# Software Tools Lab 6

Roll No.: U18CO021

## Installation

This assignment requires *symbolic* package to be installed in octave. My system is Linux.

**Run in octave:**

```
pkg install -forge symbolic
pkg load symbolic
```

**Run in terminal:** (Requires python installed)

```
$ pip install --user sympy
$ pip install --user mpmath
```

## Solutions

### A. Solve the following quadratic equation in Matlab and display their roots.

1) $x^2 - 7x + 12 = 0$

```
pkg load symbolic;
clc;
clear all;

syms x;
eqn = x^2 - 7*x + 12 == 0;

S = solve(eqn);

ans = double(S);
ans
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans =

   3
   4

>>
```

2) $(x-3)^2(x-7)=0$

```
pkg load symbolic;
clc;
clear all;

syms x;
eqn = ((x - 3)^2)*(x - 7) == 0;

S = solve(eqn);

ans = double(S);
ans
```

Command Window

```
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans =

   3
   7

>>
```

3) $x^4-7x^3+3x^2-5x+9=0$

```
pkg load symbolic;
clc;
clear all;

syms x;
eqn = x^4 - 7*x^3 + 3*x^2 - 5*x + 9 == 0;

S = solve(eqn);

ans = double(S);
ans
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans =

   6.63040 + 0.00000i
  -0.34509 - 1.07784i
  -0.34509 + 1.07784i
   1.05978 + 0.00000i

>>
```

4) $6x^2 + 11x - 35 = 0$

```
pkg load symbolic;
clc;
clear all;

syms x;
eqn = 6*x^2 + 11*x - 35 == 0;

S = solve(eqn);

ans = double(S);
ans
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans =

  -3.5000
   1.6667

>>
```

5) $(x-2)^2 - 12 = 0$

```
pkg load symbolic;
clc;
clear all;

syms x;
eqn = (x - 2)^2 -12 == 0;

S = solve(eqn);
```

```
ans = double(S);
ans
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans =

  -1.4641
   5.4641

>>
```

6) $6x^5 - 41x^4 + 97x^3 - 97x^2 + 41x - 6 = 0$

```
pkg load symbolic;
clc;
clear all;

syms x;
eqn = 6*x^5 - 41*x^4 + 97*x^3 - 97*x^2 + 41*x - 6 == 0;

S = solve(eqn);

ans = double(S);
ans
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans =

   0.33333
   0.50000
   1.00000
   2.00000
   3.00000

>>
```

7) *Find the values of* $x, y, z$ *of the equations* $x+y+z=3, x+2y+3z=4, x+4y+9z=6$

```
pkg load symbolic;
clc;
clear all;
```

```
syms x y z;
eqn = [x + y + z == 3, x + 2*y + 3*z == 4, x + 4*y + 9*z == 6];

S = solve(eqn, [x y z]);

x = double(S.x);
y = double(S.y);
z = double(S.z);
x
y
z
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
x =  2
y =  1
z = 0
>>
```

