# Nonlinear equations and systems in MATLAB

## *Algebraic Equations*

An algebraic equation or polynomial equation is an equation of the form

$P(x)=0$

where P(x) is polynomial with coefficients in some field, often the field of the rational numbers. For most authors, an algebraic equation is univariate, which means that it involves only one variable.

A polynomial in a single indeterminate can be written in the form

$p_1x^n+...+p_nx+p_{n+1}=0$

where $p_{n+1}\neq 0,\ n\geq 0$, and $p_1,...,p_n$ are numbers (coefficients) and $x$ is a variable.

An example of a first degree polynomial, is linear equation. An example of a second-degree polynomial is a quadratic equation. A third-degree polynomial is a cubic equation.

The solution of an algebraic equation in the Matlab consists of two steps: we define the polynomial using the **poly**, then we find its roots, using the command **roots**.

Definition of a polynomial by command

p = **poly**(**r**)

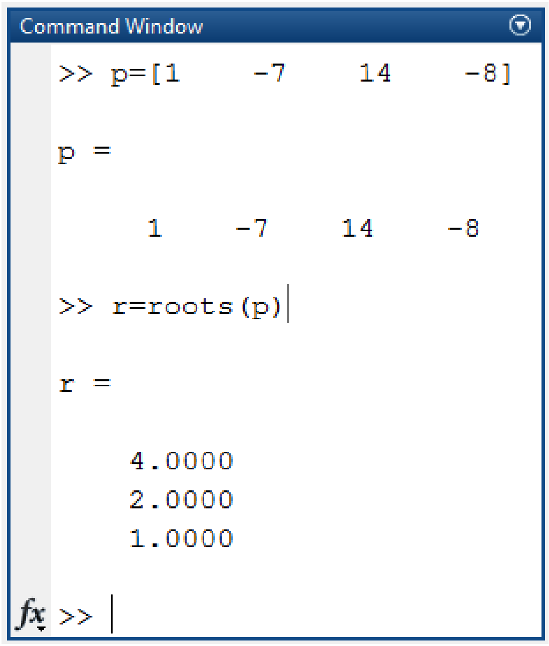
where r is a vector, returns the coefficients of the polynomial whose roots are the elements of r.

r = **roots**(p)

returns the roots of the polynomial represented by p as a column vector. Input p is a vector containing n+1 polynomial coefficients, starting with the coefficient of xn. A coefficient of 0 indicates an intermediate power that is not present in the equation. For example, p = [3 2 -2] represents the polynomial $3x^2+2x-2$.

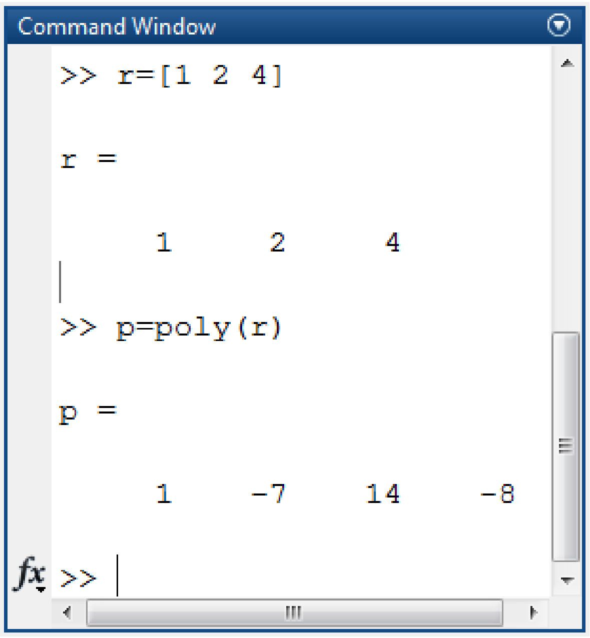
The roots function solves polynomial equations of the form $p_1x^n+...+p_nx+p_{n+1}=0$. Polynomial equations contain a single variable with nonnegative exponents.

For example, solve polynomial equation: $x^3-7x^2+14x-8=0$



```
>> p=[1     -7     14     -8]

p =

     1    -7    14    -8

>> r=roots(p)

r =

    4.0000
    2.0000
    1.0000

>> 
```

Find coefficients of the polynomial whose roots are the elements of $r_1=1$, $r_2=2$; $r_3=4$:

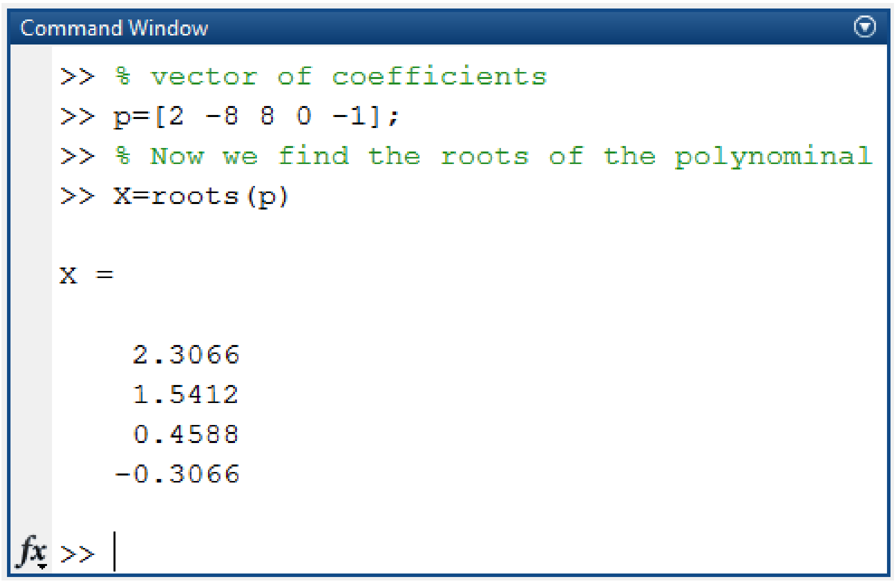


```
>> r=[1 2 4]

r =

     1     2     4

>> p=poly(r)

p =

     1    -7    14    -8

>> 
```

Problem 5.1.

