EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Worksheets for Numerical methods

Zuzana Morávková, Jiří Krček

VSB TECHNICAL
UNIVERSITY
OF OSTRAVA

Introduction

The study material is designed for students of VSB - Technical University of Ostrava.

The worksheets consist of several theoretical sheets, some solved problems and some sheets with unsolved problems for practicing. The materials should support classwork and they are not recommended for self-study or as a replacement for textbooks.

Thanks

Technology for the Future 2.0, CZ.02.2.69/0.0/0.0/18_058/0010212

ISBN 978-80-248-4493-0

DOI 10.31490/9788024844930

Contents

Worksheets for Numerical methods

Root-finding for non-linear equations

5 – Root-finding for non-linear equations, separation of roots

Given a continuous function $y = f(x)$, we find $\tilde{x} \in D_f$ such that

$$f(\tilde{x}) = 0.$$

The value $\tilde{x}$ is called **root** or **zero** of the function f.

Separation of roots

At first we have to determine number of roots and their positions, i.e. we need to find such intervals that each of these includes only one root. We can use the following theorem.

Theorem

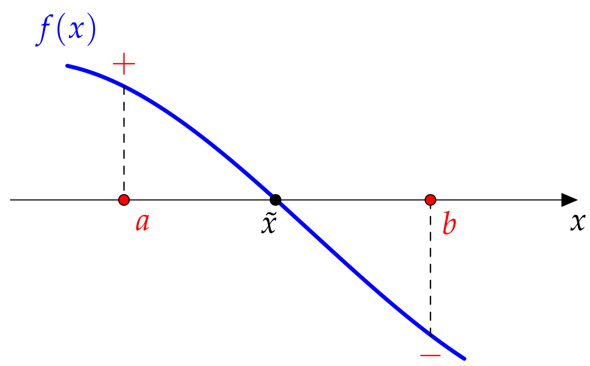
If the function f is continuous on the interval $[a, b]$ and

$$f(a) \cdot f(b) < 0$$

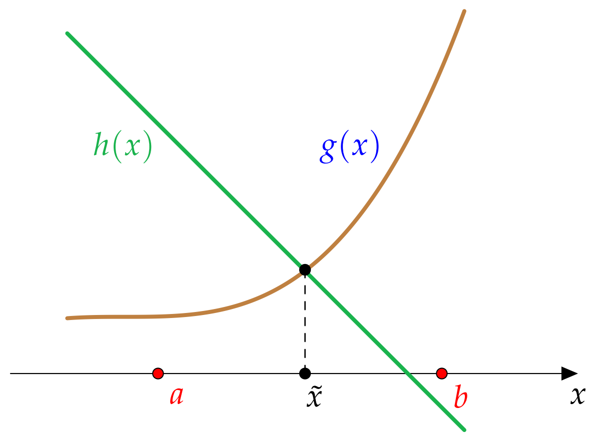
then there is $\tilde{x} \in (a, b)$ such that $f(\tilde{x}) = 0$.

There are several ways how to find intervals such that each one of these includes only one root.

- We plot the graph of the function f and find points of intersection of this graph and the x-axis.



- We convert the equation $f(x) = 0$ into a form $h(x) = g(x)$ and we find points of intersection of graphs of functions h and g.



- We tabulate values of the function f a find where their signs change.

Worksheets for Numerical methods

6 – Root-finding for non-linear equations, separation of roots

Example

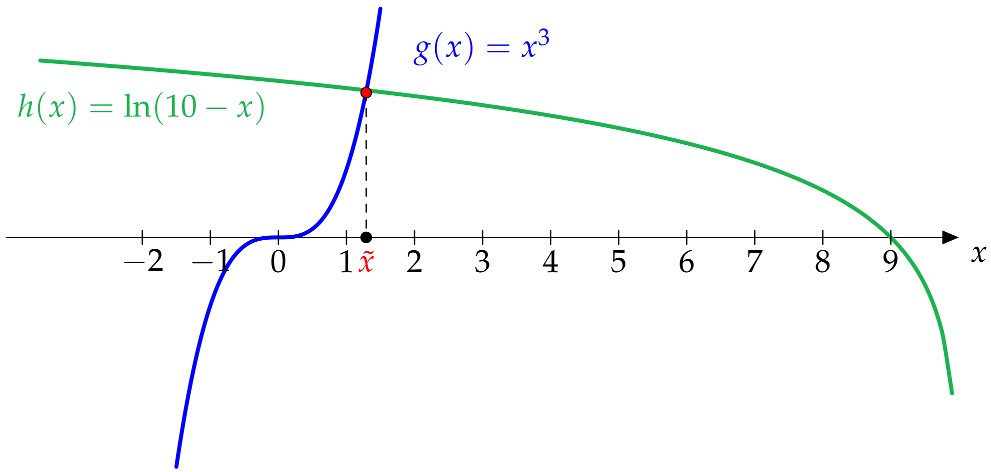
Separate all roots of the equation

$$x^3 - \ln(10 - x) = 0.$$

We convert the given equation into such form to be able to plot graphs of functions on both sides of this equation.

$$x^3 - \ln(10 - x) = 0$$
$$x^3 = \ln(10 - x)$$

We plot graphs of functions $g(x) = x^3$ and $h(x) = \ln(10 - x)$ and find points of intersection of these graphs.



It is obvious that the point of intersection is unique and lies within the interval $[1, 2]$. To determine the root more precisely we tabulate function values of the function $f(x) = x^3 - \ln(10 - x)$ on this interval with the step 0.1. We round all values to two decimal places.

x	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
$f(x)$	-1.19	-0.85	-0.44	0.03	0.59	1.23	1.96	2.79	3.72	4.76	5.92

We can observe that sings of these function values change between 1.2 and 1.3. The function f is also evidently continuous on the mentioned interval. So the equation $x^3 - \ln(10 - x) = 0$ has unique root in the interval $[1.2, 1.3]$.

7 – Root-finding for non-linear equations, separation of roots

Example

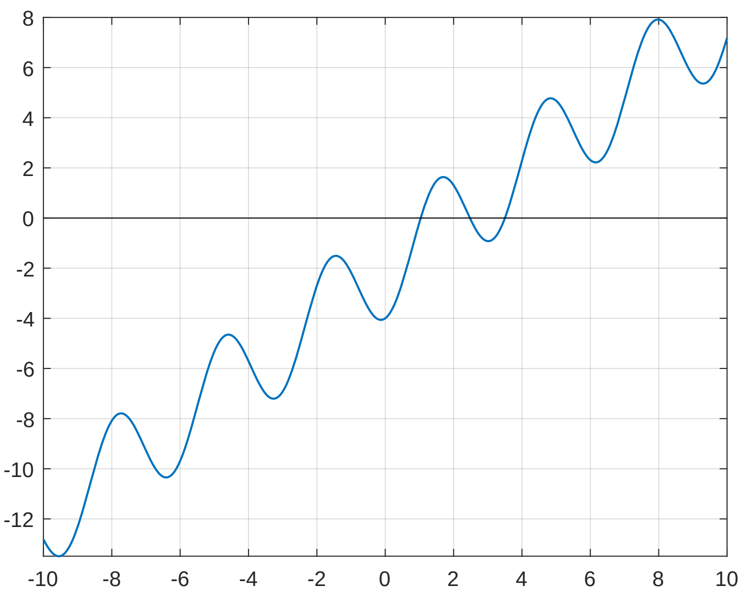
Separate all roots of the equation

$$x - 4\cos^2(x) = 0.$$

Use the MATLAB to solve the problem.

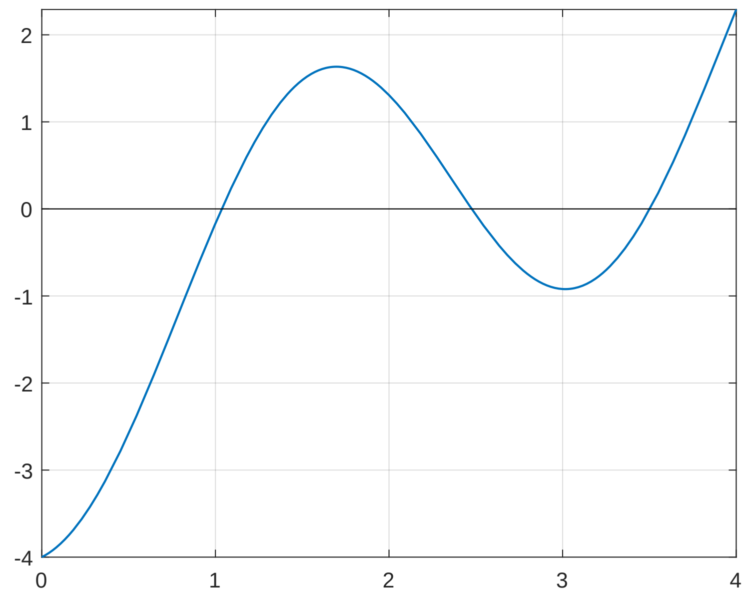
We plot graph of the function $f(x) = x - 4\cos^2(x)$ and locate its point of intersection with the x-axis. We plot the graph on sufficiently long interval that naturally must be a subset of the function domain. We choose the interval $[-10, 10]$.

```
>> fplot(@(x)x-4*cos(x).^2,[-10,10])
>> grid on
```



We can observe that all roots lie in the interval $[0, 4]$. So we plot the graph once more only on this shorter interval to determine the roots position more precisely.

```
>> fplot(@(x)x-4*cos(x).^2,[0,4])
>> grid on
```



There are three roots separated in intervals $[1, 2]$, $[2, 3]$ and $[3, 4]$.

Worksheets for Numerical methods

8 – Root-finding for non-linear equations, separation of roots

Example

Separate all roots of the equation

$$x^4 - 5x^3 - 10x^2 + 1 = 0.$$

We calculate function values of the function $f(x) = x^4 - 5x^3 - 10x^2 + 1$ for the x values $-4, -3.5, -3, \ldots, 7$. All values are rounded to one decimal place.

x	-4	-3.5	-3	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7
$f(x)$	417	242.9	127	55.7	17	0.4	-3	-0.8	1	-2.1	-13	-33.3	-63	-100.6	-143	-185.8	-223	-247.1	-249	-218.3	-143	-9.6	197

Using this table we can localize intervals where the function values change signs.

There are four roots of the given equation separated in intervals $[-1.5, -1]$, $[-0.5, 0]$, $[0, 0.5]$ and $[6.5, 7]$.

9 – Root-finding for non-linear equations, separation of roots

Exercise

Separate all roots of the equation

$$2x^3 - x^2 - x - 1 = 0.$$

10 – Root-finding for non-linear equations, separation of roots

Exercise

Separate all roots of the equation

$$2x^3 - x^2 - x - 1 = 0.$$

Use the MATLAB to solve the problem.

11 – Root-finding for non-linear equations, bisection method

Bisection method

The aim of all methods is to find a sequence of numbers $x^{(1)},\ x^{(2)}, x^{(3)}, \ldots, x^{(k)}$ that converges to the searched root $\tilde{x}$.

Assume that the root is separated in the interval $[a, b]$. We denote $a^{(1)} = a,\ b^{(1)} = b,\ k = 1$ in the beginning of calculations.
We determine the value $x^{(k)}$ as the mid-point of the interval $[a^{(k)}, b^{(k)}]$, i.e.

$$x^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}.$$

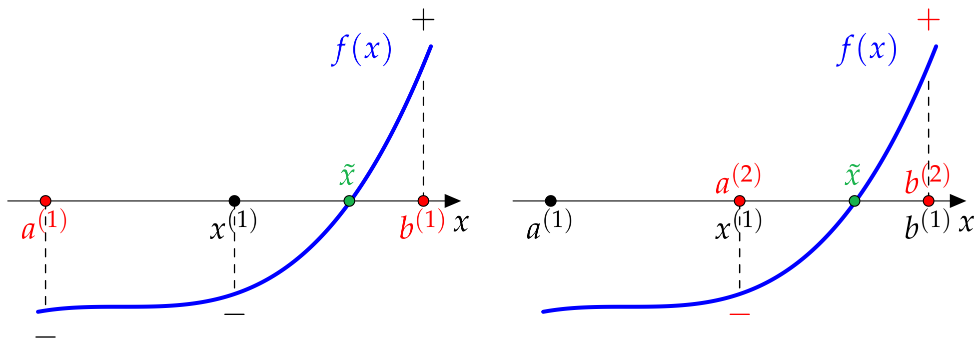
The next interval is chosen in accordace with signs of function values $f(a^{(k)}), f(x^{(k)}), f(b^{(k)})$.

If $f(a^{(k)})f(x^{(k)}) < 0$ then $a^{(k+1)} := a^{(k)},\ b^{(k+1)} := x^{(k)}$.

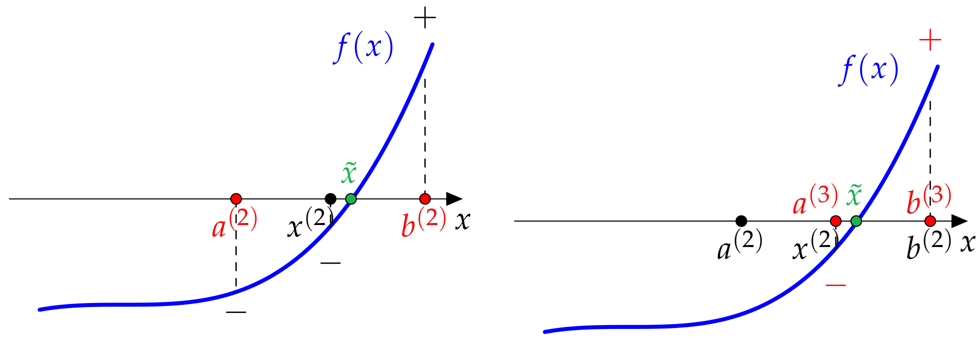
If $f(x^{(k)})f(b^{(k)}) < 0$ then $a^{(k+1)} := x^{(k)},\ b^{(k+1)} := b^{(k)}$.

Thus, we successively bisect intervals and their mid-points $\{x^{(k)}\}$ converge to the root $\tilde{x}$.

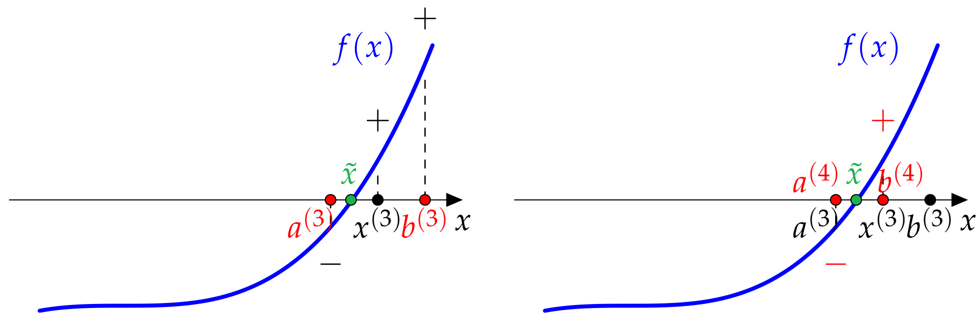
For $k = 1$



For $k = 2$



For $k = 3$



The calculation is terminated when the given accuracy is obtained, i.e. when the following holds

$$\frac{b^{(k)} - a^{(k)}}{2} \leq \varepsilon\,.$$

The last mid-point $x^{(k)}$ approximate the searched root $\tilde{x}$ with the accuracy ε.

12 – Root-finding for non-linear equations, bisection method

Example

Find all roots of the equation

$$x^3 - \ln(10 - x) = 0$$

using the bisection method with the accuracy $\varepsilon = 10^{-2}$.

We already know that the root lies in the interval $[1.2, 1.3]$, i.e. $a^{(1)} = 1.2$, $b^{(1)} = 1.3$.
The approximation error is $\frac{b^{(1)} - a^{(1)}}{2} = 0.05 > \varepsilon = 10^{-2}$ so we continue in calculations.
We calculate the first approximation $x^{(1)}$ as the mid-point of this interval:

$$x^{(1)} = \frac{b^{(1)} + a^{(1)}}{2} = \frac{1.3 + 1.2}{2} = 1.25\,.$$

Then we calculate values of the function $f(x) = x^3 - \ln(10 - x)$ at points $a^{(1)}, x^{(1)}, b^{(1)}$:

$$f(a^{(1)}) = -0.446, \quad f(x^{(1)}) = -0.2159, \quad f(b^{(1)}) = 0.0337$$

and determine interval $[a^{(2)}, b^{(2)}]$:

$$f(x^{(1)}) \cdot f(b^{(1)}) < 0 \Rightarrow a^{(2)} = x^{(1)} = 1.25, \quad b^{(2)} = b^{(1)} = 1.3.$$

The approximation error is $\frac{b^{(2)} - a^{(2)}}{2} = 0.025 > \varepsilon = 10^{-2}$ so we continue in calculations.
We calculate the second approximation

$$x^{(2)} = \frac{a^{(2)} + b^{(2)}}{2} = \frac{1.25 + 1.3}{2} = 1.275$$

and determine interval $[a^{(3)}, b^{(3)}]$:

$$f(a^{(2)}) = -0.2159, \quad f(x^{(2)}) = -0.0935, \quad f(b^{(2)}) = 0.0337,$$
$$f(x^{(2)}) \cdot f(b^{(2)}) < 0 \Rightarrow a^{(3)} = x^{(2)} = 1.275 \quad b^{(3)} = b^{(2)} = 1.3.$$

The approximation error is $\frac{b^{(3)} - a^{(3)}}{2} = 0.0125 > \varepsilon = 10^{-2}$ so we continue in calculations.
We calculate the third approximation

$$x^{(3)} = \frac{a^{(3)} + b^{(3)}}{2} = \frac{1.275 + 1.3}{2} = 1.2875$$

and determine interval $[a^{(4)}, b^{(4)}]$:

$$f(a^{(3)}) = -0.0935, \quad f(x^{(3)}) = -0.0305, \quad f(b^{(3)}) = 0.0337,$$
$$f(x^{(3)}) \cdot f(b^{(3)}) < 0 \Rightarrow a^{(4)} = x^{(3)} = 1.2875, \quad b^{(4)} = b^{(3)} = 1.3.$$

Because the approximation error fulfil $\frac{b^{(4)} - a^{(4)}}{2} = 0.0062 \leq \varepsilon = 10^{-2}$, we can terminate our calculations.
We calculate the last approximation

$$x^{(4)} = \frac{a^{(4)} + b^{(4)}}{2} = \frac{1.2875 + 1.3}{2} = 1.2938$$

k	$a^{(k)}$	$f(a^{(k)})$	$x^{(k)}$	$f(x^{(k)})$	$b^{(k)}$	$f(b^{(k)})$	$\frac{\lvert b^{(k)} - a^{(k)}\rvert}{2}$
1	1.2	$-$	1.25	$-$	1.3	$+$	$0.05 > 10^{-2}$
2	1.25	$-$	1.275	$-$	1.3	$+$	$0.02 > 10^{-2}$
3	1.275	$-$	1.2875	$-$	1.3	$+$	$0.0125 > 10^{-2}$
4	1.2875	$-$	1.2938	$+$	1.3	$+$	$0.0062 \leq 10^{-2}$

The resulting approximation of the given equation root is

$$\tilde{x} = 1.29 \pm 0.01.$$

13 – Root-finding for non-linear equations, bisection method

Example

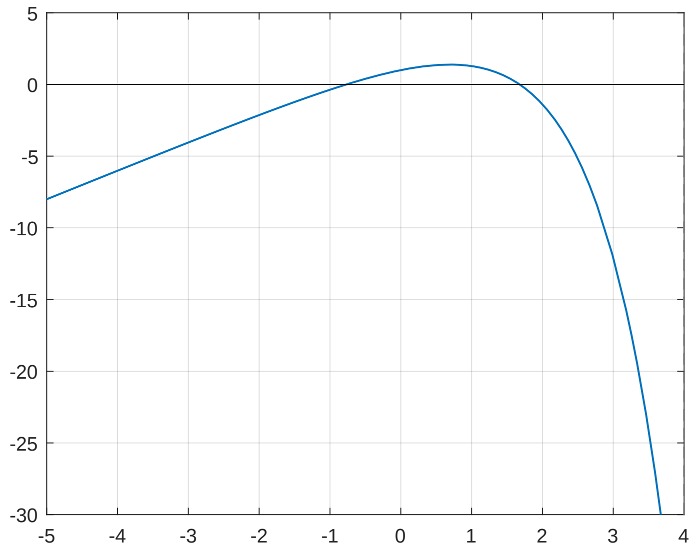
Find all roots of the equation

$$2x + 2 - e^x = 0$$

using the bisection method with the accuracy $\varepsilon = 10^{-2}$.
Use the MATLAB to solve the problem.

The first step is to separate roots. We define function f and plot its graph on a sufficient interval that we choose according to the domain of this function. Let us note that the domain of the function $f(x) = 2x + 2 - e^x$ is $D_f = \mathbb{R}$.

```
>> f=@(x)(2*x+2-exp(x))
f =
    @(x)(2*x+2-exp(x))
>> fplot(f,[-5,4])
>> grid on
```



It is obvious that there are two points of intersection of the function f graph and the x-axis included in intervals $[-1, 0]$ and $[1, 2]$.

Now we will calculate a root in the interval $[1, 2]$.
We input end-points of the interval as the variables `a` and `b` and set up the starting value of the approximation index `k`.

```
>> k=0; a=1; b=2;
```

In each step we increase the index `k` by one, calculate `x(k)` and the approximation error. We use the `if` statement to choose an interval for the next step.

```
>> k=k+1
k =
     1
>> x(k)=(a(k)+b(k))/2
x =
    1.5000
>> (b(k)-a(k))/2
ans =
    0.5000
>> if f(a(k))*f(x(k))<0, a(k+1)=a(k);b(k+1)=x(k);
else a(k+1)=x(k); b(k+1)=b(k);end
```

We repeat the following four statements until the approximation error is less than the given accuracy.

```
>> k=k+1
>> x(k)=(a(k)+b(k))/2
>> (b(k)-a(k))/2
>> if f(a(k))*f(x(k))<0, a(k+1)=a(k);b(k+1)=x(k);
else a(k+1)=x(k); b(k+1)=b(k);end
```

The given accuracy is achieved in the seventh step.

```
>> k=k+1
k =
     7
>> x(k)=(a(k)+b(k))/2
x =
  1.5000    1.7500    1.6250    1.6875    1.6563    1.6719    1.6797
>> (b(k)-a(k))/2
ans =
    0.0078
```

We write obtained data to a table.

k	$x^{(k)}$	$\frac{\lvert b^{(k)}-a^{(k)}\rvert}{2}$
1	1.5000	$0.5 > 10^{-2}$
2	1.7500	$0.25 > 10^{-2}$
3	1.6250	$0.125 > 10^{-2}$
4	1.6875	$0.0625 > 10^{-2}$
5	1.6563	$0.0313 > 10^{-2}$
6	1.6719	$0.0156 > 10^{-2}$
7	1.6797	$0.0078 \leq 10^{-2}$

We round the last approximation to two decimal places according to the given accuracy.

The resulting approximation of the searched root is:

$$\tilde{x} = 1.68 \pm 10^{-2}.$$

The second root lying. in the interval $[-1, 0]$ can be found in analogous way. Approximations of all roots of the given equation are:

$$-0.77 \pm 10^{-2}, \qquad 1.68 \pm 10^{-2}.$$

15 – Root-finding for non-linear equations, bisection method

Exercise

Find all roots of the equation

$$2x^3 - x^2 - x - 1 = 0$$

using the bisection method with the accuracy $\varepsilon = 10^{-2}$.

16 – Root-finding for non-linear equations, bisection method

Exercise

Find all roots of the equation

$$2x^3 - x^2 - x - 1 = 0$$

using the bisection method with the accuracy $\varepsilon = 10^{-2}$.
Use the MATLAB to solve the problem.

17 – Root-finding for non-linear equations, Newton method

Newton method

We find a sequence of numbers $x^{(0)}, x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(k)}$ that converges to the searched root $\tilde{x}$. The initial approximation $x^{(0)}$ is an arbitrary number from the interval $[a, b]$ that we obtain by previous separation of roots. The principle of the Newton method is to construct a tangent line to the graph of the given function f at the point $[x^{(0)}, f(x^{(0)})]$. The point of intersection of this tangent line and the x-axis is the next approximation $x^{(1)}$. This process is repeated until the given accuracy is achieved.

Let the following assumptions be fulfilled:

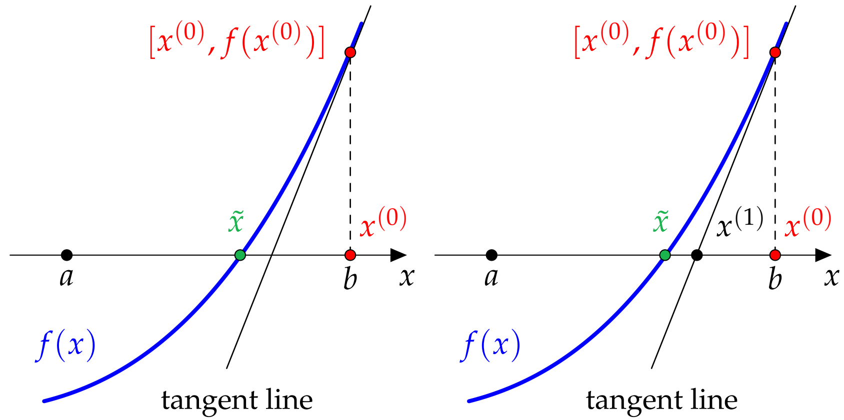
1. the first derivative f' does not change sign on the interval (a, b) (i.e. function f is either increasing or decreasing on (a, b));
2. the second derivative f'' does not change sign on the interval (a, b) (i.e. function f is either convex or concave on (a, b));
3. it holds $f(a) \cdot f(b) < 0$;
4. it holds $\left|\dfrac{f(a)}{f'(a)}\right| < b - a$ and $\left|\dfrac{f(b)}{f'(b)}\right| < b - a$.

Then the sequence $\{x^k\}$ calculated using the formula

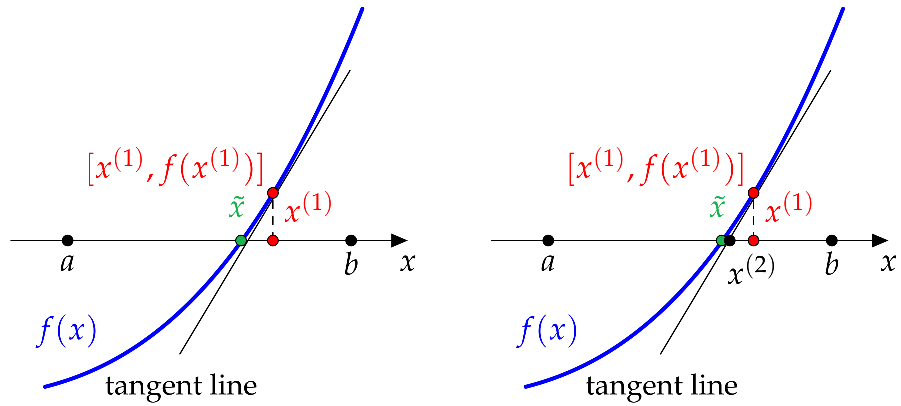
$$x^{(k+1)} = x^{(k)} - \frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}$$

converges for an arbitrary initial approximation $x^{(0)} \in [a, b]$.