8) *For* $f(x) = 8x^8 - 7x^7 + 12x^6 - 5x^5 + 8x^4 + 13x^3 - 12x + 9$

*compute* $f(2)$, *roots of* $f(x)$ *and plot f or* $0\ 20$

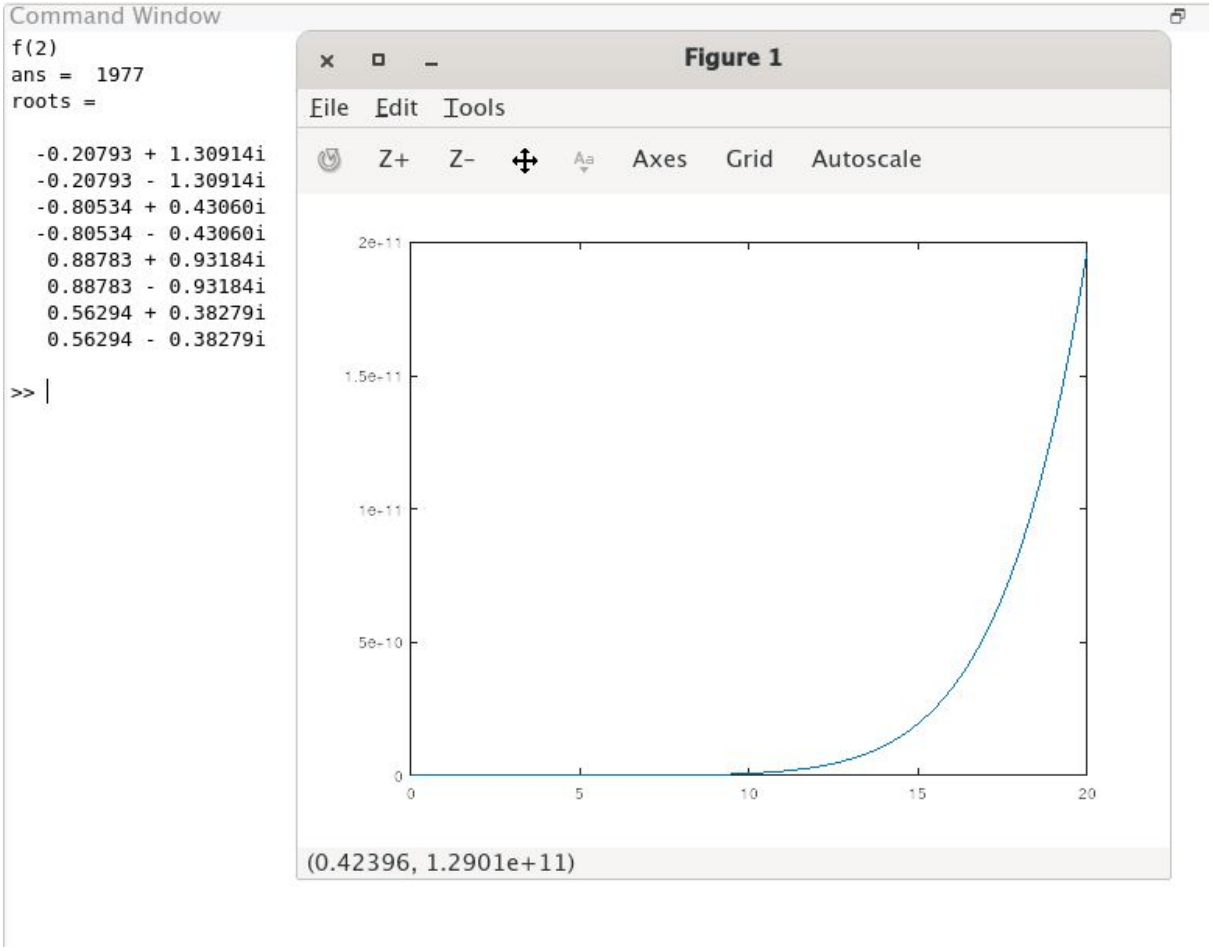
```
clc;
clear all;

p = [8 -7 12 -5 8 13 0 -12 9];

disp("f(2)");
polyval(p, 2)

roots = roots(p)

t = [0:0.1:20];
plot(t, polyval(p, t))
```




**B. Solve the following equation in Matlab.**

1) $5x + 9y = 5;\ 3x - 6y = 4$

```
pkg load symbolic;
clc;
clear all;

syms x y;
eqn = [5*x + 9*y == 5, 3*x - 6*y == 4];

S = solve(eqn, [x y]);

x = double(S.x);
y = double(S.y);
x
y
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
x =  1.1579
y = -0.087719
>>
```

2) $x+3y-2z=5;\ 3x+5y+6z=7;\ 2x+4y+3z=8$

```
pkg load symbolic;
clc;
clear all;

syms x y z;
eqn = [x + 3*y - 2*z == 5, 3*x + 5*y + 6*z == 7, 2*x + 4*y + 3*z == 8];

S = solve(eqn, [x y z]);

x = double(S.x);
y = double(S.y);
z = double(S.z);
x
y
z
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
x = -15
y =  8
z =  2
>>
```