Find the roots of the polynomial $2x^4 - 8x^3 + 8x^2 - 1 = 0$

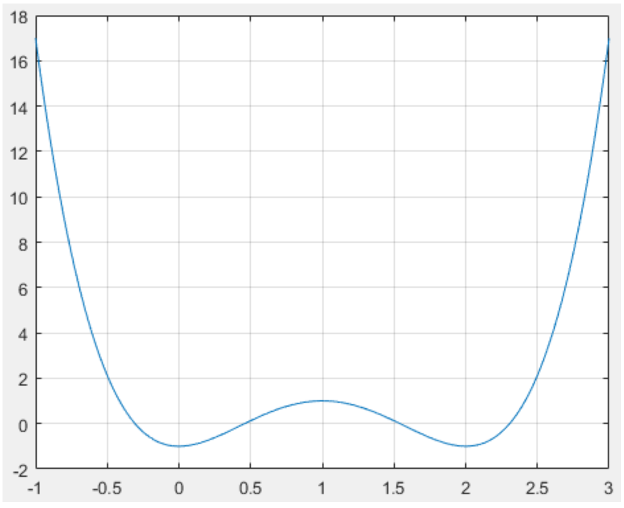
First, we define the vector of coefficients V. Then we find the roots of the polynomial.



```
>> % vector of coefficients
>> p=[2 -8 8 0 -1];
>> % Now we find the roots of the polynominal
>> X=roots(p)

X =

    2.3066
    1.5412
    0.4588
   -0.3066

>>
```

Graphical solution of the polynomial we can see below.

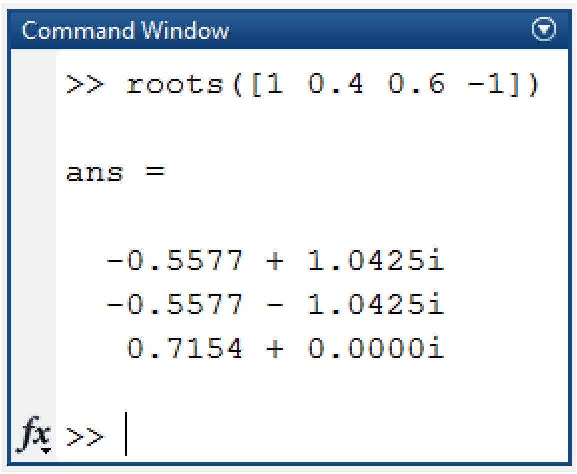
```
x=-1:0.001:3;
p=-1+8*x.^2-
8*x.^3+2*x.^4;
plot(x,p);
grid on;
```



Problem 5.2.

Find the roots of the polynomial $x^3 + 0.4x^2 + 0.6x - 1 = 0$

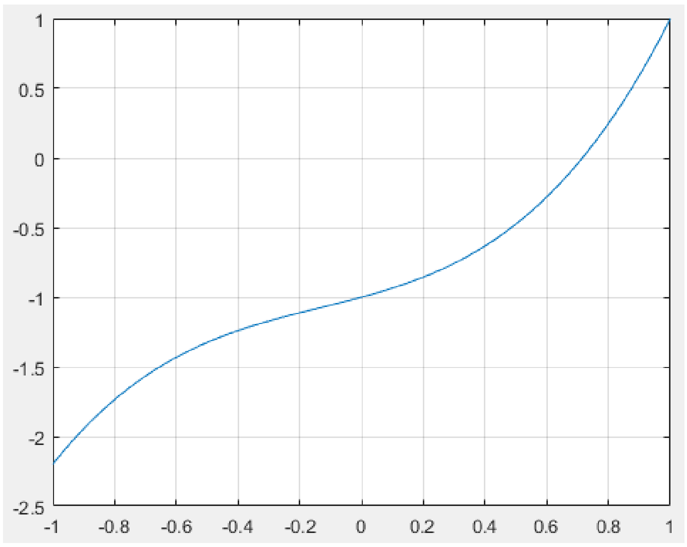
Solution of this problem is different from the previous method of determining of a polynomial.



```
>> roots([1 0.4 0.6 -1])

ans =

  -0.5577 + 1.0425i
  -0.5577 - 1.0425i
   0.7154 + 0.0000i

>>
```

Graphical solution of the polynomial we can see in Fig.

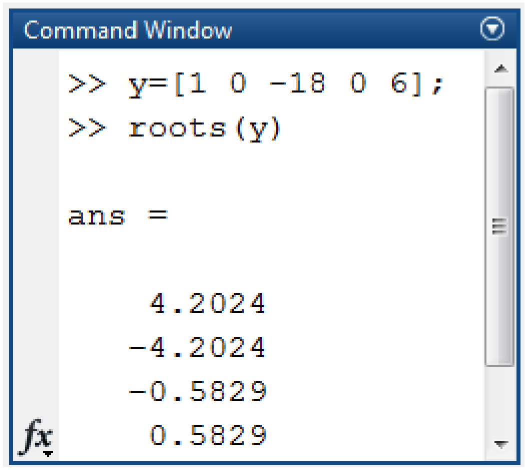
```
x1=-1:0.001:1;
p1=-
1+0.6*x1+0.4*x1.^2+x1
 .^3;
```



Problem 5.3.

Solve the algebraic equation $y(x) = 0$, if $y(x) = x^4 - 18x^2 + 6$

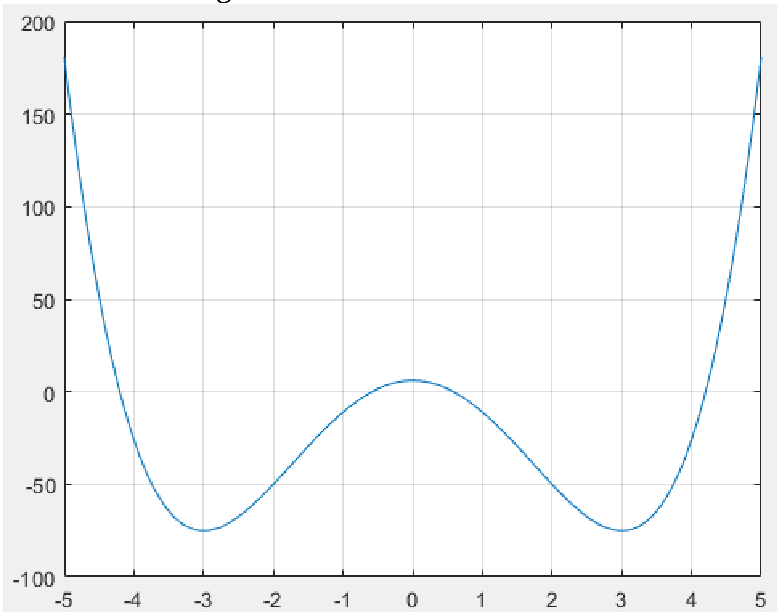
Solution of this problem is different from the previous method of determining of a polynomial.



```
>> y=[1 0 -18 0 6];
>> roots(y)

ans =

    4.2024
   -4.2024
   -0.5829
    0.5829
```

Graphical solution of the polynomial we can see in Fig.

```
x=-5:0.01:5;
y=x.^4-
18.*x.^2+6;
plot(x,y);
```



***Transcendental equation***

A transcendental equation is an equation containing a transcendental function of the variable(s) being solved for.

