1.	**Calculator**	(2/4)+(4*1)-5.5	-->2/4+4*1-5.5 ans = - 1.
2.	**Square root**	sqrt(9)	-->sqrt(9) ans = 3.
3.	**Important numbers** (π, e, i)	%pi, %e, %i	-->%pi, %e, %i %pi = 3.1415927 %e = 2.7182818 %i = i
4.	**Exponential (pl.: e^2)**	exp(2) *or* %e^2	-->exp(2) ans = 7.3890561
5.	**Variable defining**	a=3	-->a=3 a = 3.
6.	**Row matrix defining**	A=[1 2 3]	-->A=[1 2 3] A = 1. 2. 3.
7.	**Column matrix defining**	B=[4;5;6]	-->B=[4;5;6] B = 4. 5. 6.

8.	**3x3 matrix defining**	C=[1 2 3;4 3 6;1 8 2]	-->C C = 1. 2. 3. 4. 3. 6. 1. 8. 2.
9.	**Matrix addition**	A+B’	-->A+B' ans = 5. 7. 9.
10.	**Matrix multiplication**	A*B	-->A*B ans = 32.
11.	**Matrix element-wise multiplication**	A.*B’	-->A.*B' ans = 4. 10. 18.
12.	**Transpose of a matrix**	A’	-->A' ans = 1. 2. 3.
13.	**c_{21} element of $\underline{\underline{C}}$ matrix**	C(2,1)	-->C(2,1) ans = 4.
14.	**3rd row of $\underline{\underline{C}}$ matrix**	C(3,:)	-->C(3,:) ans = 1. 8. 2.
15.	**1st column of $\underline{\underline{C}}$ matrix**	C(:,1)	-->C(:,1) ans = 1. 4. 1.

16.	**1st and 2nd columns of $\underline{\underline{C}}$ matrix**	C(:,1:2)	`-->C(:,1:2)`
			`ans =`
			`1. 2.`
			`4. 3.`
			`1. 8.`

17.	**2nd and 3rd rows of $\underline{\underline{C}}$ matrix**	C(2:3,:)	`-->C(2:3,:)`
			`ans =`
			`4. 3. 6.`
			`1. 8. 2.`

18.	**Add a new row to $\underline{\underline{C}}$ matrix**	D=[C; [0 0 0]]
		or
		D=[C; [0; 0; 0]']
			`-->D = [C; [0 0 0]]`
			`D =`
			`1. 2. 3.`
			`4. 3. 6.`
			`1. 8. 2.`
			`0. 0. 0.`

19.	**Add a new column to $\underline{\underline{C}}$ matrix**	E=[C, [0;0;0]]
		or
		E=[C, [0 0 0]']
			`-->E = [C, [0; 0; 0]]`
			`E =`
			`1. 2. 3. 0.`
			`4. 3. 6. 0.`
			`1. 8. 2. 0.`

20.	**Add new rows and new columns to $\underline{\underline{C}}$ matrix**	C=[[[0 0 0;0 0 0]; C(1,:)], [0; 0; 0], [0; 0; 0]]	`-->C=[[[0 0 0;0 0 0]; C(1,:)], [0; 0; 0], [0; 0; 0]]`
			`C =`
			`0. 0. 0. 0. 0.`
			`0. 0. 0. 0. 0.`
			`1. 2. 3. 0. 0.`

21.	**Matrix determinant**	det(C)	`-->det(C)`
			`ans =`
			`41.`

22.	**Matrix inverse**	inv(C)	`-->inv(C)`
			`ans =`
			`- 1.0243902 0.4878049 0.0731707`
			`- 0.0487805 - 0.0243902 0.1463415`
			`0.7073171 - 0.1463415 - 0.1219512`

23.	$\underline{\underline{C}} \cdot \underline{\underline{C}}^{-1} = \underline{\underline{E}}$	C*inv(C)	`-->C*inv(C)`
			`ans =`
			`1. - 5.551D-17 - 1.110D-16`
			`0. 1. - 2.220D-16`
			`2.220D-16 0. 1.`

