UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF APPLIED COMPUTER SCIENCE
AND MODELLING

FUNDAMENTALS OF OPTIMIZATION

Constrained multi-variable optimization using non-gradient mehods

1. Aim of the exercise.

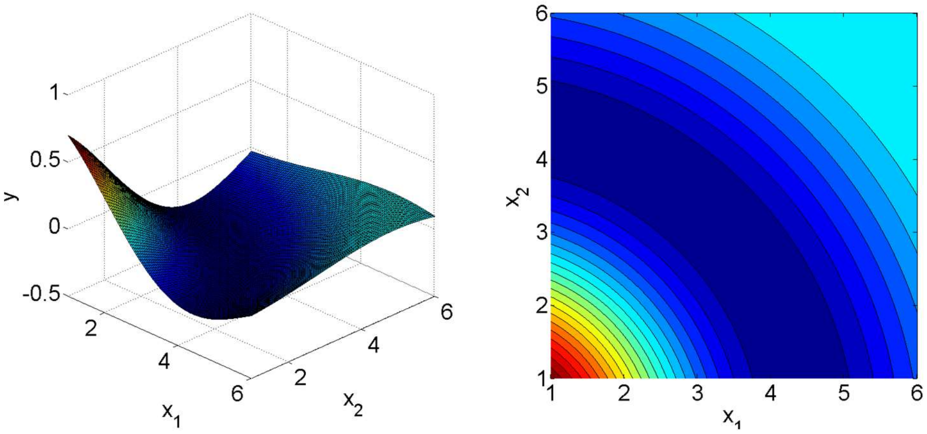
The aim of the exercise is using the non-gradient, multi-variable optimization methods to finding minimum of the given objective function taking into account the set of constraints.

2. Test function.

The test function is given by following equation:

$$f(x_1, x_2) = \frac{sin\left(\pi\sqrt{\left(\frac{x_1}{\pi}\right)^2 + \left(\frac{x_2}{\pi}\right)^2}\right)}{\pi\sqrt{\left(\frac{x_1}{\pi}\right)^2 + \left(\frac{x_2}{\pi}\right)^2}}$$

The test function is shown in the figure below.



The constraints are defined by the equations:

$$g_1(x_1) = -x_1 + 1$$

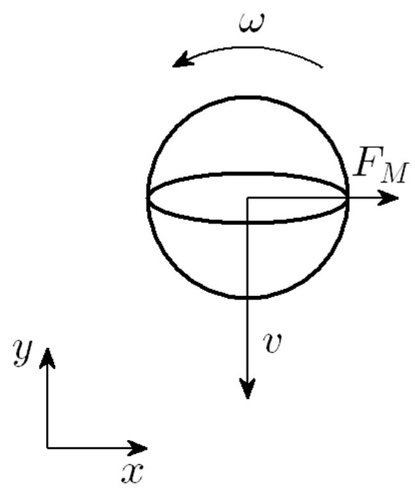
$$g_2(x_2) = -x_2 + 1$$

$$g_3(x_1, x_2) = \sqrt{x_1^2 + x_2^2} - 4$$

3. Ball throw.

The ball with a mass $m = 600g$ and a radius of $r = 12cm$ is thrown from the height of $y_0 = 100m$ (y is a vertical coordinate of the ball position). The initial horizontal coordinate of the ball is equal to 0 ($x_0 = 0$). The initial vertical velocity is equal to 0 ($v_{0y} = 0$) while the initial horizontal velocity v_{0x}

as well as its angular velocity ω is not equal to 0 (the ball is rotating around its axis; axis is perpendicular to the xy plane). Combining the linear movement of the ball and its spin results in Magnus effect. The Magnus effect results in additional force acting on the ball (see figure below). The force direction is equal to cross product of vectors $v_a \times \omega$, where v_a is a vector of air velocity flowing around the moving ball, i.e. $v_a = -v$.



Horizontal and vertical position of the ball can be calculated using the following differential equations:

$$\begin{cases} m\dfrac{d^2x}{dt^2} + D_x + F_{Mx} = 0 \\ m\dfrac{d^2y}{dt^2} + D_y + F_{My} = -mg \end{cases}$$

where: $g = 9.81\, {}^{m}/_{s^2}$ is an acceleration of gravity, D is a drag force, F_M is a Magnus force.

