

Computational Optimization
Laboratory 2 - 2/05/08- Newton's Method
Due in class Friday 2/08/04

All questions should be turned in. Download the matlab files for lab 3 from the course web page.

1. In Lab 1 we worked with the function $f(x) = x^2$. Calculate the minimum of $f(x)$ using Newton's Method by hand. Remember that the Newton's Method algorithm is defined as

$$x_{i+1} = x_i - \text{inv}(\nabla^2 f(x_i)) \nabla f(x_i)$$

For a function of one dimension this is just:

$$x_{i+1} = x_i - f'(x_i)/f''(x_i)$$

How many iterations does it take? Why is this?

2. Now let's use the Matlab routine plainnewton (PN) to solve the above problem. Plainnewton is an implementation of pure Newton's method; the search direction is the negative inverse Hessian, and the step length is one. Plainnewton requires the function, gradient, and Hessian routines, f.m, gradf.m and hessf.m respectively. In Matlab, define x0 = 5 and tol = 1e-6. Evaluate f(x0), gradf(x0) and hessf(x0) in Matlab and then verify by hand that the calculations are accurate. Now run plainnewton on f.

```
[xmin,fxmin]=plainnewton(@f,@gradf,@hessf,x0,tol)
```

Using the chart at the end of this assignment, record the final objective function value, the number of iterations, the time, and the error. Did you get the correct answer? ***What measure is being used for the error?

3. Construct the function, gradient, and Hessian routines K.m, gradK.m, and hessK.m for $K(x) = 0.5 * x^2 - 10 * \sin(x)$. Test the new routines to make sure that they are correct. Use plainnewton to find the minimum of $K(x)$ starting at $x_0 = 10$ and $\text{tol} = 1\text{e-}6$. Record your results in the table below. Did you get a local minimum (check the gradient and Hessian at the solution)? Did you get a global minimum (you can answer this by looking at a plot of the function)? Is the function value at each iteration increasing or decreasing?
4. Now minimize K starting from $x_0 = 3$. Record the results in the chart below. How does it compare with your previous answer? Can you explain this? Is the function value at each iteration increasing or decreasing? Did you get a local minimum? Did you get a global minimum?
5. Plainnewton also works on functions with more than one dimension. Verify that the routines L.m, gradL.m, and hessL.m correctly calculate the function, gradient, and Hessian of $L(x) = 3 * x(1)^2 + x(2)^4 - x(1) * x(2)$. First specify the initial values by typing

```
x0=[1 2]';
tol = 1e-6
```

then run plainnewton by typing

```
[xmin,fxmin]=plainnewton(@L,@gradL,@hessL,x0,tol)
```

Record the results in the chart below. Did the algorithm work as expected? Check the second order sufficient conditions to see if your answer is a strict local minimum.

Function	Start Point	method	f(x)	Iteration	time	$\ \nabla f(x)\ $
f	5	PN				
K	10	PN				
K	3	PN				
L	[1 2]'	PN				

6. The function below is a generalized Rosenbrock function

$f := (x_2 - x_1^2)^2 + (1 - x_1)^2 + (x_4 - x_3^2)^2 + (1 - x_3)^2 + (x_6 - x_5^2)^2 + (1 - x_5)^2$. Construct the function, gradient, and Hessian routines for the generalized Rosenbrock function.

Make a new version of plainnewton.m as follows. Find the line where the value of the search direction is calculated by solving:

$$d = \text{inv}(H) * \text{rhs};$$

This line calculates the inverse of the Hessian matrix H and then multiplies it times the right hand side of the Newton equation $(-\nabla f(x))$. An alternative method to find the search direction is to use Gaussian elimination. The command to do this in Matlab is

$$d = H \backslash \text{rhs};$$

Notice that this line is commented out by the use of the percent sign. Comment out the inverse calculation and uncomment the Gaussian elimination command. You might find it help to also comment out the lines that display the results of iteration.

Compare the plainnewton version using $\text{inv}(H)$ versus H on the generalized Rosenbrock function. Specifically compare the elapsed time using the tic toc function. See if there is any real difference and discuss what happens. Start at different points and experiment a little.