Solved Example Set-4

Solve using modified Newton's method the following system of non-linear algebraic equations

$$x_1^3 - 2x_2 - 2 = 0$$
$$x_1^3 - 5x_3^2 + 7 = 0$$
$$x_2 x_3^2 - 1 = 0$$

Solution

We have our system of non-linear equations in the following form:

$$f_1(x_1, x_2, x_3) = x_1^3 - 2x_2 - 2$$
$$f_2(x_1, x_2, x_3) = x_1^3 - 5x_3^2 + 7$$
$$f_3(x_1, x_2, x_3) = x_2 x_3^2 - 1$$

The Newton's method gives the improvement for the components of x in the s+1 th iteration as:

$$\begin{Bmatrix} x_1^{(s+1)} \\ x_2^{(s+1)} \\ x_3^{(s+1)} \end{Bmatrix} = \begin{Bmatrix} x_1^{(s)} \\ x_2^{(s)} \\ x_3^{(s)} \end{Bmatrix} - [J]^{-1} \begin{Bmatrix} f_1^{(s)} \\ f_2^{(s)} \\ f_3^{(s)} \end{Bmatrix}$$

where, the Jacobian $[J] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_2} \end{bmatrix}$

We have
$$\frac{\partial f_1}{\partial x_1} = 3x_1^2, \ \frac{\partial f_1}{\partial x_2} = -2, \ \frac{\partial f_1}{\partial x_3} = 0$$
$$\frac{\partial f_2}{\partial x_1} = 3x_1^2, \ \frac{\partial f_2}{\partial x_2} = 0, \ \frac{\partial f_2}{\partial x_3} = -10x_3$$
$$\frac{\partial f_3}{\partial x_1} = 0, \ \frac{\partial f_3}{\partial x_2} = x_3^2, \ \frac{\partial f_3}{\partial x_3} = 2x_2 x_3$$

Therefore, you have $[J] = \begin{bmatrix} 3x_1^2 & -2 & 0 \\ 3x_1^2 & 0 & -10x_3 \\ 0 & x_3^2 & 2x_2 x_3 \end{bmatrix}$

You start with some initial guess $\begin{Bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{Bmatrix} = \begin{Bmatrix} 2 \\ 1 \\ 2 \end{Bmatrix}$ and you have $\begin{Bmatrix} f_1^{(0)} \\ f_2^{(0)} \\ f_3^{(0)} \end{Bmatrix} = \begin{Bmatrix} 4 \\ -5 \\ 3 \end{Bmatrix}$

In Modified Netwon-Raphson method, we need to evaluate J only once and will retain the same J in every iteration.

