ECSE 443 – ASSIGNMENT 3

MUHAMMAD TAHA – 260505597

Chapter 5

Exercises

5.3

a) As $\frac{df(x)}{x} = 2x$, the Newton iteration is given as follows, when f(x) = 0 needs to be solved.

$$x_{k+1} = x_k - \frac{x_k^2 - y}{2x_k}$$

b) Given an initial guess of 4-bit accuracy, to obtain 24-bit accuracy, the number of iterations required is $k = log_2\left(\frac{24}{4}\right) = 3$. For 53-bit accuracy $k = log_2\left(\frac{53}{4}\right) = 4$.

5.4

Let $f(x) = x - 1 - y$. Then, $\frac{df(x)}{dx} = -x - 2$. Hence, to solve f(x) = 0 the Newton iteration is as follows:

$$x_{k+1} = x_k - \frac{x_k^{-1} - y}{-x_k^{-2}} = x_k + {x_k}^2\left(x_k^{-1} - y\right) = 2x_k - {x_k}^2 y$$

5.6

a) As $g_1'(x) = 1 - 2x$ and $g_1'\left(\sqrt{3}\right) = \left|1 - 2\sqrt{3}\right| = 2.46 > 1$, the iterative scheme is not convergent.

b) As $g_2'(x) = 1 - \frac{2x}{y}$ and $g_1'\left(\sqrt{3}\right) = \left|1 - \frac{2\sqrt{3}}{3}\right| = 0.155 < 1$, the iterative scheme is locally convergent.

c) $f'(x) = 2x$, the fixed-point iteration function given by Newton's method is $g(x) = x - \frac{x^2 - y}{2x}$.

5.9

This first iteration given by Newton's method is $x_0 + s_0$, where s_0 can be found via:

$$J_f(x_0)s_0 = \begin{bmatrix} 2x_1 & -2x_2 \\ 2x_2 & 2x_1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} -x_1^2 \\ 1 - 2x_1x_2 \end{bmatrix} = -f(x_0)$$

For $x_0 = [0 \quad 1]^T$

$$J_f(x_0)s_0 = \begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} -2s_2 \\ 2s_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Hence $s_1 = 0.5$ and $s_2 = -0.5$.

Therefore, the first iteration is = $x_0 + s_0 = \begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}0.5\\-0.5\end{bmatrix} = \begin{bmatrix}0.5\\0.5\end{bmatrix}$

5.10

If $x_k = x^*$, then $f(x_k) = f(x^*) = 0$

Hence, $x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} = x^* - 0 = x^*$

If $x_{k-1} = x^*$, then $f(x_{k-1}) = f(x^*) = 0$

Therefore,

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x^*}{f(x_k)}$$

$$x_{k+1} = x_k - (x_k - x^*)$$

$$x_{k+1} = x^*$$

Computer Problems

5.1

a)

- $|g_1'(x)| = \left|\frac{2x}{3}\right| = |\frac{4}{3}|$

Therefore, the gradient is divergent.

- $|g_2'(x)| = \left|\frac{3}{2\sqrt{3x-2}}\right| = |\frac{3}{4}|$

Therefore, the gradient is convergent at 0.75.

- $|g_3'(x)| = \left|\frac{2}{x^2}\right| = |\frac{1}{2}|$

Therefore, the gradient is convergent at 0.5.

- $|g_4'(x)| = \left|\frac{-2(x^2-2)}{(2x-3)^2} + \frac{2x}{2x-3}\right| = |-4+4| = 0$

Therefore, the gradient is quadratically convergent.

b) The MATLAB output for the functions above is given below:

```
ans =

(x^2+2)/3


error =

   1.5830e+65


ratio =

   2.2971e+32
```

```
ans =

sqrt(3*x-2)


error =

    0.0341


ratio =

    0.7437
```

```
ans =

3-2/x


error =

   4.8852e-04


ratio =

    0.4998
```

```
ans =

(x^2-2)/(2*x-3)


error =

     0


ratio =

     0
```

5.2

The termination criteria used for each function was that the computation for each function was repeated a maximum of ten times per method and the tolerance of error was set as 10^{-10}.

