CS726, Spring 2018

Homework 5 (due Wednesday 4/4/18 6:00pm)

Hand in hard copies of your code and results, and answers to the written question, through Canvas.

1. Exercise 4.7 from the text.

2. Write a Matlab routine called `DoglegTR.m` to implement the dogleg trust-region algorithm with a modified Hessian. Use the trust-region framework of Algorithm 4.1, with the dogleg procedure of p. 73-75 to find the approximate solution of the trust-region subproblem. For the subproblem Hessian B_k use the true Hessian $\nabla^2 f(x^k)$ modified in that the eigenvalues λ_i are replaced by $\max(\lambda_i, \delta)$, for $i = 1, 2, \ldots, n$, for some positive parameter δ.

   Use parameters $\hat{\Delta} = 10$, $\eta = .01$, $\Delta_0 = 1$, $\delta = .01$.

   The header line of your routine should be

   ```
   function [inform,x] = DoglegTR(fun, x, trparams)
   ```

   where the input parameters are:

   fun - a pointer to a function (such as obja, objb, objc)

   x - a strucure with fields x.p, x.f, x.g, and x.h, in which x.p contains the point x, while x.f, x.g, and x.h contain the function, gradient, and Hessian values corresponding to x. On input, x.p is set to the starting point values `x = struct('p', [-1.2, 1.0]);` while x.f, x.g, and x.h are left blank. (The latter values can be used by DoglegTR to store the function information at iterates of the algorithm.)

   trparams - a structure containing parameter values for the test:

   ```
   trparams = struct('maxit',100,'delta',.01,'hatDelta',10,...
   'eta',.01,'Delta0',1,'toler',1.0e-6);
   ```

   (The parameter names and values are discussed above and below.)

   Your routine should call on `fun` to evaluate the objective function, gradient, and Hessian at computed points, using

```
x.f = feval(fun,x.p,1);
```

and

```
x.g = feval(fun,x.p,2);
```

and

```
x.h = feval(fun,x.p,4);
```

You should terminated when either $\|\nabla f(x_k)\|_2 \leq 10^{-6}$ or 100 function evaluations have been taken, whichever comes first.

The output `inform` is a structure containing two fields: `inform.status` is 1 if the gradient tolerance is achieved and 0 if not, while `inform.iter` is the number of steps taken. The output `x` is the solution structure, with point, function, and gradient values at the final value of x_k.

3. Matlab functions that implement the three functions below can be found on Canvas, under the names `obja.m`, `objb.m`, and `objc.m`. Your program will be tested using the code `DoglegMain.m`, also posted. For each of the three functions above, the code will be run with initial point $x_0 = (-1.2, 1)^T$.

   (a) $f(x) = x_1^2 + 5x_2^2 + x_1 - 5x_2$
   (b) $f(x) = x_1^2 + 5x_1x_2 + 100x_2^2 - x_1 + 4x_2$
   (c) $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

   Note that the global variables `numf`, `numg`, and `numH` are incremented by the function evaluation routines. Be sure to set these to zero at the start of `DoglegTR.m`.

   Use the following line in your code, to print out information at each iteration:

```
fprintf(1,' iter %3d: f=%12.5e, ||Df||=%12.5e, Delta=%7.2e\n',...
            inform.iter, x.f, norm(x.g), Delta);
```