Graded Homework 2 Numerical Optimization Fall 2023

The solutions are due by 11:59PM on Sunday October 22, 2023

Written Exercises

Problem 2.1

Consider solving a linear system $Ax = b$ where A is symmetric positive definite using steepest descent.

2.1.a

Suppose you use steepest descent without preconditioning. Show that the residuals, r_k and r_{k+1} are orthogonal for all k.

2.1.b

Suppose you use steepest descent with preconditioning. Are the residuals, r_k and r_{k+1} orthogonal for all k? If not is there any vector from step k that is guaranteed to be orthogonal to r_{k+1}?

Problem 2.2

Let $A = Q\Lambda Q^T$ be a symmetric positive definite matrix where Q is an orthogonal matrix and Λ is a diagonal matrix whose diagonal elements are positive and also are the eigenvalues of A. Define

$$\tilde{x} = Q^T x \quad \text{and} \quad \tilde{b} = Q^T b$$
$$Ax = b \quad \text{and} \quad \Lambda\tilde{x} = \tilde{b}$$

Given x_0 and $\tilde{x}_0$, define the sequence x_k as the sequence of vectors produced by steepest descent applied to $Ax = b$ and the sequence $\tilde{x}_k$ as the sequence of vectors produced by steepest descent applied to $\Lambda\tilde{x} = \tilde{b}$.

Let $e_k = x_k - x$ and $\tilde{e}_k = \tilde{x}_k - \tilde{x}$. Show that if $\tilde{x}_0 = Q^T x_0$ then

$$\|e_k\|_2 = \|\tilde{e}_k\|_2, \quad k > 0$$

Problem 2.3

2.3.a

Let the cost function $f : \mathbb{R}^2 \to \mathbb{R}$ be defined by

$$f(x) = x^T d + x^T x, \quad \text{where} \quad d = \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix}$$

where $\delta_1 > 0$, $\delta_2 > 0$ and $\mu = \|d\|_2 > 1$. Consider the problem

$$\min_{x \in \mathbb{R}^2} f(x).$$

(i) Find a minimizer x^*. Is it unique?

(ii) Write the iteration that defines applying the steepest descent algorithm to solve the minimization problem.

(iii) How would you set the stepsize α_k and why?

(iv) Will your choice of α_k yield an algorithm that converges in a finite number of steps?

2.3.b

Now suppose the minimization problem is constrained so that we are only interested in $x \in \mathbb{R}^2$ on the circle of radius 1, i.e., the unit circle

$$\mathcal{S}_1 = \{x \in \mathbb{R}^2 \mid x^T x = 1\}$$

Specifically, we want to solve

$$\min_{x \in \mathcal{S}_1} f(x)$$

(i) Show that this problem can be viewed as an unconstrained minimization problem on $\mathbb{R}$ by writing the cost function over $\mathcal{S}_1$ as a function of a real variable θ.

(ii) Write the iteration that defines applying the steepest descent algorithm to solve the minimization problem over $\mathbb{R}$.

(iii) How would you set the stepsize α_k and why?

(iv) Will your choice of α_k yield an algorithm that converges in a finite number of steps when started at an initial guess $\theta_0 = 0$?

Problem 2.4

For each of the two cost functions on $\mathbb{R}^2$, i.e., $x^T = (\xi_1, \quad \xi_2)$, determine the gradient, the Hessian and the stationary points. Identify which stationary points are minima, maxima or neither.

(2.4.a) $f(x) = 8\xi_1 + 12\xi_2 + \xi_1^2 - 2\xi_2^2$

(2.4.b) $f(x) = 100(\xi_2 - \xi_1^2)^2 + (1 - \xi_1)^2$

Problem 2.5

Let $f(x) : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Let the set Γ be the set of global minimizers of f. Show that Γ must be a convex set.

Programming Exercise

Problem 2.6

2.6.a Coding Task

Implement a general descent method that can:

- choose the direction vector as a general descent direction or more specifically, as the residual at the current iterate x_k, or as a direction that is intentionally taken to be different from the residual;
- choose the optimal local stepsize, α_k^*, for the given direction vector, or select $\alpha_k = \tilde{\alpha}$ as a constant satisfying a convergence sufficient condition, or select α_k based on other specified criteria given below.

