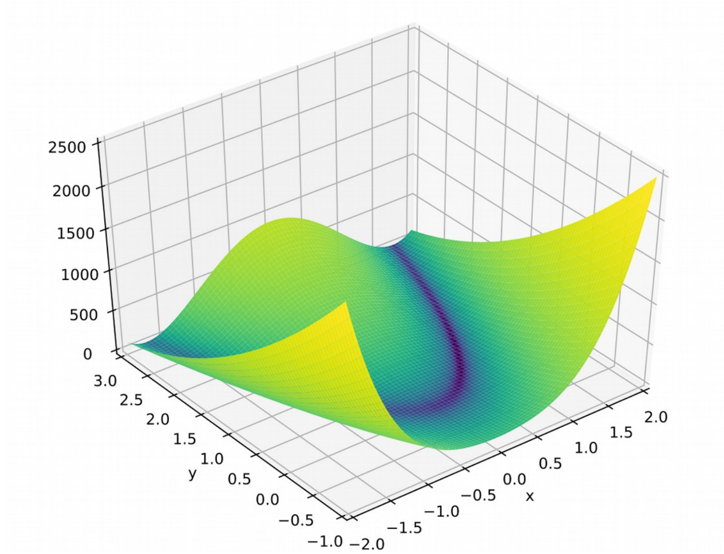
Zadanie 21 – znalezienie minimum funkcji Rosenbrocka z użyciem algorytmu Levenberga-Marquardta



Do rozwiązania tego zadania napisałam program w języku C++.

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

Najpierw wyznaczam 8 losowych punktów z przedziału [-10,10]

Przykładowe wylosowane punkty:
(-0.24614, 0.15635)
(-1.91403, 6.29896)
(9.80216, 1.08490)
(0.18202, -6.43750)
(3.18081, -3.22226)
(-4.61152, 6.03036)
(3.24455, 5.72707)
(0.27732, -2.40142)

Następnie wykorzystując wylosowane punkty algorytm podąża kierunku malejących wartości funkcji Rosenbrocka, aż osiągnie minimum globalne.

Minimum globalne funkcji znajduje się wewnątrz długiego, parabolicznego wgłębienia funkcji: w punkcie (x,y)=(1,1) dla którego funkcja przyjmuje wartość f(x,y)=0

Program realizuje algorytm Levenberga-Marquardta, którego pseudokod wygląda następująco:

Input: A vector function $f : \mathcal{R}^m \rightarrow \mathcal{R}^n$ with $n \geq m$, a measurement vector $\mathbf{x} \in \mathcal{R}^n$ and an initial parameters estimate $\mathbf{p}_0 \in \mathcal{R}^m$.
Output: A vector $\mathbf{p}^+ \in \mathcal{R}^m$ minimizing $||\mathbf{x} - f(\mathbf{p})||^2$.
Algorithm:
$k := 0$; $\nu := 2$; $\mathbf{p} := \mathbf{p}_0$;
$\mathbf{A} := \mathbf{J}^T\mathbf{J}$; $\epsilon_\mathbf{p} := \mathbf{x} - f(\mathbf{p})$; $\mathbf{g} := \mathbf{J}^T\epsilon_\mathbf{p}$;
stop:=($||\mathbf{g}||_\infty \leq \varepsilon_1$); $\mu := \tau * \max_{i=1,\dots,m}(A_{ii})$;
while (not stop) and ($k < k_{max}$)
    $k := k + 1$;
    repeat
        $\boxed{\text{Solve } (\mathbf{A} + \mu\mathbf{I})\delta_\mathbf{p} = \mathbf{g};}$
        if ($||\delta_\mathbf{p}|| \leq \varepsilon_2(||\mathbf{p}|| + \varepsilon_2)$)
            stop:=true;
        else
            $\mathbf{p}_{new} := \mathbf{p} + \delta_\mathbf{p}$;
            $\rho := (||\epsilon_\mathbf{p}||^2 - ||\mathbf{x} - f(\mathbf{p}_{new})||^2)/(\delta_\mathbf{p}^T(\mu\delta_\mathbf{p} + \mathbf{g}))$;
            if $\rho > 0$
                stop:=($||\epsilon_\mathbf{p}|| - ||\mathbf{x} - f(\mathbf{p}_{new})|| < \varepsilon_4||\epsilon_\mathbf{p}||$);
                $\mathbf{p} = \mathbf{p}_{new}$;
                $\mathbf{A} := \mathbf{J}^T\mathbf{J}$; $\epsilon_\mathbf{p} := \mathbf{x} - f(\mathbf{p})$; $\mathbf{g} := \mathbf{J}^T\epsilon_\mathbf{p}$;
                stop:=(stop) or ($||\mathbf{g}||_\infty \leq \varepsilon_1$);
                $\mu := \mu * \max(\frac{1}{3}, 1 - (2\rho - 1)^3)$; $\nu := 2$;
            else
                $\mu := \mu * \nu$; $\nu := 2 * \nu$;
            endif
        endif
    until ($\rho > 0$) or (stop)
    stop:=($||\epsilon_\mathbf{p}|| \leq \varepsilon_3$);
endwhile
$\mathbf{p}^+ := \mathbf{p}$;

Na wykresach narysowanych w Gnuplot można zobaczyć, jak dla kolejnych trzech wywołań programu program zbiża się do minimum globalnego, w zależności od początkowo wylosowanych liczb losowych.