$
$\phi(t)^{(n+1)}$ vanishes at one node, denoted by ξ_x.

Hence,

$$0 = \phi^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - q^{(n+1)}(\xi_x).$$

The $(n+1)$-times differentiation of (4.4) gives

$$q^{(n+1)}(t) = \lambda (n+1)!, \qquad \text{since } \varphi^{(n+1)}(t) = 0.$$

Thus

$$f^{(n+1)}(\xi_x) = \lambda (n+1)! = (n+1)! \frac{f(x) - \varphi(x)}{\prod_{j=0}^{n} (x - x_j)},$$

which gives (4.3). □

Example 4.3. *Let* $f(x) = \sin(x)$, $n = 1$, $x_0 = 0$ *and* $x_1 = \pi/2$. *We interpolate* f, *which gives* $\varphi(x) = (2/\pi)x$. **Draw figure** *Since* $|f''| \le 1$, *then* (4.3) *gives*

$$|\varphi(x) - f(x)| \le \frac{1}{2}|(x-0)(x-\pi/2)|.$$

The maximal value is attained for $x = \pi/4$, *thus*

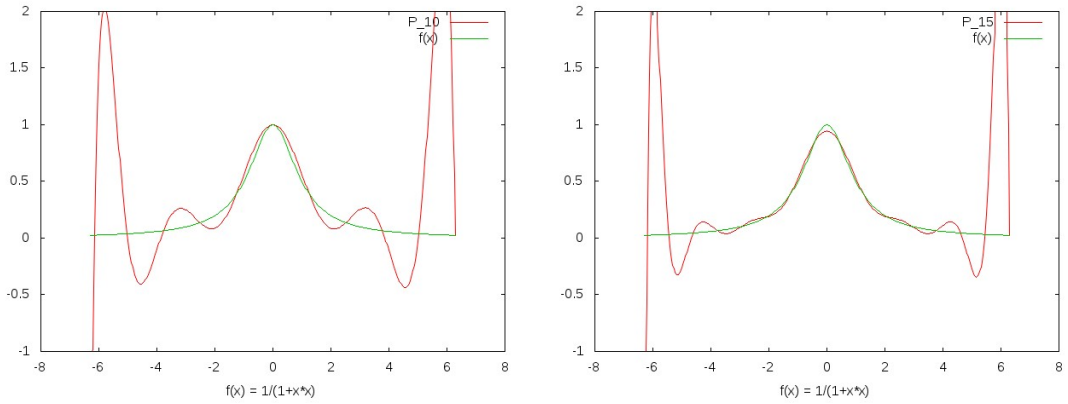
$$|\varphi(x) - f(x)| \le \frac{1}{2}(\pi/4)^2 \approx 0.308.$$

The actual error is equal to $|\sin(\pi/4) - (2/\pi)\pi/4| = (\sqrt{2}-1)/2 \approx 0.207$.

Example 4.4. *Let* $f(x) = sin(x) + 1/2$ *on* $[-2\pi, 2\pi]$. *Then* $\varphi = \varphi_n$ *converges* f *for* $n \to \infty$. **Show on the computer** *Video from* `~/vyuka/ZNM/LAGRANG/Lag-sin.avi` *It converges uniformly.*

Hint for exercise: More examples of an analytically estimation of the interpolation error.

Example 4.5. *The function* $f(x) = \frac{1}{1+x^2}$ *on* $(-5, 5]$. **Show on the computer** *Video from* `~/vyuka/ZNM/LAGRANG/Lag-ratio.avi` *Divergence at* $x = \pm 5$, *term* $f^{(n+1)}(\xi_x)\prod_{j=1}^{n}(x - x_j)$ *grows faster then* $1/(n+1)!$.



Possible solution of such problems:

- Chebyshev interpolation: the nodes are more clustered near the endpoints, namely $x_j = \cos\left(\pi\frac{2j-1}{2n}\right)$, $j = 1, \ldots, n$ on $(-1, 1)$. These values minimizes the term $\prod_{j=0}^{n}(x - x_j)$ on $(-1, 1)$ (in the max-norm).
- Spline functions – see bellow

There exists also, e.g., Hermit interpolation, where we require

$$\varphi(x_i) = f(x_i), \quad i = 0, \ldots, n \qquad \& \qquad \varphi'(x_i) = f'(x_i), \quad i = 0, \ldots, n.$$

4.3 Spline interpolation

Spline interpolation – piecewise polynomial approximation. The most used are the cubic splines.

Problem 4.6. *Let $a = x_0 < x_1 < \cdots < x_n = b$ and $y_i \in \mathbb{R}$, $i = 0, \ldots, n$ be given, we seek $\varphi : C^2([a, b])$ such that*

- $\varphi(x_i) = y_i (= f(x_i))$, $i = 0, \ldots, n$,
- $\varphi|_{[x_{i-1}, x_i]}$ *is a cubic polynomial function on* (x_{i-1}, x_i), $i = 1, \ldots, n$.

4.3.1 Construction of splines

Observation:

- n intervals, piecewise cubic function $\Rightarrow 4n$ unknowns,
- $\varphi|_{(x_{i-1}, x_i)}$ is given at endpoints $\Rightarrow 2n$ conditions,
- φ' is continuous in interior nodes $\Rightarrow n - 1$ conditions,
- φ'' is continuous in interior nodes $\Rightarrow n - 1$ conditions.

Hence, 2 conditions are missing, we prescribe, e.g.,

a) $\varphi'(x_0) = \alpha (= f'(x_0))$, $\quad \varphi'(x_n) = \beta (= f'(x_n))$,

b) $\varphi''(x_0) = \alpha (= f''(x_0))$, $\quad \varphi''(x_n) = \beta (= f''(x_n))$,

c) $\varphi''(x_0) = 0$, $\quad \varphi''(x_n) = 0$ (natural cubic spline).

The case c) is a special variant of b). We describe one (possible) efficient construction of the cubic spline.

Let $M_i := \varphi''(x_i)$, $i = 0, \ldots, n$ (M_i are called momentums), M_0 and M_n are known. Let $i = 0, \ldots, n - 1$. We denote $\varphi_i := \varphi|_{[x_i, x_{i+1}]}$ and $h_i = x_{i+1} - x_i$. Since the spline function φ_i is cubic on $[x_i, x_{i+1}]$, then φ_i'' is linear on $[x_{i-1}, x_i]$. Thus

$$\varphi_i''(x) = M_i + (M_{i+1} - M_i)\frac{x - x_i}{x_{i+1} - x_i} = M_i \frac{x_{i+1} - x}{h_i} + M_{i+1}\frac{x - x_i}{h_i}.$$

Integration gives

$$\begin{aligned}
\varphi_i'(x) &= -M_i \frac{(x_{i+1} - x)^2}{2h_i} + M_{i+1}\frac{(x - x_i)^2}{2h_i} + A_i \\
\varphi_i(x) &= \quad M_i \frac{(x_{i+1} - x)^3}{6h_i} + M_{i+1}\frac{(x - x_i)^3}{6h_i} + A_i(x - x_i) + B_i
\end{aligned}$$

where A_i and B_i are the integration constants.

We have the conditions: $\varphi_i(x_i) = y_i$, $\varphi_i(x_{i+1}) = y_{i+1}$, then

$$\begin{aligned}
\varphi_i(x_i) =& M_i \frac{(x_{i+1} - x_i)^3}{6h_i} + B_i = y_i \qquad \Rightarrow \qquad B_i = y_i - M_i \frac{h_i^2}{6}, \\
\varphi_i(x_{i+1}) =& M_{i+1} \frac{(x_{i+1} - x_i)^3}{6h_i} + A_i(x_{i+1} - x_i) + B_i = y_{i+1} \\
\Rightarrow \quad A_i =& \frac{1}{h_i}\left(y_{i+1} - M_{i+1}\frac{h_i^2}{6} - B_i\right) \quad \Rightarrow \quad A_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}\left(M_{i+1} - M_i\right).
\end{aligned}$$

It rests to determine $M_i,\ i = 1, \ldots, n-1$. We use the continuity of the first derivatives, i.e.,

$$\varphi_i(x_i^+) = \varphi_{i-1}(x_i^-), \qquad i = 1, \ldots, n-1 \tag{4.5}$$

Thus

$$\begin{aligned}
\varphi'_{i-1}(x_i^-) =& \frac{1}{2}M_i h_{i-1} + A_{i-1} = \frac{1}{2}M_i h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{6}(M_i - M_{i-1}), \\
\varphi'_i(x_i^+) =& -\frac{1}{2}M_i h_i + A_i = -\frac{1}{2}M_i h_i + \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(M_{i+1} - M_i).
\end{aligned}$$

From the condition (4.5), we have

$$\frac{1}{2}M_i h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{6}(M_i - M_{i-1}) = -\frac{1}{2}M_i h_i + \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(M_{i+1} - M_i),$$

which gives

$$\frac{1}{2}M_i h_{i-1} - \frac{h_{i-1}}{6}(M_i - M_{i-1}) + \frac{1}{2}M_i h_i + \frac{h_i}{6}(M_{i+1} - M_i) = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}$$

and

$$\frac{h_{i-1}}{6}M_{i-1} + \frac{1}{3}(h_{i-1} + h_i)M_i + \frac{h_i}{6}M_{i+1} = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}.$$

Denoting

$$\lambda_i := \frac{h_{i-1}}{h_{i-1} + h_i}, \qquad \mu_i := 1 - \lambda_i = \frac{h_i}{h_{i-1} + h_i}, \qquad g_i = \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)\frac{6}{h_{i-1} + h_i},$$

we obtain

$$\lambda_i M_{i-1} + 2M_i + \mu_i M_{i+1} = g_i, \qquad i = 1, \ldots, n-1. \tag{4.6}$$

The relations (4.6) can be written in the matrix form

$$\mathbb{A}M = g,$$

namely

$$\begin{pmatrix} 2 & \mu_1 & & & \\ \lambda_2 & 2 & \mu_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \lambda_{n-2} & 2 & \mu_{n-2} \\ & & & \lambda_{n-1} & 2 \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} g_1 - \lambda_1 M_0 \\ g_2 \\ \vdots \\ g_{n-2} \\ g_{n-1} - \mu_{n-1} M_n \end{pmatrix} \tag{4.7}$$

The matrix $\mathbb{A}$ is tri-diagonal, $a_{ii} = 2$, $i = 1, \dots, n-1$, $\lambda_i < 1$, $\mu_i < 1$ and $\lambda_i + \mu_i = 1$, hence $\mathbb{A}$ is strictly diagonally dominant, it is regular and there exists unique values $M_1, \dots, M_{n-1}$. Hence the cubic spline exists.

Tri-diagonal system can be solved efficiently, e.g., Gauss elimination (Thomas algorithm).

4.3.2 Interpolation error estimates

Theorem 4.7. *Let $f \in C^4[a,b]$. Then there exists constant $C > 0$ such that: Let $K > 0$ be a constant, let D be a partition of $[a,b]$ formed by $a = x_0 < \cdots < x_n = b$ satisfying condition*

$$\frac{\max h_i}{\min h_i} \le K, \tag{4.8}$$

where $h_i = x_{i+1} - x_i$. Let the cubic splines satisfies boundary conditions $\varphi''(x_0) = f''(x_0)$ and $\varphi''(x_n) = f''(x_n)$. Then

$$\left| f^{(k)}(x) - \varphi^{(k)}(x) \right| \le C K h^{4-k}, \quad x \in [a,b],\ k = 0,1,2,3,$$

where $h = \max h_i$. For $k = 3$ we consider the left- and the right-hand side derivatives.

Consequence 4.8. *If a sequence of partitions satisfies (4.8) such that $h \to 0$, then*

$$\varphi^{(k)} \rightrightarrows f^{(k)},\ k = 0,1,2,3.$$

Remark 4.9. *If we consider natural cubic spline $\varphi''(x_0) = 0$ and $\varphi''(x_n) = 0$, then*

$$|f(x) - \varphi(x)| \le C K h^2, \quad x \in [a,b],\ k = 0,1.$$

4.3.3 Cubic spline with a tension

In order to interpolate a singular function, we consider $\varphi \in C^2([a,b])$ and $\varphi|_{(x_i, x_{i+1})}$ are the solutions of

$$\varphi^{(4)} - \tau \varphi'' = 0,$$

where $\tau > 0$ is the tension parameter.
If $\tau = 0$ then φ is a cubic spline.
If $\tau \to \infty$ then φ is a linear function.

4.3.4 Hermit spline

Problem 4.10. *Let $a = x_0 < x_1 < \cdots < x_n = b$ and $y_i \in \mathbb{R},\ i = 0, \ldots, n$ be given, we seek $\varphi : C^1([a, b])$ such that*

- $\varphi|_{(x_{i-1}, x_i)}$ *is a cubic polynomial function on* (x_{i-1}, x_i), $i = 0, \ldots, n$,
- $\varphi(x_i) = f(x_i)$, $i = 0, \ldots, n$,
- $\varphi'(x_i) = f'(x_i)$, $i = 0, \ldots, n$.

We may use an approximation

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}.$$

We have

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} + O(h^2).$$

4.3.5 NURBS

Non-uniform rational basis spline (NURBS): the basis functions are not polynomials, but rational polynomial functions. They are widely used in practice (CAD).

Chapter 5

Numerical integration (1 week)

Some integrals can not be evaluated analytically, e.g.,

$$\operatorname{erf}(x) := \frac{2}{\pi} \int_0^x e^{-t^2} \, \mathrm{d}t$$

is the error function used in mathematical statistics. This integral can be evaluated only numerically.

Our aim is to evaluate $Q(f)$ such that

$$Q(f) \approx I(f) := \int_a^b f(x) \, \mathrm{d}x, \tag{5.1}$$

where f is an integrable function. We need

$$|Q(f) - I(f)|$$

small and the evaluations of $Q(f)$ should be fast (= a small number of mathematical operations).