For $k = 0$



For $k = 1$



The calculation is terminated when the given accuracy ε is achieved, i.e. when

$$|x^{(k)} - x^{(k-1)}| \leq \varepsilon\,.$$

18 – Root-finding for non-linear equations, Newton method

Example

Find all roots of the equation

$$x^3 - \ln(10 - x) = 0$$

using the Newton method with the accuracy $\varepsilon = 10^{-6}$.

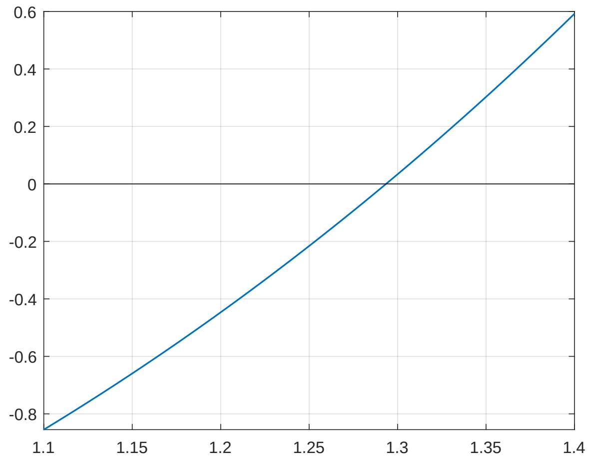
We already know that the root lies in the interval $[1.2, 1.3]$. In the beginning we have to verify the Newton method assumptions, so we calculate the first and the second derivative of the function f,

$$f(x) = x^3 - \ln(10-x), \quad f'(x) = 3 \cdot x^2 + \frac{1}{10-x}, \quad f''(x) = 6 \cdot x + \frac{1}{(10-x)^2},$$

farther we check condition

$$\left|\frac{f(a)}{f'(a)}\right| = \left|\frac{-0.4468}{4.4336}\right| = 0.1008 > 0.1 = b - a\,.$$

Thes condition is not fulfilled and that is why we have to shorten the interval in which the searched root is separated.



We can see that the searched root lies in the interval $[1.25, 1.3]$. Now we start to verify the assumptions on the new interval $[1.25, 1.3]$:

$$\left|\frac{f(a)}{f'(a)}\right| = \left|\frac{-0.2159}{4.8018}\right| = 0.0450 < 0.05 = b - a$$

$$\left|\frac{f(b)}{f'(b)}\right| = \left|\frac{0.0337}{5.1849}\right| = 0.0065 < 0.05 = b - a$$

We tabulate values of the first and the second derivative rounded to two decimal place.

x	$f'(x)$	$f''(x)$
1.25	4.80	7.51
1.26	4.88	7.57
1.27	4.95	7.63
1.28	5.03	7.69
1.29	5.11	7.75
1.3	5.19	7.81

From the table we can deduce that

$$f'(x) > 0 \text{ on } [1.25, 1.3]$$

$$f''(x) > 0 \text{ on } [1.25, 1.3]\,.$$

Thus, all assumptions of the Newton method are verified.

We choose the initial approximation $x^{(0)} = b = 1.3$.
We calculate the first approximation $x^{(1)}$

$$x^{(1)} = x^{(0)} - \frac{f\left(x^{(0)}\right)}{f'\left(x^{(0)}\right)} = 1.3 - \frac{0.03367697433946}{5.18494252873563} = 1.29350485098864$$

and the approximation error $|x^{(1)} - x^{(0)}| \doteq 0.007 > \varepsilon = 10^{-6}$. Calculations must continue.
We calculate the second approximation $x^{(2)}$

$$x^{(2)} = x^{(1)} - \frac{f\left(x^{(1)}\right)}{f'\left(x^{(1)}\right)} = 1.29350485098864 - \frac{0.00016453367995}{5.13432117883453} = 1.29347280513989$$

and the approximation error $|x^2 - x^1| \doteq 0.00003 > \varepsilon = 10^{-6}$. Calculations must continue.
We calculate the third approximation $x^{(3)}$

$$x^{(3)} = x^{(2)} - \frac{f\left(x^{(2)}\right)}{f'\left(x^{(2)}\right)} = 1.29347280513989 - \frac{0.00000000399178}{5.13407205040059} = 1.29347280436238$$

and the approximation error $|x^{(3)} - x^{(2)}| \doteq 0.0000000008 = 8 \cdot 10^{-10} < \varepsilon = 10^{-6}$. The given accuracy is achieved and calculations can be terminated.
We write obtained data into the table:

k	$x^{(k)}$	$\lvert x^{(k)} - x^{(k-1)}\rvert$
0	1.3	—
1	1.29350485098864	$0.007 > 10^{-6}$
2	1.29347280513989	$0.00003 > 10^{-6}$
3	1.29347280436238	$0.0000000008 \leq 10^{-6}$

The resulting approximation of the searched root is

$$\tilde{x} = 1.293473 \pm 10^{-6}.$$

20 – Root-finding for non-linear equations, Newton method

Example

Find all roots of the equation

$$x - 4\cos^2(x) = 0$$

using the Newton method with the accuracy $\varepsilon = 10^{-8}$.

From the previous examples we know that there are three roots in intervals $[1,2]$, $[2,3]$ and $[3,4]$.
At first we find the root that lies in the interval $[1,2]$. Because the Newton method assumptions are quite strong, it is better to work with shorter intervals. Therefore we tabulate function values (rounded to two decimal places) of the function $f(x) = x - 4\cos^2(x)$ on the interval $[1,2]$ with the step 0.1 .

x	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
$f(x)$	-0.17	0.28	0.68	1.01	1.28	1.48	1.60	1.63	1.60	1.48	1.30

We can see that the root has to lie in the interval $[1,1.1]$. We calculate the first and the second derivative

$$\begin{aligned}
f(x) &= x - 4\cos^2(x),\\
f'(x) &= 1 + 8\sin(x)\cos(x) = 1 + 4\sin(2x),\\
f''(x) &= 8\cos(2x)
\end{aligned}$$

and verify the assumptions for the chosen interval $[1,1.1]$:

$$f(a)\cdot f(b) = -0.17\cdot 0.28 < 0$$

$$\left|\frac{f(a)}{f'(a)}\right| = \left|\frac{1-4\cos^2(1)}{1+4\sin(2\cdot 1)}\right| = 0.04 < 0.1 = b-a$$

$$\left|\frac{f(b)}{f'(b)}\right| = \left|\frac{1.1-4\cos^2(1.1)}{1+4\sin(2\cdot 1.1)}\right| = 0.07 < 0.1 = b-a$$

Next we tabulate values of the first and the second derivative rounded to two decimal places

x	$f'(x)$	$f''(x)$
1.00	4.64	-3.33
1.01	4.60	-3.47
1.02	4.57	-3.62
1.03	4.53	-3.76
1.04	4.50	-3.90
1.05	4.45	-4.04
1.06	4.41	-4.18
1.07	4.37	-4.32
1.08	4.33	-4.45
1.09	4.28	-4.58
1.10	4.23	-4.71

and we can observe that both derivatives do not change signs on the mentioned interval

$$f'(x) > 0 \text{ on } [1,1.1]$$

$$f''(x) < 0 \text{ on } [1,1.1]\,.$$

Thus, all assumptions of the Newton method are verified.

We choose the initial approximation $x^{(0)} = a = 1$.
We calculate the first approximation $x^{(1)}$

$$x^{(1)} = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})} = 1 - \frac{1-4\cos^2(1)}{1+4\sin(2\cdot 1)} = 1.0361655092$$

and the approximation error $|x^{(1)} - x^{(0)}| \doteq 0.04 > \varepsilon = 10^{-8}$. Calculations must continue.

21 – Root-finding for non-linear equations, Newton method

We calculate the second approximation $x^{(2)}$

$$x^{(2)} = x^{(1)} - \frac{f(x^{(1)})}{f'(x^{(1)})} = 1.0361655092 - \frac{1.0361655092 - 4\cos^2(1.0361655092)}{1 + 4\sin(2 \cdot 1.0361655092)} = 1.0366737657$$

and the approximation error $|x^{(2)} - x^{(1)}| \doteq 0.0005 > \varepsilon = 10^{-8}$. Calculations must continue.
We calculate the third approximation $x^{(3)}$

$$x^{(3)} = x^{(2)} - \frac{f(x^{(2)})}{f'(x^{(2)})} = 1.0366737657 - \frac{1.0366737657 - 4\cos^2(1.0366737657)}{1 + 4\sin(2 \cdot 1.0366737657)} = 1.0366738760$$

and the approximation error $|x^{(3)} - x^{(2)}| \doteq 0.0000001 > \varepsilon = 10^{-8}$. Calculations must continue.
We calculate the fourth approximation $x^{(4)}$:

$$x^{(4)} = x^{(3)} - \frac{f(x^{(3)})}{f'(x^{(3)})} = 1.0366738760 - \frac{1.0366738760 - 4\cos^2(1.0366738760)}{1 + 4\sin(2 \cdot 1.0366738760)} = 1.0366738760$$

and the approximation error $|x^{(4)} - x^{(3)}| \doteq 0.00000000000001 < \varepsilon = 10^{-8}$. The given accuracy is achieved and calculations can be terminated.
We write obtained data into the table:

k	$x^{(k)}$	$\lvert x^{(k)} - x^{(k-1)}\rvert$
0	1	—
1	1.0361655092	$0.04 > 10^{-8}$
2	1.0366737657	$0.0005 > 10^{-8}$
3	1.0366738760	$0.0000001 > 10^{-8}$
4	1.0366738760	$0.00000000000001 \leq 10^{-8}$

The resulting approximation of the searched root is

$$\tilde{x} = 1.03667388 \pm 10^{-8}.$$

The other two roots can be found in analogous way. Approximations of all roots of the given equation are:

$$1.03667388 \pm 10^{-8}, \qquad 2.47646805 \pm 10^{-8}, \qquad 3.50214739 \pm 10^{-8}.$$

22 – Root-finding for non-linear equations, Newton method

Example

Find all roots of the equation

$$x - 4\cos^2(x) = 0$$

using the Newton method with the accuracy $\varepsilon = 10^{-8}$.
Use the MATLAB to solve the problem.

Due to previous separation we know that there are three roots in intervals $[1,2]$, $[2,3]$ and $[3,4]$.
At first we attend to the root from the interval $[3,4]$.
We input end-points of this interval.

```
>> a=3;
>> b=4;
```

Then we calculate the first and the second derivative of the given function.

$$\begin{aligned} f(x) &= x - 4\cos^2(x) \\ f'(x) &= 1 + 8\cos(x)\sin(x) \\ f''(x) &= -8\sin^2(x) + 8\cos^2(x) \end{aligned}$$

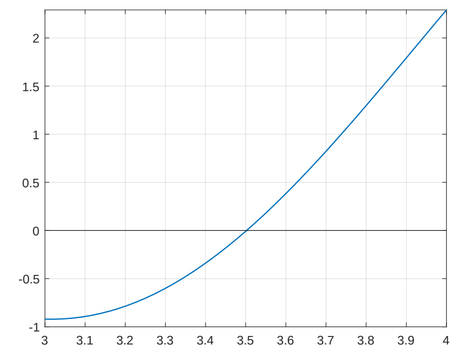
```
>> df=@(x)1+8*cos(x).*sin(x)
df =
    @(x)1+8*cos(x).*sin(x)
>> ddf=@(x)-8*sin(x).^2+8*cos(x).^2
ddf =
    @(x)-8*sin(x).^2+8*cos(x).^2
```

Now we can start verifying the assumptions.
We want to verify that values of derivatives do not change signs on the interval $[3,4]$.

Therefore we generate points from this interval with the step 0.1, input these as the vector x and calculate corresponding values of the first derivative.

```
>> x=a:0.1:b
x =
  Columns 1 through 6
 3.0000    3.1000    3.2000    3.3000    3.4000    3.5000
  Columns 7 through 11
 3.6000    3.7000    3.8000    3.9000    4.0000
>> df(x)
ans =
  Columns 1 through 6
-0.1177    0.6676    1.4662    2.2462    2.9765    3.6279
  Columns 7 through 11
 4.1747    4.5948    4.8717    4.9942    4.9574
```

However, the values of the first derivative change signs on $[3,4]$, so we have to shorten the interval of separation and to verify assumptions for this shorter one. We plot the graph of the given function on $[3,4]$.



It is obvious that the searched root lies in the interval $[3.4, 3.6]$ where we again try to verify assumptions.

We input end-points of the new interval.

```
>> a=3.4;
>> b=3.6;
```

We generate points from $[3.4, 3.6]$ with the step 0.01 and calculate values of the first derivative at these points

```
>> x=a:0.01:b;
>> df(x)
ans =
  Columns 1 through 6
 2.9765    3.0456    3.1139    3.1814    3.2480    3.3138
  Columns 7 through 12
 3.3785    3.4424    3.5053    3.5671    3.6279    3.6877
  Columns 13 through 18
 3.7464    3.8040    3.8605    3.9159    3.9701    4.0230
  Columns 19 through 21
 4.0748    4.1254    4.1747
```

as well as values of the second derivative

```
>> ddf(x)
ans =
  Columns 1 through 6
 6.9552    6.8747    6.7915    6.7056    6.6170    6.5258
  Columns 7 through 12
 6.4320    6.3355    6.2366    6.1351    6.0312    5.9249
  Columns 13 through 18
 5.8162    5.7052    5.5919    5.4764    5.3587    5.2388
  Columns 19 through 21
 5.1168    4.9928    4.8668
```

We can see that both derivatives do not change signs on the whole interval $[3.4, 3.6]$.

The next step is to verify the condition $f(a) \cdot f(b) < 0$ that guarantees the root existence.

```
>> f(a)*f(b)
ans =
   -0.1299
```

In the end we check validity of conditions $\left|\frac{f(a)}{f'(a)}\right| < b - a$ and $\left|\frac{f(b)}{f'(b)}\right| < b - a$.

```
>> abs(f(a)/df(a))
ans =
    0.1138
>> abs(f(b)/df(b))
ans =
    0.0918
```

Both values are less than $b - a = 3.6 - 3.4 = 0.2$.

Thus, all assumptions of the Newton method are verified for the interval $[3.4, 3.6]$ and the sequence of approximations will converge to the given equation root for an arbitrary initial approximation $x^{(0)} \in [3.4.3.6]$.

Because the given accuracy is 10^{-8}, we need to know all output values with higher precision. Thas is why we set up longer form of outputs using the statement `format long`.
We input the initial approximation that can be arbitrary chosen from the interval $[3.4, 3.6]$. We choose the left end-point that is saved as a.

```
>> format long
>> x0=a
x0 =
   3.40000000000000
```

We calculate the first approximation and the approximation error. If this error is greater than given ε then the calculation continue.

```
>> x1=x0-f(x0)/df(x0)
x1 =
   3.51382505776211
>> abs(x1-x0)
ans =
   0.11382505776211
```

We calculate next approximations and corresponding approximation errors. We test if the error is greater than given ε in each step.

```
>> x2=x1-f(x1)/df(x1)
x2 =
   3.50225628403900
>> abs(x2-x1)
ans =
   0.01156877372312
```

```
>> x3=x2-f(x2)/df(x2)
x3 =
   3.50214740099497
>> abs(x3-x2)
ans =
    1.088830440254540e-004
>> x4=x3-f(x3)/df(x3)
x4 =
   3.50214739121355
>> abs(x4-x3)
ans =
    9.781422338761558e-009
```

We write the obtained data into a table.

k	$x^{(k)}$	$\lvert x^{(k)} - x^{(k-1)} \rvert$
0	3.4	—
1	3.51382505776211	$0.11382505776211 > 10^{-8}$
2	3.50225628403900	$0.01156877372312 > 10^{-8}$
3	3.50214740099497	$1.088830440254540e-004 > 10^{-8}$
4	3.50214739121355	$9.781422338761558e-009 \leq 10^{-8}$

The given accuracy $\varepsilon = 10^{-8}$ is achieved in the fourth step where the calculation is terminated. We round the value of $x^{(4)}$ to eight decimal places.

The resulting approximation of the searched root is

$$\tilde{x} = 3.50214739 \pm 10^{-8}.$$

The other two roots can be found in analogous way. Approximations of all roots of the given equation are:

$$1.03667388 \pm 10^{-8}, \qquad 2.47646805 \pm 10^{-8}, \qquad 3.50214739 \pm 10^{-8}.$$

25 – Root-finding for non-linear equations, Newton method

Exercise

Find all roots of the equation

$$2x^3 - x^2 - x - 1 = 0$$

using the Newton method with the accuracy $\varepsilon = 10^{-8}$.

26 – Root-finding for non-linear equations, Newton method

Exercise

Find all roots of the equation

$$2x^3 - x^2 - x - 1 = 0$$

using the Newton method with the accuracy $\varepsilon = 10^{-8}$.
Use the MATLAB to solve the problem.

27 – Root-finding for non-linear equations

Exercise

Find all roots of the equation

$$2x^3 - x^2 - x - 1 = 0$$

using the bisection method and the Newton method, both with the accuracy $\varepsilon = 10^{-4}$.
Compare obtained results.
Use the MATLAB to solve the problem.

28 – Root-finding for non-linear equations

Exercise

Find all roots of the given equation using the bisection method with the accuracy 10^{-3} and using the Newton method with the accuracy 10^{-8}. Compare obtained results.
Use the MATLAB to solve the problem.

1. $$5e^x - xe^x - 4 = 0$$

2. $$(x-1)^2 - \cos(x) - 2 = 0$$

3. $$x - 3\sin(x) - 1 = 0$$

4. $$\ln(x) + x^2 - 5x + 5 = 0$$

5. $$x^2 - \ln(x+1) - 0.2 = 0$$

6. $$4\cos^2(x) - x^2 + x = 0$$

7. $$x^2 - x - \sin(x) - 1 = 0$$

8. $$(x-0.1)^4 - \sin^2(x) - 1 = 0$$

9. $$\ln(x) - (x-3)^2 = 0$$

10. $$x^2 - 7\ln(x) - 3 = 0$$

11. $$(x-1)^2 - 2\sin(x) = 0$$

12. $$e^x - 7x^2 + 2 = 0$$

13. $$e^x - x - 4 = 0$$

14. $$\ln(x+2) - x^2 + 2.5 = 0$$

15. $$(x-1)^4 - \ln(x) - 1 = 0$$

16. $$5\ln(x^2) + (x-2)^3 = 0$$

17. $$x^2 - 3x + 2 - e^{-x^2} = 0$$

18. $$x^2 - 2x - e^{-x^2} = 0$$

19. $$3\cos(x) + 1 + \sqrt{x} = 0$$

20. $$3\sin^2(x) - 1 - \sqrt{x} = 0$$

21. $$7\ln^2(x) - 1 - \sqrt{x} = 0$$

22. $$\arccos\left(\frac{x}{2}\right) + 3x^2 - 4 = 0$$

Worksheets for Numerical methods

Polynomial Interpolation

30 – Interpolation

The interpolation problem
Given $n+1$ pairs (x_i, y_i) of distinct nodes x_i and corresponding values y_i, the problem consists of finding a polynomial $p_n = p_n(x)$ that fulfils **the interpolation equalities**

$$p_n(x_i) = y_i, \quad i = 0, \ldots, n\,,$$

i.e. a polynomial whose graph passes the given points.

There exists unique interpolating polynomial of degree at most n. We introduce three different ways how to find this polynomial.

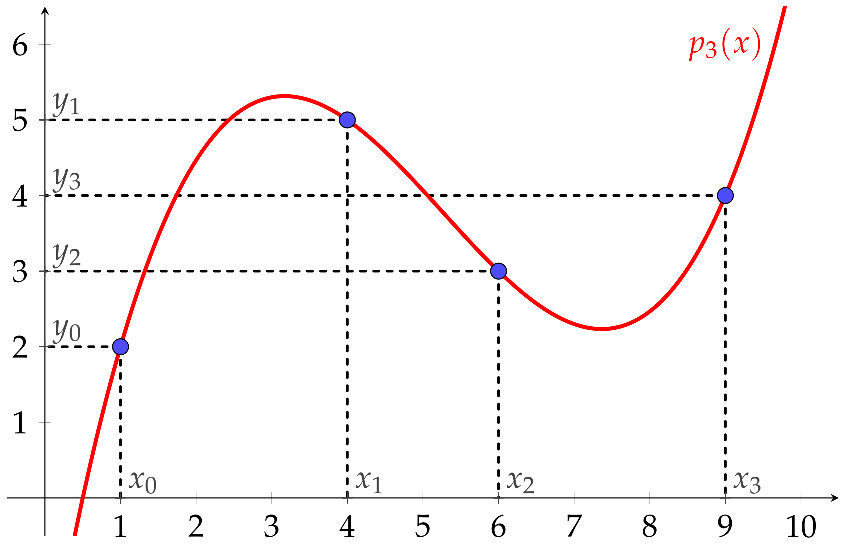


Figure: Interpolating polynomial of degree 3 for given data

	i=0	i=1	i=2	i=3
x_i	1	4	6	9
y_i	2	5	3	4

31 – Interpolating polynomial in the standard form

Interpolating polynomial in the standard form

The distinct nodes x_i and corresponding values y_i, $i = 0, \dots, n$ are given. Substituting the standard form of a polynomial

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

into the interpolation equalities $p_n(x_i) = y_i$ we obtain the system of linear equations

$$a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n = y_i, \quad i = 0, \dots, n,$$

that can be written in the matrix form as

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

The solution of this system of linear equations represents the coefficients $a_0, a_1, \dots, a_n \in \mathbb{R}$ of the interpolating polynomial.

32 – Interpolating polynomial in the standard form

Example

Find the interpolating polynomial in the standard form for the data

	i=0	i=1
x_i	2	4
y_i	1	-5

We seek for a linear interpolating polynomial of the form

$$p_1(x) = a_0 + a_1 x$$

that we substitute into the interpolation equalities $p_n(x_i) = y_i$:

$$\begin{aligned} p_1(2) &= 1 \quad \Rightarrow \quad a_0 + a_1 \cdot 2 = 1 \\ p_1(4) &= -5 \quad \Rightarrow \quad a_0 + a_1 \cdot 4 = -5 \end{aligned}$$

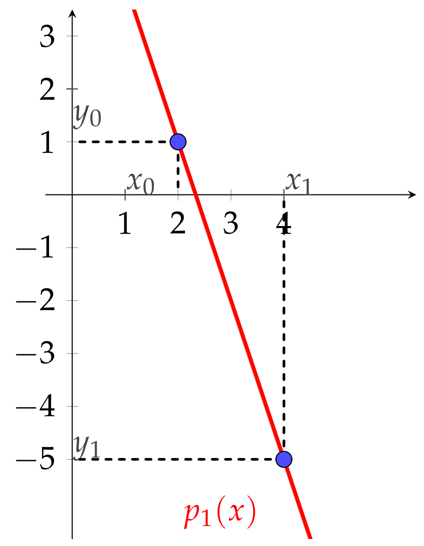
The obtained system of linear equations can be written in the matrix form:

$$\begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 \\ -5 \end{pmatrix}$$

The solution of this system is $a_0 = 7,\ a_1 = -3$.

The result is

$$p_1(x) = 7 - 3x\ .$$



Finally we check that the interpolation equalities are really fulfilled. We substitute the given nodes x_i into the interpolating polynomial.

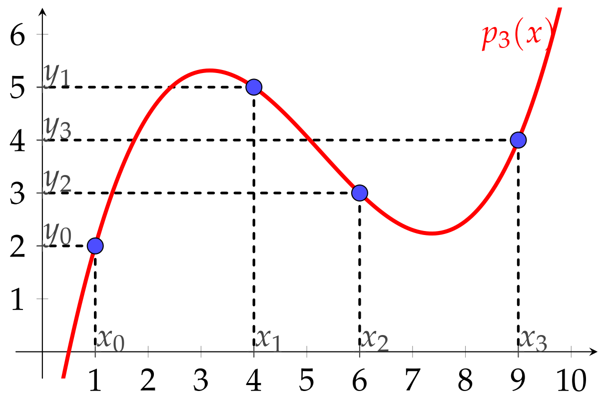
$$\begin{aligned} p_1(2) &= 7 - 3 \cdot 2 = 1 \\ p_1(4) &= 7 - 3 \cdot 4 = -5 \end{aligned}$$

33 – Interpolating polynomial in the standard form

Example

Find the interpolating polynomial in the standard form for the data

	i=0	i=1	i=2	i=3
x_i	1	4	6	9
y_i	2	5	3	4



We seek for a cubic interpolating polynomial

$$p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

that we substitute into the interpolation equalities $p_n(x_i) = y_i$

$$\begin{aligned}
p_3(1) &= 2 \quad \Rightarrow \quad a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3 = 2 \\
p_3(4) &= 5 \quad \Rightarrow \quad a_0 + a_1 \cdot 4 + a_2 \cdot 4^2 + a_3 \cdot 4^3 = 5 \\
p_3(6) &= 3 \quad \Rightarrow \quad a_0 + a_1 \cdot 6 + a_2 \cdot 6^2 + a_3 \cdot 6^3 = 3 \\
p_3(9) &= 4 \quad \Rightarrow \quad a_0 + a_1 \cdot 9 + a_2 \cdot 9^2 + a_3 \cdot 9^3 = 4
\end{aligned}$$

The system of linear equations can be written in the matrix form:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 16 & 64 \\ 1 & 6 & 36 & 216 \\ 1 & 9 & 81 & 729 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 3 \\ 4 \end{pmatrix}$$

We solve the system and we obtain the interpolating polynomial coefficients $a_0 = -2.6$, $a_1 = 5.8\overline{3}$, $a_2 = -1.31\overline{6}$, $a_3 = 0,08\overline{3}$.

The resulting interpolating polynomial is (coefficients are rounded to three decimal places)

$$p_3(x) = -2.6 + 5.833x - 1.317x^2 + 0.083x^3 \, .$$

34 – Interpolating polynomial in the standard form

Exercise

Find the interpolating polynomial in the standard form for the data

	i=0	i=1	i=2
x_i	0	3	4
y_i	2	1	5

35 – Interpolating polynomial in the standard form

Example

Find the interpolating polynomial in the standard form for the data

	i=0	i=1	i=2
x_i	0	3	4
y_i	2	1	5

Use the MATLAB to solve the problem.

At first we input the nodes x_i as the vector `x` and the values y_i as the vector `y`.

```
>> x = [0; 3; 4]
>> y = [2; 1; 5]
```

The coefficients of the interpolating polynomial are the solution of the system of linear equations.

```
>> M = [ones(3,1) x x.^2]
M =
    1    0    0
    1    3    9
    1    4   16
>> a = M\y
>> a
a =
   2.0000   3.5833   1.0833
```

The coefficients a_i are obviously rational numbers, that is why we write these as fractions.

