

Homework 6

Please answer the following questions in complete sentences in a typed manuscript and submit the solution to me in class on February 28th, 2012.

Problem 0: List your collaborators.

Please identify anyone, whether or not they are in the class, with whom you discussed your homework. This problem is worth 1 point, but on a multiplicative scale. (Note that collaboration is not allowed on the bonus question below.)

Problem 1: Implement a Quasi-Newton method

1. Nocedal and Wright state the BFGS update to the *inverse Hessian* as:

$$\mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}^T) \mathbf{H}_k (\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T.$$

Describe the amount of work required for this update.

2. Create a new function called `quasi_newton_bfgs.m` based on `newtons_method_1.m` and implement a Quasi-Newton method with the BFGS update to the *inverse Hessian*, use the Strong Wolfe line search routine from the Poblano toolbox. Make sure your function has an appropriate help description (the comments at the top of the file) and a reasonable set of parameters. *There is a good chance you'll use this code later in the course.* This function should record:

- number of function evaluations in iteration `i`
- the time taken from the start until the end of iteration `i`
- the value of the function at iteration `i`
- the infinity norm of the gradient at iteration `i`
- the size of the step at iteration `i`

1. Modify the `gradient_descent_1.m` code to use the same line search routine.

2. I've provided a list of problems in `matlab/testfuncs.zip`. Look at the file `driver_eval_methods.m`. It shows how to loop over all the test cases. For test case `ii`, `function_min_values(ii)` is the exact minimizer, and the code creates `f` and `g` to evaluate the function and gradient at a point; also `x0` is the starting value. Use `matlab/fcombine.m` to join `f` and `g` into a single function for use in your routines above.

Show how many function evaluations, iterations, and seconds is required to find a point where the infinity norm of the gradient is smaller than 10^{-5} for each problem.

Problem 2: A little theory

According to the book, the SR1 BFGS update is a special case of the Broyden class of quasi-Newton updates (pages 150, 151). Show that this is the case.

Problem 3: Comparing optimization codes.

BONUS We are getting close to the midterm. This is a bonus question worth 15 points of extra-credit.

- **You may not collaborate on this question.**
- **You must complete all parts of this question.**

We've discussed optimization algorithms frequently in this class. So far, we have seen:

- Steepest Descent with a line-search
 - Newton's Method with a line-search
 - Hessian based Trust Region methods
 - Non-linear CG methods with a line search
 - DFP Quasi-Newton method with a line search
 - BFGS Quasi-Newton method with a line search
 - SR1 Quasi-Newton method with a line search
- Given two methods, how should we compare them?

1. Please fill in the blank in the following question, keeping in mind the requirements on this bonus question.

"I –fill this in– this problem entirely by myself and did not discuss with anyone."

2. Read the following paper http://www.tops-scidac.org/papers/DM_benchmark.pdf

3. Implement a function *yourself* to compute and display a performance profile. **I'll post the best solution to the class web-page as a reference.**

4. Show a performance profile for number of function evaluations for the steepest descent and quasi-newton methods computed in problem 1.