CIE 6010/MDS 6118 Midterm Exam: Part 2

Problem Solving in Matlab: CIE 25 points / MDS 35 points

Take-home. Due 5:00pm, November 9, 2018

— Room 511, Dao Yuan Building —

No discussions are allowed except possibly with the instructor or TAs. Please include this page as the cover page for your submission.

Name (print): ____________________________

Student Number: ____________________________

Course Number: ____________________________

Your Matlab function should have the following interface:

```
function [x,hist,iter] = myphase(A,B,d,x0,tol,maxit,itGN)
%
% Solve the phase retrieval least squares problem
%        min f(x) = 0.5*norm(|(A + iB)x|^2 - d)^2
% input:
%      A,B,d = real-valued data
%         x0 = initial guess
%        tol = tolerance
%      maxit = maximum number of iterations in total
%       itGN = min number of Gauss-Newton iterations before
%              possibly switching to Newton iterations
% output:
%          x = computed solution
%       hist = history vector for c_k = norm(g_k)/(1+f_k)
%       iter = total number of iterations taken
```

Given a complex matrix $A+iB \in \mathbb{C}^{m\times n}$ ($m > n$) with real part $A \in \mathbb{R}^{m\times n}$ and imaginary part $B \in \mathbb{R}^{m\times n}$, for a unknown "signal" $\hat{x} \in \mathbb{R}^n$, one can observe noisy data $d \approx |(A+iB)\hat{x}|^2 \in \mathbb{R}^m$, which is the squared magnitude of the complex vector, taken component-wise.

Our task is to recover a nonzero, unknown signal $\hat{x}$ from a noisy observation $d > 0$ by solving the nonlinear least squares problem:

$$\min_{x\in\mathbb{R}^n} f(x) \triangleq \frac{1}{2}\|r(x)\|^2, \tag{1}$$

$$r(x) = |(A+iB)x|^2 - d, \tag{2}$$

i.e., $r_i(x) = (a_i^T x)^2 + (b_i^T x)^2 - d_i$ where a_i^T and b_i^T are the i-th rows of A and B, respectively. This problem is often called a phase retrieval problem.

Write a Matlab function to solve the above phase retrieval (nonlinear least squares) problem to obtain a solution $x \approx \hat{x}$ from noisy observation d. See the first page for the specified interface for your Matlab function.

The suggested algorithmic approach is to use a combination of Gauss-Newton and Newton methods with the unit step size (no line search). At the beginning, use Gauss-Newton iterations, then at some iteration of your choice (after a minimum number of GS iterations), switch to Newton iterations. The termination criterion should be:

$$c_k = \frac{\|\nabla f(x^k)\|}{1+f(x^k)} \le tol.$$

Your Matlab implementation should be sufficiently efficient. A useful Matlab function for this project may be `bsxfun` (although other functions exist to attain a similar efficiency). As usual, you should not directly implement mathematical formulas as derived without a careful scrutiny on their efficiency.

Please follow the instructions below:

- Download the instructor's code `yzphase.p`, and the test script `test_phase.m`, which can be run with or without your function. Use the default n value for debugging purposes.
- For submission, run the test script with $n = 1000$, and submit the screen output (possibly truncated) and all plots.
- Submit your codes and a summary (typed, no more than one page) to document and explain your decisions and observations in reasonable details. Comment on issues of convergence and efficiency.