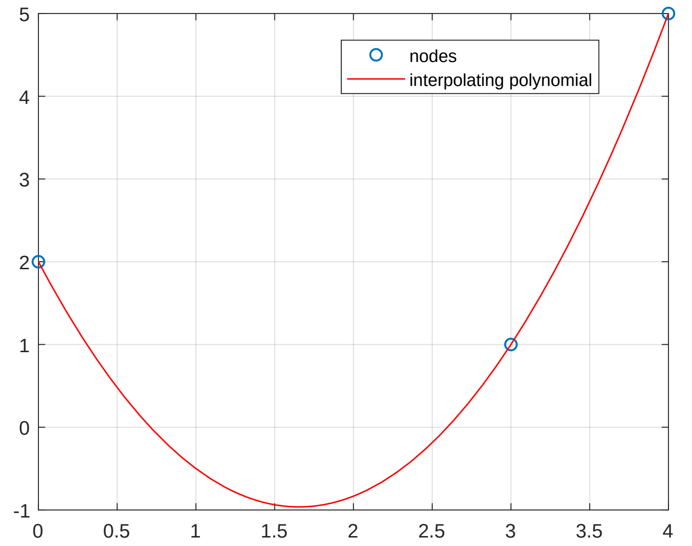
```
>> format rat
>> a
a =
       2   -43/12   13/12
```

The result is

$$p_2(x) = 2 - \frac{43}{12}x + \frac{13}{12}x^2 .$$

We input the coefficients of the interpolating polynomial as the vector `p` and plot the graph of the polynomial on the interval $[x_0, x_2] = [0, 4]$.

```
>> plot(x,y,'o')
>> grid on, hold on
>> p = @(x)a(1)+a(2)*x+a(3)*x.^2;
>> fplot(p, [0 4], 'r')
>> legend('nodes','interpolating polynomial')
```



36 – Interpolating polynomial in the standard form

Exercise

Find the interpolating polynomial in the standard form for the data

	i=0	i=1	i=2	i=3
x_i	1	4	6	9
y_i	2	5	3	4

Use the MATLAB to solve the problem.

37 – Interpolating polynomial in the Lagrange form

Interpolating polynomial in the Lagrange form

The unique interpolating polynomial can be written in the Lagrange form

$$p_n(x) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_n l_n(x),$$

where $l_0(x), l_1(x), \ldots, l_n(x)$ are the Lagrange basis of the interpolation problem, for $i = 1, \ldots, n$:

$$l_i(x) = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}\,.$$

The Lagrange basis have the following properties for $i, j = 1, \ldots, n$:

- $l_i(x)$ is the polynomial of degree n,
- $l_i(x_i) = 1$,
- $l_i(x_j) = 0$ for $i \neq j$.

38 – Interpolating polynomial in the Lagrange form

Example

Find the interpolating polynomial in the Lagrange form for the data

	i=0	i=1	i=2
x_i	0	3	4
y_i	2	1	5

Calculate the value of the polynom at the point $x = 2$.

We write the Lagrange basis corresponding to single nodes:

$$\begin{aligned} l_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-3)(x-4)}{(0-3)(0-4)} = \frac{1}{12}(x-3)(x-4), \\ l_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-0)(x-4)}{(3-0)(3-4)} = -\frac{1}{3}x(x-4), \\ l_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-0)(x-3)}{(4-0)(4-3)} = \frac{1}{4}x(x-3), \end{aligned}$$

The interpolating polynomial is

$$\begin{aligned} p_2(x) &= y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) \\ &= 2 \cdot \frac{1}{12}(x-3)(x-4) + 1 \cdot \left(-\frac{1}{3}x(x-4)\right) + 5 \cdot \frac{1}{4}x(x-3) \\ &= \frac{1}{6}(x-3)(x-4) - \frac{1}{3}x(x-4) + \frac{5}{4}x(x-3). \end{aligned}$$

The resulting form of the interpolating polynomial is

$$p_2(x) = \frac{1}{6}(x-3)(x-4) - \frac{1}{3}x(x-4) + \frac{5}{4}x(x-3)\ .$$

We calculate the value of the polynom p_2 at the point $x = 2$.

$$\begin{aligned} p_2(2) &= \frac{1}{6}(2-3)(2-4) - \frac{1}{3}2(2-4) + \frac{5}{4}2(2-3) \\ &= -\frac{5}{6} \end{aligned}$$

39 – Interpolating polynomial in the Lagrange form

Example

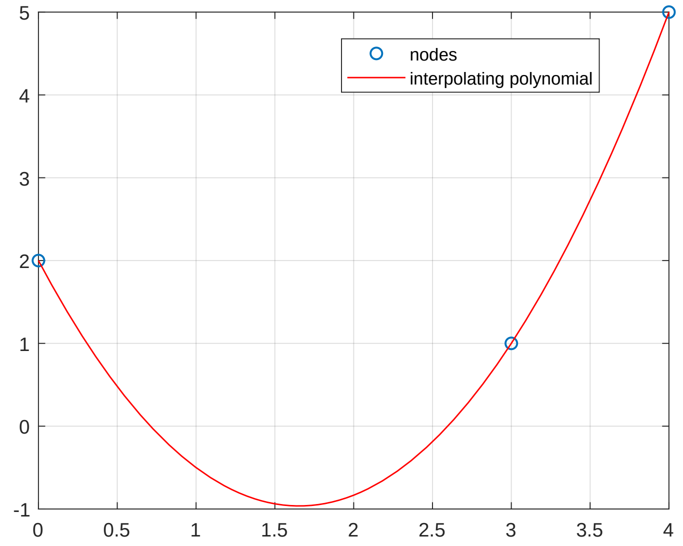
Plot a graph of the interpolating polynomial from the previous example together with the given data.
Use the MATLAB to solve the problem.

We input the given data.

```
>> x = [0; 3; 4]
>> y = [2; 1; 5]
```

We plot the given discrete points and the graph of the interpolating polynomial.

```
>> plot(x,y,'o')
>> grid on, hold on
>> p = @(x) 1/6*(x-3).*(x-4)-1/3*x.*(x-4)+5/4*x.*(x-3)
>> fplot(p,[0 4], 'r')
>> legend('nodes','interpolating polynomial')
```



40 – Interpolating polynomial in the Lagrange form

Example

Find the interpolating polynomial in the Lagrange form for the data

	i=0	i=1	i=2	i=3
x_i	1	4	6	9
y_i	2	5	3	4

The Lagrange basis corresponding to single nodes is:

$$
\begin{aligned}
l_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = \frac{(x-4)(x-6)(x-9)}{(1-4)(1-6)(1-9)} = -\frac{1}{120}(x-4)(x-6)(x-9),\\
l_1(x) &= \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} = \frac{(x-1)(x-6)(x-9)}{(4-1)(4-6)(4-9)} = \frac{1}{30}(x-1)(x-6)(x-9),\\
l_2(x) &= \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} = \frac{(x-1)(x-4)(x-9)}{(6-1)(6-4)(6-9)} = -\frac{1}{30}(x-1)(x-4)(x-9),\\
l_3(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} = \frac{(x-1)(x-4)(x-6)}{(9-1)(9-4)(9-6)} = \frac{1}{120}(x-1)(x-4)(x-6),
\end{aligned}
$$

The interpolating polynomial is:

$$
\begin{aligned}
p_3(x) =& y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) + y_3 l_3(x)\\
=& 2\cdot\left(-\frac{1}{120}(x-4)(x-6)(x-9)\right) + 5\cdot\frac{1}{30}(x-1)(x-6)(x-9) + 3\cdot\left(-\frac{1}{30}(x-1)(x-4)(x-9)\right) + 4\cdot\frac{1}{120}(x-1)(x-4)(x-6)\\
=& -\frac{1}{60}(x-4)(x-6)(x-9) + \frac{1}{6}(x-1)(x-6)(x-9) - \frac{1}{10}(x-1)(x-4)(x-9) + \frac{1}{30}(x-1)(x-4)(x-6)
\end{aligned}
$$

The resulting form of the interpolating polynomial is

$$
p_3(x) = -\frac{1}{60}(x-4)(x-6)(x-9) + \frac{1}{6}(x-1)(x-6)(x-9) - \frac{1}{10}(x-1)(x-4)(x-9) + \frac{1}{30}(x-1)(x-4)(x-6)
$$

41 – Interpolating polynomial in the Lagrange form

Exercise

Find the interpolating polynomial in the Lagrange form for the data

	i=0	i=1	i=2
x_i	1	2	3
y_i	4	5	-1

42 – Interpolating polynomial in the Newton form

Interpolating polynomial in the Newton form

The interpolating polynomial of degree n in the Newton form is defined by the formula

$$p_n(x) = y_0 + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1) + f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2) + \cdots + f[x_n, \dots, x_0](x - x_0)(x - x_1)\cdots(x - x_{n-1}),$$

where $f[x_1, x_0]$ is the 1st divided difference, $f[x_2, x_1, x_0]$ is the 2nd divided difference, up to $f[x_n, \dots, x_0]$ is the n-th divided difference.

Example for $n = 4$.
The calculation of the divided differences for $n = 4$ is realized in the following table:

i	x_i	y_i	1st $f[x_{i+1}, x_i]$	2nd $f[x_{i+2}, x_{i+1}, x_i]$	3rd $f[x_{i+3}, x_{i+2}, x_{i+1}, x_i]$	4th $f[x_{i+4}, x_{i+3}, x_{i+2}, x_{i+1}, x_i]$
0	x_0	f_0	$f[x_1, x_0] = \frac{y_1 - y_0}{x_1 - x_0}$	$f[x_2, x_1, x_0] = \frac{f[x_2,x_1] - f[x_1,x_0]}{x_2 - x_0}$	$f[x_3, x_2, x_1, x_0] = \frac{f[x_3,x_2,x_1] - f[x_2,x_1,x_0]}{x_3 - x_0}$	$f[x_4, x_3, x_2, x_1, x_0] = \frac{f[x_4,x_3,x_2,x_1] - f[x_3,x_2,x_1,x_0]}{x_4 - x_0}$
1	x_1	f_1	$f[x_2, x_1] = \frac{y_2 - y_1}{x_2 - x_1}$	$f[x_3, x_2, x_1] = \frac{f[x_3,x_2] - f[x_2,x_1]}{x_3 - x_1}$	$f[x_4, x_3, x_2, x_1] = \frac{f[x_4,x_3,x_2] - f[x_3,x_2,x_1]}{x_4 - x_1}$	
2	x_2	f_2	$f[x_3, x_2] = \frac{y_3 - y_2}{x_3 - x_2}$	$f[x_4, x_3, x_2] = \frac{f[x_4,x_3] - f[x_3,x_2]}{x_4 - x_2}$		
3	x_3	f_3	$f[x_4, x_3] = \frac{y_4 - y_3}{x_4 - x_3}$			
4	x_4	f_4				

The interpolating polynomial in the Newton form for $n = 4$ is defined as

$$p_4(x) = y_0 + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1) + f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2) + f[x_4, x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)(x - x_3)\,.$$

43 – Interpolating polynomial in the Newton form

Example

Find the interpolating polynomial in the Newton form for the data

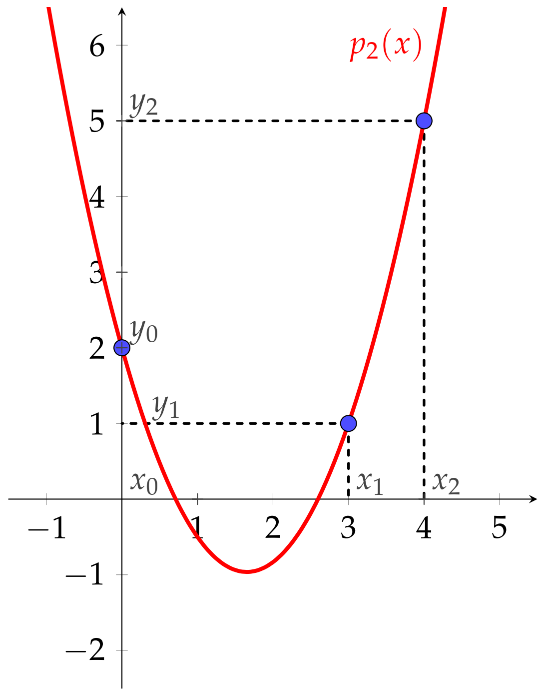
	i=0	i=1	i=2
x_i	0	3	4
f_i	2	1	5

The calculation of the divided differences is realized in the following table:

i	x_i	y_i	1st	2nd
0	0	2	$-\frac{1}{3}$	$\frac{13}{12}$
1	3	1	4	
2	4	5		

Using the values from the first row of the table we can write the interpolating polynomial.

$$p_2(x) = 2 - \frac{1}{3}(x-0) + \frac{13}{12}(x-0)(x-3) = 2 - \frac{1}{3}x + \frac{13}{12}x(x-3)$$



44 – Interpolating polynomial in the Newton form

Example

Add the node $x_3 = 1$ with the value $y_3 = \frac{3}{2}$ to the data from the previous example and find the interpolating polynomial in the Newton form.

We add the node and the value to the table of the divided differences and calculate the missing 1st divide difference $f[x_3, x_2]$, the 2nd divided difference $f[x_3, x_2, x_1]$ and the 3rd divided difference $f[x_3, x_2, x_1, x_0]$.

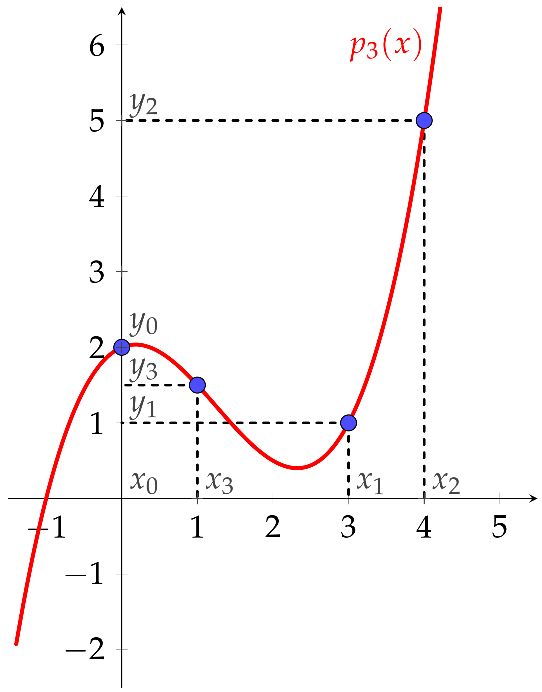
	x_i	y_i	1st	2nd	3rd
$i = 0$	0	2	$-\frac{1}{3}$	$\frac{13}{12}$	$\frac{1}{3}$
$i = 1$	3	1	4	$\frac{17}{12}$	
$i = 2$	4	5	$\frac{7}{6}$		
$i = 3$	1	$\frac{3}{2}$			

Now we add the corresponding next term to the interpolating polynomial.

$$p_3(x) = 2 - \frac{1}{3}(x-0) + \frac{13}{12}(x-0)(x-3) + \frac{1}{3}(x-0)(x-3)(x-4).$$

The resulting interpolating polynomial is

$$p_3(x) = 2 - \frac{1}{3}x + \frac{13}{12}x(x-3) + \frac{1}{3}x(x-3)(x-4).$$



Compare this graph with the graph from the previous example – the added node has changed the shape of the whole graph.

45 – Interpolating polynomial in the Newton form

Example

Find the interpolating polynomial in the Newton form for the data

	i=0	i=1	i=2
x_i	0	3	4
f_i	2	1	5

Use the MATLAB to solve the problem.

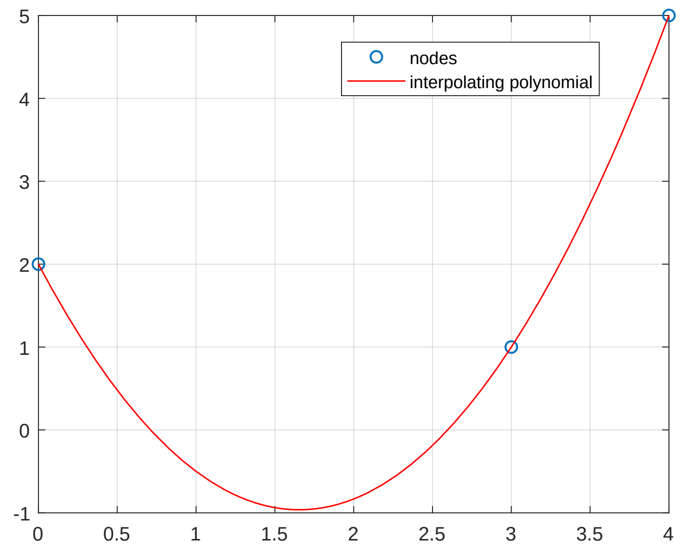
```
>> x=[0,3,4]
x =
     0     3     4
>> y=[2,1,5]
y =
     2     1     5
>> format rat
>> n=length(x);
```

The 0th divided differences are the function values in the nodes, i.e. y_i. The 1st divided differences are given by the formula $f[x_{i+1}, x_i] = \frac{y_{i+1}-y_i}{x_{i+1}-x_i}$.

```
>> df0=y
df0 =
       2               1              5
>> for i=1:n-1,df1(i)=(df0(i+1)-df0(i))/(x(i+1)-x(i));end
>> df1
df1 =
      -1/3            4
>> for i=1:n-2,df2(i)=(df1(i+1)-df1(i))/(x(i+2)-x(i));end
>> df2
df2 =
      13/12
```

Now we write the interpolating polynomial and plot its graph together with the given data.

```
>> format short
>> xg=x(1):0.01:x(3)
xg =
       0    0.0100    0.0200   .   .   .   3.9900    4.0000
>> yg=df0(1)+df1(1)*(xg-x(1))+df2(1)*(xg-x(1)).*(xg-x(2))
yg =
  2.0000    1.9643    1.9288   .   .   .   4.9493    5.0000
>> plot(x,y,'go')
>> hold on
>> plot(xg,yg)
>> legend('nodes','interpolating polynomial')
```



46 – Interpolating polynomial in the Newton form

Example

Add the node $x_3 = 1$ with the value $y_3 = \frac{3}{2}$ to the data from the previous example and find the interpolating polynomial in the Newton form.
Use the MATLAB to solve the problem.

We add the new data into the vectors `x` and `y`.

```
>> x=[x, 1]
x =
     0     3     4     1
>> y=[y, 1.5]
y =
    2.0000    1.0000    5.0000    1.5000
```

We calculate the missing 1st divided difference $f[x_3, x_2]$ and the 2nd one $f[x_3, x_2, x_1]$.

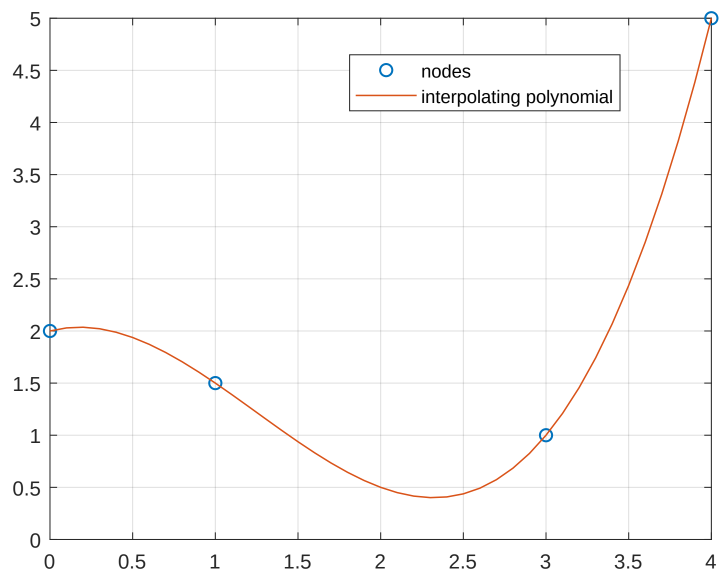
```
>> format rat
>> n=length(x);
>> df0=y
df0 =
       2      1      5     3/2
>> for i=1:n-1, df1(i)=(df0(i+1)-df0(i))/(x(i+1)-x(i)); end
>> df1
df1 =
      -1/3            4              7/6
>> for i=1:n-2, df2(i)=(df1(i+1)-df1(i))/(x(i+2)-x(i)); end
>>df2
df2 =
      13/12          17/12
```

We calculate the 3rd divided difference $f[x_3, x_2, x_1, x_0]$.

```
>> for i=1:n-3, df3(i)=(df2(i+1)-df2(i))/(x(i+3)-x(i)); end
>> df3
df3 =
       1/3
```

We extend the interpolating polynomial by the additional term and calculate the values of the polynomial in the points `xg`.

```
>> xg=min(x):0.01:max(x);
>> yg=df0(1)+df1(1)*(xg-x(1))+df2(1)*(xg-x(1)).*(xg-x(2))+df3(1)*(xg-x(1)).*(xg-x(2)).*(xg-x(3))
yg =
    2.0000    2.0040    2.0078 ... 4.9361   5.0000
>> plot(x,y,'go')
>> hold on
>> plot(xg,yg)
>> legend('nodes','interpolating polynomial')
```



48 – Interpolating polynomial in the Newton form

Exercise

Find the interpolating polynomial in the Newton form for the data

	i=0	i=1	i=2
x_i	1	2	3
y_i	4	5	-1

49 – Interpolating polynomial in the Newton form

Exercise

Find the interpolating polynomial in the Newton form for the data

	i=0	i=1	i=2	i=3
x_i	1	4	6	9
y_i	2	5	3	4

Use the MATLAB to solve the problem.

50 – Interpolating polynomial

Exercise

Find the interpolating polynomial in the standard, Lagrange and Newton forms for the data. Use the MATLAB to solve the problem.

1.

x_i	4.5	6	7.5	8	8.5
y_i	-37	44	20.1	34.7	-29.1

2.

x_i	-3	-2.5	-1	0	0.5
y_i	36.7	-12.8	-42.7	-30.1	-45.1

3.

x_i	-3	-2.5	-1.5	-1	0.5
y_i	32.3	-45.4	9.7	44.9	-21.2

4.

x_i	-6.5	-6	-5.5	-5	-3.5
y_i	-23.6	49.9	-28.9	-0.2	-21

5.

x_i	-5.5	-4	-2.5	-1.5	-1
y_i	-48.6	-21.2	31.6	48.5	-48.3

6.

x_i	-3.5	-2.5	-1.5	-1	0.5
y_i	25.3	15.9	-28.6	10.2	10.4

7.

x_i	1	1.5	2.5	3	4
y_i	18.5	17.7	37.6	-48.8	-19

8.

x_i	-6.5	-6	-4.5	-3	-2.5
y_i	-48.9	-27.3	1.6	-4.2	20.3

9.

x_i	-4	-3	-2.5	-2	-1
y_i	27	-18.7	13.8	48.6	0.2

10.

x_i	4.5	6	7.5	8	9.5
y_i	-34.4	-37.8	26.2	22.1	15.1

11.

x_i	-4.5	-3.5	-2	-1.5	-0.5
y_i	-46.5	-41.9	35	-16	-3.4

12.

x_i	-0.5	0	1.5	2.5	4
y_i	49.2	-12.7	3.1	-31.9	0.1

13.

x_i	-6	-5	-3.5	-2	-1.5
y_i	6.5	46.9	-47.7	37	-47.4

14.

x_i	-6	-5.5	-4	-3.5	-2
y_i	2.1	39.5	44.2	-16.5	-6.3

15.

x_i	-2	-1.5	-1	0	1.5
y_i	-14.2	-21.5	36.8	12.6	-25.9

16.

x_i	-5.5	-4.5	-4	-2.5	-1.5
y_i	-47.8	-23.8	-38.4	-43.1	35.2

17.

x_i	-3.5	-3	-1.5	-0.5	0
y_i	-48.8	38.9	36.6	-24.6	6.9

18.

x_i	-0.5	0.5	1	2	3.5
y_i	48	29.1	-34.8	33.3	-30.9

19.

x_i	-1	0.5	2	3	4
y_i	10.8	-32.5	-49.8	29	1.3

Worksheets for Numerical methods

Approximation by the least-squares method

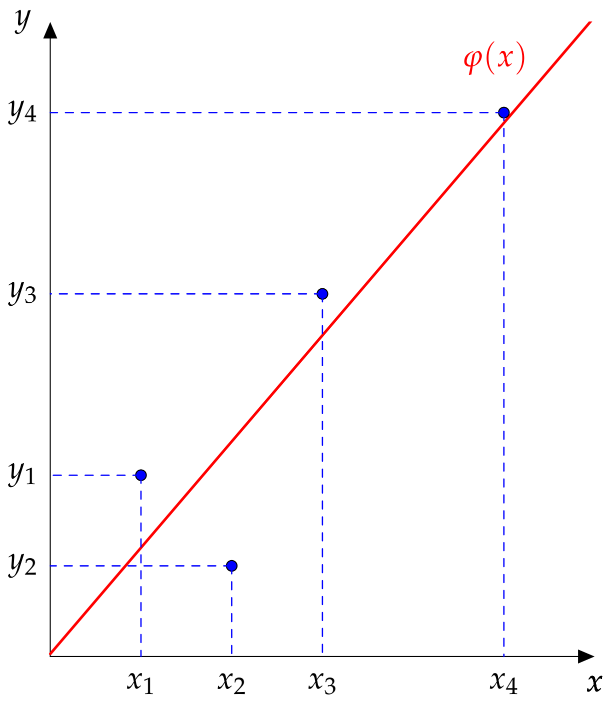
53 – Approximation by the least-squares method, linear approximation

The approximation problem
Given n pairs (x_i, y_i) of distinct nodes x_i and corresponding values y_i, the problem consists of finding a function $\varphi(x)$ that fulfils

$$\varphi(x_i) \approx y_i, \quad i = 1, \ldots, n.$$

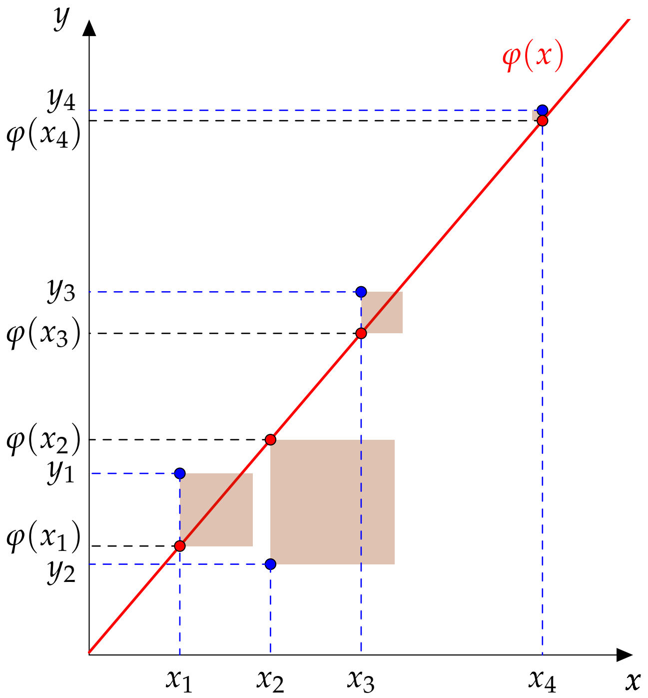
Linear approximation

Assume we are given n pairs (x_i, y_i), $i = 1, \ldots, n$ of distinct nodes x_i and corresponding values y_i. We want to find such values $c_1, c_2 \in \mathbb{R}$, that the linear function $\varphi(x) = c_1 + c_2 x$ is the best approximation of the given data in the least-squares sense.



54 – Approximation by the least-squares method, linear approximation

The following figure illustrates the given data and a straight line that represents the searched linear function $\varphi(x) = c_1 + c_2 x$.



We want to find the coefficients c_1, c_2 of the linear function $\varphi(x)$, for which the sum of areas of squares in the figure above is minimized. Because the area of the i-th square is $(c_1 + c_2 x_i - y_i)^2$, we are looking for a minimum of the **price function**

$$\Phi(c_1, c_2) = \sum_{i=1}^{n} (c_1 + c_2 x_i - y_i)^2 \,.$$

The price function Φ is quadratic, therefore its minimum exists and is unique.