Such equations often do not have closed-form solutions. Examples include:

$x = e^{-x}$, $x = \cos x$ or $x = \ln x$

Approximate numerical solutions to transcendental equations can be found using numerical, analytical approximations, or graphical methods. Numerical methods for solving arbitrary equations are called root-finding algorithms. The simplest root-finding algorithm is the bisection method. It works when f is a continuous function and it requires previous knowledge of two initial guesses, a and b, such that f(a) and f(b) have opposite signs. Although it is reliable, it converges slowly, gaining one bit of accuracy with each iteration.

Numerical solution of the nonlinear equation held in two phases: At the beginning of separate roots, ie, are sufficiently close intervals, which contain only one root. These intervals are called intervals isolation of the root, you can define them, depicting the graph of the function f (x), or by any other method. The second step is carried out clarification of separation of roots, or, in other words, finding the roots of a given accuracy.
To solve transcendental equations in Matlab used command

**fsolve(fun,x0)**

where x0 – real vector (initial value of function argument); fun – external (i.e function or list or string).

Consider a few examples.

Problem 5.4.

Solve the equation

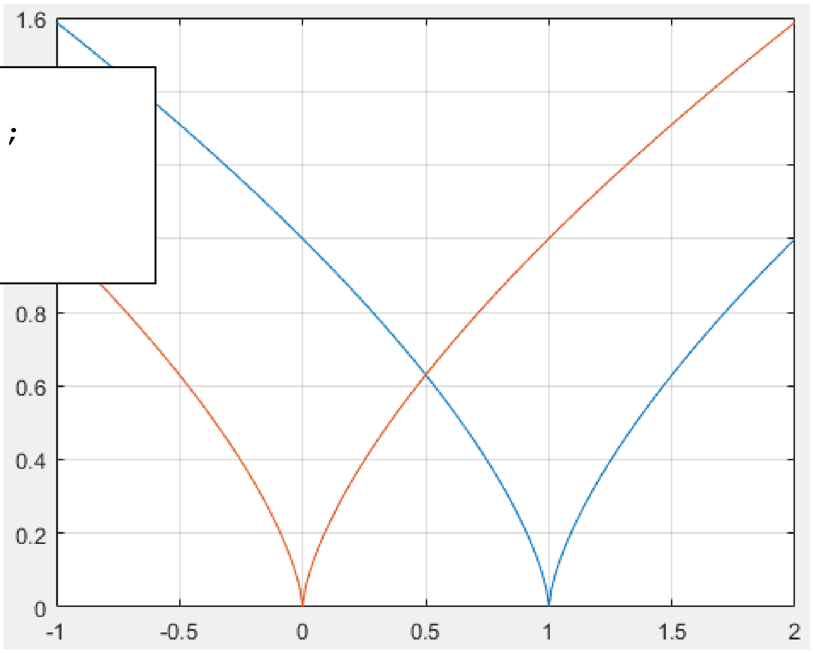
$$\sqrt[3]{(x-1)^2} - \sqrt[3]{x^2} = 0$$

We use a graphical method of separation of roots. We can submit the equation in the form f(x) − g(x) = 0. Finding a root of f(x) − g(x) = 0 is the same as solving the equation f(x) = g(x). Here, x is called the unknown in the equation. Conversely, any equation can take the canonical form f(x) = 0, so equation solving is the same thing as computing (or finding) a root of a function.

In our case

$f(x) = \sqrt[3]{(x-1)^2}$, $g(x) = \sqrt[3]{x^2}$

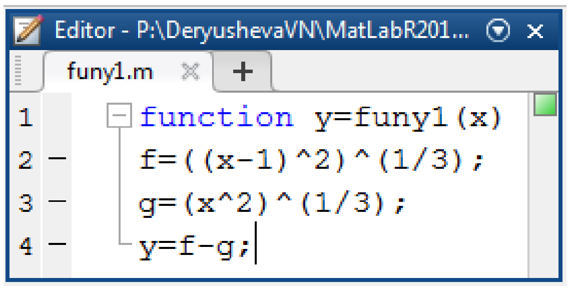
```
x=-1:0.01:2;
f=((x-1).^2).^(1/3);
g=(x.^2).^(1/3);
plot(x,f,x,g);
grid on;
```



We can see below root lies in the range [0; 1]

Choose zero as an initial approximation, given function, described of the equation and solve it.

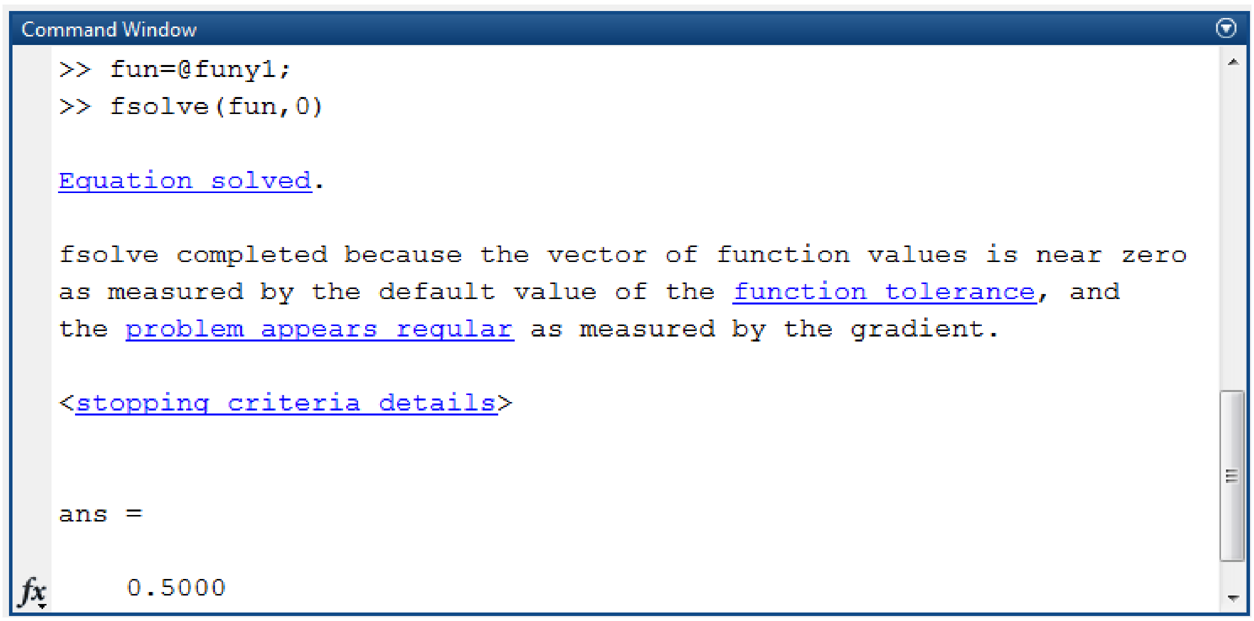
Write a function that computes of these two equations:



```
function y=funy1(x)
f=((x-1)^2)^(1/3);
g=(x^2)^(1/3);
y=f-g;
```

Save this code as a file named funy1.m on your MATLAB® path.