Idea: approximate f by a (piecewise) polynomial function φ and integrate φ. It is not necessary to explicitly construct the approximation φ.

5.1 Newton-Cotes quadrature formula

Let $n \geq 1$ and $a = x_0 < x_1 < \cdots < x_n = b$ be a uniform partition of $[a, b]$, i.e.,

$$x_i = a + \frac{i}{n}(b - a), \qquad i = 0, \ldots, n. \tag{5.2}$$

We construct the Lagrange interpolation to f at x_i, $i = 0, \ldots, n$ and integrate over $[a, b]$, i.e.,

$$I(f) \approx Q(f) := \int_a^b \sum_{i=0}^{n} f(x_i) \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \, \mathrm{d}x = \sum_{i=0}^{n} f(x_i) \int_a^b \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \, \mathrm{d}x.$$

We call $Q(f)$ the numerical quadrature (or numerical quadrature rule) and usually write

$$Q(f) = \sum_{i=0}^{n} w_i f(x_i), \tag{5.3}$$

where $x_i,\ i = 0, \dots, n$ are the quadrature nodes and $w_i,\ i = 0, \dots, n$ are the quadrature weights. If the nodes are given by (5.2) and the weights by

$$w_i = \int_a^b \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \, \mathrm{d}x,$$

then we call $Q(f)$ the Newton-Cotes quadrature rule of degree n. Putting

$$\tilde{w}_i := \frac{w_i}{b-a}, \quad i = 0, \dots, n,$$

then it is possible to show that the weights $\tilde{w}_i$ are independent of a and b.

Example 5.1. *Newton-Cotes quadrature*

- $n = 1$ *trapezoid rule* $T(f) = (b-a)\frac{f(a)+f(b)}{2}$, *i.e.,* $\tilde{w}_0 = \tilde{w}_1 = 1/2$.
- $n = 2$ *Simpson rule* $S(f) = (b-a)\frac{f(a)+4f((a+b)/2)+f(b)}{6}$, *i.e.,* $\tilde{w}_0 = \tilde{w}_2 = 1/6$, $\tilde{w}_1 = 2/3$.

Definition 5.2. *We say that the quadrature $Q(f)$ has the order p if*

$$Q(g) = I(g) \qquad \forall g \in P^p([a,b]),$$

i.e., $Q(f)$ is exact for polynomials of degree p.

Determination of the Newton-Cotes for general n: We seek $w_0, \dots, w_n$ of the quadrature

$$Q(f) = w_0 f(x_0) + w_1 f(x_1) + \cdots + w_n f(x_n).$$

The quadrature Q should integrate polynomial functions exactly. Hence, we put $f := 1$, $f := x$, $f := x^2$, $\dots$, $f := x^n$

$$\begin{aligned}
&\int_a^b 1 \, \mathrm{d}x = b - a && \Rightarrow w_0 + w_1 + \cdots + w_n = b - a, \\
&\int_a^b x \, \mathrm{d}x = \frac{b^2 - a^2}{2} && \Rightarrow w_0 x_0 + w_1 x_1 + \cdots + w_n x_n = \frac{b^2 - a^2}{2}, \\
&\int_a^b x^2 \, \mathrm{d}x = \frac{b^3 - a^3}{3} && \Rightarrow w_0 x_0^2 + w_1 x_1^2 + \cdots + w_n x_n^2 = \frac{b^3 - a^3}{3}, \\
&\vdots \\
&\int_a^b x^n \, \mathrm{d}x = \frac{b^{n+1} - a^n + 1}{n} && \Rightarrow w_0 x_0^n + \cdots + w_n x_n^n = \frac{b^{n+1} - a^{n+1}}{n},
\end{aligned}$$

we obtain a linear algebraic system, which has to be solved. The weights $w_i,\ i = 0, \dots, n$ can be found in textbooks.

5.1.1 Error estimates

From (4.3), we have

$$f(x) - \varphi(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{j=0}^{n} (x - x_j), \quad \xi_x \in [a,b]. \tag{5.4}$$

- trapezoid rule

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x - \int_a^b \varphi_1(x)\,\mathrm{d}x =& \frac{1}{2}\int_a^b f''(\xi_x)(x-a)(x-b)\,\mathrm{d}x\\ =& \frac{1}{2}f''(\eta)\int_a^b (x-a)(x-b)\,\mathrm{d}x\\ =& -\frac{1}{12}f''(\eta)(b-a)^3.\end{aligned}$$

We use the mean value theorem in the integral form:
(If $f(x)$ is continuous and $g(x) \geq 0$, then $\int_a^b f(x)g(x)\,\mathrm{d}x = f(\eta)\int_a^b g(x)\,\mathrm{d}x$.)
Hence, if f is linear, then $f'' = 0$ and the trapezoid rule is exact.

- The Simpson rule: Let $m := \frac{a+b}{2}$. Let $f \in C^4([a,b])$, the Taylor expansion at m reads

$$\begin{aligned}f(x) =& f(m) + f'(m)(x-m) + \frac{1}{2}f''(m)(x-m)^2 + \frac{1}{6}f'''(m)(x-m)^3\\ &+ \frac{1}{24}f''''(m)(x-m)^4 + \dots.\end{aligned} \tag{5.5}$$

Integration of (5.5) over (a,b) gives (the "even" terms disappears)

$$I(f) = f(m)(b-a) + \frac{1}{24}f''(m)(b-a)^3 + \frac{1}{1920}f''''(m)(b-a)^5 + \dots. \tag{5.6}$$

Moreover, we put $x := a$ and $x := b$ in (5.5) and then we sum both relations, which gives (again the "even" terms disappears)

$$f(a) + f(b) = 2f(m) + \frac{2}{2}f''(m)\frac{(b-a)^2}{4} + \frac{2}{24}f''''(m)\frac{(b-a)^4}{16} + \dots. \tag{5.7}$$

Multiplying (5.7) by $(b-a)/6$ implies

$$\frac{f(a)+f(b)}{6}(b-a) = \frac{f(m)}{3}(b-a) + \frac{1}{24}f''(m)(b-a)^3 + \frac{1}{3\cdot 384}f''''(m)(b-a)^5 + \dots. \tag{5.8}$$

Finally, (5.6) − (5.8) gives

$$\begin{aligned}I(f) =& \frac{2}{3}f(m)(b-a) + \frac{f(a)+f(b)}{6}(b-a) + \left(\frac{1}{1920} - \frac{1}{3\cdot 384}\right)f''''(m)(b-a)^5 + \dots\\ \approx& S(f) - \frac{1}{2880}f''''(m)(b-a)^5.\end{aligned}$$

Hence, the Simpson rule has the order 3!

Hint for exercise: Repeat and do similar examples.
Hint for exercise: Compute some integrals, e.g., $\int_0^2 e^{-t^2}\,\mathrm{d}t$ and estimate the error.

Theorem 5.3. *The order of the Newton-Cotes quadrature is equal to:*
n for odd n (number of nodes is even)
$n+1$ for even n (number of nodes is odd).

Remark 5.4. *Few comments:*

- *In practice, at most* $n \leq 8$.
- *Too high* n *are unstable.*
- *The Newton-Cotes quadrature are closed formulae since* $x_0 = a$ *and* $x_n = b$. *They are problematic, e.g., for* $\int_0^1 1/\sqrt{x}\,\mathrm{d}x$, *where the integrand is not defined at the end-points.*

5.2 Gauss quadrature formulae

We consider again the rule of the type

$$Q(f) = \sum_{i=0}^{n} w_i f(x_i), \tag{5.9}$$

the weights $w_i,\ i = 0, \ldots, n$ and the nodes $x_i,\ i = 0, \ldots, n$ are chosen such that the order of the quadrature is the maximal one.

Example 5.5. $n = 0$, *hence* $Q(f) = w_0 f(x_0)$. *Therefore*

$$\begin{aligned} \int_a^b 1\,\mathrm{d}x = b - a = w_0 \cdot 1 \qquad &\Rightarrow \quad w_0 = b - a, \\ \int_a^b x\,\mathrm{d}x = \frac{b^2 - a^2}{2} = w_0 \cdot x_0 = (b-a)x_0 \qquad &\Rightarrow \quad x_0 = \frac{a+b}{2}. \end{aligned}$$

Example 5.6. $n = 1$, *it is possible to derive*

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = f(1/\sqrt{3}) + f(-1/\sqrt{3}),$$

as a result of a system of solving a system of nonlinear algebraic equations.

We have $2(n+1)$ degrees of freedom, we can expect the order $(2n+1)$.

For $n \geq 1$, we can use orthogonal polynomials:

Let $q_0(x),\ q_1(x),\ q_2(x),\ \ldots (q_i(x) \in P^i[a,b],\ i = 0, 1, \ldots)$ be a sequence of orthogonal polynomials with respect to the scalar product

$$\langle p, q \rangle := \int_a^b p(x)q(x)\,\mathrm{d}x.$$

(E.g., Gram-Schmidt algorithm.)

Example 5.7. *The Legendre polynomials*

$$\begin{aligned} L_0(x) &= 1, \\ L_1(x) &= x, \\ L_k(x) &= \frac{2k-1}{k} x L_{k-1}(x) - \frac{k-1}{k} L_{k-2}(x), \quad k = 2, 3, \ldots. \end{aligned}$$

forms the orthogonal basis on $(-1, 1)$.

Theorem 5.8. *If x_i, $i = 0, \dots, n$ are the roots of $q_{n+1}(x)$ (the $(n+1)$st orthogonal polynomial on $[a, b]$), then the formula*

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n} w_i f(x_i), \tag{5.10}$$

where

$$w_i = \int_a^b \phi_i(x)\,\mathrm{d}x, \quad \phi_i := \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, 1, \dots, n$$

is exact for polynomials of degree $2n + 1$.

Proof. Let f be a polynomial of degree $2n + 1$, we divide it by q_{n+1} and obtain

$$f(x) = q_{n+1}(x)p_n(x) + r_n(x),$$

where p_n and q_n are polynomials of degree n. Then $f(x_i) = r(x_i)$, $i = 0, \dots, n$. Integrating, we obtain

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b q_{n+1}(x)p_n(x)\,\mathrm{d}x + \int_a^b r_n(x)\,\mathrm{d}x = \int_a^b r_n(x)\,\mathrm{d}x,$$

since q_{n+1} and p_n are orthogonal. From the choice of w_i, the relation (5.10) is exact for polynomials of degree n, hence

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^b r_n(x)\,\mathrm{d}x = \sum_{i=0}^{n} w_i r_n(x_i) = \sum_{i=0}^{n} w_i f(x_i).$$

□

However, the task of finding the roots of the Legendre polynomial is not easy, but it can be found in many textbooks.

Show on the computer

G_k	j	w_j	x_j
G_1	1	1.00000000000000	0.50000000000000
G_2	1	0.50000000000000	0.21132486540519
	2	0.50000000000000	0.78867513459481
G_3	1	0.27777777777778	0.11270166537926
	2	0.44444444444444	0.50000000000000
	3	0.27777777777778	0.88729833462074
G_4	1	0.17392742256873	0.06943184420297
	2	0.32607257743127	0.33000947820757
	3	0.32607257743127	0.66999052179243
	4	0.17392742256873	0.93056815579703
G_5	1	0.11846344252809	0.04691007703067
	2	0.23931433524968	0.23076534494716
	3	0.28444444444444	0.50000000000000
	4	0.23931433524968	0.76923465505284
	5	0.11846344252809	0.95308992296933
G_6	1	0.08566224618959	0.03376524289842
	2	0.18038078652407	0.16939530676687
	3	0.23395696728635	0.38069040695840
	4	0.23395696728635	0.61930959304160
	5	0.18038078652407	0.83060469323313
	6	0.08566224618959	0.96623475710158

Hint for exercise: Derive two two-point (or three-point) Gauss quadrature using the Legendre polynomials, see [GC12].

5.3 Composite rules

- it makes no sense to use too high n
- the composite rules are better: Let $a = \xi_0 < \xi_1 < \cdots < \xi_N = b$ be a partition of $[a, b]$, then

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{N} \int_{\xi_{i-1}}^{\xi_i} f(x)\,\mathrm{d}x,$$

  so we apply the quadrature on each interval $[\xi_{i-1}, \xi_i]$.