Your code should be organized so that it can be used as the basis for a general descent optimization code for unconstrained nonlinear optimization. For this assignment, it should be capable of running

1. steepest descent (SD);
2. Richardson's stationary method (RS) satisfying $\alpha < 2/\lambda_{max}$;
3. a descent method (SDslow) with the direction vector taken as the residual but with the stepsize $\alpha_k = \sigma\alpha_k^*$ where σ is constant and such that convergence is still guaranteed.
4. Conjugate Gradient without preconditioning;

5. The Barzilai and Borwein two-step methods described in the paper posted on the class webpage – the two methods differ only in the two choices of stepsize used; (It is strongly recommended you read the paper carefully before doing this assignment.)

6. and the Gauss-Southwell method (described below).

2.6.b Empirical Tasks

- Use the general descent code with the appropriate choices to compare all of the methods when applied to strictly convex and convex quadratic unconstrained optimization problems.
- In general, you are to highlight the strengths and weaknesses of the methods and their relative performances.
- A more specific goal is to empirically verify convergence rate bounds, orthogonality of successive residuals for SD, all residuals for CG, A-orthogonality for CG, and the effect of the condition number and distribution of the eigenvalues on the performance of the methods.
- Most of your experiments can be run in the eigencoordinate system but you should provide some evidence that the convergence behavior in the original coordinate system are the same when they should be.
- You should also consider local behavior of the residual and error, i.e., what happens on each step and relate it to the the size of the components of the error or residual in each eigendirection. Pay particular attention to the difference between SD and RS single step differences, e.g., you can save iterates from, say, RS and then examine what happens when you take one step of, say, SD staring from the saved RS iterates.
- The RS method requires that the constant stepsize α satisfy $\alpha < 2/\lambda_{max}$ so that the iteration will converge for all x_0. However, violating this condition **does not mean that the iteration diverges for all x_0.** Create an example where, in exact arithmetic, the iteration will converge even though $\alpha > 2/\lambda_{max}$ and explain the reasoning behind your construction. Does numerical noise due to arithmetic, and perhaps simulated here by random perturbations to the iterates or associated vectors, effect the conclusion?

2.6.c Gauss-Southwell Method

The Gauss-Southwell method is a coordinate descent method where the direction vector is $d_k = sign(e_i^T r_k)e_i$ where e_i is the $i-$th standard basis vector and i is the index of a component of r_k that has maximal magnitude, i.e.,

$$|r_k^T e_i| \geq |r_k^T e_j|, \quad 1 \leq j \leq n$$

where α_k minimizes $f(x_k + \alpha d_k)$ as a function of the scalar α. This implies

$$(r_k^T e_i)^2 \geq (r_k^T e_j)^2, \quad 1 \leq j \leq n.$$

Given the constraints

$$d_k^T g_k < 0$$

$$(d_k^T g_k)^2 \geq \beta (d_k^T d_k)(g_k^T A^{-1} g_k)$$

where $g_k = \nabla f(x_k) = -r_k$ and $0 < \beta \leq 1$, it can be shown that

$$E(x_{k+1}) \leq \left\{ 1 - \frac{1}{n-1} \frac{\lambda_{min}}{\lambda_{max}} \right\} E(x_k)$$

where

$$f(x_k) = E(x_k) = 0.5 \|x_k - x_*\|_A^2.$$

Note that this is an example of a more general constraint on the angle between the direction vector and the residual being used to choose the direction vector on a give step.

2.6.d Comments on Test Problems

As noted, the behavior of the algorithms can be demonstrated with respect to convergence, rates etc. in the eigencoordinate system (see the study questions and class notes). So you can work with the system $\Lambda \tilde{x} = \tilde{b}$ and manipulate the choice of the positive diagonal matrix Λ for strictly convex problems and nonnegative diagonal matrix Λ for convex problem that may have more than one minimizer. You should, as with the first graded homework, use a combination of a few specific problems to demonstrate results but you should also work with large numbers of randomly generated problems from various classes depending on the assumptions you are exploring.

Of course, given a diagonal matrix Λ it is easy to generate a symmetric postive definite A or positive semidefinite A by generating an orthogonal matrix $Q \in \mathbb{R}^{n \times n}$ as discussed in first graded homework assignment. For such a matrix you can verify the invariance of the behavior for the algorithm in the eigencoordinates and in the coordinates associated with A. You can also generate symmetric positive definite matrics for which you do not initially specify the eigendecomposition by generating a random lower triangular matrix $L \in \mathbb{R}^{n \times n}$ with positive diagonal elements then forming $A = LL^T$. This allows you to easily determine the true solution for any righthand side vector b (as you can also with the eigendecomposition). You can use any library, e.g., Matlab, to compute the eigendecompostion for $A = LL^T$ for your analysis.