Numerical Optimization

Homework 4

Due 02.07.2014

Give your answers with logical and/or mathematical explanations. Hand-in your homework in the beginning of a lecture on due date. Late submissions will not be accepted. Assigned points are shown in square brackets, which will be re-scaled so that the total homeworks points will be 40.

1.[5] Show that if $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex in $\mathbb{R}^n$, then the curvature condition $s_k^T y_k > 0$, with $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, holds for any x_{k+1} and x_k in $\mathbb{R}^n$.

2.[5] Show that the second strong Wolfe condition (with $0 < c_2 < 1$) with a descent direction p_k,

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f(x_k)^T p_k|$$

implies the curvature condition $s_k^T y_k > 0$.

3.[5] Show that n nonzero conjugate vectors $\{p_0, p_1, \ldots, p_{n-1}\}$, $p_i \in \mathbb{R}^n$, with respect to a symmetric positive definite matrix A are linearly independent, i.e., if $\sum_{i=0}^{n-1} c_i p_i = 0$ for any $c = (c_0, c_1, \ldots, c_{n-1})^T$, then c must be a zero vector.

4.[10] Implement the conjugate gradient algorithm (Algorithm 2 in Lecture 13), and use it to find Newton directions instead of using matrix inversion in your `R` code from homework 3 (minimization using Newton directions and backtracking linesearch). All previous parameters remain the same.

In this homework you use a high dimensional objective function provided as `highdim.R` on the cource website instead of the Rosenbrock function. Before sourcing this file, you need to define variables $n = 500$ (dimension), and $mode = 1$ (Hessian with distinct eigenvalues) or $mode = 2$ (grouped eigenvalues).

- Start minimization from the zero vector.
- CG tolerance: $1e - 20$.
- Print messages from iterations as in hw 3, but adding the number of iterations taken in your CG routine (0 if CG is not used).
- Use the infinity norm (`norminf`) for checking any covergence (including optimality of solutions, the residual of CG, etc) replacing the two-norm function `norm2`.
- Measure the total elapsed time taken by your minimization code (you can use `proc.time()` `R` function).
- Print and submit iteration messages and time measurements for all of the following cases:

  - Newton's method with matrix inversion for mode=1,
  - Newton's method with matrix inversion for mode=2,
  - Newton's method with CG for mode=1,
  - Newton's method with CG for mode=2.

Also, create and submit two plots:

- n in $seq(100, 1000, length.out = 5)$ on x-axis vs. computation time of all four cases on y-axis.
- n in $seq(100, 1000, length.out = 5)$ on x-axis vs. average CG iteration count per minimization step on y-axis.

Properly decorate the plots (xlab, ylab, legend, etc.) so that they will be understandable.

Does CG behave as you expect? From your results, can you guess what kind of function is in `highdim.R` without seeing the code? Explain.

- Correct implementation: 5pts (print & submit your CG code too).
- Plots: 3pts
- Discussion: 2pts