Math 433/833 **Assignment # 3** **Assigned: 2016.02.09**
Due: 2016.02.16

0. Read sections 2.6, 2.7, 2.8, 11.1, and 11.2 in the book.

1. Page 65, #6.1. Find the first four terms in the Taylor series expansion of $f(x) = \log(1+x) = \log_e(1+x) = \ln(1+x)$. Evaluation for $p = 0.1$ and $p = 0.01$, and compare, deriving a bound for the accuracy.

   Solution. Note that $\frac{d^n}{dx^n}\log(1+x) = \frac{(-1)^{n-1}(n-1)!}{(1+x)^n}$ for $n = 1, 2, 3, \ldots$. For the Taylor series, we have:

   $$f(x_0+p) = f(x_0) + pf'(x_0) + \tfrac{p^2}{2}f''(x_0) + \tfrac{p^3}{3!}f'''(x_0) + \tfrac{p^4}{4!}f^{(4)}(\xi)$$

   for some ξ satisfying $x_0 < \xi < x_0 + p$. Thus, for $x_0 = 0$, we have

   $$\begin{aligned}\log(1+p) &= \log(1+0) + p\frac{1}{1+0} + \tfrac{p^2}{2}\frac{-1}{(1+0)^2} + \tfrac{p^3}{3!}\frac{2}{(1+0)^3} + \tfrac{p^4}{4!}\frac{-3}{(1+\xi)^4} \\ &= p - \tfrac{p^2}{2} + \tfrac{p^3}{3} - \tfrac{p^4}{8}\frac{1}{(1+\xi)^4}\end{aligned}$$

   These are four terms, with the first term being zero. I will leave it to you to check the decimals. To find a bound on the accuracy, we see from the above equation that, since $\xi > x_0 = 0$,

   $$\left|\log(1+p) - \left(p - \tfrac{p^2}{2} + \tfrac{p^3}{3}\right)\right| = \left|\tfrac{p^4}{8}\frac{1}{(1+\xi)^4}\right| \le \left|\tfrac{p^4}{8}\frac{1}{(1+0)^4}\right| = \frac{p^4}{8}$$

   For example, if $p = 0.1$, the error in approximating by the first four terms is at most $0.1^4/8 = 1.25 \cdot 10^{-5}$.

2. Page 66, # 6.4. Find the first three terms of the Taylor series for $f(x_1, x_2) = 3x_1^4 - 2x_1^3x_2 - 4x_1^2x_2^2 + 5x_1x_2^3 + 2x_2^4$ at $\mathbf{x}_0 = \left(\begin{smallmatrix}1\\-1\end{smallmatrix}\right)$.

   Solution. Recall the Taylor series (with ∇^2 denoting the Hessian):

   $$f(\mathbf{x}_0 + \mathbf{p}) \approx f(\mathbf{x}_0) + \mathbf{p}^T\nabla f(\mathbf{x}_0) + \mathbf{p}^T\nabla^2 f(\mathbf{x}_0)\mathbf{p}$$

   We compute $f(\mathbf{x}_0) = -2$, and

   $$\nabla f(\mathbf{x}) = \begin{bmatrix} 12x_1^3 - 6x_1^2x_2 - 8x_1x_2^2 + 5x_2^3 \\ -2x_1^3 - 8x_1^2x_2 + 15x_1x_2^2 + 8x_2^3 \end{bmatrix},$$

   and

   $$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} 36x_1^2 - 12x_1x_2 - 8x_2^2 & -6x_1^2 - 16x_1x_2 + 15x_2^2 \\ -6x_1^2 - 16x_1x_2 + 15x_2^2 & -8x_1^2 + 30x_1x_2 + 24x_2^2 \end{bmatrix}.$$

Thus,

$$\nabla f(\mathbf{x}_0) = \begin{bmatrix} 5 \\ 13 \end{bmatrix}$$

and

$$\nabla^2 f(\mathbf{x}_0) = \begin{bmatrix} 40 & 25 \\ 25 & -14 \end{bmatrix}$$

We find:

$$\begin{aligned} f(\mathbf{x}_0 + \mathbf{p}) &\approx f(\mathbf{x}_0) + \mathbf{p}^T \nabla f(\mathbf{x}_0) + \tfrac{1}{2}\mathbf{p}^T \nabla^2 f(\mathbf{x}_0)\mathbf{p} \\ &= -2 + \mathbf{p}^T \begin{bmatrix} 5 \\ 13 \end{bmatrix} + \tfrac{1}{2}\mathbf{p}^T \begin{bmatrix} 40 & 25 \\ 25 & -14 \end{bmatrix} \mathbf{p} \\ &= -2 + 5p_1 + 13p_2 + 20p_1^2 + 25p_1p_2 - 7p_2^2 \end{aligned}$$

3. Page 66, # 6.6 Prove that if $\mathbf{p}^T \nabla f(\mathbf{x}_k) < 0$, then $f(\mathbf{x}_k + \epsilon\mathbf{p}) < f(\mathbf{x}_k)$ for $\epsilon > 0$ sufficiently small.

