**Numerical methods and Algorithm. Laboratory work.**

1. Try the examples provided in the files:
   1. **Interpolation.m**
   2. **Approxomation.m**
   3. **GradientOptimization.m**
2. Modify example “**Interpolation.m”** in such way, that interpolation curve will be formatted when we have 4 initial points. Provide the graphic and analytical expression of interpolated function.
3. Try it out how “**Approxomation.m**“ works using different amount of base functions. Provide graphics and analytical expression of approximation functions, when maximum degree in interpolating function is: 1, 2, 5.
4. Modify “**Approxomation.m**“ in such way, that approximating base functions be only even degree. Provide graphic and analytical expression approximation function, when three base functions are used for approximation.
5. Try it out how “**GradientOptimization.m**“ works when different amounts of initial data are provided. Determine, when computational became inefficient? Does the parallel computing can be used?
6. Modify “**GradientOptimization.m”** in such way, that steepest gradient descend for optimization will be used. Compare, the results by the means of the value of objective function, computational time, times, how many times gradient is recalculated.

Steepest Gradient Descend (for task number 6):

1. After calculating the gradient vector, it moves in the opposite direction **until the function decreases continuously**;
2. As the **function begins to increase again**, we recalculate the gradient vector and further minimize it in the opposite direction;

**Andrius Kriščiūnas,** [**andrius.krisciunas@ktu.lt**](mailto:andrius.krisciunas@ktu.lt)

Faculty of Informatics | Department of Applied Informatics

Kaunas University of Technology, Kaunas, Lithuania