Solve the system of equations starting at the point



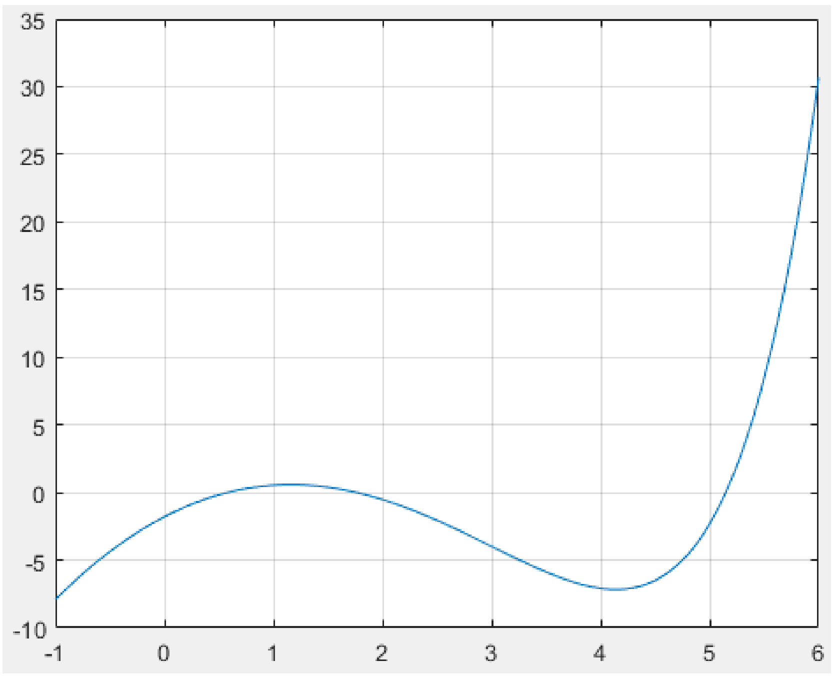
```
>> fun=@funy1;
>> fsolve(fun,0)

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


ans =

    0.5000
```

Problem 5.5.

Solve the equation

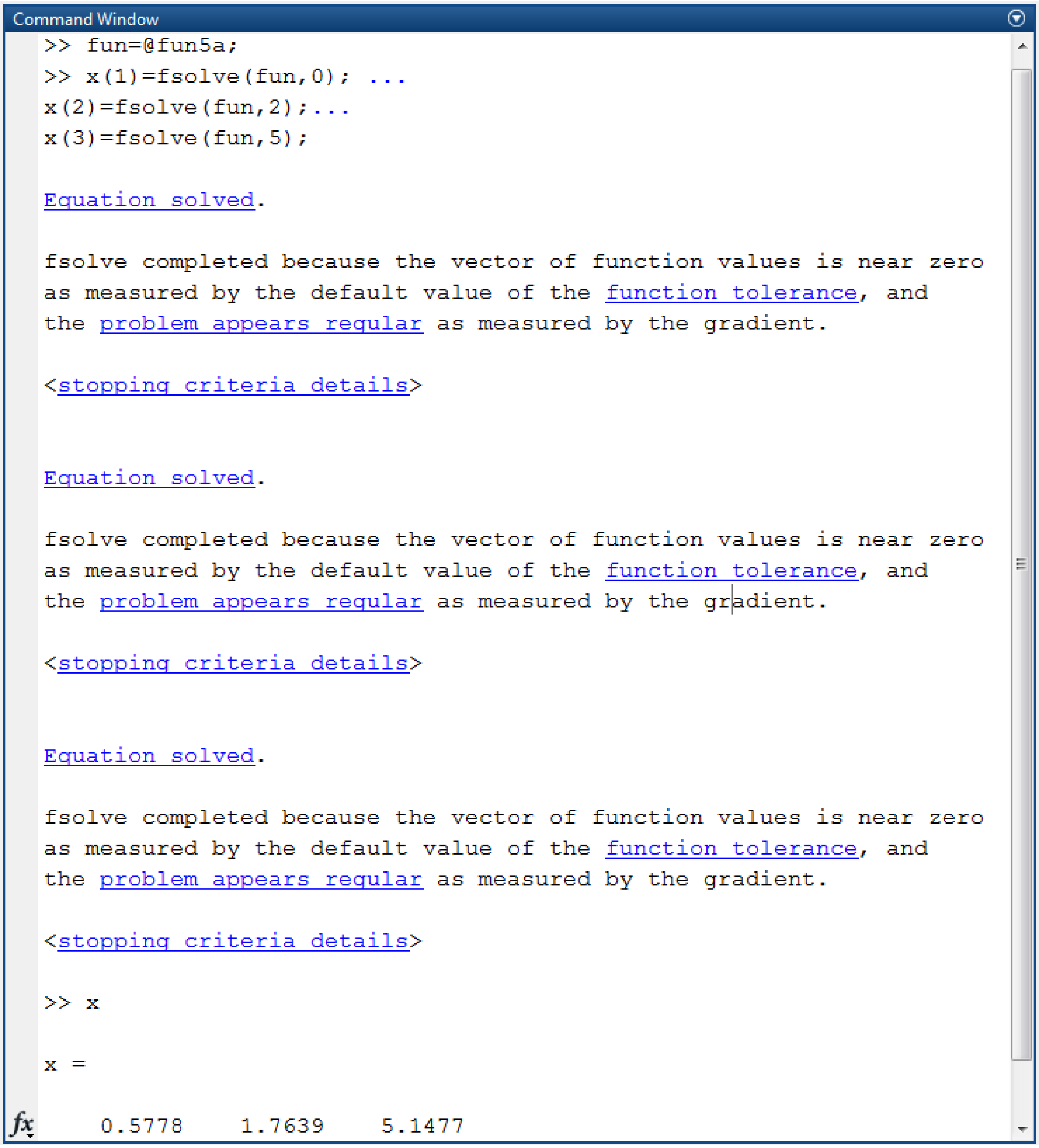
$$f(x) = \frac{e^x}{5} - 2(x-1)^2$$

```
x=-1:0.01:6;
f=((exp(x))./5-2*((x-
1).^2));
plot(x,f);
```