Theorem 5.9. *Let Q be a quadrature rule of order p (i.e., it integrates the polynomials of degree p exactly). Let $\xi_k = a + hk,\ k = 0, \ldots, N$ with $h := (b-a)/N$ be a partition of $[a, b]$. Let Q_h be the corresponding composite rule. Let $f \in C^{p+1}([a, b])$, then there exists $c > 0$ such that*

$$|I(f) - Q_h(f)| \le c \max_{\chi\in[a,b]} f^{(p+1)}(\chi) h^{p+1}(b-a) + o(h^{p+1}). \tag{5.11}$$

Proof. **ONLY for the Newton-Cotes formulae with ODD number of nodes!** Let $k = 1, \ldots, r$. Then using Theorem 4.2 (the error of the Lagrangian interpolation), we have

$$f(x) = \varphi_p(x) + \frac{1}{(p+1)!} f^{(p+1)}(\chi_x) \prod_{j=0}^{p} (x - x_{k,j}), \quad x \in (\xi_{k-1}, \xi_k), \tag{5.12}$$

where $\chi_x \in (\xi_{k-1}, \xi_k)$, $\varphi_p(x)$ is the Lagrange interpolation at nodes $x_{k,j} := \xi_{k-1} + js$, $s = (\xi_k - \xi_{k-1})/(p+1)$, $j = 0, \ldots, p$ (p is even). We have

$$\int_{\xi_{k-1}}^{\xi_k} f(x)\,\mathrm{d}x \approx Q_h(f)|_{(\xi_{k-1},\xi_k)} := \int_{\xi_{k-1}}^{\xi_k} \varphi_p(x)\,\mathrm{d}x.$$

Then, from (5.12), we have

$$\begin{aligned} I(f)|_{(\xi_{k-1},\xi_k)} := \int_{\xi_{k-1}}^{\xi_k} f(x)\,\mathrm{d}x &= \int_{\xi_{k-1}}^{\xi_k} \varphi_p(x)\,\mathrm{d}x + \int_{\xi_{k-1}}^{\xi_k} \frac{1}{(p+1)!} f^{(p+1)}(\chi_x) \prod_{j=0}^{p} (x - x_{k,j})\,\mathrm{d}x \\ &= Q_h(f)|_{(\xi_{k-1},\xi_k)} + \int_{\xi_{k-1}}^{\xi_k} \frac{1}{(p+1)!} f^{(p+1)}(\chi_x) \prod_{j=0}^{p} (x - x_{k,j})\,\mathrm{d}x. \end{aligned}$$

Then

$$\left|(I(f) - Q_h(f))|_{(\xi_{k-1},\xi_k)}\right| \le \max_{\chi\in[a,b]} f^{(p+1)}(\chi) h^{p+1} \frac{\xi_k - \xi_{k-1}}{(p+1)!}. \tag{5.13}$$

Summing (5.13) over $k = 1, \ldots, N$, we have

$$|I(f) - Q_h(f)| = \sum_{k=1}^{N} \left|(I(f) - Q_h(f))|_{(\xi_{k-1},\xi_k)}\right| \le \max_{\chi\in[a,b]} f^{(p+1)}(\chi) h^{p+1} \frac{b-a}{(p+1)!},$$

since the sum contains $N = (b-a)/h$ terms. □

Remark 5.10. *If $Q(f)$ has order 3 (e.g., Simpson rule), then the half partition reduce the error 16-times.*

Example 5.11. *Evaluation of $\int_0^1 \exp(x)\,\mathrm{d}x = e - 1$, composite rules with $N = 2^m$ intervals, $R_h(f)$ is the true error :*

Trapezoid rule, $I(f) = 1.718281828459045$*:*

m	h	N	$T_h(f)$	$R_h(f)$	R_{2h}/R_h	*order*
0	*1.000000E+00*	*1*	*1.859140914229523E+00*	*1.408591E-01*	—	—
1	*5.000000E-01*	*2*	*1.753931092464825E+00*	*3.564926E-02*	*3.9512*	*1.9823*
2	*2.500000E-01*	*4*	*1.727221904557517E+00*	*8.940076E-03*	*3.9876*	*1.9955*
3	*1.250000E-01*	*8*	*1.720518592164302E+00*	*2.236764E-03*	*3.9969*	*1.9989*
4	*6.250000E-02*	*16*	*1.718841128579994E+00*	*5.593001E-04*	*3.9992*	*1.9997*
5	*3.125000E-02*	*32*	*1.718421660316327E+00*	*1.398319E-04*	*3.9998*	*1.9999*
6	*1.562500E-02*	*64*	*1.718316786850093E+00*	*3.495839E-05*	*4.0000*	*2.0000*
7	*7.812500E-03*	*128*	*1.718290568083479E+00*	*8.739624E-06*	*4.0000*	*2.0000*
8	*3.906250E-03*	*256*	*1.718284013366820E+00*	*2.184908E-06*	*4.0000*	*2.0000*
9	*1.953125E-03*	*512*	*1.718282374686094E+00*	*5.462270E-07*	*4.0000*	*2.0000*
10	*9.765625E-04*	*1024*	*1.718281965015814E+00*	*1.365568E-07*	*4.0000*	*2.0000*

Simpson rule, $I(f) = 1.718281828459045$*:*

m	h	N	$S_h(f)$	$R_h(f)$	R_{2h}/R_h	*order*
0	*1.000000E+00*	*1*	*1.718861151876593E+00*	*5.793234E-04*	—	—
1	*5.000000E-01*	*2*	*1.718318841921747E+00*	*3.701346E-05*	*15.6517*	*3.9682*
2	*2.500000E-01*	*4*	*1.718284154699897E+00*	*2.326241E-06*	*15.9113*	*3.9920*
3	*1.250000E-01*	*8*	*1.718281974051891E+00*	*1.455928E-07*	*15.9777*	*3.9980*
4	*6.250000E-02*	*16*	*1.718281837561772E+00*	*9.102727E-09*	*15.9944*	*3.9995*
5	*3.125000E-02*	*32*	*1.718281829028015E+00*	*5.689702E-10*	*15.9986*	*3.9999*
6	*1.562500E-02*	*64*	*1.718281828494606E+00*	*3.556089E-11*	*15.9999*	*4.0000*
7	*7.812500E-03*	*128*	*1.718281828461268E+00*	*2.223111E-12*	*15.9960*	*3.9996*
8	*3.906250E-03*	*256*	*1.718281828459185E+00*	*1.394440E-13*	*15.9427*	*3.9948*
9	*1.953125E-03*	*512*	*1.718281828459054E+00*	*8.881784E-15*	*15.7000*	*3.9727*
10	*9.765625E-04*	*1024*	*1.718281828459047E+00*	*1.776357E-15*	*5.0000*	*2.3219*

Gauss rule ($n = 1$), $I(f) = 1.718281828459045$*:*

m	h	N	$G_h(f)$	$R_h(f)$	R_{2h}/R_h	*order*
0	*1.000000E+00*	*1*	*1.717896378007504E+00*	*3.854505E-04*	—	—
1	*5.000000E-01*	*2*	*1.718257165052592E+00*	*2.466341E-05*	*15.6284*	*3.9661*
2	*2.500000E-01*	*4*	*1.718280277824108E+00*	*1.550635E-06*	*15.9054*	*3.9914*
3	*1.250000E-01*	*8*	*1.718281731400156E+00*	*9.705889E-08*	*15.9762*	*3.9979*
4	*6.250000E-02*	*16*	*1.718281822390608E+00*	*6.068437E-09*	*15.9940*	*3.9995*
5	*3.125000E-02*	*32*	*1.718281828079732E+00*	*3.793128E-10*	*15.9985*	*3.9999*
6	*1.562500E-02*	*64*	*1.718281828435338E+00*	*2.370726E-11*	*15.9999*	*4.0000*
7	*7.812500E-03*	*128*	*1.718281828457563E+00*	*1.481926E-12*	*15.9976*	*3.9998*
8	*3.906250E-03*	*256*	*1.718281828458953E+00*	*9.237056E-14*	*16.0433*	*4.0039*
9	*1.953125E-03*	*512*	*1.718281828459038E+00*	*7.327472E-15*	*12.6061*	*3.6560*
10	*9.765625E-04*	*1024*	*1.718281828459046E+00*	*1.332268E-15*	*5.5000*	*2.4594*

Gauss rule ($n = 6$), $I(f) = 1.718281828459045$*:*

m	h	N	$G_h(f)$	$R_h(f)$	R_{2h}/R_h	*order*
0	*1.000000E+00*	*1*	*1.718281828459045E+00*	*0.000000E+00*	—	—
1	*5.000000E-01*	*2*	*1.718281828459045E+00*	*0.000000E+00*	*0.0000*	*0.0000*
2	*2.500000E-01*	*4*	*1.718281828459045E+00*	*0.000000E+00*	*0.0000*	*0.0000*
3	*1.250000E-01*	*8*	*1.718281828459046E+00*	*4.440892E-16*	*0.0000*	*0.0000*
4	*6.250000E-02*	*16*	*1.718281828459045E+00*	*0.000000E+00*	*0.0000*	*0.0000*
5	*3.125000E-02*	*32*	*1.718281828459045E+00*	*0.000000E+00*	*0.0000*	*0.0000*
6	*1.562500E-02*	*64*	*1.718281828459045E+00*	*0.000000E+00*	*0.0000*	*0.0000*
7	*7.812500E-03*	*128*	*1.718281828459046E+00*	*4.440892E-16*	*0.0000*	*0.0000*
8	*3.906250E-03*	*256*	*1.718281828459045E+00*	*2.220446E-16*	*0.0000*	*0.0000*
9	*1.953125E-03*	*512*	*1.718281828459046E+00*	*6.661338E-16*	*0.0000*	*0.0000*
10	*9.765625E-04*	*1024*	*1.718281828459047E+00*	*1.554312E-15*	*0.0000*	*0.0000*

Example 5.12. *Evaluation of* $\int_0^1 \sqrt(x)\,\mathrm{d}x = \frac{2}{3}$*, composite rules with* $N = 2^m$ *intervals,* $R_h(f)$ *is the true error :*

Trapezoid rule, $I(f) = 0.666666666666667$*:*

n	h	N	$T_h(f)$	$R_h(f)$	$R_{h/2}/R_h$	*order*
0	*1.000000E+00*	*1*	*5.000000000000000E-01*	*1.666667E-01*	—	—
1	*5.000000E-01*	*2*	*6.035533905932737E-01*	*6.311328E-02*	*2.6408*	*1.4010*
2	*2.500000E-01*	*4*	*6.432830462427466E-01*	*2.338362E-02*	*2.6990*	*1.4324*
3	*1.250000E-01*	*8*	*6.581302216244542E-01*	*8.536445E-03*	*2.7393*	*1.4538*
4	*6.250000E-02*	*16*	*6.635811968772282E-01*	*3.085470E-03*	*2.7667*	*1.4681*
5	*3.125000E-02*	*32*	*6.655589362789417E-01*	*1.107730E-03*	*2.7854*	*1.4779*
6	*1.562500E-02*	*64*	*6.662708113785069E-01*	*3.958553E-04*	*2.7983*	*1.4846*
7	*7.812500E-03*	*128*	*6.665256572968257E-01*	*1.410094E-04*	*2.8073*	*1.4892*
8	*3.906250E-03*	*256*	*6.666165489765280E-01*	*5.011769E-05*	*2.8136*	*1.4924*
9	*1.953125E-03*	*512*	*6.666488815499515E-01*	*1.778512E-05*	*2.8180*	*1.4946*
10	*9.765625E-04*	*1024*	*6.666603622189838E-01*	*6.304448E-06*	*2.8210*	*1.4962*

Simpson rule, $I(f) = 0.666666666666667$*:*

n	h	N	$S_h(f)$	$R_h(f)$	$R_{h/2}/R_h$	*order*
0	*1.000000E+00*	*1*	*6.380711874576983E-01*	*2.859548E-02*	—	—
1	*5.000000E-01*	*2*	*6.565262647925707E-01*	*1.014040E-02*	*2.8200*	*1.4957*
2	*2.500000E-01*	*4*	*6.630792800850236E-01*	*3.587387E-03*	*2.8267*	*1.4991*
3	*1.250000E-01*	*8*	*6.653981886281528E-01*	*1.268478E-03*	*2.8281*	*1.4998*
4	*6.250000E-02*	*16*	*6.662181827461796E-01*	*4.484839E-04*	*2.8284*	*1.5000*
5	*3.125000E-02*	*32*	*6.665081030783619E-01*	*1.585636E-04*	*2.8284*	*1.5000*
6	*1.562500E-02*	*64*	*6.666106059362655E-01*	*5.606073E-05*	*2.8284*	*1.5000*
7	*7.812500E-03*	*128*	*6.666468462030957E-01*	*1.982046E-05*	*2.8284*	*1.5000*
8	*3.906250E-03*	*256*	*6.666596590744270E-01*	*7.007592E-06*	*2.8284*	*1.5000*
9	*1.953125E-03*	*512*	*6.666641891086617E-01*	*2.477558E-06*	*2.8284*	*1.5000*
10	*9.765625E-04*	*1024*	*6.666657907176324E-01*	*8.759490E-07*	*2.8284*	*1.5000*

Gauss rule ($n = 1$), $I(f) = 0.666666666666667$*:*

n	h	N	$G_h(f)$	$R_h(f)$	$R_{h/2}/R_h$	*order*
0	*1.000000E+00*	*1*	*6.738873386790492E-01*	*7.220672E-03*	—	—
1	*5.000000E-01*	*2*	*6.692395023997495E-01*	*2.572836E-03*	*2.8065*	*1.4888*
2	*2.500000E-01*	*4*	*6.675777701535970E-01*	*9.111035E-04*	*2.8239*	*1.4977*
3	*1.250000E-01*	*8*	*6.669888871745580E-01*	*3.222205E-04*	*2.8276*	*1.4996*
4	*6.250000E-02*	*16*	*6.667805949572163E-01*	*1.139283E-04*	*2.8283*	*1.4999*
5	*3.125000E-02*	*32*	*6.667069467851046E-01*	*4.028012E-05*	*2.8284*	*1.5000*
6	*1.562500E-02*	*64*	*6.666809078632009E-01*	*1.424120E-05*	*2.8284*	*1.5000*
7	*7.812500E-03*	*128*	*6.666717016914930E-01*	*5.035025E-06*	*2.8284*	*1.5000*
8	*3.906250E-03*	*256*	*6.666684468168600E-01*	*1.780150E-06*	*2.8284*	*1.5000*
9	*1.953125E-03*	*512*	*6.666672960448092E-01*	*6.293781E-07*	*2.8284*	*1.5000*
10	*9.765625E-04*	*1024*	*6.666668891854427E-01*	*2.225188E-07*	*2.8284*	*1.5000*

Gauss rule ($n = 6$), $I(f) = 0.666666666666667$*:*

n	h	N	$G_h(f)$	$R_h(f)$	$R_{h/2}/R_h$	*order*
0	*1.000000E+00*	*1*	*6.669130850887391E-01*	*2.464184E-04*	—	—
1	*5.000000E-01*	*2*	*6.667537887353612E-01*	*8.712207E-05*	*2.8284*	*1.5000*
2	*2.500000E-01*	*4*	*6.666974689694490E-01*	*3.080230E-05*	*2.8284*	*1.5000*
3	*1.250000E-01*	*8*	*6.666775569252534E-01*	*1.089026E-05*	*2.8284*	*1.5000*
4	*6.250000E-02*	*16*	*6.666705169545143E-01*	*3.850288E-06*	*2.8284*	*1.5000*
5	*3.125000E-02*	*32*	*6.666680279489899E-01*	*1.361282E-06*	*2.8284*	*1.5000*
6	*1.562500E-02*	*64*	*6.666671479526478E-01*	*4.812860E-07*	*2.8284*	*1.5000*
7	*7.812500E-03*	*128*	*6.666668368269568E-01*	*1.701603E-07*	*2.8284*	*1.5000*
8	*3.906250E-03*	*256*	*6.666667268274141E-01*	*6.016075E-08*	*2.8284*	*1.5000*
9	*1.953125E-03*	*512*	*6.666666879367035E-01*	*2.127004E-08*	*2.8284*	*1.5000*
10	*9.765625E-04*	*1024*	*6.666666741867594E-01*	*7.520093E-09*	*2.8284*	*1.5000*

Hint for exercise: Examples verifying the order of convergence for both types of quadrature rules, regular and singular functions.