We are to solve the problem to find a minimum of the function of two variables.
The minimum $[c_1, c_2]$ of the price function Φ must fulfil the equations

$$\frac{\partial}{\partial c_1}\Phi(c_1, c_2) = 0,$$
$$\frac{\partial}{\partial c_2}\Phi(c_1, c_2) = 0.$$

Having calculated the partial derivatives we obtain

$$2\sum_{i=1}^{n} (c_1 + c_2 x_i - y_i) = 0,$$
$$2\sum_{i=1}^{n} (c_1 + c_2 x_i - y_i) x_i = 0,$$

that is the system of linear equations for the unknown coefficients c_1, c_2:

$$c_1 \sum_{i=1}^{n} 1 + c_2 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i,$$
$$c_1 \sum_{i=1}^{n} x_i + c_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i,$$

This system is called the **normal system of equations** and can be rewritten in a matrix form:

$$\begin{pmatrix} \sum_{i=1}^{n} 1 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i x_i \end{pmatrix}$$

55 – Approximation by the least-squares method, linear approximation

Example

Approximate the data from the table

x_i	-2	-1	1	2
y_i	10	4	6	3

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x.$$

We write the normal system of equations in the matrix form

$$\begin{pmatrix} \sum_{i=1}^{4} 1 & \sum_{i=1}^{4} x_i \\ \sum_{i=1}^{4} x_i & \sum_{i=1}^{4} x_i^2 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{4} y_i \\ \sum_{i=1}^{4} y_i x_i \end{pmatrix}$$

and calculate the sums:

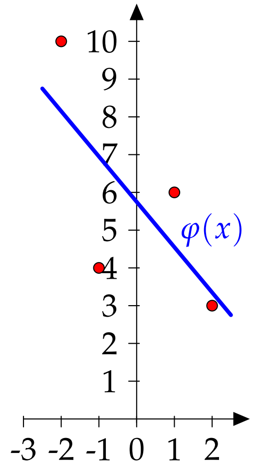
$$\begin{aligned}
\sum_{i=1}^{4} 1 &= 1 + 1 + 1 + 1 = 4 \\
\sum_{i=1}^{4} x_i &= -2 + (-1) + 1 + 2 = 0 \\
\sum_{i=1}^{4} x_i^2 &= (-2)^2 + (-1)^2 + 1^2 + 2^2 = 10 \\
\sum_{i=1}^{4} y_i &= 10 + 4 + 6 + 3 = 23 \\
\sum_{i=1}^{4} y_i x_i &= 10 \cdot (-2) + 4 \cdot (-1) + 6 \cdot 1 + 3 \cdot 2 = -12
\end{aligned}$$

The normal system of equations is:

$$\begin{pmatrix} 4 & 0 \\ 0 & 10 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 23 \\ -12 \end{pmatrix}$$

The unique solution of this system is $c_1 = \frac{23}{4}$, $c_2 = -\frac{6}{5}$.
Therefore the linear function that represents the best linear approximation of the given data in the least-squares method sense is

$$\varphi(x) = \frac{23}{4} - \frac{6}{5}x = 5.75 - 1.2x\ .$$



56 – Approximation by the least-squares method, linear approximation

Example

Approximate the data from the table

x_i	-2	-1	1	2
y_i	10	4	6	3

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x.$$

Use the MATLAB to solve the problem.

We input the given data in the MATLAB.

```
>> x=[-2 -1 1 2];
>> y=[10 4 6 3];
>> n=length(x)
n =
     4
```

We also need the matrix of the normal system of equations and the right hand sides vector.

```
>> M=[n sum(x); sum(x) sum(x.^2)]
M =
     4     0
     0    10

>> v=[sum(y); sum(y.*x)]
v =
    23
   -12
```

We solve the normal system of equations using one of many methods that the MATLAB offers.

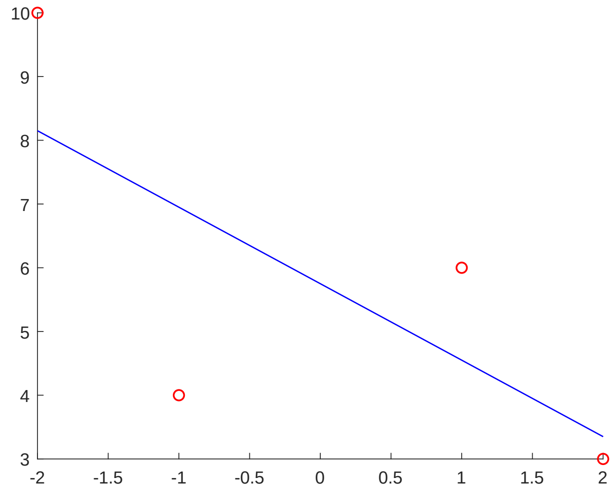
```
>> c=M\v
c =
    5.7500
   -1.2000
```

The best approximation in the least-squares sense has the form:

$$\varphi(x) = 5.75 - 1.2x$$

Finally we plot the given data as well as the found linear approximation

```
>> hold on
>> plot(x,y,'ro')
>> xg=x(1):0.01:x(n)
>> yg=c(1)+c(2)*xg;
>> plot(xg,yg)
```



57 – Approximation by the least-squares method, linear approximation

Exercise

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x.$$

58 – Approximation by the least-squares method, linear approximation

Exercise

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x.$$

Use the MATLAB to solve the problem.

59 – Approximation by the least-squares method, two functions

Approximation by two functions

Assume we are given n pairs (x_i, y_i), $i = 1, \ldots, n$ of distinct nodes x_i and corresponding values y_i as well as two functions $\varphi_1(x)$ a $\varphi_2(x)$. We want to find such values $c_1, c_2 \in \mathbb{R}$, that the function $\varphi(x) = c_1\varphi_1(x) + c_2\varphi_2(x)$ is the best approximation of the given data in the least-squares sense. Analogously to the case of linear approximation, we obtain the unknown coefficients $c_1, c_2 \in \mathbb{R}$ as the minimum of the price function

$$\Phi(c_1, c_2) = \sum_{i=1}^{n} (c_1\varphi_1(x_i) + c_2\varphi_2(x_i) - y_i)^2 ,$$

i.e. as the solution of the normal system of equations

$$\begin{aligned} c_1 \sum_{i=1}^{n} (\varphi_1(x_i))^2 + c_2 \sum_{i=1}^{n} \varphi_1(x_i) \cdot \varphi_2(x_i) &= \sum_{i=1}^{n} y_i \cdot \varphi_1(x_i), \\ c_1 \sum_{i=1}^{n} \varphi_2(x_i) \cdot \varphi_1(x_i) + c_2 \sum_{i=1}^{n} (\varphi_2(x_i))^2 &= \sum_{i=1}^{n} y_i \cdot \varphi_2(x_i) \end{aligned}$$

or in the matrix form

$$\begin{pmatrix} \sum_{i=1}^{n} (\varphi_1(x_i))^2 & \sum_{i=1}^{n} \varphi_1(x_i) \cdot \varphi_2(x_i) \\ \sum_{i=1}^{n} \varphi_2(x_i) \cdot \varphi_1(x_i) & \sum_{i=1}^{n} (\varphi_2(x_i))^2 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \cdot \varphi_1(x_i) \\ \sum_{i=1}^{n} y_i \cdot \varphi_2(x_i) \end{pmatrix} .$$

For example, if we want to approximate by the function

$$\varphi(x) = c_1 x^2 + c_1 \sin(x) ,$$

then the normal system of equations is:

$$\begin{aligned} c_1 \sum_{i=1}^{n} x_i^4 + c_2 \sum_{i=1}^{n} x_i^2 \sin(x_i) &= \sum_{i=1}^{n} y_i x_i^2, \\ c_1 \sum_{i=1}^{n} x_i^2 \sin(x_i) + c_2 \sum_{i=1}^{n} \sin^2(x_i) &= \sum_{i=1}^{n} y_i \sin(x_i) \end{aligned}$$

or in the matrix form

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^2 \sin(x_i) \\ \sum_{i=1}^{n} x_i^2 \sin(x_i) & \sum_{i=1}^{n} \sin^2(x_i) \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} y_i \sin(x_i) \end{pmatrix} .$$

60 – Approximation by the least-squares method, two functions

Example

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 \ln(x) + c_2 x.$$

We write the normal system of equations in the matrix form

$$\begin{pmatrix} \sum_{i=1}^{6} \ln^2(x_i) & \sum_{i=1}^{6} x_i \ln(x_i) \\ \sum_{i=1}^{6} x_i \ln(x_i) & \sum_{i=1}^{6} x_i^2 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{6} y_i \ln(x_i) \\ \sum_{i=1}^{6} y_i x_i \end{pmatrix}$$

and calculate the sums:

$$\sum_{i=1}^{6} \ln^2(x_i) = \ln^2(1) + \ln^2(2) + \ln^2(3) + \ln^2(5) + \ln^2(7) + \ln^2(10) = 13.3662$$

$$\begin{aligned} \sum_{i=1}^{6} x_i \ln(x_1) =& 1 \cdot \ln(1) + 2 \cdot \ln(2) + 3 \cdot \ln(3) + 5 \cdot \ln(5) \\ & + 7 \cdot \ln(7) + 10 \cdot \ln(10) = 49.3765 \end{aligned}$$

$$\sum_{i=1}^{6} x_i^2 = 1^2 + 2^2 + 3^2 + 5^2 + 7^2 + 10^2 = 188$$

$$\begin{aligned} \sum_{i=1}^{6} y_i \ln(x_i) =& 0 \cdot \ln(1) + 3 \cdot \ln(2) + 5 \cdot \ln(3) + 8 \cdot \ln(5) \\ & + 8 \cdot \ln(7) + 7 \cdot \ln(10) = 52.1334 \end{aligned}$$

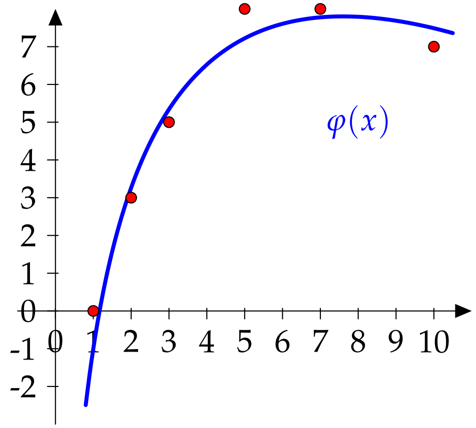
$$\sum_{i=1}^{6} y_i x_i = 0 \cdot 1 + 3 \cdot 2 + 5 \cdot 3 + 8 \cdot 5 + 8 \cdot 7 + 7 \cdot 10 = 187$$

The normal system of equations is:

$$\begin{pmatrix} 13.3662 & 49.3765 \\ 49.3765 & 188 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 52.1334 \\ 187 \end{pmatrix}$$

The unique solution of this system is $c_1 = 7.5896$, $c_2 = -0.9987$.
Therefore the best approximation of the given data in the least-squares method sense has the form

$$\varphi(x) = 7.58961 \ln(x) - 0.9987x\,.$$



61 – Approximation by the least-squares method, two functions

Example

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 \ln(x) + c_2 x.$$

Use the MATLAB to solve the problem.

We input the given data in the MATLAB.

```
>> x=[1,2,3,5,7,10]
>> y=[0,3,5,8,8,7];
>> n=length(x)
n =
     6
```

We also need the matrix of the normal system of equations and the right hand sides vector.

```
>> M=[sum(log(x).^2 sum(log(x)*x); sum(x.*log(x)) sum(x.^2)]
M =
   13.3662   49.3765
   49.3765  188.0000
>> v=[sum(y.*log(x)); sum(y.*x)]
v =
   52.1334
  187.0000
```

We solve the normal system of equations using one of many methods that the MATLAB offers.

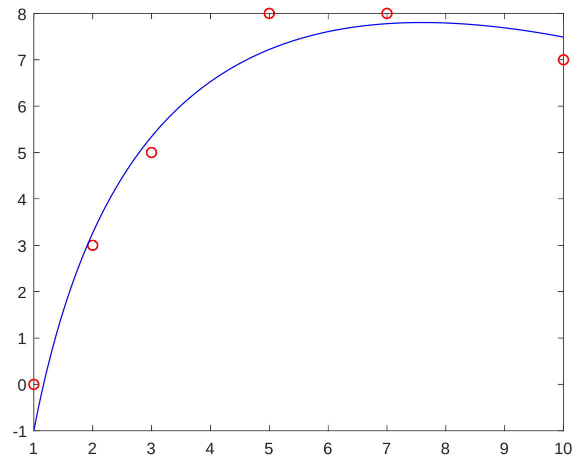
```
>> c=M\v
c =
    7.5896
   -0.9987
```

The best approximation in the least-squares sense has the form:

$$\varphi(x) = 7.58961 \ln(x) - 0.9987x\ .$$

Finally we plot the given data and the found approximation

```
>> hold on
>> plot(x,y,'ro')
>> xg=x(1):0.01:x(n)
>> yg=c(1)*log(xg)+c(2)*xg;
>> plot(xg,yg)
```



62 – Approximation by the least-squares method, two functions

Exercise

Approximate the data from the table

x_i	0	1	2	3	4
y_i	0	2	4	3	1

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 \sin(x) + c_2 x.$$

63 – Approximation by the least-squares method, two functions

Exercise

Approximate the data from the table

x_i	0	1	2	3	4
y_i	0	2	4	3	1

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 \sin(x) + c_2 x.$$

Use the MATLAB to solve the problem.

64 – Approximation by the least-squares method, k functions

Approximation by k functions

Assume we are given n pairs (x_i, y_i), $i = 1, \ldots, n$ of distinct nodes x_i and corresponding values y_i as well as k functions $\varphi_1(x)$, $\varphi_2(x)$, …, $\varphi_k(x)$. We want to find such values $c_1, c_2, \ldots c_k \in \mathbb{R}$, that the function $\varphi(x) = c_1\varphi_1(x) + c_2\varphi_2(x) + \cdots + c_k\varphi_k(x)$ is the best approximation of the given data in the least-squares sense.

These coefficients $c_1, c_2, \ldots, c_k \in \mathbb{R}$ we obtain as the solution of the normal system of equations

$$
\begin{aligned}
c_1 \sum_{i=1}^{n} (\varphi_1(x_i))^2 + c_2 \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_2(x_i) + c_3 \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_3(x_i) + \cdots + c_k \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_k(x_i) &= \sum_{i=1}^{n} y_i\cdot\varphi_1(x_i),\\
c_1 \sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_1(x_i) + c_2 \sum_{i=1}^{n} (\varphi_2(x_i))^2 + c_3 \sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_3(x_i) + \cdots + c_k \sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_k(x_i) &= \sum_{i=1}^{n} y_i\cdot\varphi_2(x_i),\\
&\vdots\\
c_1 \sum_{i=1}^{n} \varphi_k(x_i)\cdot\varphi_1(x_i) + c_2 \sum_{i=1}^{n} \varphi_k(x_i)\cdot\varphi_2(x_i) + c_3 \sum_{i=1}^{n} \varphi_k(x_i)\cdot\varphi_3(x_i) + \cdots + c_k \sum_{i=1}^{n} (\varphi_k(x_i))^2 &= \sum_{i=1}^{n} y_i\cdot\varphi_k(x_i).
\end{aligned}
$$

65 – Approximation by the least-squares method, k functions

For $k = 3$: We want to find the values $c_1, c_2, c_3 \in \mathbb{R}$ for the function $\varphi(x) = c_1\varphi_1(x) + c_2\varphi_2(x) + c_k\varphi_3(x)$. The normal system of equations is

$$\begin{aligned}
c_1 \sum_{i=1}^{n} (\varphi_1(x_i))^2 + c_2 \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_2(x_i) + c_3 \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_3(x_i) &= \sum_{i=1}^{n} y_i \cdot \varphi_1(x_i),\\
c_1 \sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_1(x_i) + c_2 \sum_{i=1}^{n} (\varphi_2(x_i))^2 + c_3 \sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_3(x_i) &= \sum_{i=1}^{n} y_i \cdot \varphi_2(x_i),\\
c_1 \sum_{i=1}^{n} \varphi_3(x_i)\cdot\varphi_1(x_i) + c_2 \sum_{i=1}^{n} \varphi_3(x_i)\cdot\varphi_2(x_i) + c_3 \sum_{i=1}^{n} (\varphi_3(x_i))^2 &= \sum_{i=1}^{n} y_i \cdot \varphi_3(x_i)
\end{aligned}$$

or in the matrix form

$$\begin{pmatrix}
\sum_{i=1}^{n} (\varphi_1(x_i))^2 & \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_2(x_i) & \sum_{i=1}^{n} \varphi_1(x_i)\cdot\varphi_3(x_i)\\
\sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_1(x_i) & \sum_{i=1}^{n} (\varphi_2(x_i))^2 & \sum_{i=1}^{n} \varphi_2(x_i)\cdot\varphi_3(x_i)\\
\sum_{i=1}^{n} \varphi_3(x_i)\cdot\varphi_1(x_i) & \sum_{i=1}^{n} \varphi_3(x_i)\cdot\varphi_2(x_i) & \sum_{i=1}^{n} (\varphi_3(x_i))^2
\end{pmatrix} \cdot \begin{pmatrix} c_1\\ c_2\\ c_3 \end{pmatrix} = \begin{pmatrix}
\sum_{i=1}^{n} y_i \cdot \varphi_1(x_i)\\
\sum_{i=1}^{n} y_i \cdot \varphi_2(x_i)\\
\sum_{i=1}^{n} y_i \cdot \varphi_3(x_i)
\end{pmatrix}.$$

66 – Approximation by the least-squares method, k functions

Example

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x + c_3 x^2.$$

We write the normal system of equations in the matrix form

$$\begin{pmatrix} \sum_{i=1}^{6} 1 & \sum_{i=1}^{6} x_i & \sum_{i=1}^{6} x_i^2 \\ \sum_{i=1}^{6} x_i & \sum_{i=1}^{6} x_i^2 & \sum_{i=1}^{6} x_i^3 \\ \sum_{i=1}^{6} x_i^2 & \sum_{i=1}^{6} x_i^3 & \sum_{i=1}^{6} x_i^4 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{6} y_i \\ \sum_{i=1}^{6} y_i x_i \\ \sum_{i=1}^{6} y_i x_i^2 \end{pmatrix}$$

and calculate the sums:

$$\sum_{i=1}^{6} 1 = 1+1+1+1+1+1 = 6$$

$$\sum_{i=1}^{6} x_i = 1+2+3+5+7+10 = 28$$

$$\sum_{i=1}^{6} x_i^2 = 1^2+2^2+3^2+5^2+7^2+10^2 = 188$$

$$\sum_{i=1}^{6} x_i^3 = 1^3+2^3+3^3+5^3+7^3+10^3 = 1504$$

$$\sum_{i=1}^{6} x_i^4 = 1^4+2^4+3^4+5^4+7^4+10^4 = 13124$$

$$\sum_{i=1}^{6} y_i = 0+3+5+8+8+7 = 31$$

$$\sum_{i=1}^{6} y_i x_i = 0\cdot 1+3\cdot 2+5\cdot 3+8\cdot 5+8\cdot 7+7\cdot 10 = 187$$

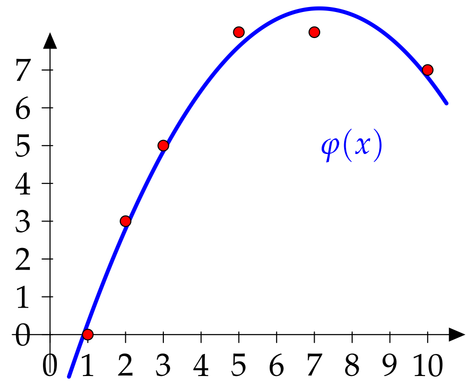
$$\sum_{i=1}^{6} y_i x_i^2 = 0\cdot 1^2+3\cdot 2^2+5\cdot 3^2+8\cdot 5^2+8\cdot 7^2+7\cdot 10^2 = 1349$$

The normal system of equations is:

$$\begin{pmatrix} 6 & 28 & 188 \\ 28 & 188 & 1504 \\ 188 & 1504 & 13124 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 31 \\ 187 \\ 1349 \end{pmatrix}$$

The unique solution of this system is $c_1 = -2.63963$, $c_2 = 3.16090$, $c_3 = -0.22163$.
The best quadratic approximation of the given data in the least-squares method sense) has the form

$$\varphi(x) = -2.63963 + 3.16090x - 0.22163x^2\ .$$



67 – Approximation by the least-squares method, k functions

Example

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x + c_3 x^2.$$

Use the MATLAB to solve the problem.

We input the given data in the MATLAB.

```
>> x=[1,2,3,5,7,10];
>> y=[0,3,5,8,8,7];
>> n=length(x)
n =
     6
```

We also need the matrix of the normal system of equations and the right hand sides vector.

```
>> M=[n sum(x) sum(x.^2); sum(x) sum(x.^2) sum(x.^3); sum(x.^2) sum(x.^3) sum(x.^4)]
M =
           6          28         188
          28         188        1504
         188        1504       13124


>> v=[sum(y); sum(y.*x); sum(y.*x.^2)]
v =
          31
         187
        1349
```

68 – Approximation by the least-squares method, k functions

We solve the normal system of equations using one of many methods that the MATLAB offers.

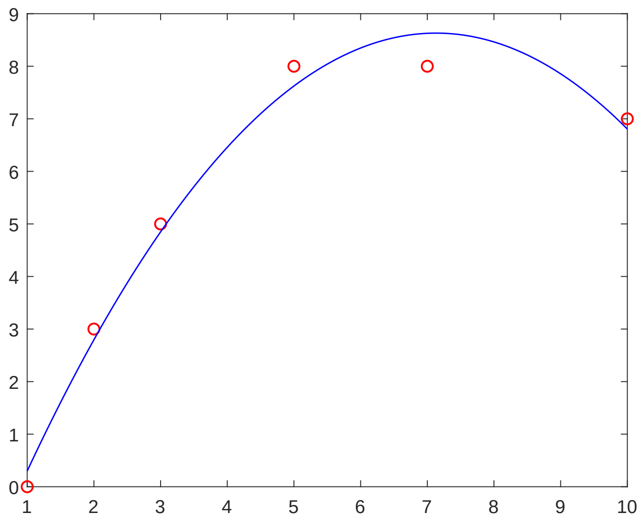
```
>> c=M\v
c =
   -2.6396
    3.1609
   -0.2216
```

The best approximation in the least-squares sense has the form:

$$\varphi(x) = -2.6396 + 3.1609x - 0.2216x^2\,.$$

We plot the given data and the found quadratic approximation

```
>> hold on
>> plot(x,y,'ro')
>> xg=x(1):0.01:x(n);
>> yg=c(1)+c(2)*xg+c(3)*x.^2;
>> plot(xg,yg)
```



69 – Approximation by the least-squares method, k functions

Exercise

Approximate the data from the table

x_i	0	1	2	3	4
y_i	0	2	4	3	1

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x + c_3 x^2.$$

70 – Approximation by the least-squares method, k functions

Exercise

Approximate the data from the table

x_i	0	1	2	3	4
y_i	0	2	4	3	1

in the sense of the least-squares method by the function

$$\varphi(x) = c_1 + c_2 x + c_3 x^2.$$

Use the MATLAB to solve the problem.

71 – Approximation by the least-squares method

Example

Approximate the data from the table

x_i	-2.3	-1.3	0.6	1.5	2.8	3.3	4.6	5.9	7.8	9.3
y_i	-51	-15	8	31	-47	-11	-101	-110	-223	-307

in the sense of the least-squares method

a) by the linear function
$$\zeta(x) = d_2 + d_1 x,$$

b) by the function
$$\varphi(x) = c_1 \sin(x) + c_2 x^2.$$

Compare the obtained results both graphically and numerically.
Use the MATLAB to solve the problem.

a)
We define the functions $\zeta_1(x) = 1$ and $\zeta_2(x) = x$ in the MATLAB as the variables `p1` and `p2`, the matrix of the normal system $\boldsymbol{G_a}$ and the right hand sides vector $\boldsymbol{d_a}$.

```
>> p1=@(x)x.^0;
>> p2=@(x)x;
>> Ga=[sum(p1(x).^2) sum(p1(x).*p2(x));sum(p2(x).*p1(x)) sum(p2(x).^2)]
Ga =
   10.0000   32.2000
   32.2000  231.6200
>> da=[sum(p1(x).*y);sum(p2(x).*y)]
da =
  1.0e+003 *
   -0.8260
   -5.6879
```

We used the equality $1 = x^0$ to define the function `p1` in the MATLAB. If we had not applied this little technical trick, we would have had to define the element in the first row and the first column of the matrix $\boldsymbol{G_a}$ manually as $\sum_1^n \zeta_2^2(x) = \sum_1^n 1 = 10$ (the number of the nodes in the table).
We solve the normal system of equations.

```
>> d=Ga\da
   -6.3842
  -23.6695
```

The best approximation in the least-squares sense has the form (the coefficients are rounded to four decimal places):

$$\zeta(x) = -6.3842 - 23.6695x.$$

b)
In this case, we denote $\varphi_1(x) = \sin x$ and $\varphi_2(x) = x^2$. We write the normal system of equations in the matrix form. We define the auxiliary matrix $\boldsymbol{G_b}$ and the vector $\boldsymbol{d_b}$.

$$\begin{pmatrix} \sum_{i=1}^n (\varphi_1(x_i))^2 & \sum_{i=1}^n \varphi_1(x_i) \cdot \varphi_2(x_i) \\ \sum_{i=1}^n \varphi_2(x_i) \cdot \varphi_1(x_i) & \sum_{i=1}^n (\varphi_2(x_i))^2 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n y_i \cdot \varphi_1(x_i) \\ \sum_{i=1}^n y_i \cdot \varphi_2(x_i) \end{pmatrix}$$

$$\boldsymbol{G_a} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \boldsymbol{d_a}$$

We solve this system of linear equations and obtain the values of the coefficients c_1, c_2 such that the function $\varphi(x)$ is the best approximation of the given data in the least-squares sense.
We input the given data in the MATLAB.

```
>> x=[-2.3 -1.3 0.6 1.5 2.8 3.3 4.6 5.9 7.8 9.3];
>> y=[-51 -15 8 31 -47 -11 -101 -110 -223 -307];
```

Then we define the functions $\varphi_1(x) = \sin x$ and $\varphi_2(x) = x^2$ in the MATLAB as the variables `f1` and `f2`.

```
>> f1=@(x)sin(x);
>> f2=@(x)x.^2;
```

73 – Approximation by the least-squares method

We also need the matrix of the normal system of equations $\boldsymbol{G_a}$ and the right hand sides vector $\boldsymbol{d_a}$.

```
>> Gb=[sum(f1(x).^2) sum(f1(x).*f2(x));
sum(f2(x).*f1(x)) sum(f2(x).^2)]
Gb =
  1.0e+004 *
    0.0005    0.0035
    0.0035    1.3058

>> db=[sum(f1(x).*y);sum(f2(x).*y)]
db =
  1.0e+004 *
   -0.0045
   -4.6797
```

We solve the normal system of equations using one of many methods that the MATLAB offers.

```
>> c=Gb\db
   16.2406
   -3.6277
```

The best approximation in the least-squares sense has the form (the coefficients are rounded to four decimal places):

$$\varphi(x) = 16.2406 \sin x - 3.6277x^2.$$

The comparison of the approximations
Now we can compare the sums of squares of the distances between the obtained approximations and the given data (i.e. the values of the price functions)

$$\sum_{i=1}^{10}(\varphi(x_i)-y_i)^2, \quad \sum_{i=1}^{10}(\zeta(x_i)-y_i)^2.$$

We input the obtained approximations as the variables `f` and `p`. Let us note that we have input the calculated coefficients c_1, c_2 (or d_1, d_2) as the variable (vector) `c` (or `d`). Hence, the coefficients values are components of these vectors `c(1)` and `c(2)` (or `d(1),d(2)`).

```
>> f=@(x)c(1)*f1(x)+c(2)*f2(x);
>> p=@(x)d(1)*p1(x)+d(2)*p2(x);
```

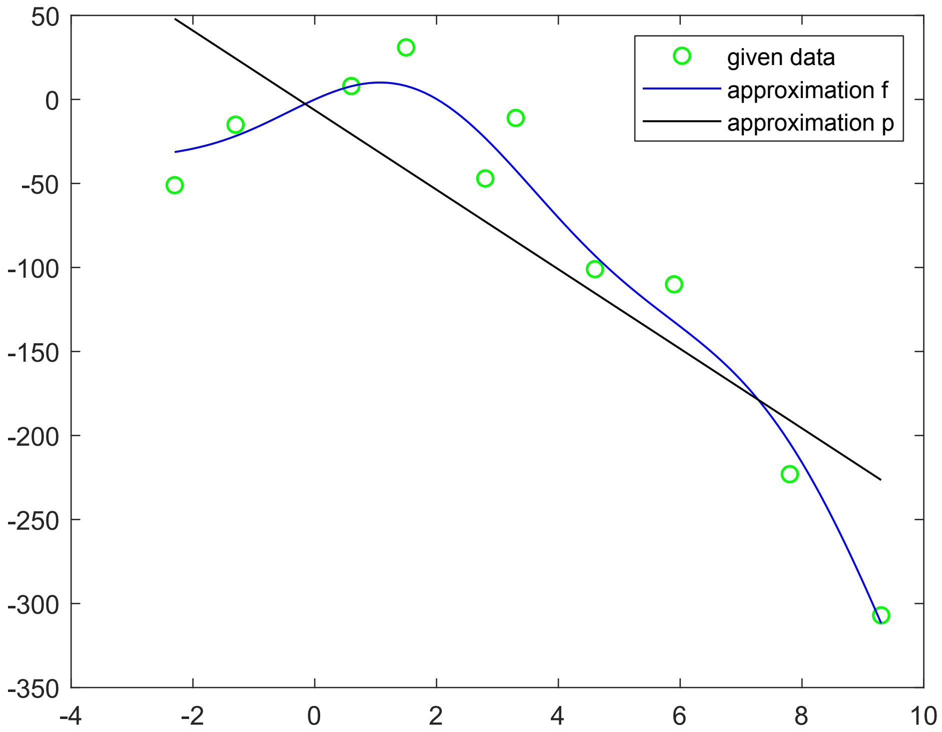
We calculate the sums in the MATLAB.

```
>> sum((f(x)-y).^2)
ans =
  3.4327e+003
>> sum((p(x)-y).^2)
ans =
  3.2557e+004
```

Because $3432 < 32557$, the function $\varphi(x)$ is better approximation than the function $\zeta(x)$.