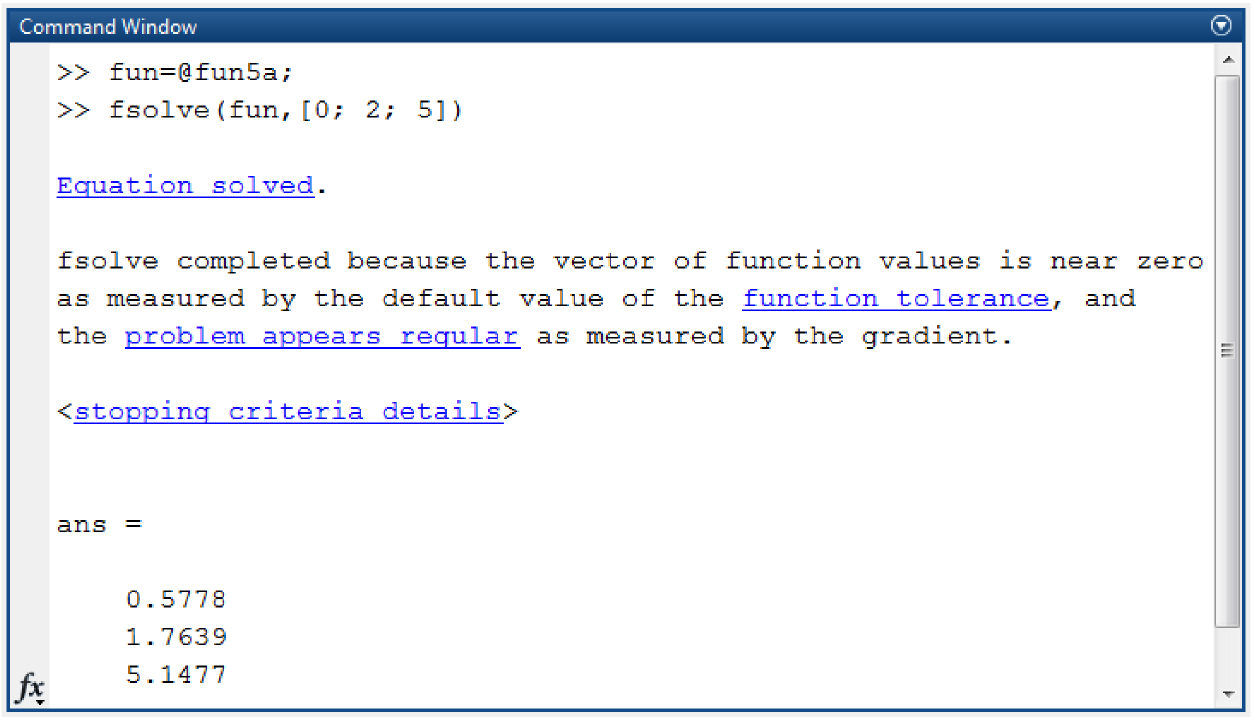
In the figure we see the graph crosses the x-axis three times, ie, the equation has three roots.

We obtain all the solutions by causing a sequence of command **fsolve** with various initial approximation.



```
>> fun=@fun5a;
>> x(1)=fsolve(fun,0); ...
x(2)=fsolve(fun,2);...
x(3)=fsolve(fun,5);

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>

>> x

x =

    0.5778    1.7639    5.1477
```

Or initial approximation can be specified as a vector.

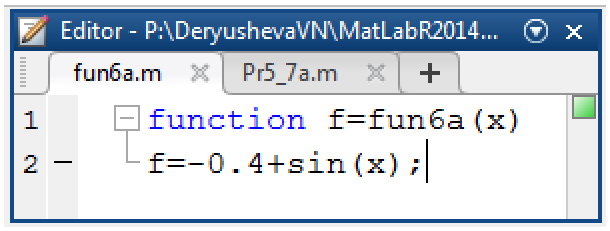


```
>> fun=@fun5a;
>> fsolve(fun,[0; 2; 5])

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


ans =

    0.5778
    1.7639
    5.1477
```

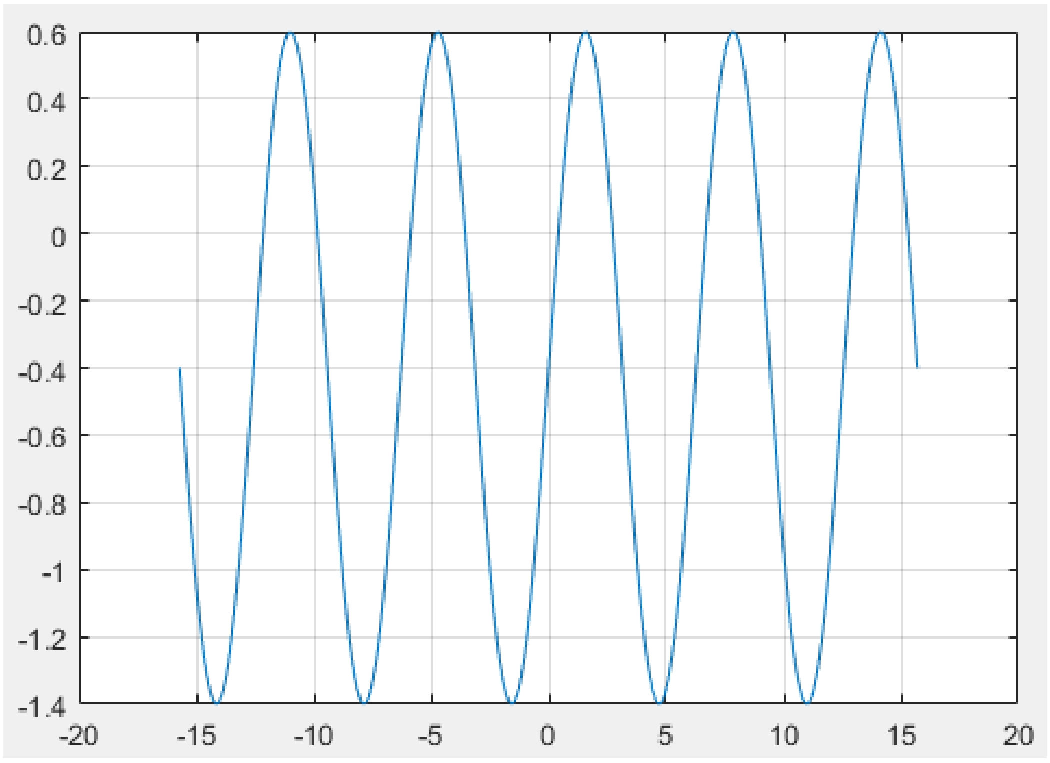
Problem 5.6.