5.4 Half-step size method

- How can we evaluate $|I(f) - Q_h(f)|$?
- In Theorem 5.9, we do not know $f^{(p+1)}$.
- We use the half-step size method.

We assume that

$$I(f) \approx Q_h(f) + Ch^{p+1}, \tag{5.14}$$

where C is an unknown constant (depending on f), i.e., **the order of the method is** p, compare with (5.11). Repeating the computation with $h/2$ we get

$$I(f) \approx Q_{h/2}(f) + C\frac{h^{p+1}}{2^{p+1}}.$$

Then, subtracting this relations, we have

$$Q_{h/2}(f) - Q_h(f) \approx \left(1 - \frac{1}{2^{p+1}}\right) Ch^{p+1} = \left(2^{p+1} - 1\right) C \left(\frac{h}{2}\right)^{p+1} \approx \left(2^{p+1} - 1\right) \left(I(f) - Q_{h/2}(f)\right)$$

$$\Rightarrow I(f) - Q_{h/2}(f) \approx \frac{Q_{h/2}(f) - Q_h(f)}{2^{p+1} - 1}.$$

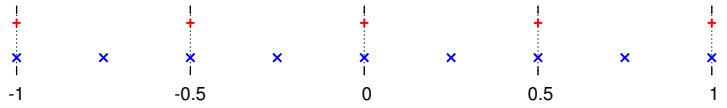
We carry out computation two-times and from the difference we estimate the error.

Example 5.13. *Evaluation of* $\int_0^1 \exp(x)\,\mathrm{d}x = e - 1$, *composite rules with* $N = 2^n$ *intervals,* $R_h(f)$ *is the true error, "estim" is the error estimate by the half-step size method:*

Simpson rule:

n	h	N	$S_h(f)$	$R_h(f)$	*estim*
0	*1.000000E+00*	*1*	*1.718861151876593E+00*	*5.793234E-04*	—
1	*5.000000E-01*	*2*	*1.718318841921747E+00*	*3.701346E-05*	*3.615400E-05*
2	*2.500000E-01*	*4*	*1.718284154699897E+00*	*2.326241E-06*	*2.312481E-06*
3	*1.250000E-01*	*8*	*1.718281974051891E+00*	*1.455928E-07*	*1.453765E-07*
4	*6.250000E-02*	*16*	*1.718281837561772E+00*	*9.102727E-09*	*9.099341E-09*
5	*3.125000E-02*	*32*	*1.718281829028015E+00*	*5.689702E-10*	*5.689171E-10*
6	*1.562500E-02*	*64*	*1.718281828494606E+00*	*3.556089E-11*	*3.556062E-11*
7	*7.812500E-03*	*128*	*1.718281828461268E+00*	*2.223111E-12*	*2.222518E-12*
8	*3.906250E-03*	*256*	*1.718281828459185E+00*	*1.394440E-13*	*1.389111E-13*
9	*1.953125E-03*	*512*	*1.718281828459054E+00*	*8.881784E-15*	*8.704149E-15*
10	*9.765625E-04*	*1024*	*1.718281828459047E+00*	*1.776357E-15*	*4.736952E-16*

The Newton-Cotes formulae are suitable for the half-step size method:



Hint for exercise: Show examples comparing the computational error with its estimate using the half-step size method.

Chapter 6

Numerical solution of ODE (2 weeks)

Let $n \geq 1$, we consider ordinary differential equation (ODE)

$$\begin{aligned} y'(t) &= f(t, y(t)), \qquad t \in (a, b) \\ y(a) &= \eta, \end{aligned} \tag{6.1}$$

where $y : [a, b] \to \mathbb{R}^n$, $f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ and $\eta \in \mathbb{R}^n$. The problem is called initial-value problem, η is called the initial condition.

Example 6.1. *The growth of a population. The increase is proportional to the size of the population*

$$y' = ky, \quad k > 0 \qquad \Longrightarrow \qquad y(t) = e^{kt}.$$

Unrealistic model.

Example 6.2. **Draw figure** *Hook's law: acceleration of an object on a spring is proportional to the distance of the object from the equilibrium:*

$$y''(t) = -ky(t), \qquad k > 0.$$

Equivalent to

$$\begin{aligned} y'(t) &= z(t) \\ z'(t) &= -ky(t), \end{aligned}$$

solution $y(t) = c_1 \sin(\sqrt{k}t) + c_2 \cos(\sqrt{k}t)$.

Example 6.3. *The use of the numerical solution of ODE:*

- *CFD (computational fluid dynamics)*
- *animation "Star Wars"*

The existence and uniqueness of the solution of (6.1) follows from the Picard theorem (if f is Lipschitz continuous with respect to y).

We need also

Definition 6.4. *The system* (6.1) *is well-posed (or stable), if the solution of* (6.1) *depends continuously on the data (i.e., initial condition η). This means that*

$$\begin{aligned} y'(t) &= f(t, y(t)), \quad y(a) = \eta \\ z'(t) &= f(t, z(t)), \quad z(a) = \eta + \delta, \quad \delta \in \mathbb{R}^n, \end{aligned}$$

then

$$|y(t) - z(t)| \leq \delta C(t),$$

where $C(t)$ is an (exponentially increasing function) of t, but independent of δ.

- This stability is called the zero-stability. If f is Lipschitz continuous, then (6.1) is zero-stable.
- The stability is a key for numerical solution of (not only) ODE. Discretization and rounding errors cause some inaccuracy, they should be under control.

6.1 Basic idea of numerical solution of ODE

- we define a partition of (a, b) : $a = t_0 < t_1 < \ldots, t_r = b$,
- we approximate $y(t_k)$ by y_k, $k = 0, \ldots, r$
- we derive some formulas **Draw figure**
  - $y_{k+1} = F(t_{k+1}, t_k, y_k)$ – one step method
  - $y_{k+1} = F(t_{k+1}, t_k, \ldots, t_m; y_k, y_{k-1}, \ldots, y_m)$ – multi-step method (m-step method)
- possibly, we reconstruct function $\tilde{y}$ from $[t_0, y_0], \ldots, [t_r, y_r]$ by an interpolation

Always, y_k depends on all y_i, $i = 0, 1, \ldots, k-1$, different from numerical integration.

6.2 Examples of numerical methods

6.2.1 The Euler method

Let $t_0, \ldots, t_r$ be a uniform partition of $[a, b]$ and $h := t_{k+1} - t_k$, $k = 0, \ldots, r-1$. Let $y \in C^2([a, b])$, Taylor at t_k:

$$y(t_{k+1}) = y(t_k) + hy'(t_k) + \frac{1}{2}h^2 y''(\xi_k), \qquad \xi_k \in [t_k, t_{k+1}].$$

Omitting the last term, using $y'(t_k) = f(t_k, y_k)$ and $y_k \approx y(t_k)$, $k = 0, \ldots, r$, we obtain the Euler method

$$\begin{aligned} y_{k+1} &= y_k + hf(t_k, y_k), \qquad k = 0, 1, \ldots, r-1, \\ y_0 &= \eta. \end{aligned} \tag{6.2}$$

We may write

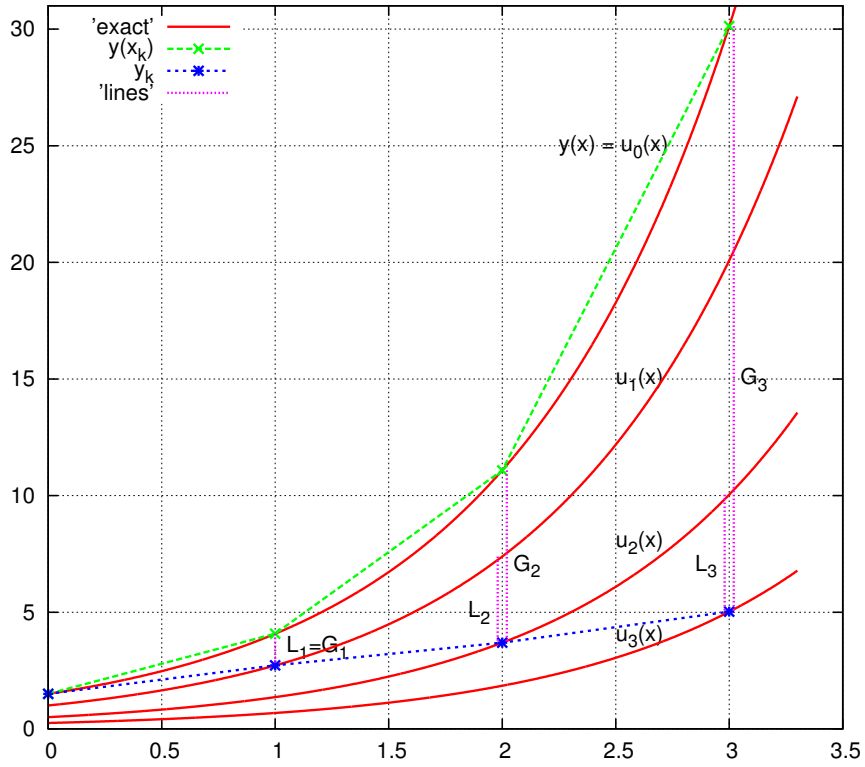
$$\begin{aligned}\frac{y(t_{k+1})-y(t_k)}{h} &= f(t_k, y(t_k)) + \frac{1}{2}hy''(\xi_k),\\ \frac{y_{k+1}-y_k}{h} &= f(t_k, y_k),\end{aligned}$$

the omitted term $\frac{1}{2}hy''(\xi_k) =: L_k$ corresponds to the local discretization (or the local truncation) error (see below), it is $O(h)$, so the Euler method is a first order method.

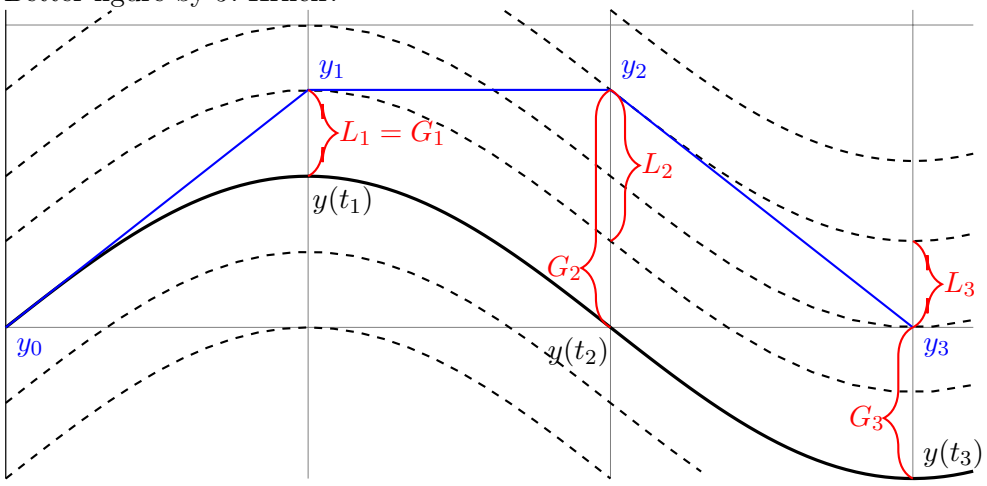
The global error

$$G_k := y(t_k) - y_k, \qquad k = 0, 1, \ldots, r.$$

Remark 6.5. $\sum_{k=0}^{N} L_k \neq G_N$.



Better figure by J. Hrnčíř:



6.2.2 Midpoint formula

Using the relations

$$\begin{aligned} y'(t_k + \frac{h}{2}) =& f(t_k + \frac{h}{2}, y(t_k + \frac{h}{2})), \\ y(t_k + \frac{h}{2}) =& y(t_k) + \frac{h}{2} y'(t_k) + O(h^2), \\ y'(t_k) =& f(t_k, y(t_k)), \end{aligned}$$

we get

$$\left(y'(t_k + \frac{h}{2}) \approx \right) \qquad \frac{y_{k+1} - y_k}{h} = f(t_k + \frac{h}{2}, y_k + \frac{h}{2} f(t_k, y_k)).$$

6.2.3 Heun's method

Integration of (6.1) over $(t, t+h)$ yields

$$y(t+h) = y(t) + \int_t^{t+h} f(s, y(s)) \, \mathrm{d}s.$$

We approximate the integral by the trapezoid rule and put

$$y_{k+1} = y_k + \frac{h}{2} \left(f(t_k, y_k) + f(t_{k+1}, y_{k+1}) \right).$$

This is an implicit method, we can use an approximation (by the Euler method)

$$f(t_{k+1}, y_{k+1}) \approx y_k + h f(t_k, y_k),$$

which gives the Heun's method

This can be rewritten in more usual form

$$\begin{aligned} y_0 =& \eta, \\ y_{k+1} =& y_k + h(q_1 + q_2), \\ q_1 =& \frac{1}{2} f(t_k, y_k), \\ q_2 =& \frac{1}{2} f(t_k + h, y_k + 2hq_1). \end{aligned}$$

It is a second order Runge-Kutta method, see bellow.