The output obtained for each function with all 3 methods is shown below.

```
bisection method
iterationCnt =

    10


a =

    2.0938


fa =

   -0.0089


b =

    2.0947


fb =

    0.0020
```

```
Newton method
iterationCnt =

     3


x =

    2.0946


fx =

  -8.8818e-16
```

```
Secant method
x =

    2.0946


fx =

  -8.6926e-11


exitflag =

     1


output =

    intervaliterations: 0
            iterations: 3
             funcCount: 5
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [2.09375, 2.09473]'
```

Figure 1: Part a

```
bisection method
iterationCnt =

    10


a =

    0.5664


fa =

    0.0012


b =

    0.5674


fb =

  -3.7535e-04
```

```
Newton method
iterationCnt =


     3


x =

    0.5671


fx =

   1.1102e-16
```

```
Secant method
x =

    0.5671


fx =

   1.1102e-16


exitflag =

     1


output =

    intervaliterations: 0
            iterations: 3
             funcCount: 5
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0.566406, 0.567383]'
```

Figure 2: Part b

```
bisection method
iterationCnt =

    10


a =

    1.1133


fa =

   -0.0012


b =

    1.1143


fb =

   1.3981e-04
```

```
Newton method
iterationCnt =


     3


x =

    1.1142


fx =

  -2.2204e-16
```

```
Secant method
x =

    1.1142


fx =

  -2.2204e-16


exitflag =

     1


output =

    intervaliterations: 0
            iterations: 3
             funcCount: 5
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [1.11328, 1.11426]'
```

Figure 3: Part c

```
>> convergence
bisection method
iterationCnt =

    10

a =

    0.9992

fa =

  -4.7684e-10

b =

    1.0001

fb =

   9.3126e-13
```

```
Newton method
iterationCnt =

    10

x =

    1.0000

fx =

  -2.6645e-15
```

```
Secant method
x =

    1.0000

fx =

     0

exitflag =

     1

output =

    intervaliterations: 0
            iterations: 14
             funcCount: 16
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0.999219, 1.0001]'
```

Figure 4: Part d

5.13

The following screenshots shows the result obtained by the implementation in MATLAB compared with the built-in MATLAB solver.

```
>> NewtonLinearEquationSolver

x =

    0.8780
    0.6768
    1.3309


iteration =

     7
```

```
x =

    0.8780
    0.6768
    1.3309


output =

        iterations: 6
         funcCount: 28
         algorithm: 'trust-region-dogleg'
    firstorderopt: 2.5674e-16
           message: 'Equation solved, inaccuracy possible.…'
```

5.17

a)

$$R_{k+1} = I - AX_{k+1}$$

$$R_{k+1} = I - A(X_k + X_k(I - AX_k))$$

$$R_{k+1} = I - A(X_k + X_k R_k)$$

$$R_{k+1} = I - AX_k(I + R_k)$$

$$R_{k+1} = I - AX_k - AX_k R_k$$

$$R_{k+1} = R_k - AX_k R_k$$

$$R_{k+1} = R_k(I - \ AX_k)$$

$$R_{k+1} = R_k^2$$

$$E_{k+1} = A^{-1} - X_{k+1}$$

$$E_{k+1} = A^{-1} - X_k + X_k(I - AX_k)$$

$$E_{k+1} = A^{-1} - X_k + X_k - AX_k^2$$

$$E_{k+1} = A^{-1} - AX_k^2$$

$$A^{-1}E_{k+1} = A^{-1}A^{-1} - X_k X_k$$

$$A^{-1}E_{k+1} = E_k(A^{-1} - X_k)$$

$$E_{k+1} = E_k A(A^{-1} - X_k)$$

$$E_{k+1} = E_k A E_k$$

b)

```
A =

    85    76    66
    93    74    17
    68    39    71

>> NewtonMatrixInverse(A)

X =

    0.0015    0.0017    0.0012
    0.0014    0.0013    0.0007
    0.0012    0.0003    0.0013

Inverse calculated by MATLAB
ans =

   -0.0394    0.0242    0.0308
    0.0468   -0.0133   -0.0403
    0.0121   -0.0159    0.0067
```