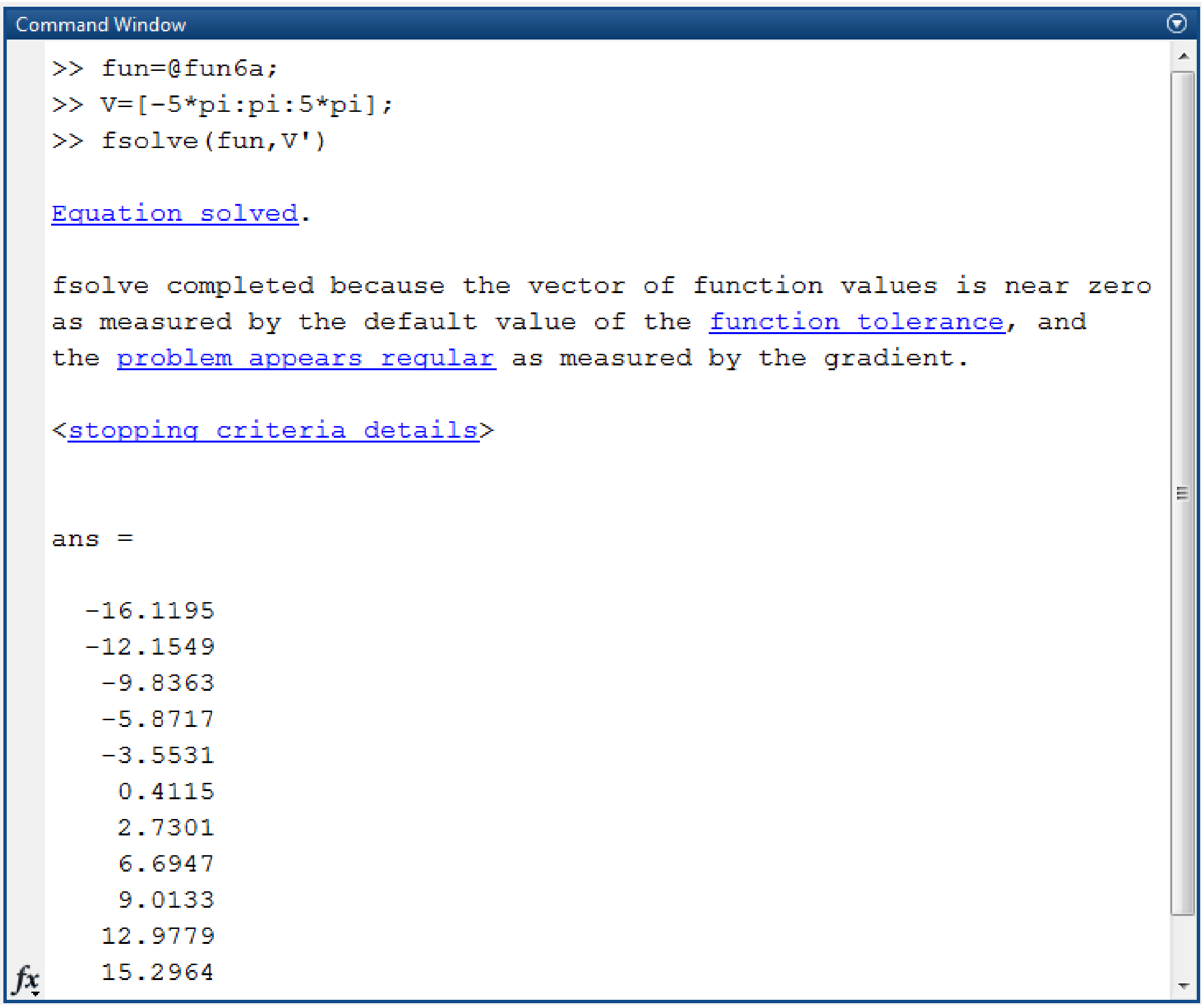
Solve the equation $\sin x - 0.4 = 0$ in the range [-5π; 5π].

solution is shown below



```
function f=fun6a(x)
f=-0.4+sin(x);
```





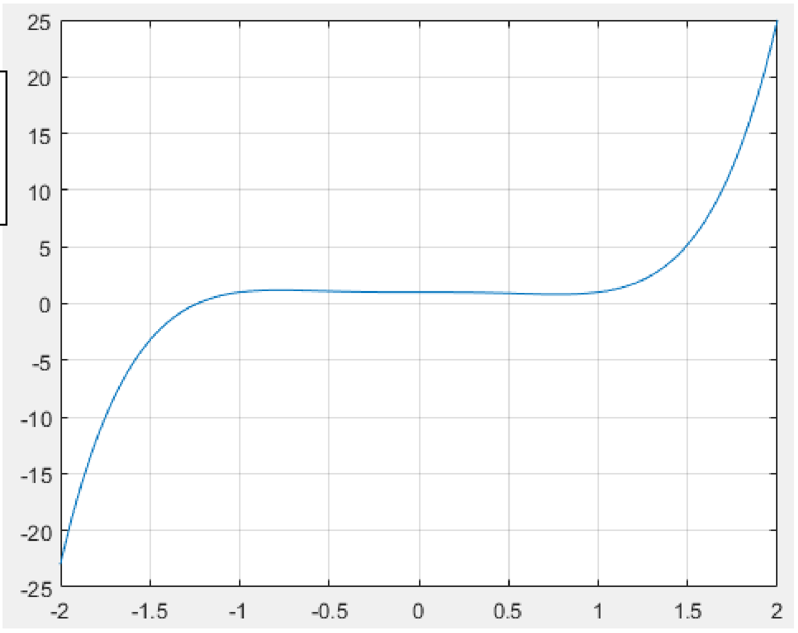
```
>> fun=@fun6a;
>> V=[-5*pi:pi:5*pi];
>> fsolve(fun,V')

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


ans =

  -16.1195
  -12.1549
   -9.8363
   -5.8717
   -3.5531
    0.4115
    2.7301
    6.6947
    9.0133
   12.9779
   15.2964
```

Problem 5.7.