$$[J] = \begin{bmatrix} 12 & -2 & 0 \\ 12 & 0 & -20 \\ 0 & 4 & 4 \end{bmatrix}$$

We are going to demonstrate a screen-based iteration using Matlab to improve the values of x and reduce f_1, f_2, and f_3 to zero.

```
>> f = [4;-5;3]; x1=[2;1;2];Ja=[12 -2 0; 12 0 -20; 0 4 4]
Ja =
    12    -2     0
    12     0   -20
     0     4     4

>> x2=x1-inv(Ja)*f
x2 =
    1.6212
    0.7273
    1.5227

>> f=[x2(1)^3-2*x2(2)-2; x2(1)^3-5*x2(3)^2+7;x2(2)*x2(3)^2-1]
f =
    0.8065
   -0.3324
    0.6863

>> format('long')
>> x3=x2-inv(Ja)*f
x3 =
   1.53663222244568
   0.62305981490335
   1.45535866744075

>> f=[x3(1)^3-2*x3(2)-2; x3(1)^3-5*x3(3)^2+7;x3(2)*x3(3)^2-1]
f =
   0.38223566794924
   0.03801104328139
   0.31968358619114

>> x4=x3-inv(Ja)*f
x4 =
   1.49527778567500
   0.56605102825391
   1.43244655754241

>> f=[x4(1)^3-2*x4(2)-2; x4(1)^3-5*x4(3)^2+7;x4(2)*x4(3)^2-1]
f =
   0.21112323838291
   0.08370959381520
   0.16148188239618

>> x5=x4-inv(Ja)*f
x5 =
```

```
   1.47253269938996
   0.53514212973512
   1.42298498546215

>> f=[x5(1)^3-2*x5(2)-2; x5(1)^3-5*x5(3)^2+7;x5(2)*x5(3)^2-1]
f =
   0.12268577534815
   0.06853869056486
   0.08360195038418

>> x6=x5-inv(Ja)*f
x6 =
   1.45955234942051
   0.51860291759251
   1.41862371000872

>> f=[x6(1)^3-2*x6(2)-2; x6(1)^3-5*x6(3)^2+7;x6(2)*x6(3)^2-1]
f =
   0.07206840651232
   0.04680808870280
   0.04368486102376

>> x7=x6-inv(Ja)*f
x7 =
   1.45208328533760
   0.50982273635118
   1.41648267599411

>> f=[x7(1)^3-2*x7(2)-2; x7(1)^3-5*x7(3)^2+7;x7(2)*x7(3)^2-1]
f =
   0.04213873794457
   0.02966835368973
   0.02292015151719

>> x8=x7-inv(Ja)*f
x8 =
   1.44779800889274
   0.50518044665431
   1.41539492781168

>> f=[x8(1)^3-2*x8(2)-2; x8(1)^3-5*x8(3)^2+7;x8(2)*x8(3)^2-1]
f =
   0.02439612923242
   0.01804301416584
   0.01204961135189

>> x9=x8-inv(Ja)*f
x9 =
   1.44535670311054
   0.50273067657736
   1.41483229505066

>> f=[x9(1)^3-2*x9(2)-2; x9(1)^3-5*x9(3)^2+7;x9(2)*x9(3)^2-1]
f =
   0.01396973850071
   0.01067897606386
   0.00634134455329
```

```
>> x10=x9-inv(Ja)*f
x10 =
   1.44397728580874
   0.50143904201692
   1.41453859347277

>> f=[x10(1)^3-2*x10(2)-2; x10(1)^3-5*x10(3)^2+7;x10(2)*x10(3)^2-1]
f =
   0.00791621566748
   0.00619713758168
   0.00333912334770

>> x11=x10-inv(Ja)*f
x11 =
   1.44320414254363
   0.50075829025998
   1.41438456439279
```

(*See the results are almost converging*)

```
>> f=[x11(1)^3-2*x11(2)-2; x11(1)^3-5*x11(3)^2+7;x11(2)*x11(3)^2-1]
f =
   0.00444413371806
   0.00354223427514
   0.00175879529821

>> x12=x11-inv(Ja)*f
x12 =
   1.44277400960464
   0.50039955948507
   1.41430359634315

>> f=[x12(1)^3-2*x12(2)-2; x12(1)^3-5*x12(3)^2+7;x12(2)*x12(3)^2-1]
f =
   0.00247470428329
   0.00200051010758
   0.00092655203759

>> x13=x12-inv(Ja)*f
x13 =
   1.44253627995973
   0.50021053375724
   1.41426098406158

>> f=[x13(1)^3-2*x13(2)-2; x13(1)^3-5*x13(3)^2+7;x13(2)*x13(3)^2-1]
f =
   0.00136842850743
   0.00111884082774
   0.00048816127301

>> x14=x13-inv(Ja)*f
x14 =
   1.44240564410922
   0.50011093290791
   1.41423854459266
```

```
>> f=[x14(1)^3-2*x14(2)-2; x14(1)^3-5*x14(3)^2+7;x14(2)*x14(3)^2-1]
f =
  1.0e-003 *

   0.75217935444893
   0.62074011237723
   0.25720416022978

>> x15=x14-inv(Ja)*f
x15 =
   1.44233421566636
   0.50005845192795
   1.41422672453256
```

Now you can see that the results have converged to a tolerance, which is suggested that $\max\left|x_j^{(s+1)} - x_j^{(s)}\right| \le 1\times 10^{-4}$. Here we took 15 iterations to converge to the solution.

Therefore, the one solution of the system is: $\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1.44233 \\ 0.50006 \\ 1.41423 \end{Bmatrix}$