24.	**Diagonal matrix defining**	x=[1 2 3] A=diag(x)	-->x=[1 2 3] x = 1. 2. 3. -->A=diag(x) A = 1. 0. 0. 0. 2. 0. 0. 0. 3.
25.	**Diagonal matrix defining**	x=[1 2 3] B=diag(x,-1)	-->x=[1 2 3] x = 1. 2. 3. -->B=diag(x,-1) B = 0. 0. 0. 0. 1. 0. 0. 0. 0. 2. 0. 0. 0. 0. 3. 0.
26.	**Copy one diagonal of a matrix to an other matrix**	y=diag(B,1)	B = 0. 0. 0. 0. 1. 0. 0. 0. 0. 2. 0. 0. 0. 0. 3. 0. -->y=diag(B,1) y = 0. 0. 0.
27.	**Upper triangular part of $\underline{\underline{C}}$ matrix** ***(triu command)***	C=[1 2 3;4 3 6;1 8 2] triu(C)	-->C=[1 2 3;4 3 6;1 8 2] C = 1. 2. 3. 4. 3. 6. 1. 8. 2. -->triu(C) ans = 1. 2. 3. 0. 3. 6. 0. 0. 2.
28.	**Lower triangular part of $\underline{\underline{C}}$ matrix** ***(tril parancs)***	C=[1 2 3;4 3 6;1 8 2] tril(C)	-->C=[1 2 3;4 3 6;1 8 2] C = 1. 2. 3. 4. 3. 6. 1. 8. 2. -->tril(C) ans = 1. 0. 0. 4. 3. 0. 1. 8. 2.

29.	**Vector cross product**	x=[1 2 3] y=[-1 0 1] axb=cross(x,y)	-->x=[1 2 3] x = 1. 2. 3. -->y=[-1 0 1] y = - 1. 0. 1. -->axb=cross(x,y) axb = 2. - 4. 2.
30.	**Vector dot product**	x=[1 2 3] y=[-1 0 1] ab=x*y'	-->x=[1 2 3] x = 1. 2. 3. -->y=[-1 0 1] y = - 1. 0. 1. -->ab=x*y' ab = 2.
31.	**Vector dyadic product**	x=[1 2 3] y=[-1 0 1] aob=x'*y	-->x=[1 2 3] x = 1. 2. 3. -->y=[-1 0 1] y = - 1. 0. 1. -->aob=x'*y aob = - 1. 0. 1. - 2. 0. 2. - 3. 0. 3.
32.	**Identity matrix / unit matrix defining** **3x3**	eye(3,3)	-->eye(3,3) ans = 1. 0. 0. 0. 1. 0. 0. 0. 1.
33.	**Zero matrix defining**	zeros(2,2)	-->zeros(2,2) ans = 0. 0. 0. 0.

34.	**Matrix of ones**	ones(2,3)	`-->ones(2,3)`
			`ans =`
			`1. 1. 1.`
			`1. 1. 1.`

35.	**Random matrix generate**	rand(3,2)	`-->rand(3,2)`
			`ans =`
			`0.5718074 0.5855573`
			`0.2141770 0.4204123`
			`0.6895462 0.4277572`

36.	**Size of a matrix**	size(C)	`-->size(C)`
			`ans =`
			`3. 3.`

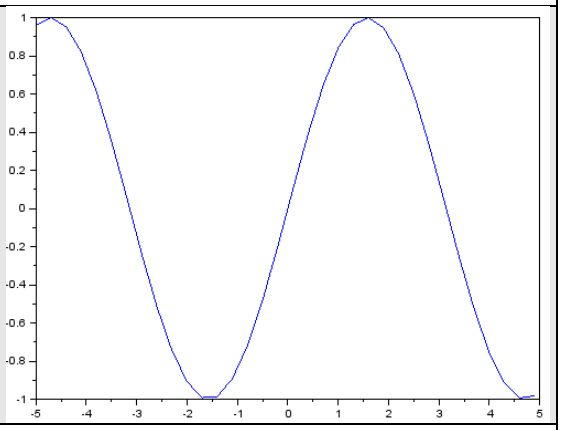
37.	**Lenght of a matrix (the number of elements)**	length(ones(2,3))	`-->length(ones(2,3))`
			`ans =`
			`6.`

38.	**Linear equation system solving**
	$x_1 + 2x_2 + 3x_3 = 3$
	$2x_1 + x_2 - 4x_3 = 9$
	$x_1 + x_2 + x_3 = 2$
		A=[1 2 3;2 1 -4;1 1 1];
		Y=[3;9;2];
		X=A\Y
		or
		X=inv(A)*Y
			`-->A=[1 2 3;2 1 -4;1 1 1];`
			`-->Y=[3;9;2];`
			`-->X=A\Y`
			`X =`
			`- 0.5`
			`4.`
			`- 1.5`