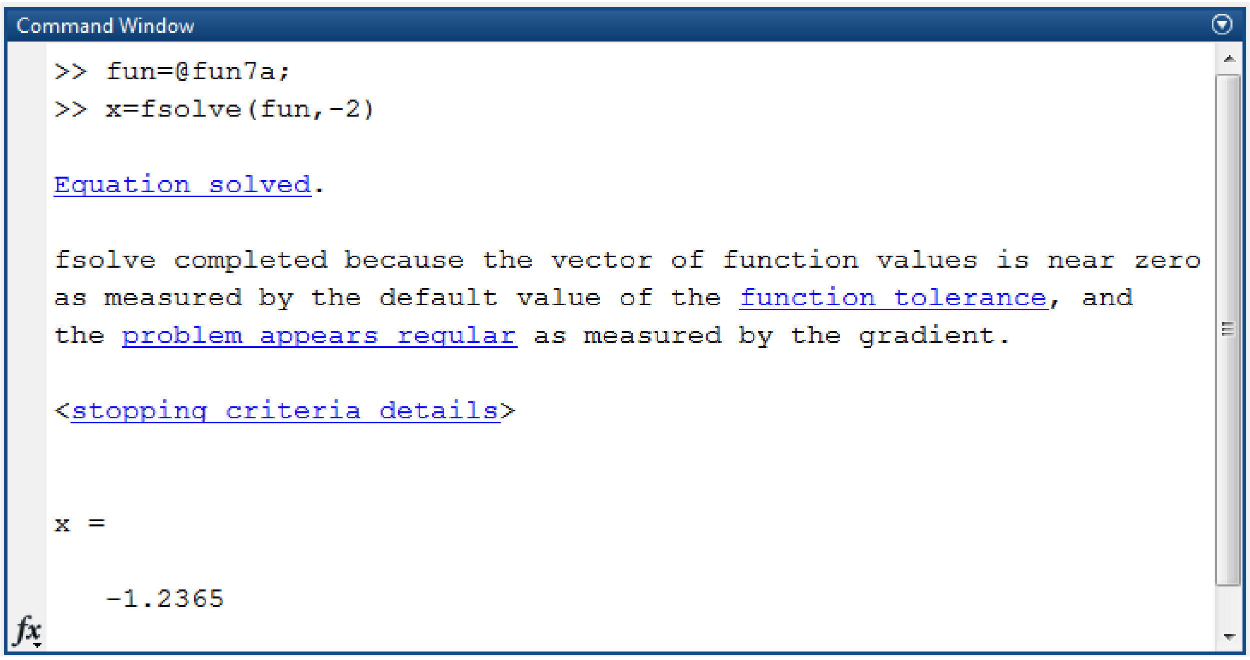
Solve the equation

$y(x) = 0$, if $y(x) = x^5 - x^3 + 1$

```
x=-2:0.01:2;
f=x.^5-x.^3+1;
plot(x,f);
grid on;
```



solution is shown below.



Command Window

```
>> fun=@fun7a;
>> x=fsolve(fun,-2)

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


x =

   -1.2365
```

As you can see, the given equation has a real root and imaginary roots. Therefore, finding the roots of all the best to use the command **roots**.

***The system of equations***

If given m equations in n unknowns, and want to find a sequence of n numbers, which simultaneously satisfy each of the m equations, we speak of a system of equations. To solve systems of equations used in the skill and function ***fsolve (fun,x0)*.**

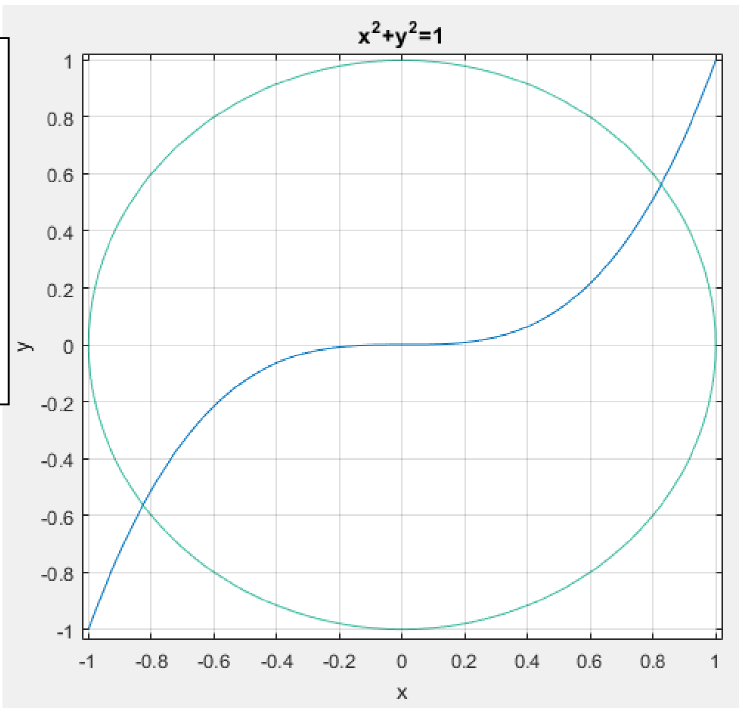
Problem 5.8.

Solve a system of equations

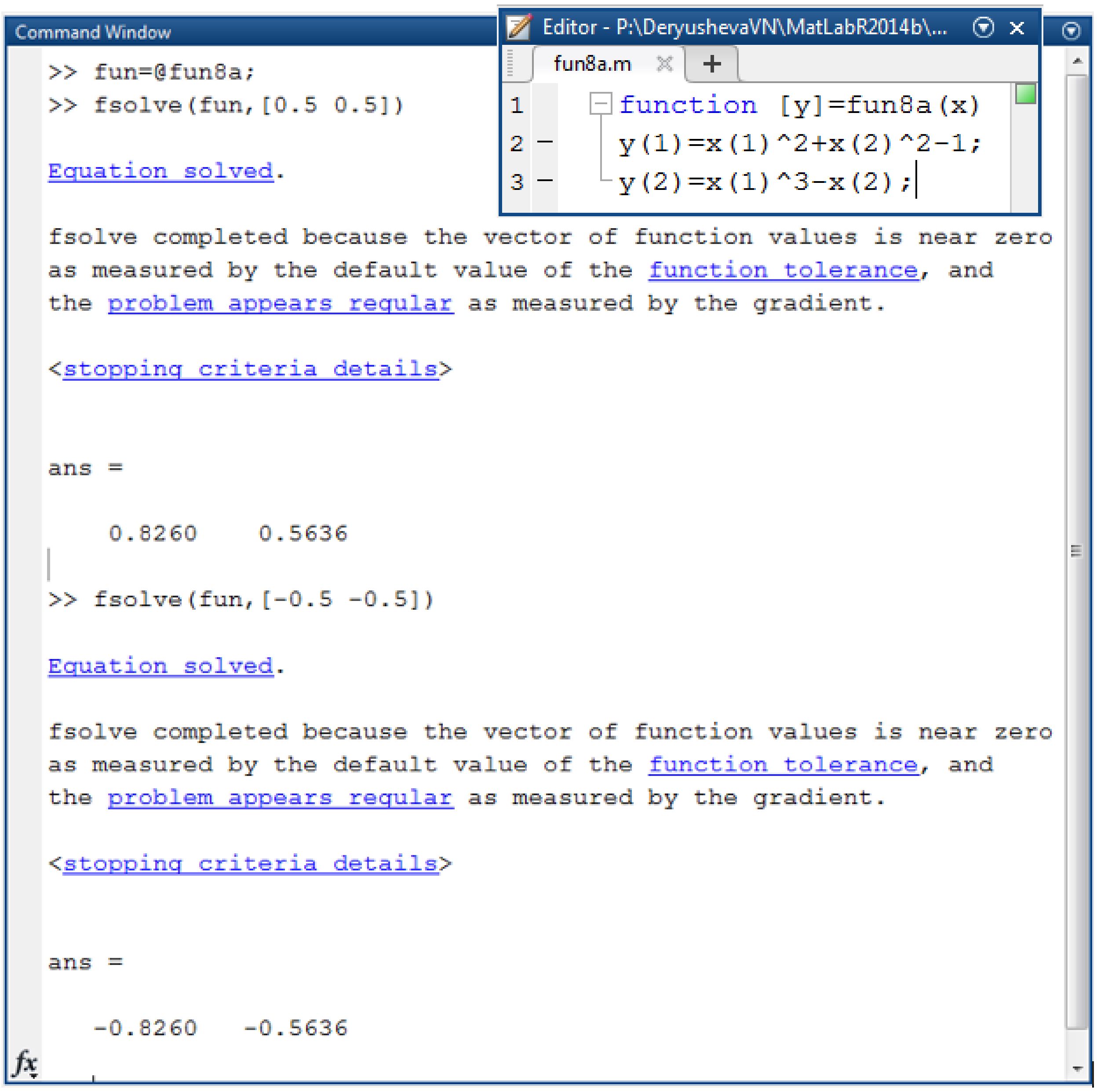
$$\begin{cases} x^2 + y^2 = 1 \\ x^3 - y = 0 \end{cases}.$$