Finally we plot the graphs of the obtained approximations $\varphi(x)$, $\zeta(x)$ and the discrete point of the given data. Both of these graphs illustrate that the function $\varphi(x)$ is better approximation of the the given data.

```
>> x1=-2.3:0.1:9.3;  %we make the vector covering all given nodes to depict the graph, the step is 0.1
>> plot(x,y,'go',x1,f(x1),'b-',x1,p(x1),'k-')  %we depict the given data as the green circles and the obtained
   approximations as the blue (or black) line
>> legend('given data','approximation f','approximation p')  %we add the legend to the graph
```



76 – Approximation by the least-squares method

Exercise

Approximate the data from the table

x_i	1	2	3	5	7	10
y_i	0	3	5	8	8	7

in the sense of the least-squares method

a) by the linear function

$$\zeta(x) = a_1 + a_2 x,$$

b) by the function

$$\varphi(x) = b_1 \ln(x) + b_2 x,$$

c) by the quadratic function

$$\theta(x) = c_1 + c_2 x + c_3 x^2.$$

Compare the obtained results both graphically and numerically.
Use the MATLAB to solve the problem.

77 – Approximation by the least-squares method

Exercise

Approximate the data from the table

x_i	1	2	3.5	4.5	6	7	7.5
y_i	4.2	5	5.5	7	7.8	8.5	8.1

in the sense of the least-squares method

a) by the linear function

$$\zeta(x) = a_1 + a_2 x,$$

b) by the function

$$\varphi(x) = b_1 \mathrm{e}^x + b_2 x,$$

c) by the quadratic function

$$\theta(x) = c_1 + c_2 x + c_3 x^2.$$

Compare the obtained results both graphically and numerically.
Use the MATLAB to solve the problem.

78 – Approximation by the least-squares method

Exercise

Approximate the data from the table in the sense of the least-squares method

a) by the linear function

$$\zeta(x) = d_1 + d_2 x,$$

b) by the function

$$\varphi(x),$$

c) by the quadratic function

$$\theta(x) = a_1 + a_2 x + a_3 x^2.$$

Compare the obtained results both graphically and numerically.
Use the MATLAB to solve the problem.

1.

$$\varphi(x) = c_1 \frac{1}{x} + c_2$$

x_i	-3	-2.5	-1.5	-1	0.5
y_i	32.3	-45.4	9.7	44.9	-21.2

2.

$$\varphi(x) = c_1 \mathrm{e}^x + c_2 \frac{1}{x}$$

x_i	-6.5	-6	-5.5	-5	-3.5
y_i	-23.6	49.9	-28.9	-0.2	-21

3.

$$\varphi(x) = c_1 + c_2 \sin(x)$$

x_i	-3.5	-2.5	-1.5	-1	0.5
y_i	25.3	15.9	-28.6	10.2	10.4

4.

$$\varphi(x) = c_1 \cos(x) + c_2$$

x_i	1	1.5	2.5	3	4
y_i	18.5	17.7	37.6	-48.8	-19

5.

$$\varphi(x) = c_1 x + c_2 \mathrm{e}^x$$

x_i	-6.5	-6	-4.5	-3	-2.5
y_i	-48.9	-27.3	1.6	-4.2	20.3

6.

$$\varphi(x) = c_1 \cos(x) + c_2 x$$

x_i	-4	-3	-2.5	-2	-1
y_i	27	-18.7	13.8	48.6	0.2

7.

$$\varphi(x) = c_1 x^2 + c_2 \ln(x)$$

x_i	4.5	6	7.5	8	8.5
y_i	-37	44	20.1	34.7	-29.1

79 – Approximation by the least-squares method

8.

$\varphi(x) = c_1 x^2 + c_2$

x_i	-3	-2.5	-1	0	0.5
y_i	36.7	-12.8	-42.7	-30.1	-45.1

9.

$\varphi(x) = c_1 e^x + c_2 x^2$

x_i	-5.5	-4	-2.5	-1.5	-1
y_i	-48.6	-21.2	31.6	48.5	-48.3

10.

$\varphi(x) = c_1 + c_2 \cos(x)$

x_i	-4.5	-3.5	-2	-1.5	-0.5
y_i	-46.5	-41.9	35	-16	-3.4

11.

$\varphi(x) = c_1 \ln(x) + c_2$

x_i	4.5	6	7.5	8	9.5
y_i	-34.4	-37.8	26.2	22.1	15.1

12.

$\varphi(x) = c_1 x^3 + c_2$

x_i	-6	-5.5	-4	-3.5	-2
y_i	2.1	39.5	44.2	-16.5	-6.3

13.

$\varphi(x) = c_1 x^2 + c_2 x$

x_i	-2	-1.5	-1	0	1.5
y_i	-14.2	-21.5	36.8	12.6	-25.9

14.

$\varphi(x) = c_1 x^2 + c_2 \sin(x)$

x_i	-5.5	-4.5	-4	-2.5	-1.5
y_i	-47.8	-23.8	-38.4	-43.1	35.2

15.

$\varphi(x) = c_1 \sin(x) + c_2 x$

x_i	-3.5	-3	-1.5	-0.5	0
y_i	-48.8	38.9	36.6	-24.6	6.9

16.

$\varphi(x) = c_1 + c_2 e^x$

x_i	-0.5	0.5	1	2	3.5
y_i	48	29.1	-34.8	33.3	-30.9

17.

$\varphi(x) = c_1 \cos(x) + c_2 x^2$

x_i	-1	0.5	2	3	4
y_i	10.8	-32.5	-49.8	29	1.3

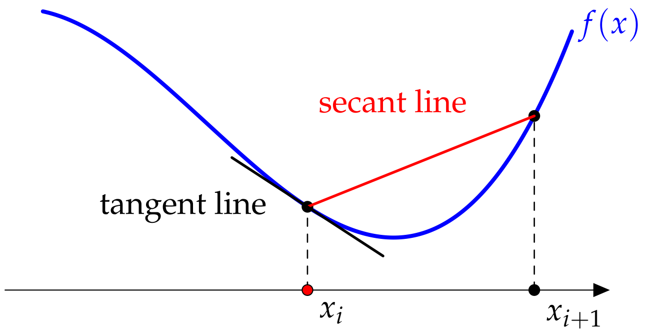
Worksheets for Numerical methods

Numerical differentiation

81 – Numerical differentiation, the first derivative

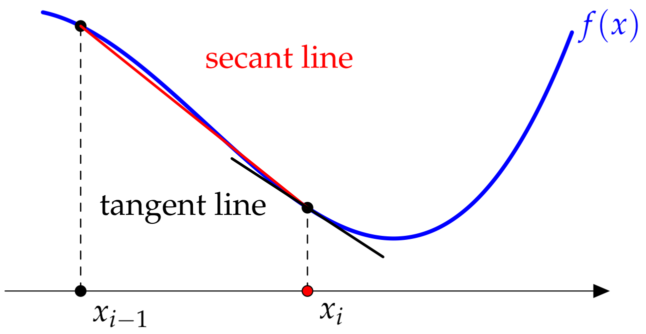
Given distinct nodes x_i and corresponding function values $f(x_i)$ of a function f, the problem is to approximate values of the first derivative $f'(x_i)$ at the given nodes. Because the value $f'(x_i)$ is the slope of the tangent line to the graph of the function f at the point $[x_i, f(x_i)]$, we can approximate it by a slope of a proper secant line.
Using a secant line corresponding to nodes x_i and x_{i+1} we obtain the **forward difference**

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



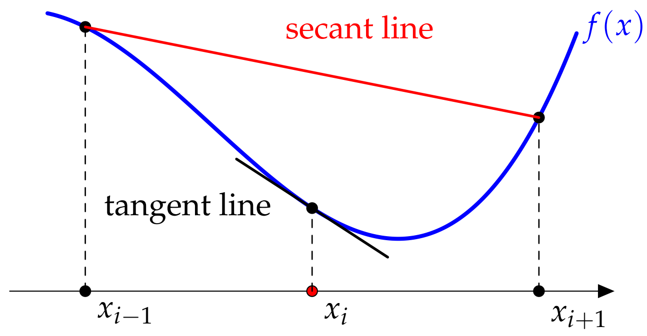
Using a secant line corresponding to nodes x_{i-1} and x_i we obtain the **backward difference**

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



Using a secant line corresponding to nodes x_{i-1} and x_{i+1} we obtain the **central difference**

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



In practice, the central difference is usually the most accurate approximation of the derivative value.

82 – Numerical differentiation, the first derivative

Example

Approximate the derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

using the forward difference.

We calculate the approximate value of the derivative $y'(x_1)$

$$y'(x_1) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{50 - 40}{1 - 0} = 10\,,$$

approximate value of the derivative $y'(x_3)$

$$y'(x_2) = \frac{y_3 - y_2}{x_3 - x_2} = \frac{20 - 50}{2 - 1} = -30\,,$$

approximate value of the derivative $y'(x_3)$

$$y'(x_3) = \frac{y_4 - y_3}{x_4 - x_3} = \frac{25 - 20}{3 - 2} = 5\,,$$

approximate value of the derivative $y'(x_4)$

$$y'(x_4) = \frac{y_5 - y_4}{x_5 - x_4} = \frac{30 - 25}{4 - 3} = 5.$$

Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	1	2	3	4
$y'(x_i)$	10	-30	5	5	—

The approximate value of the derivative $y'(x_5)$ cannot be calculated by the forward difference because there is no following node there.

Example

Approximate the derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

using the backward difference.

We calculate the approximate value of the derivative $y'(x_2)$

$$y'(x_2) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{50 - 40}{1 - 0} = 10\,,$$

approximate value of the derivative $y'(x_3)$

$$y'(x_3) = \frac{y_3 - y_2}{x_3 - x_2} = \frac{20 - 50}{2 - 1} = -30\,,$$

approximate value of the derivative $y'(x_4)$

$$y'(x_4) = \frac{y_4 - y_3}{x_4 - x_3} = \frac{25 - 20}{3 - 2} = 5\,,$$

approximate value of the derivative $y'(x_5)$

$$y'(x_5) = \frac{y_5 - y_4}{x_5 - x_4} = \frac{30 - 25}{4 - 3} = 5.$$

Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	1	2	3	4
$y'(x_i)$	—	10	-30	5	5

The approximate value of the derivative $y'(x_1)$ cannot be calculated by the backward difference because there is no previous node there.

83 – Numerical differentiation, the first derivative

Example

Approximate the derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

using the central difference.

We calculate the approximate value of the derivative $y'(x_2)$

$$y'(x_2) = \frac{y_3 - y_1}{x_3 - x_1} = \frac{20 - 40}{2 - 0} = -10\,,$$

approximate value of the derivative $y'(x_3)$

$$y'(x_3) = \frac{y_4 - y_2}{x_4 - x_2} = \frac{25 - 50}{3 - 1} = -\frac{25}{2} = -12.5\,,$$

approximate value of the derivative $y'(x_4)$

$$y'(x_4) = \frac{y_5 - y_3}{x_5 - x_3} = \frac{30 - 20}{4 - 2} = 5.$$

Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	1	2	3	4
$y'(x_i)$	—	-10	-12.5	5	—

The approximate values of the derivatives $y'(x_1)$ and $y'(x_5)$ cannot be calculated by the central difference because there are no required near by nodes there.

Example

Approximate the derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

using the central difference at the interior nodes, the forward difference at the first node and the backward difference at the last node.

All needed approximate values of the derivative $y'(x_i)$ were calculated in previous examples.
Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	1	2	3	4
$y'(x_i)$	10	-10	-12.5	5	5

84 – Numerical differentiation, the first derivative

Example

Approximate the derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

using the central difference at the interior nodes, the forward difference at the first node and the backward difference at the last node.
Use the MATLAB to solve the problem.

We input the nodes x and the function values y.

```
>> x=[0 1 2 3 4]
x =
     0     1     2     3     4
>> y=[40 50 20 25 30]
y =
    40    50    20    25    30
```

We define the number n of the nodes x_i.

```
>> n=length(x)
n =
     5
```

We calculate the approximate value of the derivative at the node x_1 using the forward difference.

```
>> yd(1)=(y(2)-y(1))/(x(2)-x(1))
yd =
     10
```

Using the cycle statement we calculate the approximate values of the derivatives at the interior nodes x_2, x_3, x_4 by the central difference.

```
>> for i=2:n-1, yd(i)=(y(i+1)-y(i-1))/(x(i+1)-x(i-1)); end
```

We calculate the approximate value of the derivative at x_n using the backward difference.

```
>> yd(n)=(y(n)-y(n-1))/(x(n)-x(n-1))
yd =
         10.000   -10.0000  -12.5000     5.0000     5.0000
```

Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	1	2	3	4
$y'(x_i)$	10.0000	-10.0000	-12.5000	5.0000	5.0000

85 – Numerical differentiation, the first derivative

Exercise

Approximate the derivative of the data

x_i	2	4	6	8	10	12	14
y_i	12.4	5.3	3.2	4.5	7.1	8.6	11.6

using the central difference at the interior nodes, the forward difference at the first node and the backward difference at the last node.

86 – Numerical differentiation, the first derivative

Exercise

Approximate the derivative of the data

x_i	2	4	6	8	10	12	14
y_i	12.4	5.3	3.2	4.5	7.1	8.6	11.6

using the central difference at the interior nodes, the forward difference at the first node and the backward difference at the last node.
Use the MATLAB to solve the problem.

87 – Numerical differentiation, the first derivative

Example

Approximate the derivative of the function
$$f(x) = \sin\left(x^2\right)$$
on the interval $[0, 2]$ using the central difference with the step 0.25.

At first we define nodes x_i with the given step 0.25: $x_1 = 0,\ x_2 = 0.25,\ x_3 = 0.5, \ldots,\ x_8 = 1.75,\ x_9 = 2.$
Then we calculate the approximate value of the derivative $f'(x_2)$

$$f'(x_2) = \frac{f(x_3) - f(x_1)}{x_3 - x_1} = \frac{\sin\left(x_3^2\right) - \sin\left(x_1^2\right)}{x_3 - x_1} = \frac{\sin\left(0.5^2\right) - \sin\left(0^2\right)}{0.5 - 0} = 0.4948\,,$$

approximate value of the derivative $f'(x_3)$

$$f'(x_3) = \frac{f(x_4) - f(x_2)}{x_4 - x_2} = \frac{\sin\left(x_4^2\right) - \sin\left(x_2^2\right)}{x_4 - x_2} = \frac{\sin\left(0.75^2\right) - \sin\left(0.25^2\right)}{0.75 - 0.25} = 0.9417$$

and so on.

Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
$f'(x_i)$	—	0.4948	0.9417	1.1881	0.9333	-0.1268	-1.8419	-3.0698	—

88 – Numerical differentiation, the first derivative

Example

Approximate the derivative of the function

$$f(x) = \sin(x^2)$$

on the interval $[0,2]$ using the central difference with the step 0.25. Use the MATLAB to solve the problem.

We input the step as h and generate the nodes using the colon notation.

```
>> h=0.25
h =
    0.2500
>> x=0:h:2
x =
  Columns 1 through 5
         0    0.2500    0.5000    0.7500    1.0000
  Columns 6 through 9
    1.2500    1.5000    1.7500    2.0000
```

We define the function f.

```
>> f=@(x)sin(x.^2)
f =
    @(x)sin(x.^2)
```

At first we define the number n of the nodes x_i. Using the cycle statement we calculate the approximate values of the derivatives. Let us note that the derivatives can be calculated only at the interior nodes $x_2, \ldots, x_8$.

```
>> n=length(x)
n =
     9
>> for i=2:n-1, fd(i)=(f(x(i+1))-f(x(i-1)))/(2*h); end
>> fd
fd =
  Columns 1 through 5
         0    0.4948    0.9417    1.1881    0.9333
  Columns 6 through 8
   -0.1268   -1.8419   -3.0698
```

Table of the obtained approximate values of the derivatives at the nodes:

x_i	0	0.2500	0.5000	0.7500	1.0000	1.2500	1.5000	1.7500	2.0000
$f'(x_i)$	—	0.4948	0.9417	1.1881	0.9333	-0.1268	-1.8419	-3.0698	—

Notice the fact that creating the vector `fd` by the cycle statements only the values `fd(2) ... fd(8)` have been inserted. However, the MATLAB has itself assigned the value `0` to `fd(1)` because the component `fd(1)` could not be left empty.

89 – Numerical differentiation, the first derivative

Exercise

Approximate the derivative of the function

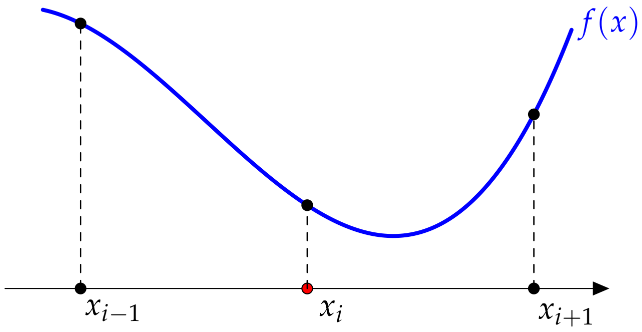
$$f(x) = \mathrm{e}^x$$

on the interval $[0, 3]$ using the central difference with the step 0.5.
Use the MATLAB to solve the problem.

90 – Numerical differentiation, the second derivative

Given distinct nodes x_i and corresponding function values $f(x_i)$ of a function f, values of the second derivative $f''(x_i)$ at the given nodes can be approximated by the formula:

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{(x_{i+1} - x_i)(x_i - x_{i-1})}.$$



If the nodes are equidistant with the step h then the formula above can be simplified to the form:

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

91 – Numerical differentiation, the second derivative

Example

Approximate the second derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

We calculate the approximate value of the second derivative $y''(x_2)$

$$y''(x_2) = \frac{y_3 - 2y_2 + y_1}{(x_3 - x_2)(x_2 - x_1)} = \frac{20 - 2 \cdot 50 + 40}{(2-1)(1-0)} = -40\,,$$

approximate value of the second derivative $y''(x_3)$

$$y''(x_3) = \frac{y_4 - 2y_3 + y_2}{(x_4 - x_3)(x_3 - x_2)} = \frac{25 - 2 \cdot 20 + 50}{(3-2)(2-1)} = 35\,,$$

approximate value of the second derivative $y''(x_4)$

$$y''(x_4) = \frac{y_5 - 2y_4 + y_3}{(x_5 - x_4)(x_4 - x_3)} = \frac{30 - 2 \cdot 25 + 20}{(4-3)(3-2)} = 0.$$

Table of the obtained approximate values of the second derivatives at the nodes:

x_i	0	1	2	3	4
$y''(x_i)$	—	-40	35	0	—

92 – Numerical differentiation, the second derivative

Example

Approximate the second derivative of the data

	i=1	i=2	i=3	i=4	i=5
x_i	0	1	2	3	4
y_i	40	50	20	25	30

Use the MATLAB to solve the problem.

We input the nodes x and the function values y.

```
>> x=[0 1 2 3 4]
x =
     0     1     2     3     4
>> h=
     1
>> y=[40 50 20 25 30]
y =
    40    50    20    25    30
```

We define the number n of the nodes x_i. Using the cycle statement we calculate the approximate values of the derivatives in accordance with the given formula. Let us mention that the derivatives can be calculated only at the interior nodes x_2, x_3, x_4.

```
>> n=length(x)
n =
     5
>> for i=2:n-1, ydd(i)=(f(x(i+1))-2*f(x(i))+f(x(i-1)))/(h^2); end
ydd =
     0   -40    35     0
```

Table of the obtained approximate values of the second derivatives at the nodes:

x_i	0	1	2	3	4
$y''(x_i)$	—	-40	35	0	—

Notice also the fact that creating the vector `ydd` by the cycle statements only the values `ydd(2)`, `ydd(3)`, `ydd(4)` have been inserted. However, the MATLAB has itself assigned the value `0` to `ydd(1)` because the component `ydd(1)` could not be left empty.

93 – Numerical differentiation, the second derivative

Exercise

Approximate the second derivative of the data

x_i	2	4	6	8	10	12	14
y_i	12.4	5.3	3.2	4.5	7.1	8.6	11.6

94 – Numerical differentiation, the second derivative

Exercise

Approximate the second derivative of the data

x_i	2	4	6	8	10	12	14
y_i	12.4	5.3	3.2	4.5	7.1	8.6	11.6

Use the MATLAB to solve the problem.

95 – Numerical differentiation, the second derivative

Example

Approximate the second derivative of the function
$$f(x) = \sin\left(x^2\right)$$
on the interval $[0, 2]$ with the step 0.25.

At first we define nodes x_i with the given step $h = 0.25$: $x_1 = 0,\ x_2 = 0.25,\ x_3 = 0.5, \ldots, x_8 = 1.75,\ x_9 = 2.$
Then we calculate the approximate value of the second derivative $f''(x_2)$

$$f''(x_2) = \frac{f(x_3) - 2f(x_2) + f(x_1)}{h^2} = \frac{\sin\left(x_3^2\right) - 2\sin\left(x_2^2\right) + \sin\left(x_1^2\right)}{0.25^2} = \frac{\sin\left(0.5^2\right) - 2\sin\left(0.25^2\right) + \sin\left(0^2\right)}{0.25^2} = 1.9598\,,$$

approximate value of the derivative $f''(x_3)$

$$f''(x_3) = \frac{f(x_4) - 2f(x_3) + f(x_2)}{h^2} = \frac{\sin\left(x_4^2\right) - 2\sin\left(x_3^2\right) + \sin\left(x_2^2\right)}{0.25^2} = \frac{\sin\left(0.75^2\right) - 2\sin\left(0.5^2\right) + \sin\left(0.25^2\right)}{0.25^2} = 1.6153$$

and so on.

Table of the obtained approximate values of the second derivatives at the nodes:

x_i	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
$f''(x_i)$	—	1.9598	1.6153	0.3563	-2.3948	-6.0862	-7.6347	-2.1880	—

96 – Numerical differentiation, the second derivative

Example

Approximate the second derivative of the function

$$f(x) = \sin\left(x^2\right)$$

on the interval $[0, 2]$ with the step 0.25. Use the MATLAB to solve the problem.

We input the step $h = 0.25$, generate the nodes using the colon notation and define the function f.

```
>> h=0.25
h =
    0.2500
>> x=0:h:2
x =
        0    0.2500    0.5000    0.7500    1.0000    1.2500    1.5000    1.7500    2.0000
>> f=@(x)sin(x.^2)
f =
    @(x)sin(x.^2)
```

At first we calculate the number n of the nodes x_i. Using the cycle statement we calculate the approximate values of the second derivatives.

```
>> n=length(x)
n =
     9
>> for i=2:n-1, fdd(i)=(f(x(i+1))-2*f(x(i))+f(x(i-1)))/(h^2); end
>> fdd
fdd =
     0    1.9598    1.6153    0.3563   -2.3948   -6.0862   -7.6347   -2.1880
```

Table of the obtained approximate values of the second derivatives at the nodes:

x_i	0	0.2500	0.5000	0.7500	1.0000	1.2500	1.5000	1.7500	2.0000
$f''(x_i)$	—	1.9598	1.6153	0.3563	-2.3948	-6.0862	-7.6347	-2.1880	—

Notice also the fact that creating the vector `fdd` by the cycle statements only the values `fdd(2) ... fdd(8)` have been inserted. However, the MATLAB has itself assigned the value `0` to `fdd(1)`.

97 – Numerical differentiation, the second derivative

Exercise

Approximate the second derivative of the function

$$f(x) = \mathrm{e}^x$$

on the interval $[0, 3]$ with the step 0.5.
Use the MATLAB to solve the problem.

98 – Numerical differentiation

Exercise

Approximate the first and the second derivative of the given data.
Use the MATLAB to solve the problem.