   Solution. Recall that, by first-order Taylor approximation:

   $$f(\mathbf{x}_k + \epsilon\mathbf{p}) = f(\mathbf{x}_k) + \epsilon\mathbf{p}^T \nabla f(\xi)$$

   for some ξ lying on the line between $\mathbf{x}_k$ and $\mathbf{x}_k + \epsilon\mathbf{p}$. (This is also known as the Mean-Value Theorem.) If f is smooth (as we usually assume), then ∇f is continuous, and therefore, so is $\mathbf{p}^T \nabla f(\mathbf{x})$. For a function is continuous and negative at a point $\mathbf{x}_k$, then it is negative for all $\mathbf{x}$ sufficiently close to $\mathbf{x}_k$ (to see this, draw a picture). Moreover, as ϵ gets small, $\mathbf{x}_k + \epsilon\mathbf{p}$ get closer to $\mathbf{x}_k$, so ξ approaches $\mathbf{x}_k$ (since again, ξ lies on the line between $\mathbf{x}_k$ and $\mathbf{x}_k + \epsilon\mathbf{p}$). Thus, for sufficiently small $\epsilon > 0$, $\mathbf{p}^T \nabla f(\xi) < 0$, since $\mathbf{p}^T \nabla f(\mathbf{x}_k) < 0$. Therefore,

   $$f(\mathbf{x}_k + \epsilon\mathbf{p}) = f(\mathbf{x}_k) + \epsilon\mathbf{p}^T \nabla f(\xi) < f(\mathbf{x}_k) + \epsilon 0 = f(\mathbf{x}_k).$$

4. Page 74, # 7.1. For this problem, write a code to use Newton's method. You can write a function[*] to do this. Here is an example of how a function might start:

```
function x = Newton(f,df,x0,tolerence,maxIter)
% A program to solve f(x) = 0 using Newton's method.
```

[*]See the Matlab Intro, Section 10, if you need a refresher on Matlab functions.

where `maxIter` is the maximum number of iterations you will allow, and `tolerence` is the maximum error between steps you will allow. Once you write the code, you can call it from the command line, as in the following example, which uses Newton's method with initial guess $x_0 = 2$ to compute $\sqrt{3}$ out to 15 decimal places. (Note that the derivative $\frac{d}{dx}(x^2 - 3) = 2x$ has been computed by hand.)

```
>> format long g
>> x = Newton(@(x) x^2 - 3, @(x) 2*x, 2, 1e-15,1000)
```

(The first line makes Matlab display more decimal places; you only need to enter it once per session.) Include your answers for page 74, # 7.1, and also print your Newton code and staple it to your homework.

Solution. One possible solution is included at the end of this document.

5. Page 362, # 2.3. First, we compute when the gradient is equal to zero:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} 16x_1 + 3x_2 - 25 \\ 3x_1 + 14x_2 + 31 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Solving this system, we find the solution to be $\mathbf{x}_* = (443/215, -571/215)$. Thus, there is a single critical point. To check that $\mathbf{x}_*$ is a minimizer, we compute the Hessian at $\mathbf{x}_*$:

$$H_f(\mathbf{x}_*) = \nabla^2 f(\mathbf{x}_*) = \begin{pmatrix} 16 & 3 \\ 3 & 14 \end{pmatrix}$$

Note that since the determinate and upper left corner are both positive, by Sylvester's Criterion, $H_f(\mathbf{x}^*)$ is positive-definite. By a theorem we learned in class, any local minimizer of a convex function is a global minimizer, and moreover, it is unique. (Note: there are convex functions without any local or global minimizers; consider $f(x) = e^x$.)

One solution to problem #4:

```
function x = Newton(f,df,x0,tolerence,maxIter)
% A program to solve f(x) = 0 using Newton's method:
%     x_0 = initial guess;
%     x_{n+1} = x_n - f(x_n)/f'(x_n);
% Until either |x_{n+1} - x_n| < tolerence
%                  and |f(x_n)| < tolerence
%            or maxIter is reached.
% We calculate f' by hand and input it as df.
%
% Example run (to find sqrt(3)):
% x = Newton(@(x) x^2 -3, @(x)2*x, 3.0, 10^(-15), 1000)
%
% Example of run that doesn't converge:
% x = Newton(@(x) sin(x), @(x) cos(x), pi/2, 10^(-15), 1000)

% Initialize
x = x0;

for count = 1:maxIter
    x_old = x; % Store the old value to check for error later.

    % Newton iteration:
    x = x - f(x)/df(x);

    if ((abs(x-x_old) < tolerence) && (abs(f(x)) < tolerence))
        break;
    end
end

if (count < maxIter)
    display(sprintf('Exited after %d iterations.',count));
else
    warning('Maximum iterations reached. May not be converged.');
end
```