6.2.4 Two-step method

Let $y \in C^3([a,b])$, Taylor:

$$\begin{aligned} y(t_{k+1}) =& y(t_k) + hy'(t_k) + \frac{1}{2}h^2 y''(t_k) + O(h^3), \\ y(t_{k+2}) =& y(t_k) + 2hy'(t_k) + 2h^2 y''(t_k) + O(h^3). \end{aligned}$$

The -4 multiple of the first relation added to the second equation gives

$$y(t_{k+2}) - 4y(t_{k+1}) = -3y(t_k) - 2hy'(t_k) + O(h^3),$$

which allows us to define

$$y_{k+2} - 4y_{k+1} + 3y_k = -2hf(t_k, y_k)$$

and $y_0 = \eta$. Here y_1 has to be computed by a one-step method.

- it is a second order method,
- more economical than the Runge-Kutta method
- a little less useful.

6.3 Analysis of a one-step methods

A general one-step method can be written in the form

$$\begin{aligned} y_{k+1} =& y_k + h\psi(t_k, y_k, h), \qquad k = 0, 1, \dots \\ y_0 =& \eta, \end{aligned} \tag{6.3}$$

where $\psi : [a,b] \times \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}$ is the relative incremental function.

Example 6.6. *The form of ψ:*

- *the Euler method:* $\psi(t,y,h) = f(t,y)$
- *Heun's method:* $\psi(t,y,h) = \frac{1}{2}\left[f(t,y) + f(t+h, y+hf(t,y))\right]$

Relation (6.3) implies

$$\frac{y_{k+1} - y_k}{h} = \psi(t_k, y_k, h), \qquad k = 0, 1, \dots \tag{6.4}$$

Definition 6.7. *The one-step method* (6.3) *is consistent if* $\lim_{h\to 0} \psi(t,y,h) = f(t,y)$.

Example 6.8. *The Euler method is consistent.*

Hint for exercise: Proof of the consistency of the midpoint formula?

Definition 6.9. *The local truncation error is given by*

$$\tau(t,h) := \frac{y(t+h)-y(t)}{h} - \psi(t,y,h).$$

The ratio $\frac{y(t+h)-y(t)}{h}$ *is called the exact relative increment.*

Lemma 6.10. *If* $\tau(t,h) \to 0$ *for* $h \to 0$ *for all* $t \in [a,b]$*, then* (6.3) *is consistent.*

Definition 6.11. *The one-step method* (6.3) *is zero-stable, if the numerical solution of* (6.1) *depends continuously on the data (i.e., initial condition* η*). This means that*

$$\begin{aligned} y_{k+1} &= y_k + h\psi(t_k, y_k, h), \quad k = 0, 1, \ldots, \qquad y_0 = \eta, \\ z_{k+1} &= z_k + h\psi(t_k, z_k, h), \quad k = 0, 1, \ldots, \qquad z_0 = \eta + \delta, \end{aligned}$$

then

$$|y_k - z_k| \le \delta \tilde{C}(t_k),$$

where $\tilde{C}(t)$ *is an (exponentially increasing function) of* t*, but independent of* δ*.*

Lemma 6.12. *If* ψ *is Lipschitz continuous, then the one-step method* (6.3) *is zero-stable.*

Definition 6.13. *The method is convergent, if* $G_k \to 0$ *for* $h \to 0$*,*
i.e., $\max_{k;a+kh\le b} |y_k - y(t_k)| \to 0$ *for* $h \to 0$*.*

Theorem 6.14. *Let us consider the one-step method* (6.3)*. Let function* ψ *be Lipschitz continuous in* y*: exists* $L > 0$

$$|\psi(t,y,h) - \psi(t,\tilde{y},h)| \le L|y-\tilde{y}| \qquad \forall y, \tilde{y} \in \mathbb{R}^n,\ t \in [a,b],\ h \in (0,h_0],$$

where $h_0 > 0$ *is the given maximal time step and* $|\tau(t,h)| \le Ch^p$ *(i.e, the one-step method* (6.3) *is consistent and zero-stable). Then the global error is bounded by*

$$\max_{k;a+kh\le b} |y_k - y(t_k)| \le Ch^p \frac{e^{L(b-a)}-1}{L}, \quad h \in (0,h_0], \tag{6.5}$$

Proof. We have

$$\begin{aligned} y_{k+1} &= y_k + h\psi(t_k, y_k, h), \\ y(t_{k+1}) &= y(t_k) + h\psi(t_k, y(t_k), h) + h\tau(t_k, h). \end{aligned}$$

Subtracting them and putting $G_k := y(t_k) - y_k$ we have

$$G_{k+1} = G_k + h[\psi(t_k, y(t_k), h) - \psi(t_k, y_k, h)] + h\tau(t_k, h).$$

The use of the Lipschitz continuity gives

$$|G_{k+1}| \le |G_k| + hL|G_k| + h|\tau(t_k,h)| \le (1+hL)|G_k| + Ch^{p+1}.$$

Hence, using the same estimates for G_k, G_{k-1}, etc., we have

$$\begin{aligned}
|G_{k+1}| \leq & (1+hL)|G_k| + Ch^{p+1} \\
\leq & (1+hL)((1+hL)|G_{k-1}| + Ch^{p+1}) + Ch^{p+1} \\
= & (1+hL)^2|G_{k-1}| + Ch^{p+1}\sum_{j=0}^{1}(1+hL)^j \\
& \vdots \\
\leq & (1+hL)^{k+1}|G_0| + Ch^{p+1}\sum_{j=0}^{k}(1+hL)^j.
\end{aligned}$$

In our case $G_0 = 0$. Further, summing the geometric series

$$|G_{k+1}| \leq Ch^{p+1}\frac{(1+hL)^{k+1}-1}{hL} = Ch^p\frac{(1+hL)^{k+1}-1}{L}.$$

Using the fact that $(1+hL)^{k+1} \leq e^{(k+1)hL}$ and $kh \leq b-a$ for all $k = 1, 2, \ldots$, we have

$$|G_{k+1}| \leq Ch^p\frac{e^{(k+1)hL}-1}{L} \leq Ch^p\frac{e^{(b-a)L}-1}{L}.$$

□

The assertion of Theorem 6.14 says:

- method (6.3) converges for $h \to 0$,
- order of convergence is $O(h^p)$ the same as the local truncation error $\tau(t,h)$.
- the constant in the estimate exponentially growths.

A generalization of Theorem 6.14:

Theorem 6.15. *If* (6.3) *is zero-stable and consistent then it is convergence.*

Example 6.16. *Let us consider*

$$\begin{aligned}
y'(t) = & -100y + 100t + 101, \\
y(0) = & 1.
\end{aligned}$$

The exact solution is $y(t) = 1 + t$. *The Euler method*

$$\begin{aligned}
y_0 &= 1, \\
y_{k+1} &= y_k + h\left(-100y_k + 100hk + 101\right), \quad k = 0, 1, \ldots.
\end{aligned}$$

diverges for $h = 0.1$, *see Example 1.11.*

It is in a contradiction with (6.14)? NO.
The zero-stability is not enough for the computations with fixed h.

6.3.1 A-Stability of the Euler method

Let $f(t,y) \in C^1$. For the Euler method, we have

$$\begin{aligned} y_{k+1} =& y_k + hf(t_k, y_k), \\ y(t_{k+1}) =& y(t_k) + hf(t_k, y(t_k)) + h\tau(t_k, h). \end{aligned}$$

Subtracting them and putting $G_k := y(t_k) - y_k$ we have

$$G_{k+1} = G_k + h[f(t_k, y(t_k)) - f(t_k, y_k)] + h\tau(t_k, h).$$

The use of the mean value theorem gives: $\exists\zeta$ such that

$$f(t_k, y(t_k)) - f(t_k, y_k) = \frac{\partial f}{\partial y}(t_k, \zeta)[y(t_k)) - y_k] =: f'(\zeta)G_k.$$

Hence,

$$G_{k+1} = G_k + hf'(\zeta)G_k + h\tau(t_k, h).$$

and thus

$$|G_{k+1}| \leq \underbrace{|1 + hf'||G_k|}_{\text{propagation of err.}} + \underbrace{|h\tau(t_k, h)|}_{\text{local error}}.$$

The term $A := 1 + h_k f'$ is called the amplification factor.

Definition 6.17. *A numerical method is absolute stable (or A-stable) if the magnitude of the amplification factor is strictly less than* 1.

This means that for a stable numerical method, the propagation of the errors from previous time step is limited. Therefore, the rounding errors do not destroy the approximate solution.

For the explicit Euler equation, we have the stability condition

$$|1 + hf'| < 1,$$

which means that

- $f' < 0$, we say that the problem (6.1) is A-stable,
- $hf' \in (-2, 0) \Rightarrow h_k < -2/f'$.

6.4 Construction of numerical methods for ODE

6.4.1 Method based on the Taylor expansion

Let $y \in C^{p+1}$. Then the Taylor

$$y(t+h) = y(t) + \sum_{i=1}^{p} \frac{y^{(i)}(t)}{i!} h^i + \frac{y^{(p+1)}(\tilde{t})}{(p+1)!} h^{p+1}$$

and hence

$$\frac{y(t+h)-y(t)}{h} = \sum_{i=1}^{p} \frac{y^{(i)}(t)}{i!} h^{i-1} + \frac{y^{(p+1)}(\tilde{t})}{(p+1)!} h^p. \tag{6.6}$$

Now, we have

$$\begin{aligned}
y'(t) =& f(t, y(t)), \\
y''(t) =& \frac{\mathrm{d}}{\mathrm{d}t} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + y'(t) \frac{\partial f}{\partial y}(t, y(t)) \\
=& \frac{\partial f}{\partial t}(t, y(t)) + f(t, y(t)) \frac{\partial f}{\partial y}(t, y(t)), \\
y'''(t) =& \frac{\mathrm{d}}{\mathrm{d}t} \left[\left(\frac{\partial f}{\partial t} + f \frac{\partial f}{\partial y} \right) (t, y(t)) \right] = \dots
\end{aligned}$$

Defining the differential operator

$$D\varphi := \frac{\partial \varphi}{\partial t} + \varphi \frac{\partial \varphi}{\partial y},$$

and

$$D^0 \varphi = \varphi, \qquad D^{i+1} \varphi = D(D^i \varphi), \ i = 0, 1, \dots,$$

we find that

$$\begin{aligned}
y'(t) =& (D^0 f)(t, y(t)), \\
y''(t) =& (D^1 f)(t, y(t)), \\
& \vdots \\
y^{(i)}(t) =& (D^{i-1} f)(t, y(t)).
\end{aligned}$$

Then (6.6), can be written

$$\frac{y(t+h)-y(t)}{h} = \sum_{i=1}^{p} \frac{(D^{i-1} f)(t, y(t))}{i!} h^{i-1} + \frac{y^{(p+1)}(\tilde{t})}{(p+1)!} h^p.$$

Hence, we define one-step method (6.3) with

$$\psi(t, y, h) := \sum_{i=1}^{p} \frac{(D^{i-1} f)(t, y(t))}{i!} h^{i-1},$$

the truncation error is

$$\tau(t, h) = \frac{y^{(p+1)}(\tilde{t})}{(p+1)!} h^p,$$

i.e., the p^{th}-order method.

However, the evaluation of $D^i f$ is expensive.

6.4.2 Runge-Kutta methods

Idea is to define the relative incremental function ψ by

$$\psi(t,y,h) = \sum_{i=1}^{s} w_i q_i(t,y,h),$$

where

$$\begin{aligned}
q_1(t,y,h) =& f(t,\ y),\\
q_2(t,y,h) =& f(t+\alpha_2 h,\ y+\beta_{21} h q_1(t,y,h)),\\
&\vdots\\
q_i(t,y,h) =& f\left(t+\alpha_i h,\ y+h\sum_{j=1}^{i-1}\beta_{ij} q_j(t,y,h)\right), \quad i=2,\dots,s.
\end{aligned}$$

The values $s \in \mathbb{N}$, $w_i \in \mathbb{R},\ i=1,\dots,s$, $\alpha_i \in \mathbb{R},\ i=2,\dots,s$ and $\beta_{ij} \in \mathbb{R},\ j=1,\dots,i-1,\ i=1,\dots,s$ have to be suitably chosen. Sometimes, we call the *s*-stage method. The increase of the accuracy is obtained by "intermediate states".

