Exercises for Week 9

1 Powell's problem

To solve nonlinear least squares data fitting problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2, \tag{1}$$

we had introduced the Gauss-Newton method and the Levenberg-Marquardt method. In fact, both methods also can be used for solving nonlinear system of equations, i.e.,

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{0}, \qquad \text{where } \boldsymbol{r} : \mathbb{R}^n \to \mathbb{R}^n. \tag{2}$$

The main reason is that a solution to (2) is a global minimizer of the function f given in (1).
In this exercise, we will apply several different methods to solve Powell's problem:

$$\boldsymbol{r}(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ \frac{10x_1}{x_1+0.1} + 2x_2^2 \end{bmatrix} = \boldsymbol{0}. \tag{3}$$

1. Verifty the unique root of (3) is $\boldsymbol{x}^* = [0, 0]^T$.
2. Calculate the Jacobian J of $\boldsymbol{r}$, and show that at $\boldsymbol{x}^*$ J is singular.
3. Implement a Matlab function to calculate $\boldsymbol{r}$ and J with a given $\boldsymbol{x}$. You can start your function with

```
function [r,J]=fun_rJ_exe(x)
```

4. **Newton's method.** In the course "02601 introduction to numerical algorithms" in Block 4, we had introduce the Newton's method to solve a nonlinear system of equations like (3). The Newton iteration step is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - J(\boldsymbol{x}_k)^{-1}\boldsymbol{r}(\boldsymbol{x}_k). \tag{4}$$

   (a) If the Jacobian J is not a square matrix, then can we still apply Newton's method?

   (b) Revise your implementation of Newton's method for solving a minimizatoin problem $\min_{\boldsymbol{x}} f(\boldsymbol{x})$, `newton`, to solve the nonlinear system (3) with the newton iteration step given in (4). Set the stopping criteria as

$$\|\boldsymbol{r}_k\|_\infty < 10^{-10} \quad \text{or} \quad k \geq 100n.$$

   Note that for solving nonlinear systems you only need the Jacobian and no need for the Hessian.

(c) Set the starting point $\boldsymbol{x}_0 = [3, 1]^T$, and apply Newton's method to solve (3). Plot $\boldsymbol{e}_k = \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2$ and $\frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x}_k)\|_2^2$ as functions of the iteration number. Which convergence rate can you see? If the method did not converge quadratically, what can be the reason?

(d) To further observe the convergence rate, plot $(x_1)_k$ and $(x_2)_{k+1}/(x_2)_k$ for $k = 10, \cdots$. How did they change?

5. **Gauss-Newton method.** Apply the Gauss-Newton method to solve the corresponding minimization problem (1). The Gauss-Newton iteration step with the step length 1 is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - (J(\boldsymbol{x}_k)^T J(\boldsymbol{x}_k))^{-1} J(\boldsymbol{x}_k)^T \boldsymbol{r}(\boldsymbol{x}_k).$$

(a) Show that if $J(\boldsymbol{x}_k)$ is a square matrix and nonsingular, the Gauss-Newton iteration step is identical to the Newton step.

(b) If the Jacobian J is not a square matrix, can we still apply the Gauss-Newton method?

(c) To see this equivalence numerically, call the Matlab function `GaussNewton_line.m`, set the same starting point, turn off the line search, and set the maximum iteration number as the same as in Newton's method. Plot $\boldsymbol{e}_k$ and $f(\boldsymbol{x}_k)$ as functions of the iteration number. Do you get the same plot as in Newton?

6. **Levenberg-Marquardt method.** Apply the Levenberg-Marquardt method to solve the minimization problem (1). The L-M iteration step is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - (J(\boldsymbol{x}_k)^T J(\boldsymbol{x}_k) + \lambda_k I)^{-1} J(\boldsymbol{x}_k)^T \boldsymbol{r}(\boldsymbol{x}_k).$$

(a) Call your implementation of the L-M method from Week 7, and set the same starting point. Plot $\boldsymbol{e}_k$ and $f(\boldsymbol{x}_k)$ as functions of the iteration number. To see the convergence rate, plot $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_2 / \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2$. Does the method converge linearly?

(b) Download the Matlab function `Levenberg_Marquardt_yq.m`, where I used another updating strategy for λ. In this function, the initial value $\lambda_0 = \tau\|J(\boldsymbol{z}_0)^T J(\boldsymbol{z}_0)\|_2$ and τ is an input. Set $\tau = 1$. Use this Matlab function to solve (1), and plot $\boldsymbol{e}_k$ and $f(\boldsymbol{x}_k)$ as functions of the iteration number. Which convergence rate do you get now? Comparing with the previous result, which one is better?

(c) In the figures from `Levenberg_Marquardt_yq.m`, you should see that the iteration process seems to stall between the step 20 and 30. What can be the reason?

7. **Change of variables.** In Powell's problem (2) the variable x_2 occurs only as x_2^2, so we can change the variables to $\boldsymbol{z} = [x_1, x_2^2]^T$. Then, the problem takes the form: Find $\boldsymbol{z}^* \in \mathbb{R}^2$ such that $\boldsymbol{r}(\boldsymbol{z}^*) = \boldsymbol{0}$, where

$$\boldsymbol{r}(\boldsymbol{z}) = \begin{bmatrix} z_1 \\ \frac{10 z_1}{z_1 + 0.1} + 2 z_2 \end{bmatrix}.$$

(a) Calculate the new Jacobian J of $\boldsymbol{r}$, and show that J is nonsingular for all $\boldsymbol{z}$.

(b) Apply the Levenberg-Marquardt method by using `Levenberg_Marquardt_yq.m` to solve the new minimization problem

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{z})\|_2^2.$$

We set the same starting point $\boldsymbol{z} = [3,1]^T$. Since J is nonsingular for all $\boldsymbol{z}$, we should set λ very small. We set its initial value as $\lambda_0 = 10^{-16}\|J(\boldsymbol{z}_0)^T J(\boldsymbol{z}_0)\|_2$. Plot $\boldsymbol{e}_k$ and $f(\boldsymbol{x}_k)$ as functions of the iteration number. How many iterations did you need?