

## Computational Optimization

Laboratory 5 - 2/12/08 - Newton's Method and Steepest Descent. Due Friday 2/18/08.

In this lab you will examine some variations of Newton's method and compare Newton's Method with steepest descent. Since time is short, it is suggested that you team up with a partner on this lab. You may want to look at the matlab results with more precision. Try typing `format long`. **Turn in all m files that you write or modify.**

1. Get the files `newtonexact.m`, `plainnewton.m` from the previous labs. Download `f0.m`, `gradf0.m`, `hessf0.m` for this lab.
2. Run the original newton's method (`plainnewton.m`) on the problem

$$\min_x \log(\exp(-x) + \exp(x))$$

from the starting point 1.1 with tolerance 1e-6. For your convenience, `f0`, `gradf0`, and `hessf0` provide the necessary function, gradient and hessian routines. What solution does the program get? Is this the correct solution?

3. The file `newtonexact.m` does Newton's method with an exact linesearch on the interval  $[0, 1]$  using exact line search. Run `newton.m` on the problem

$$\min_x \log(\exp(-x) + \exp(x))$$

from the starting point 1.1 with tolerance 1e-6. What solution does Newton's method with exact linesearch get? Trying plotting the function `fplot('f',[-5,5])`. Why did the plain newton's method's mess up? Why does this method work?

4. Copy `newton.m` to `steepest.m` and modify it to do steepest descent. Make sure you remove any mention of the hessian. To make sure everything is working okay, test steepest descent on the function `f0` and the function from previous labs. No need to turn in these results.
5. Run the following experiment using the `newtonexct.m` and the Steepest Descent Code from the given starting point on the following problems . Feel free to split up coding the functions with other people in the class. Use tolerance 1e-6. Record the condition number of the Hessian of the solution found by the code, e.g. `cond(hessf(xmin))`.

(a)  $f_1(x) = x_1^2 + x_2^2 + x_3^2$ ,  $x_0 = (1, 1, 1)^T$

(b)  $f_2(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ ,  $x_0 = (0, 0)^T$

(c)  $f_3(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ ,  $x_0 = (-1.2, 1)^T$

(d)  $f_4(x) = (x_1 + x_2)^4 + x_2^2$ ,  $x_0 = (2, -2)^T$

(e)  $f_5(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + c(x_1^2 + x_2^2 - 0.25)^2$ ,  $x_0 = (1, -1)^T$  with  $c = 1$ , and  $c = 100$ .

Function	Cond(Q)	method	f(x)	Iterations	time	$\ \nabla f(x)\ $
f1		SD				
		NM				
f2		SD				
		NM				
f3		SD				
		NM				
f4		SD				
		NM				
f5a		SD				
		NM				
f5b		SD				
		NM				

- How does Steepest Descent compare with Newton's Method? Compute the condition number of each problem i.e. the ratio of the largest and smallest of the Hessian evaluated at the optimal solution. Does the condition number affect the results? If so how?
- For graduate credit only** Modify the exact Newton's Method code and your steepest descent code to perform a backtracking linear search as discussed on page 37 of NW with  $\rho = 1/2$  and  $c = .6$ . Use tolerance  $1e-6$  and select  $\bar{\alpha} = 1$ . Repeat the above experiment with the modified codes and complete the chart below.

Function	Cond(Q)	method	f(x)	Iterations	time	$\ \nabla f(x)\ $
f1		SD-BT				
		NM-BT				
f2		SD-BT				
		NM-BT				
f3		SD-BT				
		NM-BT				
f4		SD-BT				
		NM-BT				
f5a		SD-BT				
		NM-BT				
f5b		SD-BT				
		NM-BT				

How does the backtracking search change your results? Discuss. The book suggested using an initial step size (before backtracking) of  $\bar{\alpha} = \alpha_{k-1} \frac{\nabla f(x_{k-1})' p_{k-1}}{\nabla f(x_k)' p_k}$ . Does this help your results?

8. **Hand in all m files that you write (steepest.m) and all functions. Graduate students, don't forget to hand in your m codes with the backtracking search.**