MTMM.00.005 Numerical Methods
Practical lesson 10: Taylor polynomial

April 16, 2019

To get a positive result, student has to solve all of the exercises given in practical lessons, explain the programming code and answer the questions.

The deadline for the tenth set of exercises is April 23rd.

NB! In order to use the nonstandard packages as *numpy* and *matplotlib* one could use Command Prompt and type

```
H:\nm> "C:\Program Files\Anaconda3\python" <filename>.py
```

Here `H:\nm` is the folder for `<filename>.py`.

Taylor's theorem

Let $f \in C^n[a,b]$ (i.e., f is n times continuously differentiable) and suppose it exists $f^{(n+1)}(x)$ for all $x \in (a,b)$ (i.e., $f^{(n+1)}$ need not to be continuous). Take $x_0 \in [a,b]$. Then for all $x \in [a,b]$ it exists $\xi = \xi(x) \in (x, x_0)$ such that

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \ldots + \frac{f^n(x_0)}{n!}(x-x_0)^n + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}.$$

Here

$$P_n(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \ldots + \frac{f^n(x_0)}{n!}(x-x_0)^n$$

is the n-th order Taylor polynomial at the point x_0. It approximates function f well if x is close to x_0 (for the remainder $R_n(x) = \dfrac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}$ we have $\lim\limits_{x\to x_0} R_n(x) = 0$).

Exercise. Let

$$f(x) = e^x \cos x$$

and $x_0 = 0$. Approximate function f with Taylor polynomials $P_2(x)$, $P_3(x)$ and $P_4(x)$. For that

- plot the functions $f(x)$, $P_2(x)$, $P_3(x)$ ja $P_4(x)$, $x \in [0,1]$, on the same graph;
- find corresponding remainders $R_2(x)$, $R_3(x)$ ja $R_4(x)$;
- estimate the error at the point 0.5 by using the estimates from remainders and compare this by the actual error, i.e., estimate $|R_i(0.5)|$ and compute $|f(0.5) - P_i(0.5)|$, $i = 2,3,4$.

NB! Derivatives must be found analytically. To find and estimate the derivatives one could use, for example, WolframAlpha.