

---

```

%HW2-Prb4
%Navneet Singh(nsinghl@andrew.cmu.edu)

function Problem_4
clc      %clear screen
clear all %clearing all stored variables
close all %close previous plots

%We are running loop to solve equations with random initial guesses
for i = 1:10000
func = @eqn;
%using rand function to generate random numbers
%Here random numbers are between -1000 and 1000
guess = -1000 + (1000+1000).*rand(1,4); %initial guess
options = optimset('display','off');
[x(i,:), fval, exitflag] = fsolve(func, guess,options);

%(a)plotting p-norm of solution vs p-norm of guess vector
ng = norm(guess);
ns = norm(x(i,:));
hold on
plot(ng,ns,'bo')
xlabel('Norm of guess','fontsize',15,'fontname','times new roman')
ylabel('Norm of solution','fontsize',15,'fontname','times new roman')

end

%Defining 4 equations
function f = eqn(x)
f(1) = 4*x(1) - x(2) + x(3) - x(4)*x(1);
f(2) = -x(1) + 3*x(2) - 2*x(3) -x(4)*x(2);
f(3) = x(1) -2*x(2) + 3*x(3) - x(4)*x(3);
f(4) = x(1)^2 + x(2)^2 + x(3)^2 - 1 ;
end

%All solutions are stored in x matrix(dimension - i X 4 )
%rounding solutions to 4th place of decimal
x = round(x,4);

% As all solutions are stored as rows in 'x' matrix, we will check for
unique rows and
% that will give us unique solutions. We will store these values in
'c'
% matrix. Number of rows will give number of unique solution.
c = unique(x,'rows');
[m,n] = size(c);
fprintf('Number of unique solutions = %d\n',m)
fprintf('Solutions are as follows:\n')
c
end

Number of unique solutions = 6

```

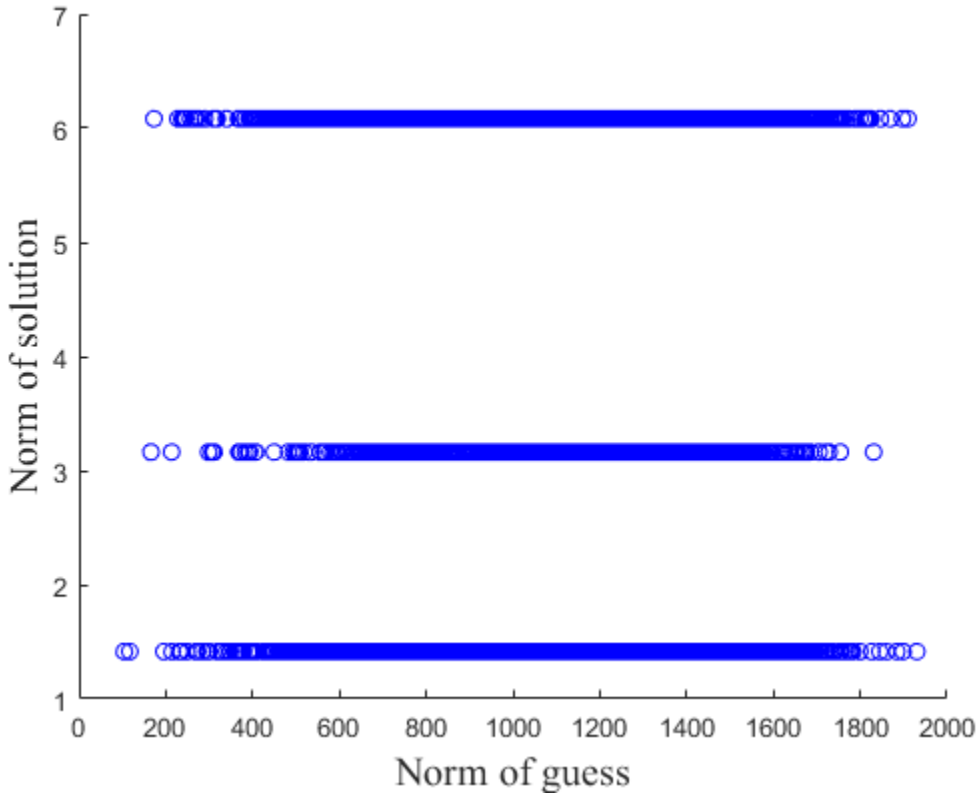
---

---

*Solutions are as follows:*

*c* =

-0.8165	-0.4082	0.4082	3.0000
-0.5774	0.5774	-0.5774	6.0000
0	-0.7071	-0.7071	1.0000
0	0.7071	0.7071	1.0000
0.5774	-0.5774	0.5774	6.0000
0.8165	0.4082	-0.4082	3.0000



*Published with MATLAB® R2016a*