2-stage Runge-Kutta method

Let

$$\tilde{\psi}(t,y,h) := \sum_{i=1}^{2} \frac{(D^{i-1}f)(t,y(t)}{i!} h^{i-1},$$

be the relative incremental function for the method based on the Taylor expansion. If we derive the Runge-Kutta method with

$$\psi(t,y,h) = \sum_{i=1}^{s} w_i q_i(t,y,h),$$

such that

$$\psi(t,y,h) - \tilde{\psi}(t,y,h) = O(h^2),$$

then the resulting Runge-Kutta method has order 2. Hence,

$$\psi(t,y,h) = w_1 f(t,y) + w_2 f(t+\alpha_2 h, y+\beta_{21} h f(t,y)), \tag{6.7}$$

$$\begin{aligned}
\tilde{\psi}(t,y,h) =& f(t,y) + \frac{h}{2}(Df)(t,y) \qquad (6.8)\\
=& f(t,y) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t,y) + f(t,y)\frac{\partial f}{\partial y}(t,y)\right).
\end{aligned}$$

The Taylor expansion for a function of several variables:

$$\begin{aligned}
& f(t+\alpha_2 h, y+\beta_{21} h f(t,y) \qquad (6.9)\\
=& f(t,y) + \frac{\partial f(t,y)}{\partial t}\alpha_2 h + \frac{\partial f(t,y)}{\partial y}\beta_{21} h f(t,y) + O(h^2).
\end{aligned}$$

From (6.7) and (6.9), we have

$$\psi(t,y,h) = (w_1+w_2)f(t,y) + w_2\frac{\partial f(t,y)}{\partial t}\alpha_2 h + w_2\frac{\partial f(t,y)}{\partial y}\beta_{21}hf(t,y) + O(h^2). \qquad (6.10)$$

Comparing (6.8) and (6.10), we have the relations

$$\begin{aligned} w_1 + w_2 &= 1, \\ w_2\alpha_2 &= \frac{1}{2}, \\ w_2\beta_{21} &= \frac{1}{2}. \end{aligned}$$

We have 3 equations for 4 unknowns, we put $w_2 := \gamma \neq 0$, then the choice

$$w_1 = 1-\gamma, \qquad w_2 = \gamma, \qquad \alpha_2 = 1/(2\gamma), \qquad \beta_{21} = 1/(2\gamma)$$

leads to the second order Runge-Kutta method. In practice, one uses $\gamma = 1$, $\gamma = 3/4$ and $\gamma = 1/2$.

Remark 6.18. *It is possible to derive the Runge-Kutta method of order $p = s$ for $s \leq 4$. In order to have the method of order 5, we need $s \geq 6$. Hence, the fourth order Runge-Kutta methods are the most used ones.*

Hint for exercise: Derive third order Runge-Kutta method.

6.5 Error estimates by the half-size method

- How large is the discretization error?
- Theorem 6.14 gives

$$|y_k - y(t_k)| \leq Ch^p \frac{e^{L(b-a)}-1}{L}. \qquad (6.11)$$

  which over-estimates the error. It takes into account the worst-case scenario.

Asymptotic error estimate by (6.11)

Euler method: $y_{n+1} = y_n + hf(x_n, y_n)$, $h = 2^{-6} = 0.015625$.

ODE	$y' = y$, $y(0) = 1$			$y' = -y$, $y(0) = 1$		
exact	$y(x) = \exp(x)$			$y(x) = \exp(-x)$		
x_n	y_n	$e_n = y_n - y(x_n)$	estim e_n	y_n	$e_n = y_n - y(x_n)$	estim e_n
1.0	2.69735	-0.02093	0.03649	0.364987	-0.002892	0.013424
2.0	7.27567	-0.11339	0.36882	0.133215	-0.002120	0.049914
3.0	19.62499	-0.46055	2.99487	0.048622	-0.001165	0.149016
4.0	52.93537	-1.66278	22.86218	0.017746	-0.000570	0.418735
5.0	142.7850	-5.6282	170.9223	0.006477	-0.000261	1.151666
	error $\approx 4\%$	**40x over-estimated**		**error** $\approx 4\%$	**10 000x over-estim.**	

We assume that

$$y_k^{(h)} - y(t_k^{(h)}) \approx \tilde{C} h^p.$$

We take the partition with $h/2$, then $t_k^{(h)} = t_{2k}^{(h/2)}$ and

$$y_{2k}^{(h/2)} - y(t_{2k}^{(h/2)}) \approx \tilde{C}(h/2)^p.$$

The subtraction gives

$$y_{2k}^{(h/2)} - y_k^{(h)} \approx \tilde{C}(h/2)^p(1-2^p) \qquad \Rightarrow \qquad \tilde{C}(h/2)^p \approx \frac{y_k^{(h)} - y_{2k}^{(h/2)}}{2^p - 1},$$

hence

$$|y_{2k}^{(h/2)} - y(t_{2k}^{(h/2)})| \approx \frac{|y_k^{(h)} - y_{2k}^{(h/2)}|}{2^p - 1}.$$

The error estimate by the the half-size method.

Error estimate by the the half-size method

ODE: $y' = 1 - y^2$, $y(0) = 5$, 4th Runge-Kutta 4. $h = 0.04$

x_n	y_n	error $y_n - y(x_n)$	estim
0.00	5.000000	0.0E+00	0.0E+00
0.04	4.200388	3.3E-05	
0.08	3.630695	3.8E-05	2.4E-05
0.12	3.205414	3.5E-05	
0.16	2.876746	3.1E-05	2.2E-05
0.20	2.615879	2.7E-05	
0.24	2.404407	2.3E-05	1.7E-05
0.28	2.230026	2.1E-05	
0.32	2.084192	1.8E-05	1.3E-05
0.36	1.960791	1.5E-05	
⋮			
0.64	1.455073	0.6E-05	0.5E-05
0.68	1.412863	0.6E-05	
0.72	1.375166	0.5E-05	0.4E-05
0.76	1.341398	0.5E-05	
0.80	1.311068	0.4E-05	0.3E-05
0.84	1.283759	0.4E-05	
0.88	1.259116	0.4E-05	0.3E-05
0.92	1.236835	0.3E-05	
0.96	1.216654	0.3E-05	0.2E-05
1.00	1.198345	0.3E-05	

6.6 Analysis of the rounding errors

The one-step method is written in the form

$$\begin{aligned} y_{k+1} =& y_k + h\psi(t_k, y_k, h), \qquad k = 0, 1, \dots \\ y_0 =& \eta, \end{aligned} \tag{6.12}$$

however, in practice we have

$$\begin{aligned} \hat{y}_{k+1} =& \hat{y}_k + h\psi(t_k, \hat{y}_k, h) + \varepsilon_{k+1}, \qquad k = 0, 1, \dots \\ \hat{y}_0 =& \eta, \end{aligned} \tag{6.13}$$

where $\hat{y}_k$ is the representation of y_k in the finite precision arithmetic. (We assume that $\hat{t}_k = t_k$, $\hat{h} = h$, etc.

- What is the rounding error, i.e., $r_k := \hat{y}_k - y_k$?

Theorem 6.19. *Let the one-step method* (6.13) *have the function ψ Lipschitz continuous and*

$$|\varepsilon_k| \le \epsilon \qquad \forall k = 1, 2, \dots$$

Then the rounding error is bounded by

$$\max_{k; a+kh \le b} |r_k| \le \frac{\epsilon}{h} \frac{e^{L(b-a)} - 1}{L}. \tag{6.14}$$

Proof. The proof is analogous to the proof of Theorem 6.14. We have

$$\begin{aligned} y_{k+1} =& y_k + h\psi(t_k, y_k, h), \\ \hat{y}_{k+1} =& \hat{y}_k + h\psi(t_k, \hat{y}_k, h) + \epsilon \end{aligned}$$

Subtracting them and putting $r_k := y(t_k) - y_k$ we have

$$r_{k+1} = r_k + h[\psi(t_k, \hat{y}_k, h) - \psi(t_k, y_k, h)] + \epsilon$$

The use of the Lipschitz continuity gives

$$|r_{k+1}| \le |r_k| + hL|r_k| + \epsilon \le (1 + hL)|r_k| + \epsilon$$

Hence, using the same estimates for r_k, r_{k-1}, etc., we have

$$\begin{aligned} |r_{k+1}| \le& (1 + hL)|r_k| + \epsilon \\ \le& (1 + hL)((1 + hL)|r_{k-1}| + \epsilon) + \epsilon \\ =& (1 + hL)^2|r_{k-1}| + \epsilon \sum_{j=0}^{1} (1 + hL)^j \\ & \vdots \\ =& (1 + hL)^{k+1}|r_0| + \epsilon \sum_{j=0}^{k} (1 + hL)^j. \end{aligned}$$

In our case $r_0 = 0$. Further, summing the geometric series

$$|r_{k+1}| \le \epsilon \frac{(1 + hL)^{k+1} - 1}{hL} = \frac{\epsilon}{h} \frac{(1 + hL)^{k+1} - 1}{L}.$$

Using the fact that $(1 + hL)^{k+1} \le e^{(k+1)hL}$ and $kh \le b - a$ for all $k = 1, 2, \dots$, we have

$$|r_{k+1}| \le \frac{\epsilon}{h} \frac{e^{(k+1)hL} - 1}{L} \le \frac{\epsilon}{h} \frac{e^{(b-a)L} - 1}{L}.$$

□

Hint for exercise: Prove this theorem in correlation with Theorem 6.14.
Influence of the rounding errors (simple accuracy)

ODE $y' = 1 - y$, $y(0) = 2$, exact solution $y = 1 + \exp(-x)$, second order method

x	h	y_n	disc. err.	round. err.	comput. err.
	1E-2	1.36789477	-0.00001527	-0.00000006	-0.00001533
	1E-3	1.36788023	-0.00000016	-0.00000063	-0.00000079
1.0	1E-4	1.36788575	0.00000000	-0.00000631	-0.00000631
	1E-5	1.36794278	0.00000000	-0.00006334	-0.00006334
	1E-6	1.36852278	0.00000000	-0.00064334	-0.00064334
	1E-2	1.13534665	-0.00001129	-0.00000008	-0.00001137
	1E-3	1.13533631	-0.00000012	-0.00000091	-0.00000103
2.0	1E-4	1.13534376	0.00000000	-0.00000848	-0.00000848
	1E-5	1.13542195	0.00000000	-0.00008667	-0.00008667
	1E-6	1.13617413	0.00000000	-0.00083885	-0.00083885
	1E-2	1.04979342	-0.00000624	-0.00000011	-0.00000635
	1E-3	1.04978815	-0.00000006	-0.00000102	-0.00000108
3.0	1E-4	1.04979648	-0.00000000	-0.00000941	-0.00000941
	1E-5	1.04988325	-0.00000000	-0.00009618	-0.00009618
	1E-6	1.05090221	-0.00000000	-0.00111514	-0.00111514

6.7 Multi-step methods

The ODE problem

$$\begin{aligned} y'(t) &= f(t, y(t)), \qquad t \in (a, b) \\ y(a) &= \eta, \end{aligned} \tag{6.15}$$

Let $x_k = a + kh$, $k = 0, 1, \ldots$, we put $f_k = f(t_k, y_k)$, the multi-step method

$$\sum_{i=0}^{m} \alpha_i y_{k+i} = h \sum_{i=0}^{m} \beta_i f_{k+i}, \qquad k = 0, 1, \ldots, \tag{6.16}$$

$\alpha_m \neq 0$ and $|\alpha_0| + |\beta_0| \neq 0$. We evaluate y_{k+m} using y_{k+m-1}, y_{k+m-2},..., y_k, m-step method.

- The value y_0 is given by the initial condition and $y_1, \ldots, y_{m-1}$ by a one-step method.
- If $\beta_m \neq 0$ then the method is implicit, we need to solve non-linear algebraic system
  - Newton method
  - predictor-corrector method
    1. by an explicit method (predictor) we evaluate y^0_{k+m},
    2. by an implicit method (corrector) we evaluate y^{l+1}_{k+m} by

$$\alpha_m y^{l+1}_{k+m} + \sum_{i=0}^{m-1} \alpha_i y_{k+i} = h\beta_m f(t_{k+m}, y^l_{k+m}) + h \sum_{i=0}^{m-1} \beta_i f_{k+i}, \; l = 0, 1, \ldots,$$

    explicit relations.
- multi-step methods are not suitable for a variable time step.

6.7.1 Adams-Bashforth methods

Integrating (6.15) over $(t_k, t_k + 1)$ gives

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(s, y(s))\,\mathrm{d}s.$$

We approximate $f(s, y(s))$ by its Lagrange interpolation at

$$[t_k, f_k], \quad [t_{k-1}, f_{k-1}], \quad [t_{k-2}, f_{k-2}], \ldots, \quad [t_{k-m+1}, f_{k-m+1}],$$

then we define the Adams-Bashforth method

$$y_{k+1} = y_k + h \sum_{i=0}^{m-1} b_i f_{k-i}, \qquad b_i = \frac{1}{h} \int_{t_k}^{t_{k+1}} \prod_{\substack{j=0 \\ j \neq i}}^{m-1} \frac{s - t_{k-j}}{t_{k-i} - t_{k-j}}\,\mathrm{d}s, \quad i = 0, \ldots, m-1.$$

Explicit formulae, the truncation error is $O(h^m)$.

Example 6.20. *Two step method*

$$y_{k+1} = y_k + h\left(\frac{3}{2} f_k - \frac{1}{2} f_{k-1}\right).$$

6.7.2 Adams-Moulton methods

Integrating (6.15) over $(t_k, t_k + 1)$ as above and we approximate $f(s, y(s))$ by its Lagrange interpolation at

$$[t_{k+1}, f_{k+1}], \quad [t_k, f_k], \quad [t_{k-1}, f_{k-1}], \ldots, \quad [t_{k-m+1}, f_{k-m+1}],$$

then we define the Adams-Moulton method

$$y_{k+1} = y_k + h \sum_{i=0}^{m} b_i f_{k+1-i}, \qquad b_i = \frac{1}{h} \int_{t_k}^{t_{k+1}} \prod_{\substack{j=0 \\ j \neq i}}^{m} \frac{s - t_{k+1-j}}{t_{k+1-i} - t_{k+1-j}}\,\mathrm{d}s, \quad i = 0, \ldots, m-1.$$

Implicit formulae, the truncation error is $O(h^{m+1})$.

