

CS726, Fall 2008
Homework 8
(due Monday 11/3/08)

Please submit your Matlab files electronically using the instructions on the course web page. Once you have set up the path in your user directory, you need to put your files for submission in a single directory and run the command

```
handin -c cs726-1 -a hwk8 -d <directory name>
```

You should hand in files with the names `BFGS.m`, `LBFGS.m`, `StepSize.m`, and `comments8.txt`. (The last file should contain your written responses on the software questions.) Your codes will be testing by running with `hwk8.m`, available from the web site.

1. (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 `hwk8.m`.

Store the approximation H_k to the inverse Hessian. Use the technique from p.143 in the text to set H_0 , after the first step has been taken.

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

$$\|\nabla f(x)\|_2 \leq \text{qnparams.toler}(1 + |f(x)|).$$

You should use line your search routine `StepSize.m` with parameter settings

```
lsparams = struct('c1', 1.0e-4, 'c2', 0.4, 'maxit', 20);
```

- (b) Repeat this process with the function $f(x) = \sum_{i=1}^3 r_i(x)$, where $x \in \mathbb{R}^2$ and the residuals are defined by

$$r_i(x) = a + Hx + 25 \left(x - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right)^T B \left(x - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) d,$$

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 `hwk8.m`.

- (c) Use your BFGS code to minimize with the function `xpowsing` from Homework 6. The starting point and dimension n are specified in `hwk8.m`.

2. Implement the LBFGS method, Algorithm 7.5 in the text. Use the `StepSize.m` routine with `lparams` 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 $x = (1, 1, \dots, 1)^T$. 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, can be changed to other positive integers.)

3. 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 Homework 6 (namely, nonlinear conjugate gradient and steepest descent)?
4. Exercise 6.4 from the text.
5. Exercise 6.7 from the text.