The forces are equal to:

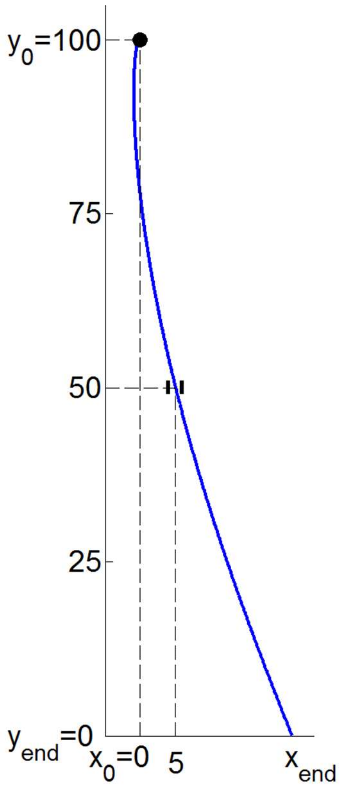
$$D_x = \frac{1}{2}C\rho S v_x^2, \qquad D_y = \frac{1}{2}C\rho S v_y^2$$

$$F_{Mx} = \rho v_y \omega \pi r^3, \qquad F_{My} = \rho v_x \omega \pi r^3$$

where: $C = 0.47$ is a drag coefficient related to the shape of moving object, $\rho = 1.2\, {}^{kg}/_{m^3}$ is an air density, $S = \pi r^2$.

Rotational friction is neglected, i.e. $\omega = const$.

An exemplary ball trajectory is shown in the figure below.



The aim of optimization is finding the values of initial horizontal velocity $v_{0x} \in [-10,10]\ ^{m}/_{s}$ and angular velocity $\omega \in [-20,20]\ ^{rad}/_{s}$ which results in maximal value of x_{end} - the horizontal position of the spot where the ball hits the ground. The additional constraint is that the center of the ball should pass the point $(5,50)$ not further than $1m$, i.e. for y coordinate of the ball center equals to $50m$ ($y = 50m$) the value of its x coordinate must be within the range $[4,6]m$ ($x \in [4,6]m$).

The differential equations describing the ball trajectory should be solved for the time interval $t_0 = 0s$, $t_{end} = 7s$ using the time step $dt = 0.01s$. In addition, it should be taken into account that the time the ball falls depends on the initial horizontal velocity and rotation of the ball (for each combination the ball will be below the ground after $t_{end} = 7s$). Therefore, the simulation should be stopped when y is zero or all values for $y < 0$ should be deleted. Stopping the simulation can be done by defining the event function (see description of the ode45 function in Matlab documentation). The event function can be also used to find the horizontal position of the ball (**x50**) at a height of 50m.

4. Optimization methods.

To perform optimization use **fmincon** function. Starting point should be chosen randomly from given range.

5. Realization of the exercise.

During the exercise six (or seven) m-files should be written:

- **start.m** – a script which runs all computations. It should:
  - display the names of the Authors of the code,

    - display the optimum found for the test function (x, y and the number of objective function calls),
    - plot the figure showing the test function (as a contour plot), starting point and found optimum,
    - display the optimum found for the ball throw problem (v0x, omega, xend, x50 and the number of objective function calls),
    - plot figure showing the ball trajectory.
- **ff_test.m** – a function which calculates and returns the test function value at given point: **y=ff_test(x)**
- **nonlcon_test.m** – a function that takes into account nonlinear constraints for the test problem : **[c, ceq]=nonlcon_test(x)**.
- **ff_ball.m** – a function which calculates and returns the objective function value in the ball throw problem: **-xend=ff_ball(x)**. The minus is necessary because our goal is to maximize **xend** while the **fmincon** function looks for the minimum of the objective function
- **sim_ball.m** – a function which returns the vectors containing samples of time, horizontal and vertical position of the ball and horizontal position at a height of 50m: **[t,x,y,x50]=sim_ball(v0x,omega)**.
- **nonlcon_ball.m** – a function that takes into account nonlinear constraints for the ball throw problem: **[c, ceq]=nonlcon_ball(x)**.
- **eventfcn.m** – function that defines events two events. The first event occurs when the ball's vertical position is 50 meters, the second the ball hits the ground: **[v,ister,dir]=eventfcn(t,y)**. This function is not obligatory.

To validate if the and **sim_ball** function is written correctly perform simulation for $v_{0x} = 5\,{}^{m}/_{s}$ and $\omega = 10\,{}^{rad}/_{s}$. The results should be approximately as presented below.

```
>> [t,x,y,x50]=sim_ball(5,10);
>> [t(end),x(end),y(end),x50]
ans =
        5.9597        41.405 -1.0658e-014         21.59
```

6. Report.

As the report, six (or seven) m-files should be sent via UPeL platform (by one Author only).