1.

x_i	5.5	6	6.5	7	7.5
y_i	-37	44	20.1	34.7	-29.1

2.

x_i	0	0.5	1	1.5	2
y_i	36.7	-12.8	-42.7	-30.1	-45.1

3.

x_i	-3	-2.5	-2	-1.5	-1
y_i	32.3	-45.4	9.7	44.9	-21.2

4.

x_i	-6.5	-6	-5.5	-5	-4.5
y_i	-23.6	49.9	-28.9	-0.2	-21

5.

x_i	1	1.5	2	2.5	3
y_i	27	-18.7	13.8	48.6	0.2

6.

x_i	-4.5	-4	-3.5	-3	-2.5
y_i	-48.6	-21.2	31.6	48.5	-48.3

7.

x_i	-3	-2.5	-2	-1.5	-1
y_i	25.3	15.9	-28.6	10.2	10.4

8.

x_i	1	1.5	2	2.5	3
y_i	18.5	17.7	37.6	-48.8	-19

9.

x_i	-6.5	-6	-5.5	-5	-4.5
y_i	-48.9	-27.3	1.6	-4.2	20.3

10.

x_i	-6	-5.5	-5	-4.5	-4
y_i	2.1	39.5	44.2	-16.5	-6.3

11.

x_i	5.5	6	6.5	7	7.5
y_i	-34.4	-37.8	26.2	22.1	15.1

12.

x_i	-4	-3.5	-3	-2.5	-2
y_i	-46.5	-41.9	35	-16	-3.4

13.

x_i	0	0.5	1	1.5	2
y_i	49.2	-12.7	3.1	-31.9	0.1

14.

x_i	-3.5	-3	-2.5	-2	-1.5
y_i	6.5	46.9	-47.7	37	-47.4

15.

x_i	0	0.5	1	1.5	2
y_i	10.8	-32.5	-49.8	29	1.3

16.

x_i	-2	-1.5	-1	-0.5	0
y_i	-14.2	-21.5	36.8	12.6	-25.9

17.

x_i	-5	-4.5	-4	-3.5	-3
y_i	-47.8	-23.8	-38.4	-43.1	35.2

18.

x_i	-3.5	-3	-2.5	-2	-1.5
y_i	-48.8	38.9	36.6	-24.6	6.9

19.

x_i	0.5	1	1.5	2	2.5
y_i	48	29.1	-34.8	33.3	-30.9

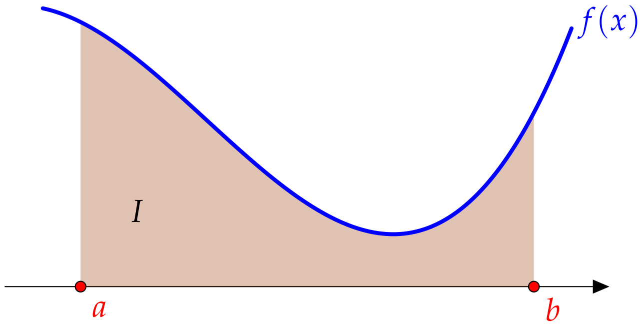
Worksheets for Numerical methods

Numerical integration

101 – Numerical integration, rectangle rule

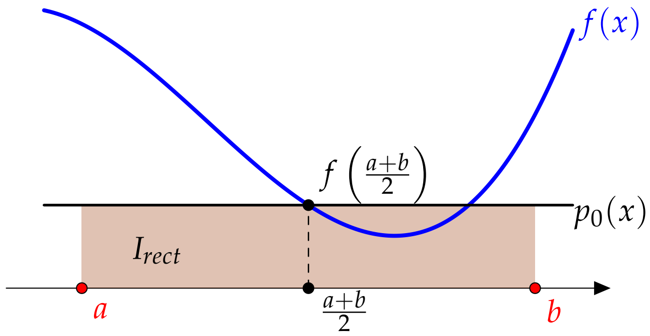
We calculate the value of the definite integral

$$\int_a^b f(x)\,\mathrm{d}x.$$



The rectangle rule

We approximate the function f by the constant interpolating polynomial $p_0(x)$ with the node $x_0 = \frac{a+b}{2}$.



This approximation can be integrated analytically

$$\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b p_0(x)\,\mathrm{d}x = \int_a^b f\left(\frac{a+b}{2}\right)\mathrm{d}x = f\left(\frac{a+b}{2}\right)\int_a^b 1\,\mathrm{d}x$$
$$= f\left(\frac{a+b}{2}\right)[x]_a^b = (b-a)f\left(\frac{a+b}{2}\right)$$

and we obtain **the rectangle rule:**

$$I_{rect} = (b-a)f\left(\frac{a+b}{2}\right)$$

102 – Numerical integration, rectangle rule

Example

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the rectangle rule.

We have $a = 0$, $b = 6$ and $f(x) = \dfrac{x}{1+x^2}$.

The rectangle rule.

$$I_{rect} = (b-a)f\left(\frac{a+b}{2}\right) = (6-0)f(3) = (6-0)\frac{3}{1+3^2}$$
$$= \frac{18}{10} = 1.8$$

Example

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the rectangle rule.
Use the MATLAB to solve the problem.

We define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f=@(x)x./(1+x.^2)
f =
 function_handle with value:
    @(x)x./(1+x.^2)
>> a=0
a =
     0
>> b=6
b =
     6
```

We calculate the approximate value of the given integral and input it as the variable `I`.

```
>> I=(b-a)*f((a+b)/2)
I =
    1.8000
```

The calculated approximate value of the given integral is 1.8.

103 – Numerical integration, rectangle rule

Exercise

Evaluate the definite integral

$$\int_1^3 \frac{2^x}{x^2 + x + 3}\,\mathrm{d}x$$

using the rectangle rule.

104 – Numerical integration, rectangle rule

Exercise

Evaluate the definite integral

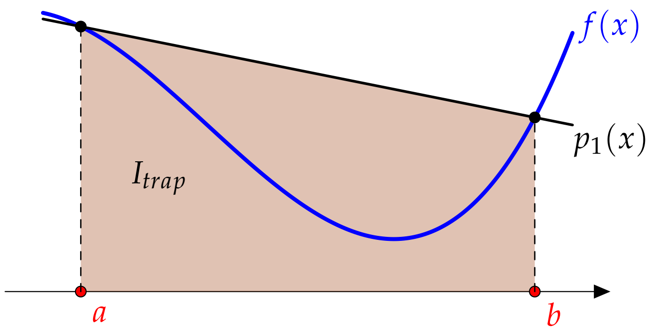
$$\int_1^3 \frac{2^x}{x^2+x+3}\,\mathrm{d}x$$

using the rectangle rule.
Use the MATLAB to solve the problem.

105 – Numerical integration, trapezoidal rule

The trapezoidal rule

We approximate the function f by the linear interpolating polynomial $p_1(x)$ with the nodes $x_0 = a$, $x_1 = b$.



The linear approximation can be also integrated analytically

$$\begin{aligned}
\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b p_1(x)\,\mathrm{d}x &= \int_a^b \left(\frac{x-b}{a-b}f(a) + \frac{x-a}{b-a}f(b)\right)\,\mathrm{d}x \\
&= \int_a^b \left(\frac{f(b)-f(a)}{b-a}x + \frac{b\cdot f(a) - a\cdot f(b)}{b-a}\right)\,\mathrm{d}x \\
&= \frac{f(b)-f(a)}{b-a}\int_a^b x\,\mathrm{d}x + \frac{b\cdot f(a) - a\cdot f(b)}{b-a}\int_a^b 1\,\mathrm{d}x \\
&= \frac{f(b)-f(a)}{b-a}\left[\frac{x^2}{2}\right]_a^b + \frac{b\cdot f(a) - a\cdot f(b)}{b-a}[x]_a^b \\
&= \frac{f(b)-f(a)}{2(b-a)}\left(b^2-a^2\right) + \frac{b\cdot f(a) - a\cdot f(b)}{b-a}(b-a) \\
&= \frac{f(b)-f(a)}{2}(b+a) + b\cdot f(a) - a\cdot f(b) \\
&= \frac{b-a}{2}\left(f(a)+f(b)\right).
\end{aligned}$$

And we obtain **the trapezoidal rule:**

$$I_{trap} = \frac{b-a}{2}\left(f(a)+f(b)\right).$$

106 – Numerical integration, trapezoidal rule

Example

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the trapezoidal rule.

We have $a=0$, $b=6$ and $f(x)=\dfrac{x}{1+x^2}$.

The trapezoidal rule.

$$\begin{aligned} I_{trap} &= \frac{b-a}{2}\left(f(a)+f(b)\right)=\frac{6-0}{2}(f(0)+f(6)) \\ &= 3\left(\frac{0}{1+0^2}+\frac{6}{1+6^2}\right)=\frac{18}{37}=0.4865 \end{aligned}$$

Example

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the trapezoidal rule.
Use the MATLAB to solve the problem.

We define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f=@(x)x./(1+x.^2)
f =
 function_handle with value:
    @(x)x./(1+x.^2)
>> a=0
a =
     0
>> b=6
b =
     6
```

We calculate the approximate value of the given integral and input it as the variable `I`.

```
>> I=(b-a)/2*(f(a)+f(b))
I =
    0.4865
```

The calculated approximate value of the given integral is 0.4865.

107 – Numerical integration, trapezoidal rule

Exercise

Evaluate the definite integral

$$\int_1^3 \frac{2^x}{x^2+x+3}\,\mathrm{d}x$$

using the trapezoidal rule.

108 – Numerical integration, trapezoidal rule

Exercise

Evaluate the definite integral

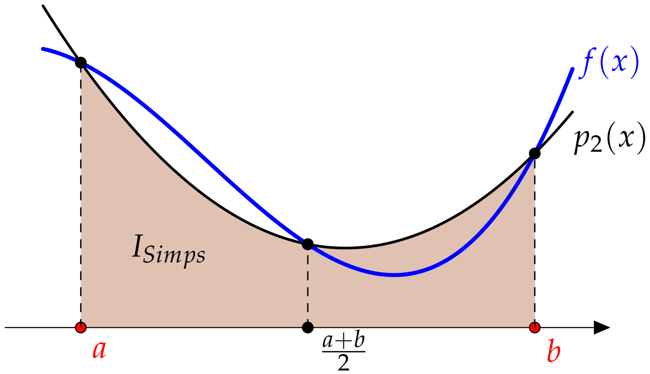
$$\int_1^3 \frac{2^x}{x^2+x+3}\,\mathrm{d}x$$

using the trapezoidal rule.
Use the MATLAB to solve the problem.

109 – Numerical integration, Simpson's rule

The Simpson's rule

We approximate the function f by the quadratic interpolating polynomial $p_2(x)$ with the nodes $x_0 = a$, $x_1 = \frac{a+b}{2}$, $x_2 = b$.



The quadratic approximation can be also integrated analytically and we obtain

$$I_{Simps} = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

110 – Numerical integration, Simpson's rule

Example

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the Simpson's rule.

We have $a = 0$, $b = 6$ and $f(x) = \dfrac{x}{1+x^2}$.

The Simpson's rule.

$$\begin{aligned} I_{Simps} &= \frac{b-a}{6}\left(f(a) + 4f\left(\tfrac{a+b}{2}\right) + f(b)\right) = \frac{6-0}{6}(f(0) + 4f(3) + f(6)) \\ &= 1\left(\frac{0}{1+0^2} + 4\frac{3}{1+3^2} + \frac{6}{1+6^2}\right) = 0 + \frac{12}{10} + \frac{6}{37} = 1.3622 \end{aligned}$$

Example

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the Simpson's rule.
Use the MATLAB to solve the problem.

We define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f=@(x)x./(1+x.^2)
f =
 function_handle with value:
    @(x)x./(1+x.^2)
>> a=0
a =
     0
>> b=6
b =
     6
```

We calculate the approximate value of the given integral and input it as the variable `I`.

```
>> I=(b-a)/6*(f(a)+4*f((a+b)/2)+f(b))
I =
    1.3622
```

The calculated approximate value of the given integral is 1.3622.

111 – Numerical integration, Simpson's rule

Exercise

Evaluate the definite integral

$$\int_1^3 \frac{2^x}{x^2+x+3}\,\mathrm{d}x$$

using the Simpson's rule.

112 – Numerical integration, Simpson's rule

Exercise

Evaluate the definite integral

$$\int_1^3 \frac{2^x}{x^2 + x + 3}\,\mathrm{d}x$$

using the Simpson's rule.
Use the MATLAB to solve the problem.

113 – Numerical integration, error estimation

Theorem

(i) Let the second derivative of the function f be continuous on the interval $[a,b]$. Then it holds:

$$\begin{aligned} I - I_{rect} &= \frac{f''(\xi)}{24}(b-a)^3, \\ I - I_{trap} &= -\frac{f''(\xi)}{12}(b-a)^3. \end{aligned}$$

(ii) Let the derivative of order 4 of the function f be continuous on the interval $[a,b]$. Then it holds:

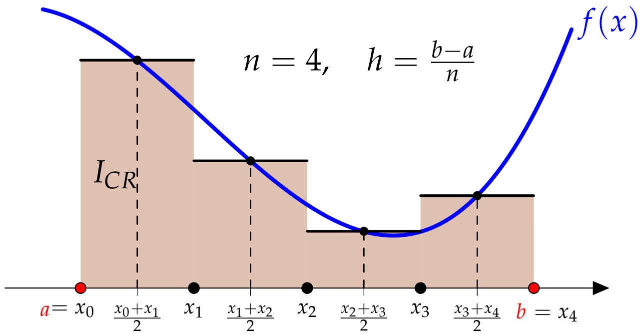
$$I - I_{Simps} = -\frac{f^{(4)}(\xi)}{90}(b-a)^5.$$

In all cases ξ lies within the interval (a,b).

114 – Numerical integration, composite rectangle rule

The composite rectangle rule

If we want to integrate the function f over the interval $[a, b]$ using the composite rectangle rule, we have to divide the given interval into n equidistant subintervals of the length $h = (b-a)/n$ with the nodes $x_i = a + ih$, $i = 0, 1, \ldots, n$.



The formula of the composite rectangle rule with the step h is then of the form

$$I_{CR} = h \sum_{i=1}^{n} f\left(\frac{x_{i-1}+x_i}{2}\right).$$

Example

Evaluate the definite integral

$$\int_{-1}^{1} \mathrm{e}^x \, \mathrm{d}x$$

using the composite rectangle rule for $n = 4$.

We equate $n = 4$, so that the step is $h = \frac{b-a}{n} = 0.5$ and we obtain the nodes $x_0 = -1$, $x_1 = -0.5$, $x_2 = 0$, $x_3 = 0.5$ a $x_4 = 1$.

$$\begin{aligned} I_{CR} &= h \sum_{i=1}^{n} f\left(\frac{x_{i-1}+x_i}{2}\right) \\ &= h\left(f\left(\frac{x_0+x_1}{2}\right) + f\left(\frac{x_1+x_2}{2}\right) + f\left(\frac{x_2+x_3}{2}\right) + f\left(\frac{x_3+x_4}{2}\right)\right) \\ &= 0.5(\mathrm{e}^{-0.75} + \mathrm{e}^{-0.25} + \mathrm{e}^{0.25} + \mathrm{e}^{0.75}) \doteq 2.3261. \end{aligned}$$

115 – Numerical integration, composite rectangle rule

Example

Evaluate the definite integral

$$\int_{-1}^{1} \mathrm{e}^x \,\mathrm{d}x$$

using the composite rectangle rule for $n = 4$.

Use the MATLAB to solve the problem.

At first we define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f = @(x)exp(x)
f =
  function_handle with value:
    @(x)exp(x)
>> a = -1
a =
    -1
>> b = 1
b =
     1
```

Then we define $n = 4$ and calculate the step h.

```
>> n = 4
n =
     4
>> h = (b-a)/n
h =
    0.5000
```

We input the vector of the midpoints between the nodes as the variable `xmid`.

```
>> xmid = a+h/2:h:b-h/2
x =
   -0.7500   -0.2500    0.2500    0.7500
>> y = f(xmid)
y =
    0.4724    0.7788    1.2840    2.1170
```

We calculate the approximate value of the given integral and input it as the variable `I`.

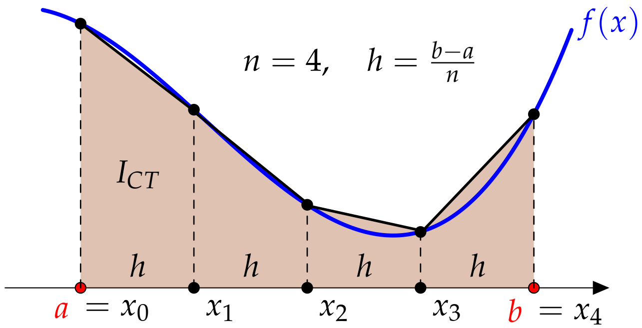
```
>> I=h*sum(y)
I =
    2.3261
```

The calculated approximate value of the given integral is 2.3261.

116 – Numerical integration, composite trapezoidal rule

The composite trapezoidal rule

If we want to integrate the function f over the interval $[a, b]$ using the composite trapezoidal rule, we have to divide the given interval into n equidistant subintervals of the length $h = (b-a)/n$ with the nodes $x_i = a + ih$, $i = 0, 1, \ldots, n$.



The formula of the composite trapezoidal rule with the step h is then of the form

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

Example

Evaluate the definite integral

$$\int_{-1}^{1} \mathrm{e}^x \, \mathrm{d}x$$

using the composite trapezoidal rule for $n = 4$.

We equate $n = 4$, so that the step is $h = \frac{b-a}{n} = 0.5$ and we obtain the nodes $x_0 = -1$, $x_1 = -0.5$, $x_2 = 0$, $x_3 = 0.5$ a $x_4 = 1$.

$$\begin{aligned} I_{CT} &= \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right) \\ &= \tfrac{h}{2}\left(f(x_0) + 2\left(f(x_1) + f(x_2) + f(x_3) \right) + f(x_n) \right) \\ &= \frac{0.5}{2}(\mathrm{e}^{-1} + 2(\mathrm{e}^{-0.5} + \mathrm{e}^{0} + \mathrm{e}^{0.5}) + \mathrm{e}^{1}) \doteq 2.3992. \end{aligned}$$

117 – Numerical integration, composite trapezoidal rule

Example

Evaluate the definite integral

$$\int_{-1}^{1} e^x \, dx$$

using the composite trapezoidal rule for $n = 4$.

Use the MATLAB to solve the problem.

At first we define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f = @(x)exp(x)
f =
  function_handle with value:
    @(x)exp(x)
>> a = -1
a =
    -1
>> b = 1
b =
     1
```

Then we define $n = 4$ and calculate the step h.

```
>> n = 4
n =
     4
>> h = (b-a)/n
h =
    0.5000
```

We input the vector of the nodes as the variable `x`.

```
>> x = a:h:b
x =
   -1.0000   -0.5000         0    0.5000    1.0000
>> y = f(x)
y =
    0.3679    0.6065    1.0000    1.6487    2.7183
```

We calculate the approximate value of the given integral and input it as the variable `I`.

```
>> I=h/2*(y(1)+2*sum(y(2:n))+y(n+1))
I =
    2.3992
```

The calculated approximate value of the given integral is 2.3992.

118 – Numerical integration, composite trapezoidal rule

Exercise

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the composite trapezoidal rule for $n = 4$ and for $n = 8$. Compare the results.

119 – Numerical integration, composite trapezoidal rule

Exercise

Evaluate the definite integral

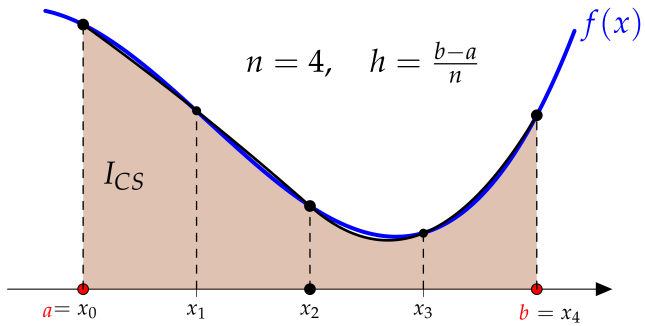
$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the composite trapezoidal rule for $n = 4$ and for $n = 8$. Compare the results.
Use the MATLAB to solve the problem.

120 – Numerical integration, composite Simpson's rule

The composite Simpson's rule

If we want to integrate the function f over the interval $[a, b]$ using the composite Simpson's rule, we have to divide the given interval into n equidistant subintervals where n has to be an even number. The length of each subinterval is $h = (b-a)/n$ and we obtain an odd number of the nodes $x_i = a + ih$, $i = 0, 1, \ldots, n$.



The formula of the composite Simpson's rule with the step h is then of the form

$$I_{CS} = \frac{h}{3}\left(f(x_0) + 4 \sum_{\substack{i=1 \\ i \text{ even}}}^{n-1} f(x_i) + 2 \sum_{\substack{i=2 \\ i \text{ odd}}}^{n-2} f(x_i) + f(x_n) \right).$$

Example

Evaluate the definite integral

$$\int_{-1}^{1} e^x \, dx$$

using the composite Simpson's rule for $n = 4$.

We euate $n = 4$, so that the step is $h = \frac{b-a}{n} = 0.5$ and we obtain the nodes $x_0 = -1$, $x_1 = -0.5$, $x_2 = 0$, $x_3 = 0.5$ a $x_4 = 1$.

$$\begin{aligned} I_{CS} &= \frac{h}{3}\left(f(x_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + f(x_{2m}) \right) \\ &= \tfrac{h}{3}\left(f(x_0) + 4\left(f(x_1) + f(x_3)\right) + 2f(x_2) + f(x_4) \right) \\ &= \tfrac{0.5}{3}\left(e^{-1} + 4(e^{-0.5} + e^{0.5}) + 2e^{0} + e^{1}\right) \doteq 2.3512. \end{aligned}$$

121 – Numerical integration, composite Simpson's rule

Example

Evaluate the definite integral

$$\int_{-1}^{1} e^x \, dx$$

using the composite Simpson's rule for $n = 4$.

Use the MATLAB to solve the problem.

At first we define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f = @(x)exp(x)
f =
  function_handle with value:
    @(x)exp(x)
>> a = -1
a =
    -1
>> b = 1
b =
     1
```

Then we define $n = 8$ and calculate the step h.

```
>> n = 4
n =
     4
>> h = (b-a)/n
h =
    0.5000
```

We input the vector of the nodes as the variable `x`.

```
>> x = a:h:b
x =
   -1.0000   -0.5000         0    0.5000    1.0000
>> y = f(x)
y =
    0.3679    0.6065    1.0000    1.6487    2.7183
```

We calculate the approximate value of the given integral and input it as the variable `I`.

```
>> I=h/3*(y(1)+4*sum(y(2:2:n))+2*sum(y(3:2:n-1))+y(n+1))
I =
    2.3512
```

The calculated approximate value of the given integral is 2.3512.

122 – Numerical integration, composite Simpson's rule

Example

Evaluate the definite integral

$$\int_{-1}^{1} \mathrm{e}^{x}\,\mathrm{d}x$$

using the composite Simpson's rule for $n = 8$.

We equate $n = 8$, so that the step is $h = \frac{b-a}{n} = 0.25$ and we obtain the nodes $x_0 = -1$, $x_1 = -0.75$, $x_2 = -0.5$, $x_3 = -0.25$, $x_4 = 0$, $x_5 = 0.25$, $x_6 = 0.5$, $x_7 = 0.75$ a $x_8 = 1$.

$$\begin{aligned} I_{CS} &= \frac{h}{3}\left(f(x_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + f(x_{2m})\right) \\ &= \frac{h}{3}\left(f(x_0) + 4\left(f(x_1) + f(x_3) + f(x_5) + f(x_7)\right) + 2\left(f(x_2) + f(x_4) + f(x_6)\right) + f(x_8)\right) \\ &= \frac{0.25}{3}(\mathrm{e}^{-1} + 4(\mathrm{e}^{-0.75} + \mathrm{e}^{-0.25} + \mathrm{e}^{0.25} + \mathrm{e}^{0.75}) + 2\left(\mathrm{e}^{-0.5} + \mathrm{e}^{0} + \mathrm{e}^{0.5}\right) + \mathrm{e}^{1}) \doteq 2.3505. \end{aligned}$$

We use the MATLAB to solve the problem.

```
>> f = @(x)exp(x);
>> a = -1;
>> b = 1;
>> n = 8;
>> h = (b-a)/n
h =
    0.2500
>> x = a:h:b
x =
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
>> y = f(x)
y =
    0.3679    0.4724    0.6065    0.7788    1.0000    1.2840    1.6487    2.1170    2.7183
>> I=h/3*(y(1)+4*sum(y(2:2:n))+2*sum(y(3:2:n-1))+y(n+1))
I =
    2.3505
```

123 – Numerical integration, composite Simpson's rule

Exercise

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the composite Simpson's rule for $n = 4$ and for $n = 8$.
Compare the results.

124 – Numerical integration, composite Simpson's rule

Exercise

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the composite Simpson's rule for $n = 4$ and for $n = 8$.
Compare the results.
Use the MATLAB to solve the problem.

125 – Numerical integration, error estimation of composite rules

Theorem

(i) Let the second derivative of the function f be continuous on the interval $[a, b]$. The composite rectangle rule and the composite trapezoidal rule have degree of exactness equal to 2, i.e. it holds:

$$\begin{aligned} |I - I_{CR}| &\leq C_1 h^2, \\ |I - I_{CT}| &\leq C_2 h^2. \end{aligned}$$

(ii) Let the derivative of order 4 of the function f be continuous on the interval $[a, b]$. The composite Simpson's rule has degree of exactness equal to 4, i.e. it holds:

$$|I - I_{CS}| \leq C_3 h^4.$$

C_1, C_2 and C_3 are some non-negative constants that are independent on the step h.

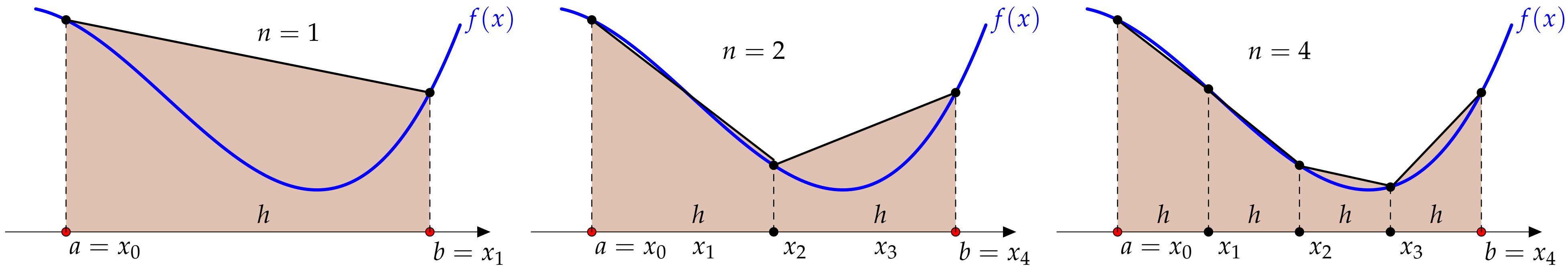
126 – Numerical integration, given accuracy

The evaluation of the integral with the given accuracy

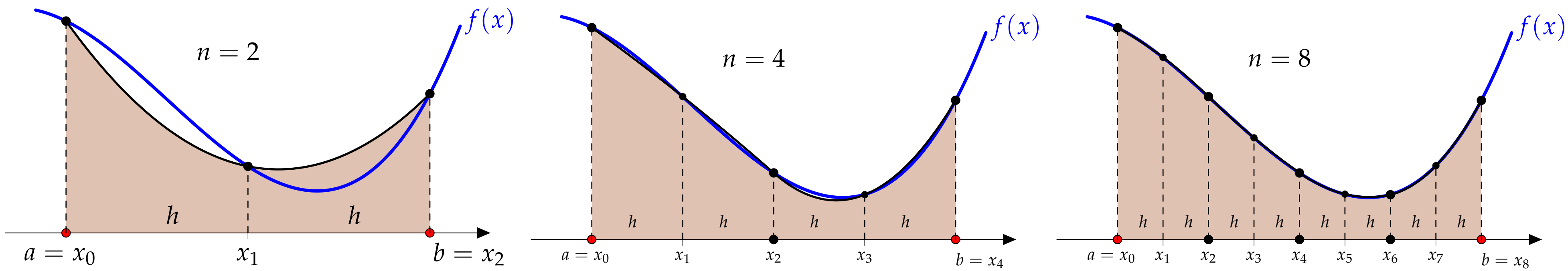
We calculate the approximate value I_h of the integral using the integration formula with the step h. Consequently, we calculate the approximate value $I_{h/2}$ with the half-length step $h/2$. We stop the calculation, if the following holds

$$|I_h - I_{h/2}| \leq \varepsilon.$$

The composite trapezoidal rules for $n = 1, 2, 4$



The composite Simpson's rules for $n = 2, 4, 8$



127 – Numerical integration, given accuracy

Example

Evaluate the definite integral

$$\int_{-1}^{1} e^x \, dx$$

using the composite trapezoidal formula with the given accuracy $\varepsilon = 10^{-4}$.
Use the MATLAB to solve the problem.

