Final Report Questions

Hand in on DTU Inside before 14 December 10pm. **The overall page limit exclude Appendix is 6.**

1 Exercise for one-page report (40%)

In this exercise, we apply several methods that we learned in the course to solve the following minimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} f(\boldsymbol{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5.$$

Please first complete the following questions, then write your one-page report.

1. (by hand) Compute the gradient $\nabla f(\boldsymbol{x})$ and the Hessian $\nabla^2 f(\boldsymbol{x})$.
2. (by hand) Show that $\boldsymbol{x}^* = [1,1]^T$ is the only local minimizer of this function.
3. (in Matlab) Make a contour plot of $f(\boldsymbol{x})$ in the interval $-5 \leq x_1, x_2 \leq 5$.
4. (in Matlab) Write a Matlab function to return the function value $f(\boldsymbol{x})$, its gradient $\nabla f(\boldsymbol{x})$ and the Hessian $\nabla^2 f(\boldsymbol{x})$ with a given $\boldsymbol{x}$.
5. (in Matlab) Apply the steepest descent method (with line search), Newton's method (with fixed step length 1), BFGS method (with $H_0 = I$ and line search), coordinate search method (with $\gamma = 1$) and nonlinear conjugate gradient method to solve this minimization problem.
   - Set the starting point as $\boldsymbol{x}_0 = [1,4]^T$.
   - Set the stopping criteria as $\|\nabla f\|_\infty \leq 10^{-6}$ or $k \geq 200$, where k is the iteration index.
   - For line search, we use the backtracking line search with $\rho = 0.5$ and $c = 0.1$.
   - Save and output the first 8 iterates, i.e., $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_8$ and plot them on the contour plot for each method.
   - Use `semilogy` to plot $e_k = \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2$, $f(\boldsymbol{x}_k)$ and $\|\nabla f(\boldsymbol{x}_k)\|_2$ (if applicable) for each method.
   - For nonlinear conjugate gradient method, you can implement it easily by modifying your code for the steepest descent method with backtracking line search. You only need change the part for calculating the search direction $\boldsymbol{p}_k$. In nonlinear conjugate gradient method, we have

     $$\boldsymbol{p}_0 = -\nabla f(\boldsymbol{x}_0), \qquad \boldsymbol{p}_{k+1} = -\nabla f(\boldsymbol{x}_{k+1}) + \beta_{k+1}\boldsymbol{p}_k,$$

     where we use Polak-Ribière method to obtain β_{k+1}, i.e,

     $$\beta_{k+1} = \frac{\nabla f(\boldsymbol{x}_{k+1})^T(\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k))}{\|\nabla f(\boldsymbol{x}_k)\|_2^2}.$$

6. **(Report)** In your report, you should include the following results, comments and explanations.

   (a) (10%) Include the recorded values of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_8$ from all methods in a table.

   (b) (20%) According to the table and the contour plots, which do not need be included in the report, comment on and explain the progress of all 5 methods.

   (c) (10%) Test Newton's method with different starting points: $\boldsymbol{x}_0 = [-3, -2]^T$ and $[0, 0.5]^T$. How did the results change? Why?

2 Convergence of CG method (in Matlab, 20%)

In this exercise, we consider a 500×500 symmetric matrix A constructed as follows. The elements in the main diagonal of A equal 1, and the off-diagonal elements are random numbers from the uniform distribution on $[-1, 1]$. Then, we replace all elements in the off-diagonals with $|a_{ij}| > \tau$ by zero, where τ is a parameter. For τ close to zero, A is close to the identity matrix, which is well-conditioned with the condition number 1.

1. (5%) Download the Matlab function `genA` from the same folder as the exercise list in DTU Inside, which generates the matrix A. Set the parameter τ as 0.01, 0.05, 0.09, and 0.3, and compute the condition number and the smallest eigenvalue of A. As τ increase, how does the condition number change? At $\tau = 0.3$, is A still positive definite? (Hint: the smallest eigenvalue of A can be calculated by calling `eigs(A,1,'sm')`).

2. (10%) Apply the CG method to solve the linear system $A\boldsymbol{x} = \boldsymbol{b}$ with $\boldsymbol{b} = [1, 1, \cdots, 1]^T$ and the starting point $\boldsymbol{x}_0 = [0, 0, \cdots, 0]^T$. Plot the four curves $\log_{10}(\|A\boldsymbol{x}_k - \boldsymbol{b}\|_2)$ corresponding to the CG iteration with $\tau = 0.01, 0.05, 0.09, 0.3$, respectively, in the same figure. How do the convergence curves look like?

