Optimization Theory and Algorithms

Matlab Problem: Newton's Method

Optimization Model

We consider an unconstrained optimization problem of minimizing a quartic objective function:

$$\min_{x \in \mathbb{R}^n} \; f(x) \equiv \frac{1}{4}\left(x^T A x - \frac{1}{4}u^T A u\right)^2 + \frac{\mu}{2}\|x - u\|_2^2, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, $u \in \mathbb{R}^n$ and $0 < \mu \in \mathbb{R}$, all 3 being given parameters.

Assignment

Implement the pure Newton's method for solving nonlinear systems of equations of the form $g(x) = 0$, where $g : \mathbb{R}^n \to \mathbb{R}^n$ is differentiable, with either exact or finite-difference approximate Jacobian. That is, write two Matlab functions (see below for details), and apply them to solving the optimization problem (1) with $g(x) = \nabla f(x)$. The 2 Matlab functions must have the following interfaces, respectively,

```
[x, iter] =   myNewton(func,x0,tol,maxit,varargin);
[x, iter] = myFdNewton(func,x0,tol,maxit,varargin);
```

where `myNewton` uses analytic Jacobian $g'(x)$ while the other uses a finite difference approximation. In the above, `x0` is an initial guess, `tol` is an tolerance value for termination, `maxit` the the maximum number of iterations allowed, and `varargin` contains possible parameters required by the input function `func`. Termination criterion is $\|g(x^k)\|/\|g(x^0)\| \le tol$. The input `func` is a function that can evaluate one or both of the values: $g(x)$ and $g'(x)$ for given x. More specifically, inside your functions, `myNewton` and `myFdNewton`, you call the input function `func` as follows, respectively,

```
[g,H] = feval(func,x,varargin{:});      (in myNewton.m)
    g = feval(func,x,varargin{:});      (in myFdNewton.m)
```

where parameters are passed to `func` through `varargin`. We note that `myNewton` uses both gradient g and Hessian H, but `myFdNewton` must only use gradient g (since Hessian will be approximated by finite differences).

A p-file `yzQuartic.p` will be provided that has the interface:

```
[g,H,f] = yzQuartic(x,A,u,mu);
```

which evaluates the related quantities for the given quartic function in (1). It can be called with one (gradient only) output or two (both gradient and Hessian) output (or all three output as well).

Ph.D. students only: Implement your own Matlab function for evaluations of the quartic function:

```
[g,H,f] = myQuartic(x,A,u,mu);
```

that has the same functionalities as the instructor's code. To do so, you will need to derive formulas for the gradient and Hessian of the quartic $f(x)$ in (1),

For given u and $A \succ 0$, establish a condition on μ that guarantees convexity of $f(x)$ *with a proof.*

Details

Download and open the zip file `yzCodeNewton.zip` which contains 3 p-files from the instructor. Also download the test script `test_quartic.m`, which will test several run cases and generate plots. You are to write and put your codes in the same folder.

When `runcase = 0`, the test script will run the instructor's codes only (please try that first). *At each iteration, you codes should print out the same information in the same format as the instructor's codes do.*

During debugging, if your code is running slowly, you may modify the test script to use a smaller m value so that you can have a quicker turn-around time while experimenting.

Once you have written and debugged your solvers, run the test script, with $m = 50$, on `runcase = 1` and `runcase = 2`. In addition, Ph.D students should also run `runcase = 3`.

Requirements

Run the test script `test_quartic.m` with the required run cases. Submit the following items:

- Copies of your codes (for Ph.D. students the total is 3).
- Matlab screen printout and plots generated by *your* codes.
- A short summary (typed, half a page or less) about your experiments and observations that you consider to be the most relevant and most important.

  In addition, Ph.D students also include your derivation and proof for your μ condition.