Example 6.21. *One step, second order method:*

$$y_{k+1} = y_k + h\left(\frac{1}{2} f_{k+1} + \frac{1}{2} f_k\right).$$

Definition 6.22. *If the multi-step method* (6.16) *has order at least one, then it is consistent, i.e., the local truncation error converges to* 0 *if* $h \to 0$.

6.7.3 Backward difference formulae

The backward difference formulae

$$\sum_{i=0}^{m} \alpha_i y_{k+i} = h f_{k+m}, \qquad k = 0, 1, \ldots \tag{6.17}$$

The best stability property.

6.8 Analysis of the multi-step methods

The multi-step method reads

$$\sum_{i=0}^{m} \alpha_i y_{k+i} = h \sum_{i=0}^{m} \beta_i f_{k+i}, \qquad k = 0, 1, \ldots \tag{6.18}$$

The local truncation error is given by

$$\tau(t, y, h) := \frac{1}{h} \sum_{i=0}^{m} \alpha_i y(t + ih) - \sum_{i=0}^{m} \beta_i f(t + ih, y(t + ih)). \tag{6.19}$$

Definition 6.23. *The method* (6.16) *has order* p *if* $\tau(t, y, h) = O(h^p)$.

Theorem 6.24. *The method* (6.16) *has order* p *if and only if*

$$\sum_{i=0}^{m} \alpha_i = 0, \qquad \sum_{i=0}^{m} i^j \alpha_i = j \sum_{i=0}^{m} i^{j-1} \beta_i, \ \ j = 1, \ldots, p.$$

Proof. We expand $\tau(t, y, h)$ in the Taylor series with respect to y. First we have

$$y(t + ih) = \sum_{j=0}^{p} y^{(j)}(t) \frac{(ih)^j}{j!} + O(h^{p+1})$$

and

$$\begin{aligned} f(t + ih, y(t + ih)) = y'(t + ih) &= \sum_{j=0}^{p-1} y^{(j+1)}(t) \frac{(ih)^{(j)}}{j!} + O(h^p) \\ &= \sum_{j=1}^{p} y^{(j)}(t) \frac{(ih)^{(j-1)}}{(j-1)!} + O(h^p). \end{aligned}$$

This gives together

$$\begin{aligned} \tau(t, y, h) =& \frac{1}{h} \sum_{i=0}^{m} \alpha_i y(t + ih) - \sum_{i=0}^{m} \beta_i y'(t + ih) \\ =& \frac{1}{h} \sum_{i=0}^{m} \alpha_i \sum_{j=0}^{p} y^{(j)}(t) \frac{(ih)^j}{j!} - \sum_{i=0}^{m} \beta_i \sum_{j=1}^{p} y^{(j)}(t) \frac{(ih)^{(j-1)}}{(j-1)!} + O(h^p) \\ =& \frac{1}{h} \sum_{i=0}^{m} \alpha_i y^{(0)}(t) + \frac{1}{h} \sum_{i=0}^{m} \alpha_i \sum_{j=1}^{p} y^{(j)}(t) \frac{(ih)^j}{j!} - \sum_{i=0}^{m} \beta_i \sum_{j=1}^{p} y^{(j)}(t) \frac{(ih)^{(j-1)}}{(j-1)!} + O(h^p) \\ =& \frac{1}{h} \sum_{i=0}^{m} \alpha_i y(t) + \sum_{j=1}^{p} \frac{h^{j-1}}{j!} \left(\sum_{i=0}^{m} i^j \alpha_i - j \sum_{i=0}^{m} \beta_i i^{j-1} \right) y^{(j)}(t) + O(h^p) = O(h^p). \end{aligned}$$

□

Lemma 6.25. *If method* (6.16) *has the order at least one, i.e.,*

$$\sum_{i=0}^{m} \alpha_i = 0 \qquad \text{and} \qquad \sum_{i=0}^{m} i\alpha_i = \sum_{i=0}^{m} \beta_i$$

then it is consistent

- Is the order of convergence sufficient for a reasonable method?

Example 6.26. *The second order method*

$$y_{k+2} - 3y_{k+1} + 2y_k = h(\frac{13}{12}f_{k+2} - \frac{5}{3}f_{k+1} - \frac{5}{12}f_k).$$

Let us consider simple problem

$$y' = 0, \quad y(0) = 1 \qquad \Rightarrow \quad y(t) = 1.$$

Let us consider a small perturbation $y_1 = 1 + \epsilon$, *then*

$$\begin{aligned} y_2 =& 3y_1 - 2y_0 = 1 + 3\epsilon, \\ y_3 =& 3y_2 - 2y_1 = 1 + 7\epsilon, \\ y_4 =& 3y_3 - 2y_2 = 1 + 15\epsilon, \\ & \dots \\ y_k =& 1 + (2^k - 1)\epsilon. \end{aligned}$$

For $\epsilon = 2^{-53}$ *then after 53 steps the error is of order 1 and after 100 steps the error* $= 2^{47}$. *The method is unstable for any* $h > 0$*!*

6.9 Stability of the multistep method

Let us consider again

$$y' = 0, \quad y(0) = 1 \qquad \Rightarrow \quad y(t) = 1.$$

Then the multistep method reads

$$\sum_{i=0}^{m} \alpha_i y_{k+i} = 0, \qquad k = 0, 1, \dots \tag{6.20}$$

Relation (6.20) represents the linear difference equation with constant coefficients. The solution is a sequence $\{y_k\}_{k=1}^{\infty}$.

Example 6.27. *Let* $m = 2$ *then* (6.20) *reads*

$$\begin{aligned} \alpha_2 y_2 + \alpha_1 y_1 + \alpha_0 y_0 &= 0, \\ \alpha_2 y_3 + \alpha_1 y_2 + \alpha_0 y_1 &= 0, \\ \alpha_2 y_4 + \alpha_1 y_3 + \alpha_0 y_2 &= 0, \\ \alpha_2 y_5 + \alpha_1 y_4 + \alpha_0 y_3 &= 0, \\ &\vdots \end{aligned}$$

$\Longleftrightarrow$

$$\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & & & \\ & \alpha_0 & \alpha_1 & \alpha_2 & & \\ & & \alpha_0 & \alpha_1 & \alpha_2 & \\ & & & \alpha_0 & \alpha_1 & \alpha_2 \\ \vdots & & & & & \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix} = 0.$$

We need to solve it. Let us seek the solution in the form $y_k = \xi^k$. Inserting into (6.20), we have

$$\sum_{i=0}^{m} \alpha_i \xi^{k+i} = \xi^k \sum_{i=0}^{m} \alpha_i \xi^i = 0. \tag{6.21}$$

If $\xi \neq 0$ then the values ξ solving (6.21) are the roots of the characteristic polynomial function

$$\rho(\xi) := \sum_{i=0}^{m} \alpha_i \xi^i. \tag{6.22}$$

Obviously, if ξ_l is a root of (6.22) then the sequence

$$\{(\xi_l)^k\}_{k=1}^{\infty}$$

is the solution of (6.21). Moreover, if $\xi_l \neq 0$ is a root of (6.22) with the multiplicity p_l then the sequence

$$\begin{aligned} &\{(\xi_l)^k\}_{k=1}^{\infty}, \qquad \{k(\xi_l)^{k-1}\}_{k=1}^{\infty}, \qquad \{k(k-1)(\xi_l)^{k-2}\}_{k=1}^{\infty}, \\ &\ldots, \{k(k-1)\ldots((k-p_l+2)(\xi_l)^{k-p_l+1}\}_{k=1}^{\infty} \end{aligned} \tag{6.23}$$

are the solutions of (6.21). This means

$$\begin{aligned} \{y_k\}_{k=0}^{\infty} = &\quad \xi_l, \quad \xi_l^2, \quad \xi_l^3, \quad \xi_l^4, \quad \ldots \\ \{y_k\}_{k=0}^{\infty} = &\quad 1, \quad 2\xi_l, \quad 3\xi_l^2, \quad 4\xi_l^3, \quad \ldots \\ \{y_k\}_{k=0}^{\infty} = &\quad 0, \quad 2, \quad 6\xi_l, \quad 12\xi_l^2, \quad \ldots \end{aligned}$$

Relation (6.23) follows from the following observation. If $\xi_l \neq 0$ is a root of $\rho(\xi)$ with the multiplicity p_l, then it is a root with the multiplicity p_l of

$$\phi_n(\xi) = \xi^n \rho(\xi) \qquad n \in \mathbb{N}_0.$$

Hence, $p_l - 1$ derivative of ϕ_n is equal to zero, thus

$$\begin{aligned} \phi_n(\xi) &= \sum_{i=0}^{m} \alpha_i \xi^{i+n} = 0 \qquad \forall n \in \mathbb{N}_0, \\ (\phi_n)'(\xi) &= \sum_{i=0}^{m} \alpha_i (i+n) \xi^{i+n-1} = 0 \qquad \forall n \in \mathbb{N}_0, \\ (\phi_n)''(\xi) &= \sum_{i=0}^{m} \alpha_i (i+n)(i+n-1) \xi^{i+n-2} = 0 \qquad \forall n \in \mathbb{N}_0, \\ &\vdots \end{aligned}$$

If $\xi_l =$ then we put $0 \cdot \xi_l^{-j} = 0$ for $j > 0$, i.e.

$$\begin{aligned} \xi_l = 0: \qquad \xi_l^k: &\quad \{y_k\}_{k=0}^{\infty} = 1, 0, 0, 0, \dots \\ k\xi_l^{k-1}: &\quad \{y_k\}_{k=0}^{\infty} = 1, 1, 0, 0, \dots \\ k(k-1)\xi_l^{k-2}: &\quad \{y_k\}_{k=0}^{\infty} = 1, 1, 1, 0, \dots \end{aligned}$$

Theorem 6.28. *Let $\xi_l,\ l = 1, \dots, z$ be the roots of $\rho(\xi)$ with the multiplicity $p_l,\ l = 1, \dots, z$. Then the solution of (6.21) reads*

$$y_k = \sum_{l=1}^{z} \left(c_{l,1}\xi_l^k + c_{l,2}k\xi_l^{k-1} + \cdots + k(k-1)\dots(k-p_l+2)\xi_l^{k-p_l+1} \right).$$

(If $\xi_l = 0$ then we put $0 \cdot \xi_l^{-j} = 0$ for $j > 0$.)

Proof. See [FK14]. □

In order to avoid a propagation of the error we require that

- All roots of $|\rho(\xi)| \leq 1$.
- If $|\rho(\xi)| = 1$ then its multiplicity is equal to 1.

Definition 6.29. *We say that (6.18) is stable (or more precisely zero-stable) if the roots of the corresponding characteristic polynomial satisfy the above conditions.*

Theorem 6.30. *The multistep method (6.16) is convergent if and only if it is stable and consistent.*

Remark 6.31. *There exists many types of stabilities, A-stability, D-stability, α-stability, etc.*

Hint for exercise: Given multistep method, decide if it is table or not, set the order of the method.

Hint for exercise: Derive the multistep method in the given form

Chapter 7

Numerical optimization (1 week)

Basic task: let $J : U \to \mathbb{R}$ be a mapping, $U \subset \mathbb{R}^n$ we seek the minimum of J on U, i.e., we seek $\bar{u} \in U$ such that

$$J(\bar{u}) \leq J(u) \quad \forall u \in U. \tag{7.1}$$

Remark 7.1. *If we need a maximum of J, we seek minimum of $-J$.*

Example 7.2. *Interpolation by the least square technique: Let (x_i, y_i), $i = 1, \ldots, n$ be the given data, we seek a curve $f = f(\alpha_1, \ldots, \alpha_r, x)$ depending on the real parameters $\alpha_1, \ldots, \alpha_r$ such that*

$$J(\alpha_1, \ldots, \alpha_r) = \sum_{i=1}^{n} (f(\alpha_1, \ldots, \alpha_r, x_i) - y_i)^2$$

is minimal.

Example 7.3. *The optimalization of the shape of a ship.*

- *The horizontal cut of a ship Γ can be parametrized by a function $\phi : [a, b] \to \mathbb{R}^2$,*

$$\Gamma = \{\phi(t) \in \mathbb{R}^2, \ t \in [a, b]\}.$$

- *The flow around the ship is described by a system of partial differential equations, Γ defines a boundary of the computational domain.*
- *Solving of this system we obtain the distribution of pressure* $\mathtt{p}$
- *The drag force is given by*

$$F_D(\phi) = \int_\Gamma \mathtt{p} n_1 \, \mathrm{d}S,$$

  where n_1 is the component of the unit outer normal to Γ in the direction of the flow.

- *The aim is to find ϕ such that $F_D(\phi)$ is minimal.*
- *In practice, we prescribe ϕ by a finite number of parameters.*

Our tasks:

- When exists a unique solution of (7.1)?
- How can we approximate the unique solution of (7.1)?

