step= 0

a=

1

1

1

step = 1

b =

-2.8333

-20.0000

3.6667

step = 2

b =

-0.9586

-9.7004

1.8455

step = 3

b =

0.0638

-4.7418

0.7415

step = 4

b =

0.6595

-2.4475

0.1340

step = 5

b =

0.9714

-1.4884

-0.1541

step = 6

b =

1.0812

-1.1960

-0.2488

step = 7

b =

1.0958

-1.1598

-0.2609

step = 8

b =

1.0960

-1.1592

-0.2611

uvw\_roots =

1.0960

-1.1592

-0.2611

a =

3

3

3

step = 1

b =

2.1667

10.6667

-1.2222

step = 2

b =

5.6905

4.4412

-3.2600

step = 3

b =

4.6675

0.3197

-1.7668

step = 4

b =

2.6463

4.4131

-2.0422

step = 5

b =

3.0752

1.5884

-1.5548

step = 6

b =

1.9632

2.0336

-1.1813

step = 7

b =

2.1125

1.1605

-1.0766

step = 8

b =

2.0051

1.0189

-1.0053

step = 9

b =

2.0001

1.0001

-1.0000

step = 10

b =

2.0000

1.0000

-1.0000

uvw\_roots =

2.0000

1.0000

-1.0000

function roots = multiNewton(a,iter)

%a is initial guess vector for (u,v,w)

%F is matrix of set of nonlinear equations

%DF is the jacobian matrix

function f = F(u,v,w)

f = [2\*u^2-4\*u+v^2+3\*w^2+6\*w+2;

u^2+v^2-2\*v+2\*w^2-5;

3\*u^2-12\*u+v^2+3\*w^2+8];

end

function df = DF(u,v,w)

df = [4\*u-4 2\*v 6\*w+6; 2\*u 2\*v-2 4\*w; 6\*u-12 2\*v 6\*w];

end

step = 1

s = gaussian(DF(a(1),a(2),a(3)),-F(a(1),a(2),a(3)));

b = a + s

for i = 1:iter

step = step + 1

a = b;

s = gaussian(DF(a(1),a(2),a(3)),-F(a(1),a(2),a(3)));

b = a + s

end

u = b(1);

v = b(2);

w = b(3);

roots = b;

end