

```

function [ x ] = CSCI3656_ProblemSet5_5( x0 )

%Parameters
%x0 = [0; 0; 0];           %Initial x value
%f = [f1; f2; f3];         %Matrix of functions with roots we're trying to find
%df = [df11 df12 df13; df21 df22 df23; df31 df32 df33]
tol = 10^(-15);            %Digit accuracy desired
n_max = 25;                %Total iterations
solved = false;            %Boolean to check if convergences
labels = ['      Step      ' '      u      ' '      v      ' '      w      '];
data = [];                 %Data to be printed

%Initial f and df values
f = zeros(size(x0));
df = zeros(length(x0), length(x0));

f(1) = 2*x0(1)^2 - 4*x0(1) + x0(2)^2 + 3*x0(3)^2 + 6*x0(3) + 2;
f(2) = x0(1)^2 + x0(2)^2 - 2*x0(2) + 2*x0(3)^2 - 5;
f(3) = 3*x0(1)^2 - 12*x0(1) + x0(2)^2 + 3*x0(3)^2 + 8;

df(1,1) = 4*x0(1) - 4;
df(1,2) = 2*x0(2);
df(1,3) = 6*x0(3) + 6;
df(2,1) = 2*x0(1);
df(2,2) = 2*x0(2) - 2;
df(2,3) = 4*x0(3);
df(3,1) = 6*x0(1) - 12;
df(3,2) = 2*x0(2);
df(3,3) = 6*x0(3);

for i = 1 : n_max

    %Solves s = DF(x_k)s * F(x_k), built-in function for solving
    s = mldivide(df, f);

    %Newton Computation: x_k+1 = x_k - (DF(x_k))^-1 * F(x_k)
    x1 = x0 - s;

    %Check if within tolerance
    if(abs(x1 - x0) <= tol * abs(x1))
        solved = true;
        break;
    end

    %Updates x0 to restart process and prints
    x0 = x1;
    f = zeros(size(x0));
    df = zeros(length(x0), length(x0));

```

CSCI3656_ProblemSet5_5.m

```
f(1) = 2*x0(1)^2 - 4*x0(1) + x0(2)^2 + 3*x0(3)^2 + 6*x0(3) + 2;  
f(2) = x0(1)^2 + x0(2)^2 - 2*x0(2) + 2*x0(3)^2 - 5;  
f(3) = 3*x0(1)^2 - 12*x0(1) + x0(2)^2 + 3*x0(3)^2 + 8;
```

```
df(1,1) = 4*x0(1) - 4;  
df(1,2) = 2*x0(2);  
df(1,3) = 6*x0(3) + 6;  
df(2,1) = 2*x0(1);  
df(2,2) = 2*x0(2) - 2;  
df(2,3) = 4*x0(3);  
df(3,1) = 6*x0(1) - 12;  
df(3,2) = 2*x0(2);  
df(3,3) = 6*x0(3);
```

```
%Outputting
```

```
data = [data; i x0(1) x0(2) x0(3)];
```

```
end
```

```
%Printing
```

```
disp(labels);  
fprintf('\n');  
disp(data);
```