We define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f = @(x)exp(x);
>> a = -1;
>> b = 1;
```

We define $n = 2$ in the first step. We input the vector of the nodes as the variable `x`.

```
>> n = 2;
>> h = (b-a)/n;
>> x = a:h:b;
>> y = f(x);
```

We calculate the approximate value of the given integral and input it as the variable `Inew`.

```
>> Inew = h/2*(y(1)+2*sum(y(2:n))+y(n+1))
Inew =
    2.5431
```

We save the obtained value as the variable `I`. Then we double the value of n and repeat all the calculation. Finally we evaluate the error approximation $|I_h - I_{2h}|$

```
>> I = Inew;
>> n = 2*n
n =
     4
>> h = (b-a)/n;
>> x = a:h:b;
>> y = f(x);
>> Inew = h/2*(y(1)+2*sum(y(2:n))+y(n+1))
Inew =
     2.3992
>> Error = abs(Inew-I)
Error =
    0.1439
```

We repeat the previous seven statements till the error is greater than 10^{-4}.

We round the result to four decimal places and write it as

$$\int_{-1}^{1} e^x \, dx = 2.3504 \pm 10^{-4}.$$

n	I_h	$\lvert I_h - I_{2h}\rvert$	$\varepsilon = 10^{-4} = 0.0001$
2	2.5430806	—	
4	2.3991662	0.1439143	$> 10^{-4}$
8	2.3626313	0.0365349	$> 10^{-4}$
16	2.3534620	0.0091693	$> 10^{-4}$
32	2.3511674	0.0022945	$> 10^{-4}$
64	2.3505936	0.0005737	$> 10^{-4}$
128	2.3504502	0.0001434	$> 10^{-4}$
256	2.3504143	0.0000358	$\leq 10^{-4}$

128 – Numerical integration, given accuracy

Exercise

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the composite trapezoidal formula with the given accuracy $\varepsilon = 10^{-4}$.
Use the MATLAB to solve the problem.

129 – Numerical integration, given accuracy

Example

Evaluate the definite integral

$$\int_1^{\mathrm{e}} \frac{\ln x}{\sqrt{9-x^2}}\,\mathrm{d}x$$

using the composite Simpson's formula with the given accuracy $\varepsilon = 10^{-8}$.
Use the MATLAB to solve the problem.

We define the function `f` and input the limits of integration as the variables `a`, `b`.

```
>> f = @(x)log(x)./sqrt(9-x.^2);
>> a = 1;
>> b = exp(1);
```

We set $n = 2$ in the first step. We input the vector of the nodes as the variable `x`.

```
>> n = 2;
>> h = (b-a)/n;
>> x = a:h:b;
>> y = f(x);
```

Because we aim at the accuracy 10^{-8}, we can not take up with the four decimal places that the MATLAB display in the standard `short` format. We have to switch the output format to `long`.

```
>> format long
```

We calculate the approximate value of the given integral and input it as the variable `Inew`.

```
>> Inew = h/3*(y(1)+4*sum(y(2:2:n))+2*sum(y(3:2:n-1))+y(n+1))
Inew =
    0.52733592
```

130 – Numerical integration, given accuracy

We save the obtained value as the variable `I`. Then we double the value of n and repeat all calculations. Finally we evaluate the error approximation $|I_h - I_{2h}|$

```
>> I = Inew;
>> n = 2*n
n =
     4
>> h = (b-a)/n;
>> x = a:h:b;
>> y = f(x);
>> Inew = h/3*(y(1)+4*sum(y(2:2:n))+2*sum(y(3:2:n-1))+y(n+1))
Inew =
    0.51036199
>> Error = abs(Inew-I)
Error =
    0.01697393
```

We repeat the previous seven statements till the error is greater than 10^{-8}.
We round the result to eight decimal places and write it as

$$\int_1^{\mathrm{e}} \frac{\ln x}{\sqrt{9-x^2}}\,\mathrm{d}x = 0.50661191 \pm 10^{-8}.$$

n	I_h	$\lvert I_h - I_{2h}\rvert$	$\varepsilon = 10^{-8} = 0.000000001$
2	0.52733592	—	
4	0.51036199	0.01697393	$> 10^{-8}$
8	0.50708297	0.00327902	$> 10^{-8}$
16	0.50665442	0.00042855	$> 10^{-8}$
32	0.50661499	0.00003943	$> 10^{-8}$
64	0.50661211	0.00000288	$> 10^{-8}$
128	0.50661192	0.00000019	$> 10^{-8}$
256	0.50661191	0.00000001	$> 10^{-8}$
512	0.50661191	0.00000000	$\leq 10^{-8}$

131 – Numerical integration, given accuracy

Exercise

Evaluate the definite integral

$$\int_0^6 \frac{x}{1+x^2}\,\mathrm{d}x$$

using the composite Simpson's formula with the given accuracy $\varepsilon = 10^{-8}$.
Use the MATLAB to solve the problem.

132 – Numerical integration

Exercise

Evaluate the definite integral using the composite Simpson's formula with the given accuracy $\varepsilon = 10^{-8}$.
Use the MATLAB to solve the problem.

1. $$\int_0^1 x^2\sqrt{1+x^2}\,\mathrm{d}x$$
2. $$\int_1^2 \frac{\ln(1+x)}{1+\cos x}\,\mathrm{d}x$$
3. $$\int_0^1 x^2\cos\left((x^2)\right)\,\mathrm{d}x$$
4. $$\int_0^1 \frac{1}{\sqrt{1+4x-x^4}}\,\mathrm{d}x$$
5. $$\int_0^2 x^2 \mathrm{e}^{-x^2}\,\mathrm{d}x$$
6. $$\int_1^2 \frac{\cos^2(4x)}{x}\,\mathrm{d}x$$
7. $$\int_0^1 \frac{\sqrt{1+x^2}}{1+\cos x}\,\mathrm{d}x$$
8. $$\int_1^2 \frac{\ln(1+x^2)}{1+x^2}\,\mathrm{d}x$$
9. $$\int_{0.5}^2 \sqrt{1+x^4}\,\mathrm{d}x$$
10. $$\int_1^2 \frac{\sin^2(3x)}{x^2}\,\mathrm{d}x$$
11. $$\int_0^1 x^3\sqrt{1+x^3}\,\mathrm{d}x$$
12. $$\int_1^3 \frac{\cos^2 x}{x^2}\,\mathrm{d}x$$
13. $$\int_1^2 x^3 \mathrm{e}^{-x^3}\,\mathrm{d}x$$
14. $$\int_1^2 \frac{\mathrm{e}^x}{x^2}\,\mathrm{d}x$$
15. $$\int_0^1 \frac{x}{\sqrt{1+x^4}}\,\mathrm{d}x$$
16. $$\int_0^1 \frac{x^2}{1+\sin x}\,\mathrm{d}x$$
17. $$\int_0^{0.8} \frac{x^3+x}{\cos^2 x}\,\mathrm{d}x$$

18.
$$\int_1^2 \frac{\sqrt{1+x^4}}{x^4}\,\mathrm{d}x$$

19.
$$\int_1^2 \frac{1}{x^2\sin^2 x}\,\mathrm{d}x$$

20.
$$\int_0^1 \sqrt{\frac{1-x^3}{1+x^3}}\,\mathrm{d}x$$

21.
$$\int_1^2 \mathrm{e}^{-x^2+2x+1}\,\mathrm{d}x$$

22.
$$\int_0^1 \cos\left(\frac{x^2}{2}\right)\,\mathrm{d}x$$

23.
$$\int_0^1 \sqrt{x}\,\cos x\,\mathrm{d}x$$

24.
$$\int_{0.5}^{2.1} \frac{\ln(1+x)}{1+\cos x}\,\mathrm{d}x$$

25.
$$\int_{-2.2}^{-4.2} \frac{3-2x}{\sqrt{3x^2-2x-1}}\,\mathrm{d}x$$

26.
$$\int_{0.3}^{0.9} \frac{\ln\left(1+\sqrt{x}\right)}{\sqrt{x}}\,\mathrm{d}x$$

Worksheets for Numerical methods

Numerical solution of ordinary differential equations

135 – Numerical solution of ordinary differential equations

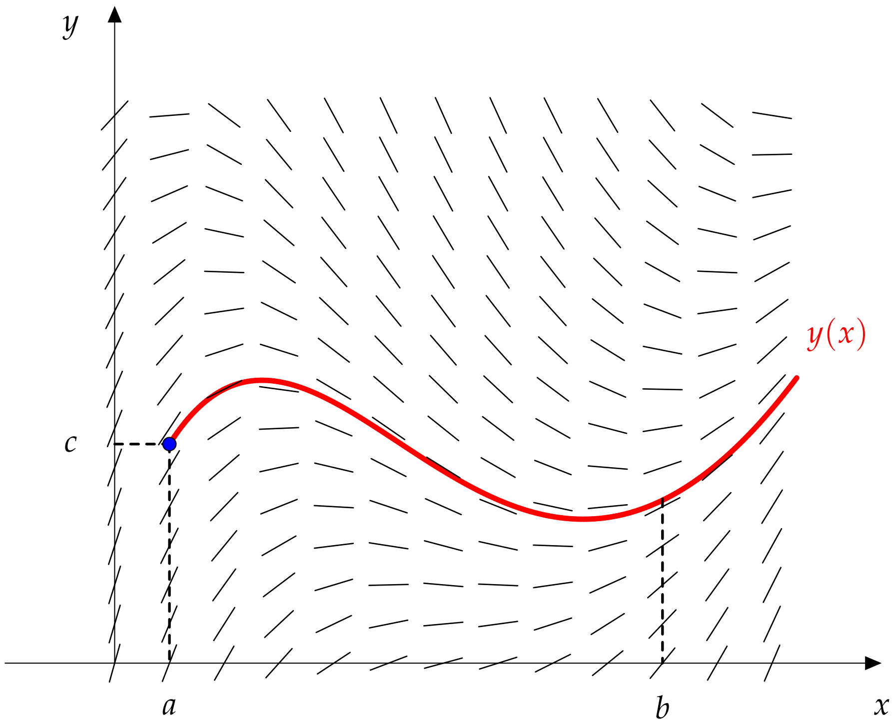
The initial-value problem for the ordinary differential equation
We find the continuous function $y = y(x)$ that on the interval $[a, b]$ fulfil the differential equation

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

and the initial condition

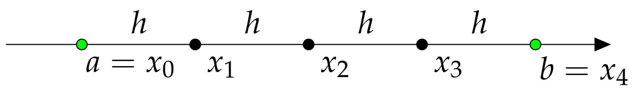
$$y(a) = c.$$

This continuous function is the **analytical solution** of the initial-value problem.

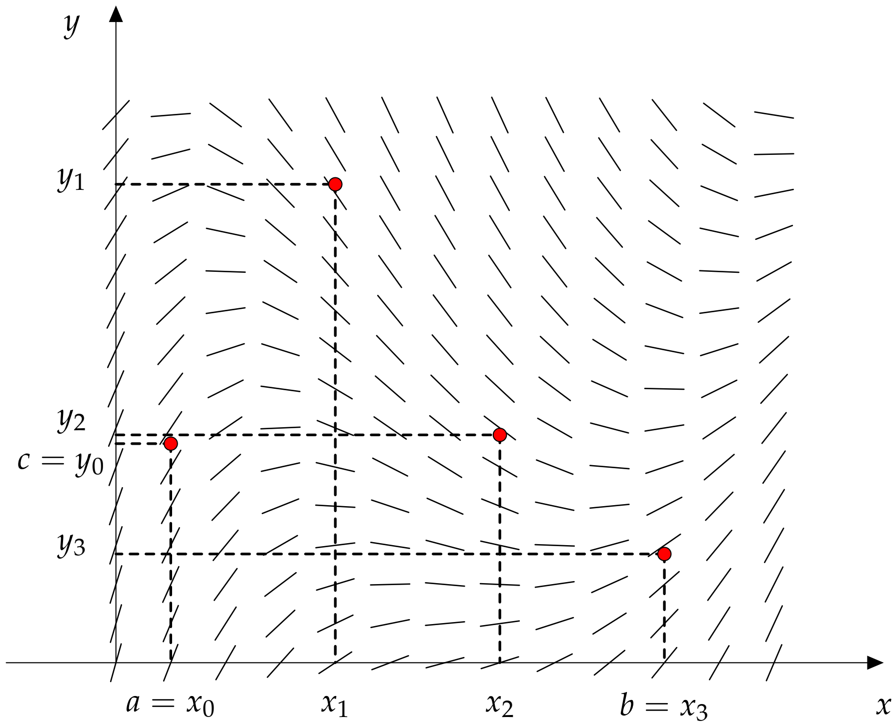


To solve the problem numerically we divide the interval $[a, b]$ into n equidistant subintervals of the length $h = (b-a)/n$ with the nodes $x_0 = a, x_1, x_2, \ldots, x_n = b$,
i.e.

$$x_i = a + ih, \quad i = 0, 1, \ldots, n.$$



To these nodes we assign values $y_0 = c, y_1, y_2, \ldots, y_n$ that approximate values of the analytical solution $y(x_0), y(x_1), y(x_2), \ldots, y(x_n)$.
Thus the **numerical solution** of the initial-value problem is a set of $n + 1$ discrete points $[x_i; y_i]$, $i = 0, 1, \ldots, n$.



136 – Numerical solution of ordinary differential equations, Euler method

Euler method

At first we calculate the nodes

$$x_i = a + ih, \quad i = 0, 1, \ldots, n.$$

We consider the differential equation

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

in a node x_i and we replace the accurate value of the solution $y(x_i)$ by its approximation y_i.

Next we approximate the derivative on the left side using the numerical formula

$$y'(x_i) = f(x_i, y(x_i)) \quad \approx \quad \frac{y_{i+1} - y_i}{h} = f(x_i, y_i).$$

If the values x_i, y_i are known, we can calculate an unknown value y_{i+1}

$$\begin{aligned} \frac{y_{i+1} - y_i}{h} &= f(x_i, y_i) \\ y_{i+1} - y_i &= h \cdot f(x_i, y_i) \\ y_{i+1} &= y_i + h \cdot f(x_i, y_i) \end{aligned}$$

The initial value y_0 is given by the initial condition and the other values y_{i+1} we can calculate by the derived formula.

$$\begin{aligned} &y_0 = c \\ &\text{for } i = 0, \ldots, n-1 \\ &\quad y_{i+1} = y_i + hf(x_i, y_i), \end{aligned}$$

The Euler method is the **method of the first order** – the global error is bounded by the product of the step size h and a constant $C > 0$ independent on h:

$$|y_i - y(x_i)| \le Ch\,, \quad \forall i = 0, 1, \ldots, n.$$

137 – Numerical solution of ordinary differential equations, Euler method

Example

Solve the initial-value problem

$$y' = x^2 - 0.2y, \quad y(-2) = -1$$

on the interval $[-2, 3]$ using the Euler method with the step $h = 1$.

At first we specify the number n of subintervals into which we divide the given interval $[a, b]$

$$n = \frac{b-a}{h} = \frac{3-(-2)}{1} = 5.$$

Then we calculate the nodes:

$$\begin{aligned}
x_0 &= a = -2\\
x_1 &= a + h = -2 + 1 = -1\\
x_2 &= a + 2h = -2 + 2 = 0\\
x_3 &= a + 3h = 1\\
x_4 &= a + 4h = 2\\
x_5 &= a + 5h = 3
\end{aligned}$$

The value $y_0 = -1$ is given by the initial condition, other values y_i for $i = 1, \dots, 5$ can be calculated by the formula $y_{i+1} = y_i + hf(x_i, y_i)$.

There it holds $f(x, y) = x^2 - 0.2y$ and $h = 1$, so the computational formula is of the form:

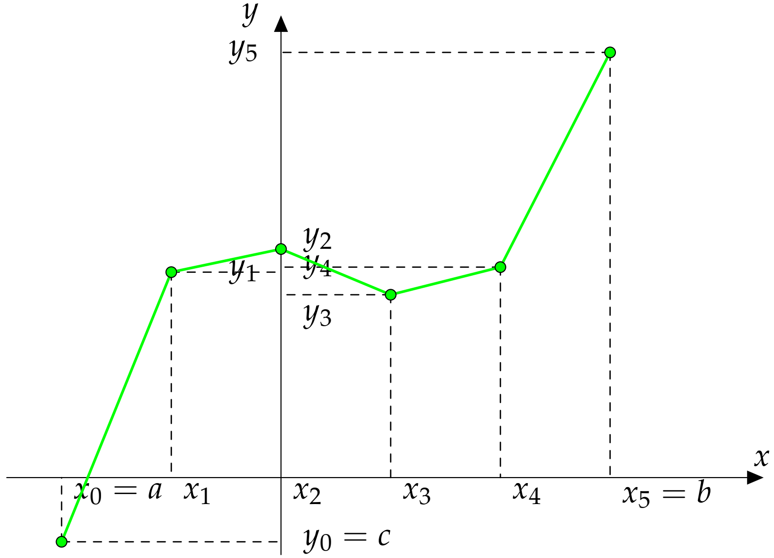
$$y_{i+1} = y_i + x_i^2 - 0.2y_i$$

$$\begin{aligned}
&y_0 = -1 \qquad \text{(given initial value)}\\
&y_1 = y_0 + hf(x_0, y_0) = y_0 + x_0^2 - 0.2y_0 = -1 + (-2)^2 - 0.2 \cdot (-1) = 3.2\\
&y_2 = y_1 + hf(x_1, y_1) = y_1 + x_1^2 - 0.2y_1 = 3.2 + (-1)^2 - 0.2 \cdot 3.2 = 3.56\\
&y_3 = y_1 + hf(x_2, y_2) = y_2 + x_2^2 - 0.2y_2 = 2.848\\
&y_4 = y_1 + hf(x_3, y_3) = y_3 + x_3^2 - 0.2y_3 = 3.2784\\
&y_5 = y_1 + hf(x_4, y_4) = y_4 + x_4^2 - 0.2y_4 = 6.6227
\end{aligned}$$

In the end we write the calculated values of the obtained numerical solution of the initial-value problem to a table

x_i	-2	-1	0	1	2	3
y_i	-1	3.2	3.56	2.848	3.2784	6.6227

and plot the graph of this numerical solution:



138 – Numerical solution of ordinary differential equations, Euler method

Example

Solve the initial-value problem

$$y' = x^2 - 0.2y, \quad y(-2) = -1$$

on the interval $[-2, 3]$ using the Euler method with the step $h = 1$.
Use the MATLAB to solve the problem.

We input both end-points of the interval $[a, b]$ as well as the value c of the initial condition and we define the function f from the right side of the differential equation.

```
>> a=-2
a =
    -2
>> b=3
b =
     3
>> c=-1
c =
    -1
>> f=@(x,y)(x.^2-0.2*y)
f =
     @(x,y)(x.^2-0.2*y)
```

Then we input the step size h and calculate the number of subintervals $n = \frac{b-a}{h} = \frac{3-(-2)}{1} = 5$.

```
>> h=1
h =
    1
>> n=(b-a)/h
n =
    5
```

Using the colon notation we generate the nodes x_i.

```
>>  x=a:h:b
x =
     -2    -1     0     1     2     3
```

We input the value y_0 and calculate other values y.

```
>> y(1)=c
y =
    -1
>> for i=1:n, y(i+1)=y(i)+h*f(x(i),y(i)); end
>> y
y =
   -1.0000   3.2000   3.5600   2.8480   3.2784   6.6227
```

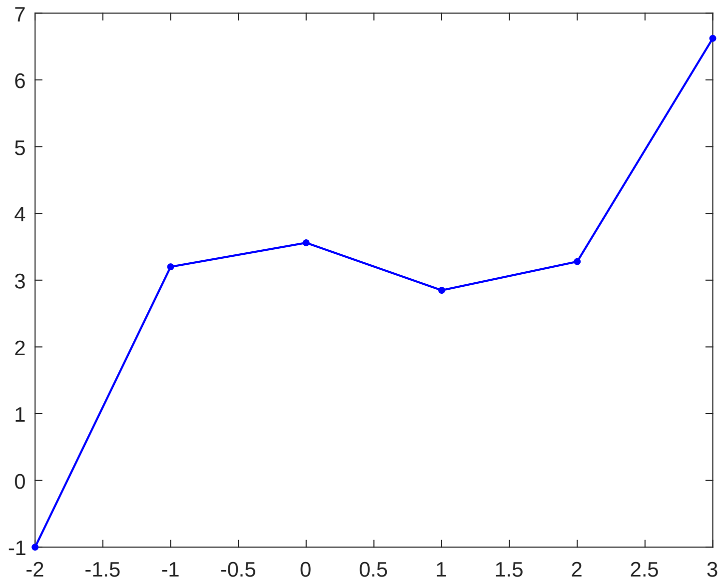
Let us note that the indexing in MATLAB goes from 1, i.e. the values $y_0, y_1, \ldots, y_n$ are input as the variables `y(1)`, `y(2)`, ..., `y(n+1)`.

We write the numerical solution values to a table and plot the solution graph.

```
>> [x;y]
ans =
  -2.0000   -1.0000         0    1.0000    2.0000    3.0000
  -1.0000    3.2000    3.5600    2.8480    3.2784    6.6227
>> plot(x,y,'b.-')
```

The resulting numerical solution of the given initial-value problem is:

x_i	-2	-1	0	1	2	3
y_i	-1	3.2	3.56	2.848	3.2784	6.6227



140 – Numerical solution of ordinary differential equations, Euler method

Exercise

Solve the initial-value problem

$$y' = \frac{3y - 2x}{x + y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Euler method with the step $h = 0.5$.

141 – Numerical solution of ordinary differential equations, Euler method

Exercise

Solve the initial-value problem

$$y' = \frac{3y - 2x}{x + y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Euler method with the step $h = 0.5$.
Use the MATLAB to solve the problem.

142 – Numerical solution of ordinary differential equations, Euler method

Example

Solve the initial-value problem

$$y' = y - x^2 + 2, \quad y(0) = -1$$

on the interval $[0, 2]$ using the Euler method with the steps $h = 0.5$ and $h = 0.1$.
Compare obtained numerical solution with the analytical one.
Use the MATLAB to solve the problem.

We input both end-points of the interval $[a, b]$ as well as the value c of the initial condition and we define the function f from the right side of the differential equation.

```
>> a=0
a =
     0
>> b=2
b =
     2
>> c=-1
c =
    -1
>> f=@(x,y)(y-x.^2+2)
f =
     @(x,y)(y-x.^2+2)
```

Then we input the step size h and calculate the number of subintervals $n = \frac{b-a}{h}$.

```
>> h=0.5
h =
    0.5000
```

```
>> n=(b-a)/h
n =
     4
```

Using the colon notation we generate the nodes $x_i = a + ih$ for $i = 0, \ldots, n$.

```
>> x=a:h:b
x =
         0    0.5000    1.0000    1.5000    2.0000
```

We input the value y_0 and calculate other values y.

```
>> y(1)=c
y =
    -1

>> for i=1:n, y(i+1)=y(i)+h*f(x(i),y(i)); end
>> y
y =
   -1.0000   -0.5000    0.1250    0.6875    0.9063
```

Let us note that the indexing in MATLAB goes from 1, i.e. the values $y_0, y_1, \ldots, y_n$ are input as the variables `y(1)`, `y(2)`, ..., `y(n+1)`.

We write the obtained numerical solution values to a table.

```
>> [x;y]
ans =
         0    0.5000    1.0000    1.5000    2.0000
   -1.0000   -0.5000    0.1250    0.6875    0.9063
```

The resulting numerical solution of the given initial-value problem is:

x_i	0	0.5	1	1.5	2
y_i	-1	-0.5	0.125	0.6875	0.9063

We save the values of this numerical solution (calculated with the step $h = 0.5$) to the variables `x05` a `y05` and clear the variables `x`, `y`, `n`, `h` to prepare these for the next calculations.

```
x05=x; y05=y;
clear x y n h
```

Then we input the step size h and calculate the number of subintervals n.

```
>> h=0.1
h =
    0.1000
>> n=(b-a)/h
n =
    20
```

Using the colon notation we generate the nodes x_i.

```
>> x=a:h:b;
```

We input the initial value y_0 and calculate other values y.

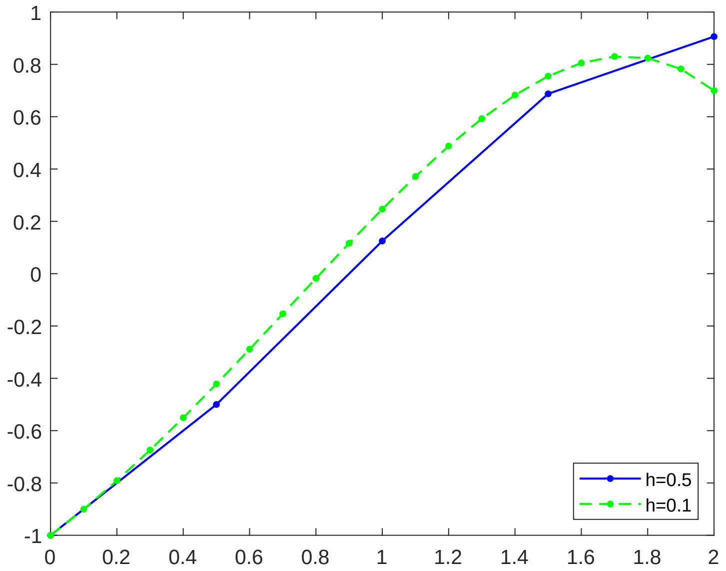
```
>> y(1)=c
y =
    -1
>> for i=1:n, y(i+1)=y(i)+h*f(x(i),y(i)); end
```

We write the numerical solution values to a table.

```
>> [x;y]
ans =
  Columns 1 through 6
        0    0.1000    0.2000    0.3000    0.4000    0.5000
  -1.0000   -0.9000   -0.7910   -0.6741   -0.5505   -0.4216

  Columns 7 through 12
   0.6000    0.7000    0.8000    0.9000    1.0000    1.1000
  -0.2887   -0.1536   -0.0179    0.1163    0.2469    0.3716

  Columns 13 through 18
   1.2000    1.3000    1.4000    1.5000    1.6000    1.7000
   0.4877    0.5925    0.6828    0.7550    0.8055    0.8301

  Columns 19 through 21
   1.8000    1.9000    2.0000
   0.8241    0.7825    0.6998
```

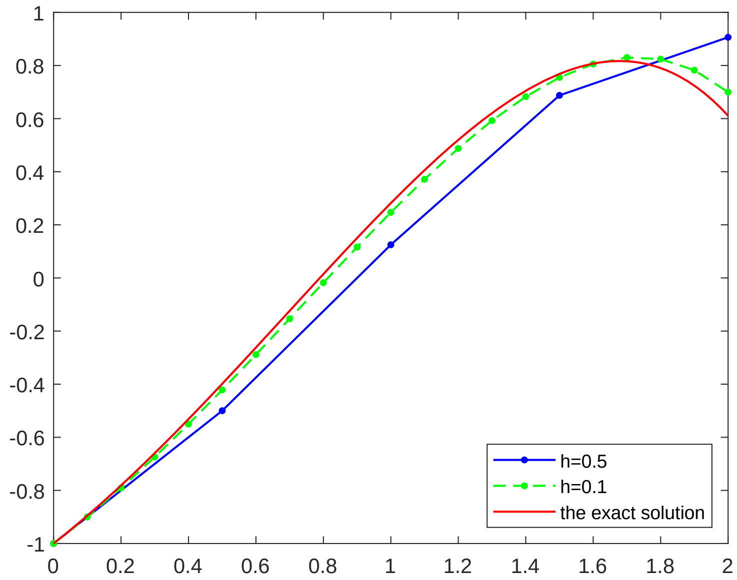
Now we can plot graphs of numerical solutions corresponding to the step size $h = 0.5$ (saved as `x05`, `y05`) and to the step size $h = 0.1$.