3. (5%) At $\tau = 0.3$, you will observe that the CG method does not converge. Why?

3 Necessary condition (20%)

In this exercise, we use one example to illustrate that the KKT conditions are only necessary not sufficient. Consider the following equality constrained problem

$$\min_{\boldsymbol{x}} \frac{1}{2}((x_1 - 1)^2 + x_2^2) \quad \text{s.t.} \quad -x_1 + \beta x_2^2 = 0, \tag{1}$$

where β is a parameter.

1. (by hand, 10%) Write done the KKT conditions for (1), and verify that $\boldsymbol{x}^* = [0, 0]^T$ and $\lambda^* = 1$ satisfy the conditions.

2. (in Matlab, 5%) On one figure, plot the feasible set of the problem (1), i.e. the equality constraint $-x_1 + \beta x_2^2 = 0$, with $\beta = 0, 0.25$ and 1, respectively. Then, on the same figure draw a contour plot of the objective function.

3. (5%) From your figure, you should see that for $\beta = 0$ and 0.25 $\boldsymbol{x}^* = \mathbf{0}$ is a unique local minimizer. In fact, it is also a global minimizer, since the objective function is strictly convex. But for $\beta = 1$ is $\boldsymbol{x}^*$ still a minimizer? If not, is it a maximizer? Where are the minimizers approximately located? (Only need give an approximation from the figure, and no calculations are needed here.)

   In fact, in order to obtain the sufficient conditions for minimizers or maximizers, we would need the 2nd order condition, similar as for unconstrained problems.

4 Review: A regularized least-squares problem (20%)

We consider a regularized least squares problem with smoothing:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{i=1}^{k}(\boldsymbol{a}_i^T\boldsymbol{x} - b_i)^2 + \delta\sum_{i=1}^{n-1}(x_{i+1} - x_i)^2,$$

where $\delta > 0$ is a pre-given parameter, and $\boldsymbol{a}_i \in \mathbb{R}^n$ for all $i = 1, \cdots, k$.

1. (by hand, 5%) Verify that the objective function in matrix-vector form is

   $$\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \delta\|L\boldsymbol{x}\|_2^2.$$

   What are the matrices A and L? What are their dimensions?

2. (by hand, 5%) According to the definition of 2-norm for vectors, the objective function can be further written as

   $$\left\| \begin{bmatrix} A \\ \sqrt{\delta}L \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2.$$

   So you can see that this minimization problem is a least-squares problem. According to the optimality condition, express its normal equation.

3. (in Matlab) Download the data file `vec_b.mat` and the Matlab function `get_AL.p` from the same folder as the exercise list. By loading `vec_b.mat`, you will obtain the vector $\boldsymbol{b}$. The p-file is a compiled function where you cannot see the codes. You can obtain the matrix A and L by using the commend

   ```
   [A, L] = get_AL(n);
   ```

   with an input integer n. Here n should be the length of the vector $\boldsymbol{b}$. You can illustrate $\boldsymbol{b}$ by running

   ```
   figure, imagesc(reshape(b,sqrt(n),sqrt(n))),
   colormap(gray),
   ```

You should see that $\boldsymbol{b}$ is in fact a blurry image.

4. (in Matlab, 5%) Now you have the matrices A and L together with the vector $\boldsymbol{b}$. Use the backslash from Matlab to solve your normal equation with $\delta = 0.005$, then show your solution $\boldsymbol{x}$ as an image. (You can call the similar commands as in the previous question and change $\boldsymbol{b}$ into $\boldsymbol{x}$.) Include this result in your answer file. You should see that in fact you are applying a simple deblurring method. :D

   Note that in this exercise we use the backslash to solve the normal equation, and do not use QR factorization to solve the least-squares problem directly. The reason is that your code for QR cannot handle large-scale sparse system efficiently.

5. (in Matlab, 5%) Play with the choices of the parameter δ by testing much larger and much smaller values than 0.005. What is your observation on how δ influences the result?

   (Optional, Bonus 5%) Can you try to explain your observation according to the minimization problem?