3) $7x+5y-3z=16\ \ 3x-5y+2z=-8\ \ 5x+3y-7z=0$

```
pkg load symbolic;
clc;
clear all;

syms x y z;
eqn = [7*x + 5*y - 3*z == 16, 3*x - 5*y + 2*z == -8, 5*x + 3*y - 7*z ==
0];

S = solve(eqn, [x y z]);

x = double(S.x);
```

```
y = double(S.y);
z = double(S.z);
x
y
z
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
x =  1
y =  3
z =  2
>>
```

4) $3x + 2y = 16;\ 7x + y = 19$

```
pkg load symbolic;
clc;
clear all;

syms x y;
eqn = [3*x + 2*y == 16, 7*x + y == 19];

S = solve(eqn, [x y]);

x = double(S.x);
y = double(S.y);
x
y
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
x =  2
y =  5
>>
```

5) $4x + 3y = -2;\ 8x - 2y = 12$

```
pkg load symbolic;
clc;
clear all;

syms x y;
eqn = [4*x + 3*y == -2, 8*x - 2*y == 12];
```

```
S = solve(eqn, [x y]);

x = double(S.x);
y = double(S.y);
x
y
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
x =  1
y = -2
>>
```

**C. Factorize and simplify the following Algebraic equation**

1) $x^2 - y^2$

```
pkg load symbolic;
clc;
clear all;

syms x y;
factor(x^2 - y^2)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym) (x - y)·(x + y)
>>
```

2) $x^3 + y^3$

```
pkg load symbolic;
clc;
clear all;

syms x y;
factor(x^3 + y^3)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym)

          ⎛ 2          2⎞
  (x + y)⋅⎝x  - x⋅y + y ⎠

>>
```

3) $(x^4 - 16)/(x^2 - 4)$

```
pkg load symbolic;
clc;
clear all;

syms x y;
factor((x^4 - 16)/(x^2 - 4))
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym)

   2
  x  + 4

>>
```

4) $x^4 + y^4$

```
pkg load symbolic;
clc;
clear all;

syms x y;
factor(x^4 + y^4)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym)

   4    4
  x  + y

>>
```

5) $x^5 - y^5$

```
pkg load symbolic;
clc;
clear all;

syms x y;
factor(x^5 - y^5)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym)

                ⎛ 4    3      2  2      3    4⎞
  (x - y)·      ⎝x  + x ·y + x ·y  + x·y  + y ⎠

>>
```

**D. Find the limit of following functions**

1) $\lim_{x \to 0} \frac{x^3+5}{x^4+7}$

```
pkg load symbolic;
clc;
clear all;

syms x;
f = (x^3 + 5)/(x^4 + 7);

limit(f, x, 0)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym) 5/7
>>
```

2) $\lim_{x \to 1} \frac{x-3}{x-1}$

```
pkg load symbolic;
clc;
clear all;

syms x;
f = (x - 3)/(x - 1);

limit(f, x, 1)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym) -∞
>>
```

3) $\lim_{x \to 1} \frac{1-\sqrt{x}}{1-x}$

```
pkg load symbolic;
clc;
clear all;

syms x;
f = (1 - x^(0.5))/(1 - x);

limit(f, x, 0)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
warning: passing floating-point values to sym is dangerous, see "help sym"
warning: called from
    double_to_sym_heuristic at line 50 column 7
    sym at line 379 column 13
    mpower at line 76 column 5
    qd3 at line 6 column 3
ans = (sym) 1
>>
```

4) $\lim_{x \to 0} \frac{\sin 5x}{3x}$

```
pkg load symbolic;
clc;
clear all;

syms x;
f = (sin(5*x))/(sin(3*x));

limit(f, x, 0)
```

```
Command Window
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
ans = (sym) 5/3
>>
```

**E. Show that limit of given function does not exist using left and right sided limits and also plot the graph for it.**

```
pkg load symbolic;
clc;
clear all;

syms x;
f = (x - 3)/abs(x - 3);

left_limit = limit(f, x, 3, 'left')
right_limit = limit(f, x, 3, 'right')
left_limit == right_limit

t = [-10:0.1:10];
y = (t - 3)./abs(t - 3);

plot(t, y)
axis([-10 10 -2 2])
```

Command Window

```
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
left_limit = (sym) -1
right_limit = (sym) 1
ans = (sym) False

>
```