Shi Qin

Math456/CMPSC456 Homework 1

Due Jan 19 2020

1. (15 points) Let $f(x_1, x_2) = e^{x_1-3x_2} + e^{x_1+3x_2} + e^{-x_1}$. Starting with $(0, 0)$, determine the approximation after one step of the Newton's iteration for the minimization problem: $\min f(x_1, x_2)$.

2. (25 points) Computer Problem: Apply the Newton's method and the modified Newton's method ($J(\vec{x}_k) \approx J(\vec{x}_0)$) to the nonlinear equations,

$$x_1^2 - x_2^2 + 2x_2 = 0, 2x_1 + x_2^2 - 6 = 0.$$

with initial guess (-5,-4). With the tolerance $TOL = 10^{-9}$, compare the number of iterations needed before convergence. (Attach the code).

1. $f(x_1, x_2) = e^{x_1-3x_2} + e^{x_1+3x_2} + e^{-x_1}$ $\quad (0,0)$

$$\frac{\partial f}{x_1} = e^{x_1-3x_2} + e^{x_1+3x_2} - e^{-x_1} = 0 = h_1(x_1, x_2)$$

$$\frac{\partial f}{x_2} = -3e^{x_1-3x_2} + 3e^{x_1+3x_2} = 0 = g(x_1, x_2)$$

$$f\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{bmatrix} \frac{\partial h}{x_1} & \frac{\partial h}{x_2} \\ \frac{\partial g}{x_1} & \frac{\partial g}{x_2} \end{bmatrix}^{-1} \begin{bmatrix} h(0,0) \\ g(0,0) \end{bmatrix}$$

$$h(0,0) = e^{0-0} + e^{0+0} - e^{0} = 1$$

$$g(0,0) = -3e^{0} + 3e^{0} = 0$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{bmatrix} e^{x_1-3x_2}+e^{x_1+3x_2}+e^{-x_1} & -3e^{x-3x_2}+3e^{x_1+3x_2} \\ -3e^{x_1-3x_2}+3e^{x_1+3x_2} & 9e^{x_1-3x_2}+9e^{x_1+3x_2} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{bmatrix} e^0+e^0+e^0 & -3e^0+3e^0 \\ -3e^0+3e^0 & 9e^0+9e^0 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \frac{1}{54} \begin{bmatrix} 18 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{18} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{bmatrix} \frac{1}{3} \\ 0 \end{bmatrix}$$

$$= -\begin{bmatrix} \frac{1}{3} \\ 0 \end{bmatrix}$$

e.g. 1.

$$x_1^2 - x_2^2 + 2x_2 = 0$$
$$2x_1 + x_2^2 - 6 = 0$$

let $f_1 = x_1^2 - x_2^2 + 2x_2$
$f_2 = 2x_1 + x_2^2 - 6$

$$\vec{x}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Newton's method.

$$\vec{f}(\vec{x}^{(0)}) = \begin{pmatrix} 2 \\ -3 \end{pmatrix}$$

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} \leftarrow D\vec{f}$$

$$= \begin{pmatrix} 2x_1 & -2x_2+2 \\ 2 & 2x_2 \end{pmatrix}$$

$$J(\vec{x}^{(0)}) = \begin{pmatrix} 2 & 0 \\ 2 & 2 \end{pmatrix}$$

$$\vec{x}^{(1)} = \vec{x}^{(0)} - \begin{pmatrix} 2 & 0 \\ 2 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ -3 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ \frac{7}{2} \end{pmatrix}$$

编辑器 - D:\OneDrive - The Pennsylvania State University\tusrau\10_学校\22SP\CMPSC456\456code

M_20220119cmpsc456hw1newton.m

```matlab
%function x = M_20220119cmpsc456hw1newton(x0,f,tol)

% x0 is initial guess
% tol is tolerance
% x0 is initial guess(-5,-4)

tol=10^-9;
dx=1.0;


syms h(x1,x2)
h(x1,x2)= x1.^2-x2.^2+2.*x2;
syms g(x1,x2)
g(x1,x2)= 2.*x1+x2.^2-6;
% 2 equation
syms J(x1,x2)
J(x1,x2)=[diff(h,x1) diff(h,x2);diff(g,x1) diff(g,x2)];

x0=[-5 -4]
%initial guess

step=0;
%reversed=inv(J(x0(1),x0(2)));
while dx>tol
    reversed=inv(J(x0(1),x0(2)));
    last=[h(x0(1),x0(2));g(x0(1),x0(2))];
    x1=x0-reversed*last;
    dx=norm(x1-x0);
    x0=x1
    step=step+1;
end
disp(step)
disp(x0)
```

count the steps

will need 6 steps to converge

and modified will need 9 steps to converge

1

命令行窗口

```
     6

[-23341192293371082286023721142542561614116860349096691750141213843673033598050 8
[ -95153742232190507675962589056140885497210276575017146136716315423850957820997 8
```

2

命令行窗口

```
     9

[  -3299504902413424707508813429997663895736346966593935498245194289128240763069 9
[-10760726702636834654768347982498119418315621978958833561606691114152506593942 6
fx >>
```

```java
package math.calculus;
package math.calculus;

import java.lang.reflect.Array;

import org.junit.Test;


import junit.framework.Assert;
import nilgiri.math.DoubleReal;
import nilgiri.math.DoubleRealFactory;

class newton {

    static final double EPSILON = 0.000000001;

    static double f(double x1,double x2){
        return x1*x1-x2*x2+2*x2;

    }
    static double h(double x1,double x2){
        return 2*x1+x2*x2-6;

    }

    static Array J(double x1,double x2){
        double[][] J={{diff(f(x1,x2),x1), diff(f(x1,x2),x2)},{diff(h(x1,x2),x1), diff(h(x1,x2),x2)}};
        return J;

    }
    static void newton(double x)
    {
        double h = func(x) / derivFunc(x);
        while (Math.abs(h) >= EPSILON)
        {
            h = func(x) / derivFunc(x);

            // x(i+1) = x(i) - f(x) / f'(x)
            x = x - h;
        }

        System.out.print("The value of the"
                + " root is : "
                + Math.round(x * 100.0) / 100.0);



    Run | Debug
    public static void main (String[] args)
    {

        // Initial values assumed
        double x0 = -20;
        newton(x0);
    }
}
```

tried to use Java but failed, do not know how to differentiation