Chapter 6

Exercises

6.1

a) $f(x) = \frac{1}{2}(x_1^2 - x_2)^2 + \frac{1}{2}(1 - x_1)^2$

$$\nabla f(x) = \begin{bmatrix} 2x_1^3 - 2x_1x_2 + x_1 - 1 \\ -x_1^2 + x_2 \end{bmatrix}$$

$$H_f(x) = \begin{bmatrix} 6x_1 - 2x_2 + 1 & -2x_1 \\ -2x_1 & 1 \end{bmatrix}$$

f has a minimum, critical point at $x^* = [1 \quad 1]$ as $H_f(x)$ is positive definite at x^*.

b) The matrix inverse is given by:

$$H^{-1} = \frac{1}{5}\begin{bmatrix} 1 & 4 \\ 21 & 4 \end{bmatrix}$$

Hence,

$$\Delta x_2 = \begin{bmatrix} 1 & 4 \\ 21 & 4 \end{bmatrix}\begin{bmatrix} -9 \\ 2 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} -1 \\ 6 \end{bmatrix}$$

Therefore,

$$x_2 = \Delta x_2 + x_1 = \frac{1}{5}\begin{bmatrix} 9 \\ 16 \end{bmatrix}$$

c) The derivative is close to zero with this step as shown below

$||f'(x_0)|| = ||f'(2,2)||^2 = ||(9, -2)||$ = 9.2 and $||f'(x_1)\Big|\Big| = ||f'\left(\frac{9}{5}, \frac{16}{5}\right)\Big|\Big| = 0.94$

d) The new iteration deviates from the solution so the chosen step is not good

$||x_0 - x^*||$ = $||(1, 1)||$ = 1.41

$||x_1 - x^*||$ = $||(\frac{6}{5}, \frac{4}{5})||$ = 2.34

6.2

a) When $Ax^* = b$:

$$\nabla f(x) = Ax - b$$

$$Hf(x) = A$$

The system for Newton's method step from an arbitrary x_0 is given below

$$H_f(x_0)s_0 = As_0 = b - Ax_0 = -\nabla f(x_0)$$

Which gives the result $x_1 = x_0 + s_0$. Hence, we get:

$$Ax_1 = A(x_0 + s_0) = Ax_0 + As_0 = (b - Ax_0) + As_0 = b$$

Implying that $x_1 = x^*$. Thus, it can be observed that Newton's method converges to the solution in one iteration from any x_0.

b) For the steepest descent method, we need to find the minimum of $\alpha f(x_0 + \alpha s_0)$**,** where $s_0 = -\nabla f(x) = b - Ax$. The minimizing value for α is:

$$\alpha = s_k^T s_k / s_k^T A s_k$$

If, $x_0 - x^*$ is an eigenvector of A, then

$$\lambda(x_0 - x) = A(x_0 - x) = Ax_0 - Ax = \mathrm{A}x_0 - b.$$

The following equations proves that s_0 is A's eigenvector:

$$As_0 = \lambda A(x^* - x_0) = \lambda A(x^* - Ax) = \lambda(b - Ax_0) = \lambda s_0$$

Therefore, $\lambda = \frac{s_0^T A s_0}{s_0^T s_0}$

As $\alpha = \frac{1}{\lambda}$,

$$x_1 = x_0 - \alpha \nabla f(x_0)$$

$$x_1 = x_0 + \left(\frac{1}{\lambda}\right)(b - Ax_0)$$

$$x_1 = x_0 + \left(\frac{1}{\lambda}\right)(Ax^* - Ax_0)$$

$$x_1 = x_0 + \left(\frac{1}{\lambda}\right)(A(x^* - x_0))$$

$$x_1 = x_0 + \left(\frac{1}{\lambda}\right)(\lambda(x^* - x_0)) = x^*$$

Showing that the steepest method converges to the solution in one iteration from the starting point.