The graphic solution of the system shows that it has two pairs of roots.

```
x=-1:0.001:1;
y=x.^3;
ezplot('x^2+y^2=1');
hold on;
plot(x, y);
grid on;
hold off;
```



The circle and the parabola intersect at points [0: 8; 0: 6], and [0: 8; 0: 6]. These values are approximate. In order for clarify the roots of the functio n we use **fsolve**, before that we define the system using the file-function:



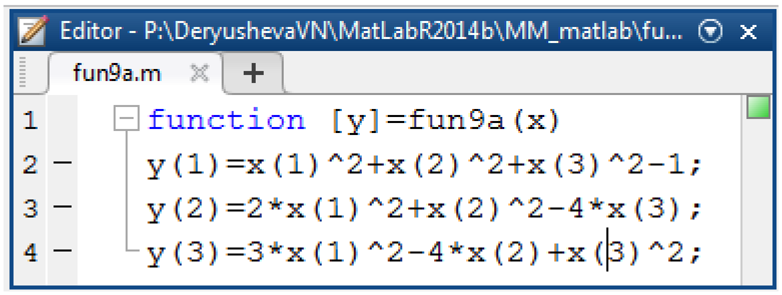
```
function [y]=fun8a(x)
y(1)=x(1)^2+x(2)^2-1;
y(2)=x(1)^3-x(2);
```

```
>> fun=@fun8a;
>> fsolve(fun,[0.5 0.5])

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


ans =

    0.8260    0.5636

>> fsolve(fun,[-0.5 -0.5])

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


ans =

   -0.8260   -0.5636
```

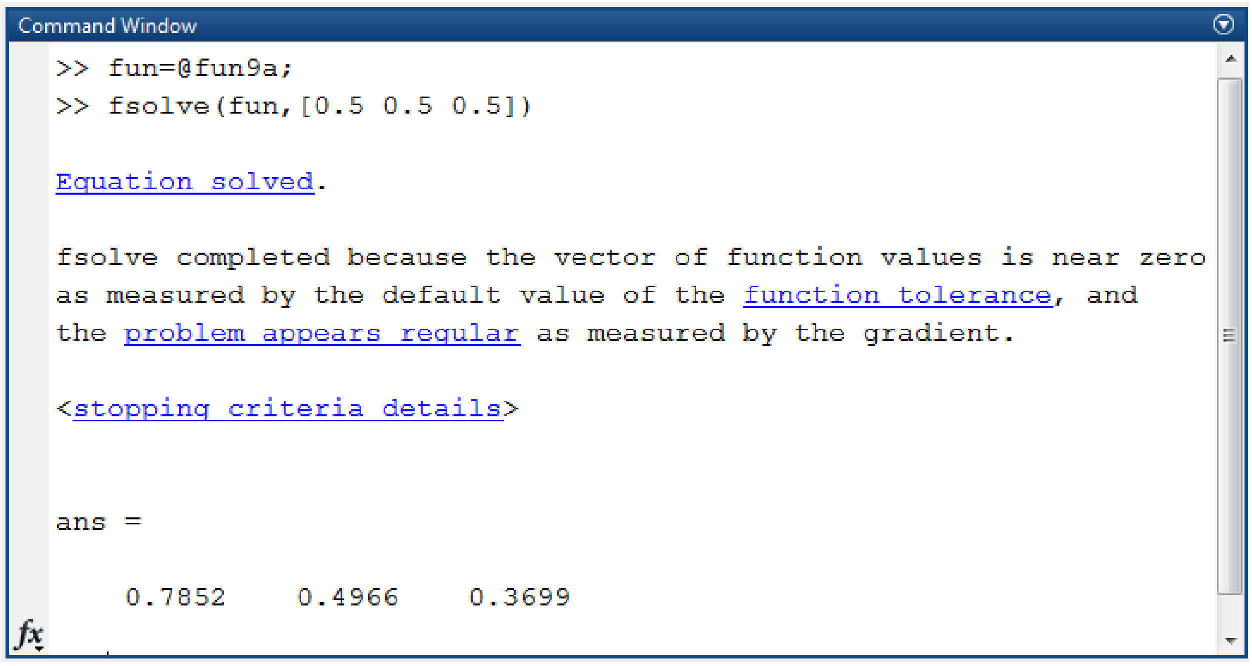
Problem 5.9.

Solve a system of equations

$$\begin{cases} x^2+y^2+z^2=1 \\ 2x^2+y^2-4z=0 \\ 3x^2-4y+z^2=0 \end{cases}$$



```
function [y]=fun9a(x)
y(1)=x(1)^2+x(2)^2+x(3)^2-1;
y(2)=2*x(1)^2+x(2)^2-4*x(3);
y(3)=3*x(1)^2-4*x(2)+x(3)^2;
```



```
>> fun=@fun9a;
>> fsolve(fun,[0.5 0.5 0.5])

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>



ans =

    0.7852    0.4966    0.3699
```