7.1 Existence of the minimum

Theorem 7.4. *Let U be a closed and bounded domain, $J : U \to \mathbb{R}$ a continuous function, then there exists a minimum of J on U.*

Definition 7.5. *Let U be unbounded domain. We say that J is coercive on U if*

$$\lim_{u\in U;\|u\|\to\infty} J(u) = \infty.$$

Theorem 7.6. *Let U be a closed and unbounded domain, $J : U \to \mathbb{R}$ a continuous and coercive function, then there exists a minimum of J on U.*

Proof. Let $a \in U$. Since f is coercive, there exists $R > 0$ such that

$$a \in U \cap B(0,R) \neq \emptyset, \qquad J(u) \geq J(a) + 1 \ \forall u \in U \setminus B(0,R),$$

where $B(0,R)$ is the closed ball with the centre at the origin and the radius R. Let $\bar{u}$ be the minimum on $U \cap B(0,R)$ (exists due to Theorem 7.4). It is a minimum on U since

$$J(u) \geq J(a) + 1 > J(a) \geq J(\bar{u}) \quad \forall u \in U \setminus B(0,R).$$

□

Definition 7.7. *Let $J \in C^1(U)$, where U is open. For $u \in U$, $\varphi \in \mathbb{R}^n$ we define the directional derivative of J at u along the direction φ by*

$$J'(u;\varphi) := \lim_{\theta\to 0} \frac{1}{\theta}(J(u+\theta\varphi) - J(u)).$$

It is valid that

$$J'(u;\varphi) = \nabla J(u)\cdot\varphi, \qquad \nabla J(u) = \left(\frac{\partial J}{\partial u_1}(u), \ldots, \frac{\partial J}{\partial u_2}(u)\right)^{\mathrm{T}}.$$

Definition 7.8. *Let $U \subset \mathbb{R}^n$ be a convex set. We say that $J : U \to \mathbb{R}$ is a convex function if*

$$J(u+\theta(v-u)) \leq J(u) + \theta(J(v) - J(u)) \qquad \forall u, v \in U \ \forall\theta \in (0,1).$$

We say that J is a strictly convex function if

$$J(u+\theta(v-u)) < J(u) + \theta(J(v) - J(u)) \qquad \forall u, v \in U,\ u \neq v,\ \forall\theta \in (0,1).$$

Lemma 7.9. *Let $U \subset \mathbb{R}^n$ be an open convex set, $J \in C^1(U)$. Then*

i) J is convex $\Leftrightarrow$ $J(v) \geq J(u) + J'(u; v-u)$ $\forall u, v, \in U$.

ii) J is strictly convex $\Leftrightarrow$ $J(v) > J(u) + J'(u; v-u)$ $\forall u, v \in U,\ u \neq u$.

Proof. Four steps.

i) $\Rightarrow$ Let J be convex. Let $u, v \in U$, then

$$J(u + \theta(v - u)) \leq J(u) + \theta(J(v) - J(u)) \qquad \forall \theta \in (0, 1),$$

hence

$$J(v) - J(u) \geq \frac{1}{\theta}(J(u + \theta(v - u)) - J(u)) \qquad \forall \theta \in (0, 1).$$

Let $\theta \to 0^+$, then

$$J(v) - J(u) \geq \lim_{\theta \to 0^+} \frac{1}{\theta}(J(u + \theta(v - u)) - J(u)) = J'(u; v - u).$$

i) $\Leftarrow$ Let

$$J(\bar{v}) \geq J(\bar{u}) + J'(\bar{u}; \bar{v} - \bar{u}) \ \forall \bar{u}, \bar{v}, \in U$$

Let $u, v \in U$, $\theta \in (0, 1)$ arbitrary. We put $\bar{v} := u$, $\bar{u} = u + \theta(v - u)$. Then

$$\begin{aligned} J(u) \geq & J(u + \theta(v - u)) + J'(u + \theta(v - u); -\theta(v - u)) \\ = & J(u + \theta(v - u)) - \theta J'(u + \theta(v - u); (v - u)) \end{aligned} \tag{7.2}$$

Similarly, we put $\bar{v} := v$, $\bar{u} = u + \theta(v - u)$,then

$$\begin{aligned} J(v) \geq & J(u + \theta(v - u)) + J'(u + \theta(v - u); (1 - \theta)(v - u)) \\ = & J(u + \theta(v - u)) + (1 - \theta) J'(u + \theta(v - u); (v - u)) \end{aligned} \tag{7.3}$$

Performing $(1 - \theta)(7.2) + \theta(7.3)$, we have

$$(1 - \theta)J(u) + \theta J(v) \geq J(u + \theta(v - u)), \quad u, v \in U, \ \theta \in (0, 1),$$

Hence, J is convex.

ii) $\Leftarrow$ is completely the same as i) $\Leftarrow$

ii) $\Rightarrow$ Let J be strictly convex. Let $u, v \in U$, $u \neq v$, $\theta \in (0, 1)$. Then

$$J(u + \theta(v - u)) < J(u) + \theta(J(v) - J(u)),$$

hence

$$J(v) - J(u) > \frac{J(u + \theta(v - u)) - J(u)}{\theta} \overset{i)}{\geq} \frac{J'(u; \theta(v - u)}{\theta} = J'(u; v - u).$$

□

Theorem 7.10. *Let U be open, $J \in C^1(U)$.*

i) Let $\bar{u} \in U$ be a local minimum of J, then $J'(u; \varphi) = 0$ for all $\varphi \in \mathbb{R}^n$ (i.e., $\nabla J(\bar{u}) = 0$).

ii) Let U be convex, J be convex. Then, the following assertion are equivalent:

a) $\bar{u}$ is a local minimum

b) $\bar{u}$ *is a minimum*

c) $\nabla J(\bar{u}) = 0$.

iii) If J *is strictly convex, then* J *has at most one minimum.*

Proof. i) Known results of the mathematical analysis.

ii)
- b)⇒ a) is obvious
- a)⇒ c) is i)
- c)⇒ b). Since J is convex then using Lemma 7.9, we have

$$J(v) \geq J(\bar{u}) + J'(\bar{u}; v - \bar{u}) = J(\bar{u}) \ \forall v \in U,$$

hence $\bar{u}$ is the local minimum of J on U.

iii) Let J has two minima $u_1 \neq u_2$. Since J is strictly convex then using Lemma 7.9, we have

$$J(u_1) > J(u_2) + J'(u_2; u_1 - u_2) = J(u_2).$$

Hence u_2 can not be the minimum.

□

Definition 7.11. *Let* $J \in C^2(U)$, $U \in\subset \mathbb{R}^n$. *Then we define the second order derivative of* J *at* $u \in U$ *along the directions* φ *and* ψ *by*

$$J''(u, \varphi, \psi) := \lim_{\theta \to 0} \frac{1}{\theta}(J'(u + \theta\psi; \varphi) - J'(u; \varphi)).$$

We have

$$J''(u; \varphi, \psi) = \varphi^{\mathrm{T}} D^2 J(u) \psi, \qquad D^2 J(u) = \left\{ \frac{\partial^2 J}{\partial u_i \partial u_j}(u) \right\}_{i,j=1}^{n}.$$

$D^2 J$ is called the Hess matrix. Then, using the Taylor series, there exists $\theta \in (0, 1)$ such that

$$J(v) = J(u) + J'(u; v - u) + \frac{1}{2} J''(u + \theta(v - u); v - u, v - u).$$

Theorem 7.12. *Let* $J : \mathbb{R}^n \to \mathbb{R}$, $J \in C^2$ *and let there exists* $\alpha > 0$ *such that*

$$J''(u; \varphi, \varphi) \geq \alpha \|\varphi\|^2 \qquad \forall u, \varphi \in \mathbb{R}^n$$

Then J *is coercive and strictly convex on* $\mathbb{R}^n$.

Proof. The Cauchy inequality gives

$$|J'(0; u)| = |\nabla J(0) \cdot u| \leq \|\nabla J(0)\| \|u\| =: M\|u\|,$$

where $M < \infty$ since $J \in C^2$. Moreover, the Taylor expansion

$$\begin{aligned} J(u) =& J(0) + J'(0; u) + \frac{1}{2} J''(\theta u; u, u) \\ \geq& J(0) - M\|u\| + \frac{\alpha}{2}\|u\|^2 \to \infty \quad \text{for} \quad \|u\| \to \infty, \end{aligned}$$

hence J is coercive.

Let $u \neq v$, using the Taylor expansion

$$\begin{aligned} J(v) =& J(u) + J'(u; v-u) + \frac{1}{2} J''(u + \theta(v-u); v-u, v-u) \\ \geq& J(u) + J'(u; v-u) + \frac{\alpha}{2}\|v-u\|^2 \\ >& J(u) + J'(u; v-u), \end{aligned}$$

hence J is strictly convex. □

Theorem 7.13. *If J satisfies the assumptions of Theorem 7.12, then there exists a unique minimum of J.*

7.2 Numerical methods seeking the minimum of J

Let $J : \mathbb{R}^n \to \mathbb{R}$ be a mapping, we seek $\bar{u} \in \mathbb{R}^n$ such that

$$J(\bar{u}) \leq J(u) \quad \forall u \in \mathbb{R}^n. \tag{7.4}$$

- The minimum of J can be sought numerically (approximate value is sufficient).
- we need to define a sequence $u_1, u_2, \ldots$, such that $u_k \to \bar{u}$, where $\bar{u}$ is the solution of (7.4).
- As usually, u_{k+1} is computed from u_k, we employ the recurrence formulae

  $$u_{k+1} = u_k + \rho_k \varphi_k, \qquad k = 0, 1, 2, \ldots, \tag{7.5}$$

  where $\varphi_k \in \mathbb{R}^n$ is the direction of the descend $\rho_k \in \mathbb{R}$ is the size of the descend.
- How to choose φ_k and ρ_k?
- **Idea:** if u_k is an approximation then u_{k+1} should be such that $J(u_{k+1}) \leq J(u_k)$.

It is suitable to choose $\varphi_k \in \mathbb{R}^n$ such that

$$J'(u_k; \varphi_k) = \nabla J(u_k) \cdot \varphi_k < 0. \tag{7.6}$$

Theorem 7.14. *Let $J \in C^2(\mathbb{R}^n)$, (7.5) and (7.6) be valid. Then there exists $\tilde{\rho} > 0$ such that*

$$J(u_{k+1}) < J(u_k) \quad \text{for } \rho_k \in (0, \tilde{\rho}).$$

Proof. The Taylor relation gives

$$J(u_{k+1}) = J(u_k) + \rho_k J'(u_k; \varphi_k) + \frac{1}{2}\rho_k^2 J''(\beta; \varphi_k, \varphi_k),$$

where $\beta \in \mathbb{R}^n$ is between u_k and $u_k + \rho_k \varphi_k$. Let $K > 0$, then there exists $M > 0$ such that

$$|J''(\beta; \varphi_k, \varphi_k)| \leq M \quad \forall \beta = u_k + \rho \varphi_k, \ \rho \in [0, K].$$

Thus

$$J(u_{k+1}) = J(u_k) + \rho_k \left(\underbrace{J'(u_k;\varphi_k)}_{<0} + \frac{1}{2}\rho_k \underbrace{J''(\beta;\varphi_k,\varphi_k)}_{\leq M} \right).$$

Hence, there exists $\tilde{\rho} > 0$ such that

$$J'(u_k) + \frac{1}{2}\rho_k J''(\beta;\varphi_k,\varphi_k) < 0 \qquad \forall \rho_k \in (0,\tilde{\rho})$$

and thus $J(u_{k+1}) < J(u_k)$. □

7.2.1 Methods of the deepest descent

Usually, we put

$$\varphi_k := -\nabla J(u_k),$$

the deepest descent. Then

$$J'(u_k,\varphi_k) = -\nabla J(u_k)\cdot\nabla J(u_k) = -\|\nabla J(u_k)\|^2 < 0$$

provided that $\nabla J(u_k) \neq 0$.

Two possibilities:

- fixed step ρ_k:

  Theorem 7.15. *Let $J \in C^2(\mathbb{R}^n)$ and let there exists $\lambda > 0$ and $\Lambda > 0$ such that*

  $$\lambda\|\varphi\|^2 \leq \varphi^{\mathrm{T}} J''(u)\varphi \leq \Lambda\|\varphi\|^2 \qquad \forall u,\varphi \in \mathbb{R}^2.$$

  Putting, $\rho_k = 2/(\lambda+\Lambda)$ and $\varphi_k := -\nabla J(u_k)$, the method (7.5) converges to the minimum of J.

- optimal step ρ_k: we put

  $$\rho_k = \arg\min_{\rho>0} J(u_k + \rho\varphi_k).$$

  Stopping criterion

  $$\|\nabla J(u_k)\| \leq \varepsilon, \qquad \varepsilon > 0 \text{ is a tolerance.}$$

7.2.2 Methods using the Newton method

In order to solve (7.1), in virtue of Theorem 7.10, we seek $\bar{u} = (\bar{u}_1,\ldots,\bar{u}_n) \in U \subset \mathbb{R}^n$ such that

$$\nabla J(\bar{u}) = 0 \qquad \Longleftrightarrow \qquad \frac{\partial J}{\partial u_i}(\bar{u}) = 0 \ \forall i = 1,\ldots,n. \tag{7.7}$$

Relation (7.7) exhibits the system of the nonlinear algebraic equations which can be written as

$$F(\bar{u}) := (F_1(\bar{u}), \dots, F_n(\bar{u})) = 0 \qquad F_i(\bar{u}) := \frac{\partial J}{\partial u_i}(\bar{u}),\ i = 1, \dots, n. \tag{7.8}$$

Using the Newton method, we have the sequence $\{\bar{u}^k\}$ approximating of $\bar{u}$, where

$$\bar{u}^{k+1} = \bar{u}^k + d^k, \qquad \frac{\mathrm{D}F(\bar{u}^k)}{\mathrm{D}\bar{u}} d^k = -F(\bar{u}^k) \tag{7.9}$$

and

$$\frac{\mathrm{D}F(u)}{\mathrm{D}u} := \left\{ \frac{\partial F_i(u)}{\partial u_j} \right\}_{i,j=1}^{n} = \left\{ \frac{\partial^2 J(u)}{\partial u_j \partial u_i} \right\}_{i,j=1}^{n}. \tag{7.10}$$

On contrary to the deepest descent methods, the second order derivatives of J are required. Convergence can be faster.

Bibliography

[FK14] M. Feistauer and V. Kučera. *Základy numerické matematiky.* MFF UK, 2014.

[GC12] Anne Greenbaum and Timothy P. Chartier. *Numerical Methods: Design, Analysis and Computer Implementation of Algorithms.* Princeton Universtity Press, 2012.