6.3

If y is a non-zero vector, then

$$\begin{bmatrix} 0 \\ y \end{bmatrix}^T \begin{bmatrix} B & J^T \\ J & O \end{bmatrix} \begin{bmatrix} 0 \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}^T \begin{bmatrix} J^T y \\ 0 \end{bmatrix} = 0$$

Hence, a 2x2 Hessian matrix of the Lagrange function for constrained optimization cannot be positive definite.

6.4

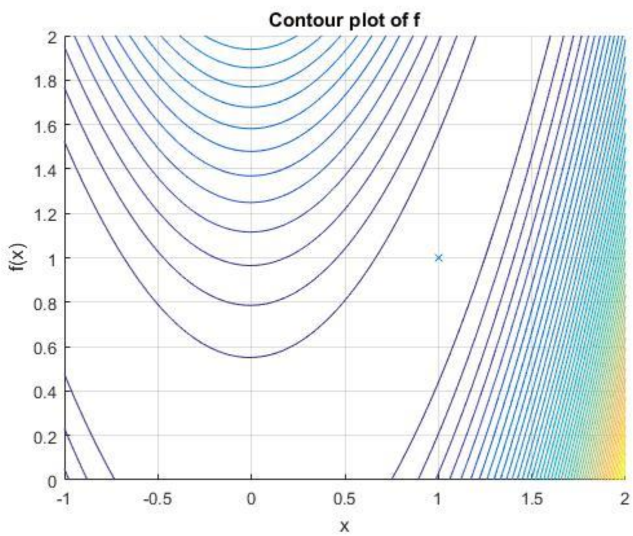
a) The following vertices are contained in the feasible region:
{(0,0), (0, 1.5), (0.857, 0.857), (1.09, 0.545), (1.2, 0)}

b) The values of the objective function at each vertex are: 0, -3, -4.29, -4.37 and -3.6 respectively.
The lowest cost is found at: (1.09, 0.545)

Computer Problems

6.9

The plot of the path taken in plane by the solutions for each method is given below



Final solutions obtained from each method after a suitable number of iterations is provided below

Steepest descent	Newton method	dampened Newton method
x =	x =	x =
0.9100	0.9100	0.9100
0.8277	0.8277	0.8277
iteration =	iteration =	iteration =
50	50	50

6.11

The following figures shows the solutions obtained from MATLAB

```
Newton method solution:
iteration =

     7


x =

   1.0e-11 *

   -0.1160
   -0.0104
    0.0045

Steepest descent solution:
iteration =

     2


x =

     0
     0
     0


solution:
iteration =

    20


x =

    0.0075
   -0.0000
   -0.0671
```

6.12

The solution below is obtained for the Fletcher Reeves algorithm.

```
>> conjugateGradient

grad =

   -1.1821
    2.4015
   -4.4803


iteration =

    20
```

The solution below is obtained for Polak Ribiere algorithm.

```
>> conjugateGradient

grad =

   -1.1821
    2.4015
   -4.4803


iteration =

    20
```

The tolerance was set as 10^{-5} and the function successfully converged to the solution in n steps for an arbitrary quadratic function of n variables.

6.13

a) The solution obtained from MATLAB is given below

```
x =

    1.0000
    1.9149
```

b) The solution obtained from MATLAB is given below

```
x =

   -0.3148
    0.7592
```

6.19 (a)

The solution obtained from MATLAB for all unknown x is shown below

```
x =

   -0.2308
    0.0769
    0.0769
    0.0769
    0.0769
    1.0000
    1.0000
    1.0000
```