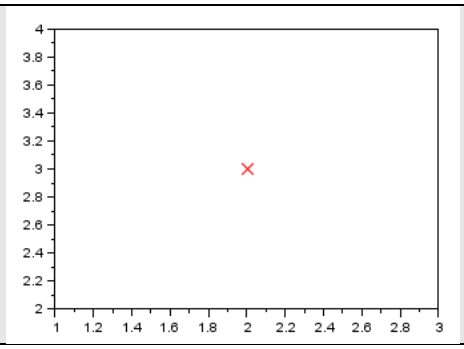
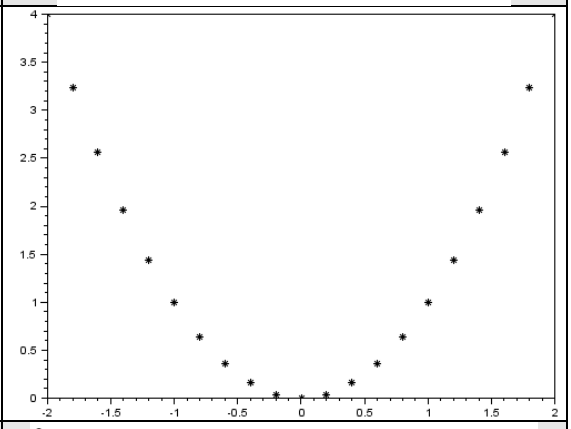
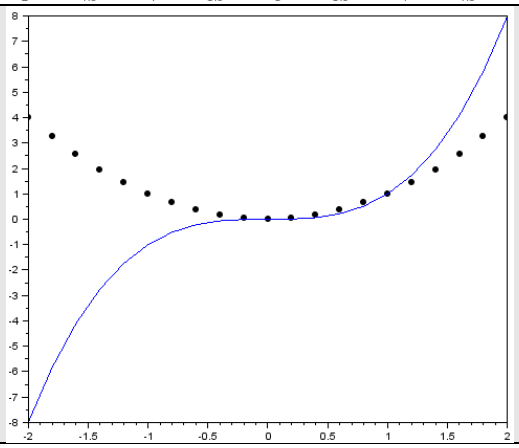
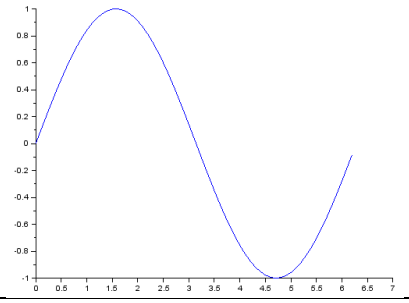
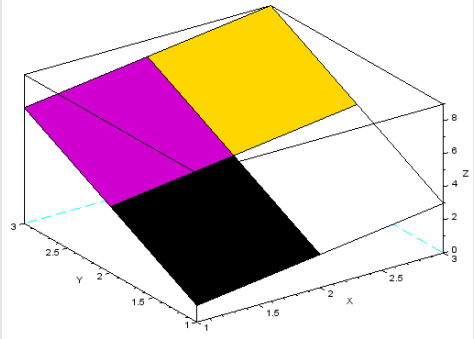
39.	**Series defining *start value:endvalue***	-2:3	`-->-2:3`
			`ans =`
			`- 2. - 1. 0. 1. 2. 3.`

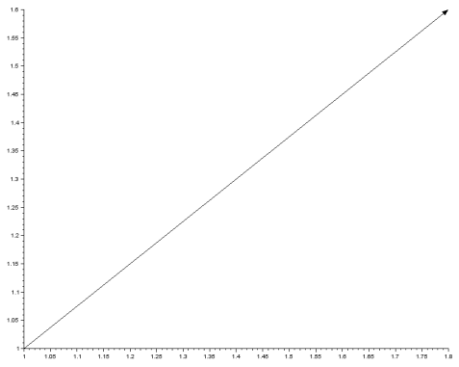
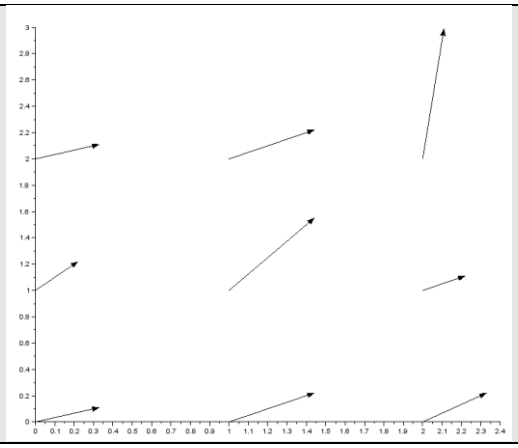
40.	**Series defining *start value: increment:endvalue***	-2:0.5:3	`-->-2:0.5:3`
			`ans =`
			`column 1 to 6`
			`- 2. - 1.5 - 1. - 0.5 0. 0.5`
			`column 7 to 11`
			`1. 1.5 2. 2.5 3.`

41.	**Linspace command *linspace(start value, endvalue, divide up the lenght into)***	linspace(-2,6,3)	-->linspace(-2,6,3) ans = - 2. 2. 6.
42.	**For loop**	for n=1:3 u(n)=n^2 end	-->for n=1:3 -->u(n)=n^2 -->end u = 1. u = 1. 4. u = 1. 4. 9.