```
>> x01=x; y01=y;
>> plot(x05,y05,'b.-',x01,y01,'g.--')
>> legend('h=0.5','h=0.1')
```



We compare both numerical solutions with the known analytical solution of given initial-value problem that is $y = x^2 + 2x - e^x$.

```
>> plot(x05,y05,'b.-',x01,y01,'g.--')
>> hold on
>> fplot(@(x)x.^2+2*x-exp(x),[0,2],'r')
>> legend('h=0.5','h=0.1','analytical solution')
```



145 – Numerical solution of ordinary differential equations, Euler method

Exercise

Solve the initial-value problem

$$y' = x^2 - 0.2y, \quad y(-2) = -1$$

on the interval $[-2, 3]$ using the Euler method with the steps $h = 1$ and $h = 0.5$.
Compare obtained numerical solution with the analytical one.
Use the MATLAB to solve the problem.

146 – Numerical solution of ordinary differential equations, Euler method

Exercise

Solve the initial-value problem

$$y' = \frac{3y - 2x}{x + y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Euler method with the steps $h = 0.5$, $h = 0.1$ and $h = 0.05$.
Use the MATLAB to solve the problem.

147 – Numerical solution of ordinary differential equations, Heun method

Heun method

The equidistant nodes are the same as for the Euler method, i.e.

$$x_i = a + ih, \quad i = 0, 1, \ldots, n.$$

The value y_0 is also given by the initial condidion $y_0 = c$.
The method principle is that for every $i = 0, \ldots, n-1$ the value y_i is already known and the value y_{i+1} is to be found. In contrast to the Euler method, in each step we first have to evaluate auxiliary constants k_1, k_2 and next we calculate the value y_{i+1} from these:

$$\begin{aligned}
&y_0 = c \\
&\text{for } i = 0, \ldots, n-1 \\
&\quad k_1 = hf(x_i, y_i) \\
&\quad k_2 = hf(x_i + h, y_i + k_1) \\
&\quad y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)
\end{aligned}$$

The Heun method is the **method of the second order** – the global error is bounded by the product of h^2 and a constant $C > 0$ independent on h:

$$|y_i - y(x_i)| \leq Ch^2, \quad \forall i = 0, 1, \ldots, n.$$

148 – Numerical solution of ordinary differential equations, Heun method

Example

Solve the initial-value problem

$$y' = x^2 - 0.2y, \quad y(-2) = -1$$

on the interval $[-2, 3]$ using the Heun method with the step $h = 1$.

At first we calculate the number of subintervals into which we divide the given interval $[a, b]$

$$n = \frac{b-a}{h} = \frac{3-(-2)}{1} = 5$$

and the nodes

$$\begin{aligned} x_0 &= a = -2 \\ x_1 &= a + h = -2 + 1 = -1 \\ x_2 &= a + 2h = -2 + 2 = 0 \\ x_3 &= a + 3h = 1 \\ x_4 &= a + 4h = 2 \\ x_5 &= a + 5h = 3\,. \end{aligned}$$

The initial condition determine the value $y_0 = -1$. In following steps we first evaluate constants k_1, k_2 and using these we calculate the required value y_{i+1}.

$$y_0 = -1 \qquad \text{(initial condition)}$$

Since it holds $f(x, y) = x^2 - 0.2y$ and $h = 1$ in this example, the calculations look like as follows:

for $i = 0$

$$\begin{aligned} k_1 &= hf(x_0, y_0) = 1 \cdot ((-2)^2 - 0.2 \cdot (-1)) = 4.2 \\ k_2 &= hf(x_0 + h, y_0 + k_1) = 1 \cdot ((-2+1)^2 - 0.2 \cdot (-1 + 4.2)) = 0.36 \end{aligned}$$

$$y_1 = y_0 + \frac{1}{2}(k_1 + k_2) = -1 + \frac{1}{2}(4.2 + 0.36) = 1.28$$

for $i = 1$

$$\begin{aligned} k_1 &= hf(x_1, y_1) = 1 \cdot ((-1)^2 - 0.2 \cdot 1.28) = 0.744 \\ k_2 &= hf(x_1 + h, y_1 + k_1) = 1 \cdot ((-1+1)^2 - 0.2 \cdot (1.28 + 0.744)) = \\ &- 0.4048 \end{aligned}$$

$$y_2 = y_1 + \frac{1}{2}(k_1 + k_2) = 1.28 + \frac{1}{2}(0.744 + (-0.4048)) = 1.4496$$

for $i = 2$

$$\begin{aligned} k_1 &= hf(x_2, y_2) = -0.2899 \\ k_2 &= hf(x_2 + h, y_2 + k_1) = 0.7681 \end{aligned}$$

$$y_3 = y_2 + \frac{1}{2}(k_1 + k_2) = 1.6887$$

for $i = 3$

$$\begin{aligned} k_1 &= hf(x_3, y_3) = 0.6623 \\ k_2 &= hf(x_3 + h, y_3 + k_1) = 3.5298 \end{aligned}$$

$$y_4 = y_3 + \frac{1}{2}(k_1 + k_2) = 3.7847$$

for $i = 4$

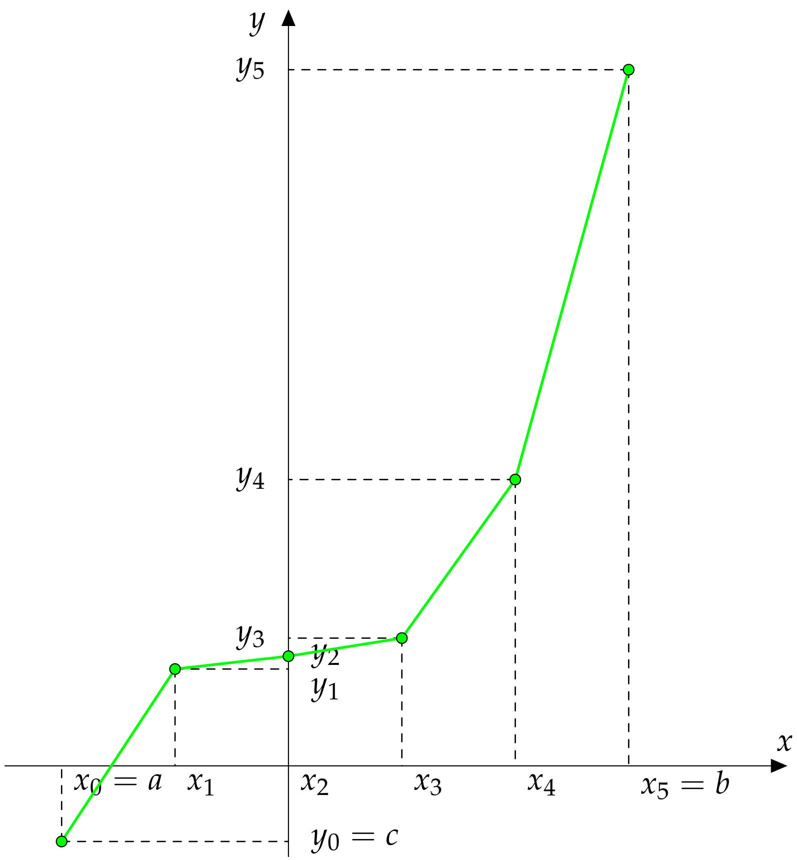
$$\begin{aligned} k_1 &= hf(x_4, y_4) = 3.2431 \\ k_2 &= hf(x_4 + h, y_4 + k_1) = 7.5944 \end{aligned}$$

$$y_5 = y_4 + \frac{1}{2}(k_1 + k_2) = 9.2035$$

We write the obtained numerical solution values to a table

x_i	-2	-1	0	1	2	3
y_i	-1	1.2800	1.4496	1.6887	3.7847	9.2035

and plot the graph of this numerical solution:



150 – Numerical solution of ordinary differential equations, Heun method

Example

Solve the initial-value problem

$$y' = x^2 - 0.2y, \quad y(-2) = -1$$

on the interval $[-2, 3]$ using the Heun method with the step $h = 1$. Use the MATLAB to solve the problem.

We input both end-points of the interval $[a, b]$ as well as the value c of the initial condition and we define the function f from the right side of the differential equation.

```
>> a=-2
a =
     -2
>> b=3
b =
     3
>> c=-1
c =
    -1
>> f=@(x,y)(x.^2-0.2*y)
f =
     @(x,y)(x.^2-0.2*y)
```

Then we input the step size h and calculate the number of subintervals $n = \frac{b-a}{h} = \frac{3-(-2)}{1} = 5$.

```
>> h=1
h =
    1
>> n=(b-a)/h
n =
    5
```

Using the colon notation we generate the nodes x_i.

```
>>  x=a:h:b
x =
     -2    -1     0     1     2     3
```

We input the value y_0 and calculate other values y.

```
>> y(1)=c
y =
    -1
>> for i=1:n,
  k1=h*f(x(i),y(i)),
  k2=h*f(x(i+1),y(i)+k1),
  y(i+1)=y(i)+1/2*(k1+k2),
end
k1 =
    4.2000
k2 =
    0.3600
y =
   -1.0000    1.2800

k1 =
    0.7440
k2 =
   -0.4048

y =
   -1.0000    1.2800    1.4496
```

Let us note that the indexing in MATLAB goes from 1, i.e. the values $y_0, y_1, \ldots, y_n$ are input as the variables `y(1)`, `y(2)`, ..., `y(n+1)`.

151 – Numerical solution of ordinary differential equations, Heun method

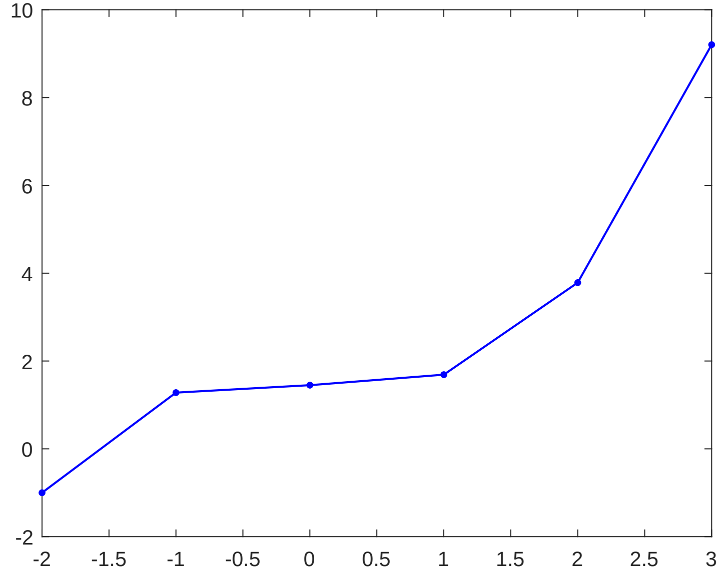

```
k1 =
   -0.2899
k2 =
    0.7681

y =
   -1.0000   1.2800   1.4496   1.6887

k1 =
    0.6623
k2 =
    3.5298
y =
   -1.0000   1.2800   1.4496   1.6887   3.7847

k1 =
    3.2431
k2 =
    7.5944
y =
   -1.0000   1.2800   1.4496   1.6887   3.7847   9.2035
```

We write the obtained numerical solution values to a table and plot this solution graph.

```
>> [x;y]
ans =
 -2.0000   -1.0000         0    1.0000    2.0000    3.0000
 -1.0000    1.2800    1.4496    1.6887    3.7847    9.2035

>> plot(x,y,'b.-')
```

The resulting numerical solution of the given initial-value problem is:

x_i	-2	-1	0	1	2	3
y_i	-1	1.2800	1.4496	1.6887	3.7847	9.2035



152 – Numerical solution of ordinary differential equations, Heun method

Exercise

Solve the initial-value problem

$$y' = \frac{3y-2x}{x+y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Heun method with the step $h = 0.5$.

153 – Numerical solution of ordinary differential equations, Heun method

Exercise

Solve the initial-value problem

$$y' = \frac{3y - 2x}{x + y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Heun method with the step $h = 0.5$.
Use the MATLAB to solve the problem.

154 – Numerical solution of ordinary differential equations, Runge-Kutta method

Runge-Kutta method RK4

The equidistant nodes are the same as before, i.e.

$$x_i = a + ih, \quad i = 0, 1, \dots, n$$

and the value y_0 is again determined by the initial condition $y_0 = c$.
The method principle is that for every $i = 0, \dots, n-1$ the value y_i is already known and the value y_{i+1} is to be found. Analogous to the Heun method, in each step we first have to evaluate auxiliary constants k_1, k_2, k_3, k_4 and next we calculate the value y_{i+1} from these:

$$\begin{aligned}
&y_0 = c \\
\text{for } &i = 0, \dots, n-1 \\
&k_1 = hf(x_i, y_i) \\
&k_2 = hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1) \\
&k_3 = hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2) \\
&k_4 = hf(x_i + h, y_i + k_3) \\
&y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).
\end{aligned}$$

The Runge-Kutta method RK4 is the **method of the fourth order** – the global error is bounded by the product of h^4 and a constant $C > 0$ independent on h:

$$|y_i - y(x_i)| \le Ch^4 , \quad \forall i = 0, 1, \dots, n$$

155 – Numerical solution of ordinary differential equations, Runge-Kutta method

Example

Solve the initial-value problem

$$y' = y - x^2 + 2, \quad y(0) = -1$$

on the interval $[0, 2]$ using the Runge-Kutta method RK4 with the step $h = 0.5$.

We calculate the number of subintervals into which we divide the given interval $[a, b]$

$$n = \frac{b-a}{h} = \frac{2-0}{0.5} = 4$$

and the nodes

$$\begin{aligned}
x_0 &= a = 0 \\
x_1 &= a + h = 0 + 0.5 = 0.5 \\
x_2 &= a + 2h = 0 + 1 = 1 \\
x_3 &= a + 3h = 1 + 1.5 = 1.5 \\
x_4 &= a + 4h = 0 + 2
\end{aligned}$$

The initial condition determine the value $y_0 = -1$. In consequent steps we first evaluate constants k_1, k_2, k_3, k_4 and using these we calculate the required value y_{i+1}.

$$y_0 = -1 \qquad \text{(initial condition)}$$

Since it holds $f(x, y) = y - x^2 + 2$ and $h = 0.5$ in this example, the calculations look like as follows:

for $i = 0$

$$\begin{aligned}
k_1 &= hf(x_0, y_0) = 0.5 \\
k_2 &= hf(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_1) = 0.5938 \\
k_3 &= hf(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}k_2) = 0.6172 \\
k_4 &= hf(x_0 + h, y_0 + k_3) = 0.6836 \\
y_1 &= y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \\
&= -1 + \frac{1}{6}(0.5 + 2 \cdot 0.5938 + 2 \cdot 0.6172 + 0.6836) = -0.3991
\end{aligned}$$

for $i = 1$

$$\begin{aligned}
k_1 &= hf(x_1, y_1) = 0.6755 \\
k_2 &= hf(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}k_1) = 0.6881 \\
k_3 &= hf(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}k_2) = 0.6912 \\
k_4 &= hf(x_1 + h, y_1 + k_3) = 0.6461 \\
y_2 &= y_1 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 0.2809
\end{aligned}$$

for $i = 2$

$$
\begin{aligned}
k_1 &= hf(x_2, y_2) = 0.6405 \\
k_2 &= hf(x_2 + \tfrac{1}{2}h, y_2 + \tfrac{1}{2}k_1) = 0.5193 \\
k_3 &= hf(x_2 + \tfrac{1}{2}h, y_2 + \tfrac{1}{2}k_2) = 0.4890 \\
k_4 &= hf(x_2 + h, y_2 + k_3) = 0.2600 \\
y_3 &= y_2 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
&= 0.7671
\end{aligned}
$$

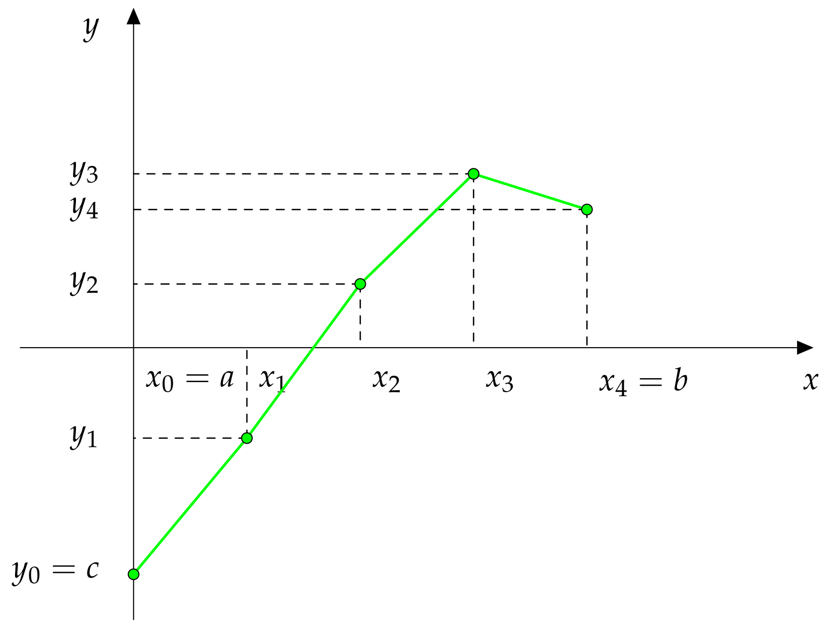
for $i = 3$

$$
\begin{aligned}
k_1 &= hf(x_2, y_2) = 0.2586 \\
k_2 &= hf(x_3 + \tfrac{1}{2}h, y_3 + \tfrac{1}{2}k_1) = -0.0830 \\
k_3 &= hf(x_3 + \tfrac{1}{2}h, y_3 + \tfrac{1}{2}k_2) = -0.1684 \\
k_4 &= hf(x_3 + h, y_3 + k_3) = -0.7007 \\
y_4 &= y_3 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
&= 0.6096
\end{aligned}
$$

We write the obtained numerical solution to a table

x_i	0	0.5	1	1.5	2
y_i	-1	-0.3991	0.2809	0.7671	0.6096

and plot this solution graph:



157 – Numerical solution of ordinary differential equations, Runge-Kutta method

Example

Solve the initial-value problem

$$y' = y - x^2 + 2, \quad y(0) = -1$$

on the interval $[0, 2]$ using the Runge-Kutta method RK4 with the step $h = 0.5$.
Use the MATLAB to solve the problem.

We input both end-points of the interval $[a, b]$ as well as the value c of the initial condition and we define the function f from the right side of the differential equation.

```
>> a=0
a =
     -2
>> b=2
b =
     3
>> c=-1
c =
    -1
>> f=@(x,y)(y-x.^2+2)
f =
     @(x,y)(y-x.^2+2)
```

Then we input the step size h and calculate the number of subintervals n.

```
>> h=0.5
h =
    0.5000
>> n=(b-a)/h
n =
    4
```

Using the colon notation we generate the nodes x_i.

```
>>  x=a:h:b
x =
    0     0.5     1     1.5     2
```

We input the value y_0 and calculate other values y.

```
>> y(1)=c
y =
    -1
>> for i=1:n
  k1=h*f(x(i),y(i))
  k2=h*f(x(i)+h/2,y(i)+1/2*k1)
  k3=h*f(x(i)+h/2,y(i)+1/2*k2)
  k4=h*f(x(i)+h,y(i)+k3)
  y(i+1)=y(i)+1/6*(k1+2*k2+2*k3+k4)
end
k1 =
    0.5000
k2 =
    0.5938
k3 =
    0.6172
k4 =
    0.6836
y =
   -1.0000   -0.3991
```

Let us note that the indexing in MATLAB goes from 1, i.e. the values $y_0, y_1, \ldots, y_n$ are input as the variables `y(1)`, `y(2)`, ..., `y(n+1)`, and that the cycle statement includes all between `for` and `end`.

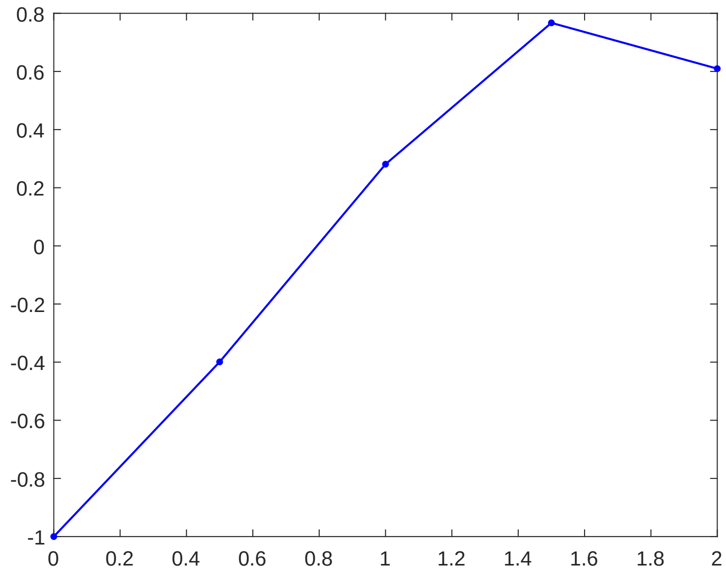
```
k1 =
    0.6755
k2 =
    0.6881
k3 =
    0.6912
k4 =
    0.6461
y =
   -1.0000   -0.3991    0.2809
k1 =
    0.6405
k2 =
    0.5193
k3 =
    0.4890
k4 =
    0.2600
y =
   -1.0000   -0.3991    0.2809    0.7671
k1 =
    0.2586
k2 =
   -0.0830
k3 =
   -0.1684
k4 =
   -0.7007
y =
   -1.0000   -0.3991    0.2809    0.7671    0.6096
```

We write the obtained numerical solution values to a table and plot this solution graph.

```
>> [x;y]
ans =
         0    0.5000    1.0000    1.5000    2.0000
   -1.0000   -0.3991    0.2809    0.7671    0.6096

>> plot(x,y,'b.-')
```

The resulting numerical solution of the given initial-value problem is:

x_i	0	0.5	1	1.5	2
y_i	-1	-0.3991	0.2809	0.7671	0.6096



159 – Numerical solution of ordinary differential equations, Runge-Kutta method

Exercise

Solve the initial-value problem

$$y' = \frac{3y - 2x}{x + y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Runge-Kutta method RK4 with the step $h = 0.5$.

160 – Numerical solution of ordinary differential equations, Runge-Kutta method

Exercise

Solve the initial-value problem

$$y' = \frac{3y - 2x}{x + y}, \quad y(3) = 2$$

on the interval $[3, 6]$ using the Runge-Kutta method RK4 with the step $h = 0.5$.
Use the MATLAB to solve the problem.

161 – Numerical solution of ordinary differential equations, Runge-Kutta method

Exercise

Solve the initial-value problem

$$y' = y - x^2 + 2, \quad y(0) = -1$$

on the interval $[0, 2]$ using the Runge-Kutta method RK4 with the steps $h = 0.5$ and $h = 0.1$.
Compare obtained numerical solution with the analytical one.
Use the MATLAB to solve the problem.

162 – Numerical solution of ordinary differential equations

Exercise

Solve the initial-value problem

$$y' = y - x^2 + 2, \quad y(0) = -1$$

on the interval $[0, 2]$ using the Euler method and the Runge-Kutta method RK4 both with the same step $h = 0.5$.
Compare values of both numerical solutions in a table as well as graphically.
Use the MATLAB to solve the problem.

163 – Numerical solution of ordinary differential equations

Exercise

Solve the initial-value problem

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

on the interval $[a,b]$ using the Euler method and the Runge-Kutta method RK4 both with the same step $h = 0.5$.
Compare values of both numerical solutions in a table as well as graphically.
Use the MATLAB to solve the problem.

1.
$$f(x,y) = \frac{y}{x^2+1}, \quad a=0,\ b=1,\ c=2$$

2.
$$f(x,y) = 3xy^2 + \frac{y}{x}, \quad a=1,\ b=2,\ c=-1$$

3.
$$f(x,y) = \frac{x^2+y}{x}, \quad a=1,\ b=2,\ c=0$$

4.
$$f(x,y) = \frac{y}{x} - y^2, \quad a=1,\ b=2,\ c=1$$

5.
$$f(x,y) = x - xy, \quad a=0,\ b=1,\ c=3$$

6.
$$f(x,y) = \frac{3x+y-2}{2-x}, \quad a=0,\ b=1,\ c=4$$

7.
$$f(x,y) = \sqrt{\frac{y}{x}}, \quad a=1,\ b=2,\ c=4$$

8.
$$f(x,y) = \frac{y}{x}(1+\ln y - \ln x), \quad a=1,\ b=2,\ c=\mathrm{e}$$

9.
$$f(x,y) = \frac{y+2}{x+3}, \quad a=0,\ b=1,\ c=1$$

10.
$$f(x,y) = \sin^2(y-x), \quad a=0,\ b=1,\ c=0$$

11.
$$f(x,y) = \frac{xy+y}{x}, \quad a=1,\ b=2,\ c=\frac{1}{\mathrm{e}}$$

12.
$$f(x,y) = \frac{1-x^2}{xy}, \quad a=1,\ b=2,\ c=2$$

13.
$$f(x,y) = \mathrm{e}^y - 1 + x, \quad a=0,\ b=1,\ c=-2$$

14.
$$f(x,y) = \frac{x^2+y^2}{xy}, \quad a=1,\ b=2,\ c=2$$

15.
$$f(x,y) = x^3 + \frac{2y}{x}, \quad a=1,\ b=2,\ c=0$$

164 – Numerical solution of ordinary differential equations

16.

$$f(x,y) = \frac{y^2 \ln x - y}{x}, \quad a = 1,\ b = 2,\ c = 1$$

17.

$$f(x,y) = x^2 + 1 + \frac{2xy}{x^2+1}, \quad a = 0,\ b = 1,\ c = 0$$

18.

$$f(x,y) = (x+y)^2, \quad a = 0,\ b = 1,\ c = 0$$

19.

$$f(x,y) = \frac{y}{2\sqrt{x}}, \quad a = 3,\ b = 4,\ c = 3$$

20.

$$f(x,y) = \frac{x^2+y^2}{2xy}, \quad a = 3,\ b = 4,\ c = 1$$

21.

$$f(x,y) = \frac{y}{x^2+1}, \quad a = 1,\ b = 2,\ c = 4$$

22.

$$f(x,y) = 3xy^2 + \frac{y}{x}, \quad a = -2,\ b = -1,\ c = 1$$

Name: Worksheets for Numerical methods

Department, Institute: Faculty of Mechanical Engineering, Department of Mathematics and Descriptive Geometry

Authors: Zuzana Morávková, Jiří Krček

Place, year of publishing: Ostrava, 2021, 1st Edition

Number of Pages: 165

Published: VSB – Technical University of Ostrava

ISBN 978-80-248-4493-0