CS726, Spring 2018

Homework 7 (due Wednesday 4/25/18 at 6:00pm)

Hand in hard copies of your code and results, and answers to the questions. Submit the files `BFGS.m` and `LBFGS.m`, as well as typeset or neatly written responses to the questions.

1. Complete the proof of Theorem 3.7 in the book by showing the converse of the results proved in the book. That is: under the conditions of the theorem, if $\{x_k\}$ converges superlinearly to x^*, then
$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|} = 0.$$

2. Question 6.7 from the text.

3. Prove the statements 2 and 3 on p. 145 of the text.

4. (a) Use the BFGS method to solve a nonlinear least squares problem in which the objective is defined by
$$f(x) = \frac{1}{2}\sum_{i=1}^{15} r_i^2(x),$$
where $x \in \mathbb{R}^3$. The function f and its gradient are calculated by the routine `nls_resida.m`, available on the web site, which has the usual calling sequence for function and gradient evaluation routines. The starting point is specified in `testqn.m`. Note that `testqn.m` also prints the number of function and gradient evaluations (`numf` and `numg`, respectively). These are already tracked in the function evaluation files, so you don't need to increment them.

   Store the approximation H_k to the inverse Hessian. Use formula (6.20) from the text to reset H_0, after the first step has been taken but before the update to H_1 is performed.

   Your calling sequence should be

   ```
   function [inform, x] = BFGS(fun, x, qnparams)
   ```

   where `qnparams = struct('toler', 1.0e-6, 'maxit', 1000)` and x is a struct with the fields `x.p` and `x.g`, as in previous homeworks.

   Use the stopping criterion
$$\|\nabla f(x)\|_2 \leq \texttt{qnparams.toler}(1 + |f(x)|).$$

   You should use your line search routine `EBLS.m` with parameter settings $c_1 = 10^{-4}$, $c_2 = 0.4$, `maxit=100`, `alpha_start=1`.

   (b) Repeat this process with the function $f(x) = \sum_{i=1}^{3} r_i(x)$, where $x \in R^2$ and the residuals are defined by
$$r_i(x) = a + Hx + 25\left(x - \begin{bmatrix}1\\1\end{bmatrix}\right)^T B \left(x - \begin{bmatrix}1\\1\end{bmatrix}\right) d,$$
where a, H, d, and B are define in the evaluation routine `nls_residb.m` for the function and its gradient is, available on the web site. The starting point is specified in `testqn.m`.

(c) Use your BFGS code to minimize with the function `xpowsing` from Homework 6. The starting point and dimension n are specified in `testqn.m`.

5. Implement the LBFGS method, Algorithm 7.5 in the text. Use the `EBLS.m` routine with parameters set as above. Test it on the function

$$f(x) = \frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}\sum_{i=1}^{n-1}(x_i - 2x_{i+1})^4,$$

with $n = 1000$ and $x = (1, 1, \ldots, 1)^T$. The evaluation routine for this function is `tridia.m`. Your calling sequence should be

```
[inform,xnew] = LBFGS(fun,x,lbfgsparams)
```

where `lbfgsparams` is defined by

```
lbfgsparams=struct('toler',1.e-4,'maxit',1000,'m',5);
```

(The value of m, which is the number of saved steps, is set to a number of different values in the calling code `testqn.m`.)

6. Comment on the following issues.

(a) How does the number of iterations of LBFGS change as a function of number of saved steps m, from different starting points?

(b) How does the performance of BFGS on `xpowsing` compare with the techniques you used in the previous homework (namely, nonlinear conjugate gradient and steepest descent)?