43.	**Function defining and plotting**	function y=f(x);
		y=sin(x);
		endfunction
		x=-5:0.3:5;
		plot(x,f)
			

44.	**Function defining**
	$y=\begin{cases} x & ha\ x\geq 2 \\ x+(x-2)^2 & ha\ x<2 \end{cases}$
		function y=f(x)
		if x>=2 then y=x;end;
		if x<2 then y=x+(x-2)^2;end;
		endfunction
			-->function y=f(x)
			-->if x>=2 then y=x;end;
			-->if x<2 then y=x+(x-2)^2;end;
			-->endfunction
			-->f(1.5)
			ans =
			1.75
			-->f(4)
			ans =
			4.

45.	**Plot command** ***parameters:*** ”b”-*blue*,”k”-*black*, ”r”-*red*, ”g”-*green*, ”y”-*yellow* *marks:* ”.”, ”+”, ”o”, ”x”, ”<”, ”v”	plot(2,3,”xr”)	
46.	**Plot command**	x=-2:0.2:2; plot(x,x^2,"*k")	
47.	**Plot command**	x=-2:0.2:2; plot(x,x^2,".k",x,x^3)	
48.	**Plot2d command**	x=[0:0.1:2*%pi]; plot2d(x,sin(x),2)	
49.	**Surf command**	A=[1 2 3;4 5 6;7 8 9] surf(A)	

50.	**Champ command** ***(Vector plotting)***	champ(1,1,4,3)	
51.	**Champ command (Vector plotting)**	champ(0:2,0:2,[3 2 3;4 4 4;3 2 1],[1 2 1;2 5 2;2 1 9])	
52.	**Help** ***(pl.: search for lot2d command)***	help plot2d	
53.	**Delete b vaiable**	clear b	
54.	**Delete all vaiables**	clear	
55.	**Clear console window**	clc	
56.	**Clear graphic window**	clf	
57.	**Jump home**	tohome	
58.	**Run time measurement**	tic *// start a stopwatch timer* toc *//Read the stopwatch timer*	

59.	**Maximum**	max(-3:4)	`--> max(-3:4)`
			`ans =`
			`4.`

60.	**Minimum**	min(-3:4)	`--> min(-3:4)`
			`ans =`
			`-3.`

61.	**Principal Stress (eigenvalues) and principal stress directions (eigenvectors)** $\underline{\underline{F}} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & -8 & 10 \\ 0 & 10 & 12 \end{bmatrix} MPa$	Ftenz=[20 0 0;0 -8 10;0 10 12]; [feszfoirany,fofesz]= spec(Ftenz) // feszfoirany1=feszfoirany(:,1); sqrt(feszfoirany1(1)^2+feszfoirany1(2)^2+feszfoirany1(3)^2)	-->Ftenz=[20 0 0;0 -8 10;0 10 12]; -->[feszfoirany,fofesz]= spec(Ftenz) fofesz = - 12.142136 0. 0. 0. 16.142136 0. 0. 0. 20. feszfoirany = 0. 0. 1. - 0.9238795 0.3826834 0. 0.3826834 0.9238795 0. -->feszfoirany1=feszfoirany(:,1); -->sqrt(feszfoirany1(1)^2+feszfoirany ans = 1.
62.	**matrix command**	W=matrix(11:42,8,4)	-->W=matrix(11:42,8,4) W = 11. 19. 27. 35. 12. 20. 28. 36. 13. 21. 29. 37. 14. 22. 30. 38. 15. 23. 31. 39. 16. 24. 32. 40. 17. 25. 33. 41. 18. 26. 34. 42.
63.	**Remainder after division**	modulo(9,7) modulo(-9,5)	--> modulo(9,7) ans = 2. --> modulo(-9,5) ans = -4.
64.	**Round to nearest integer**	round([1.3,6.5])	--> round([1.3,6.5]) ans = 1. 7.
65.	**Round down to the next integer**	floor([1.3,5.9]) int([1.3,6.8])	--> floor([1.3,5.9]) ans = 1. 5.
66.	**Round up to the next integer**	ceil([1.3,6.8])	--> ceil([1.3,6.8]) ans = 2. 7.