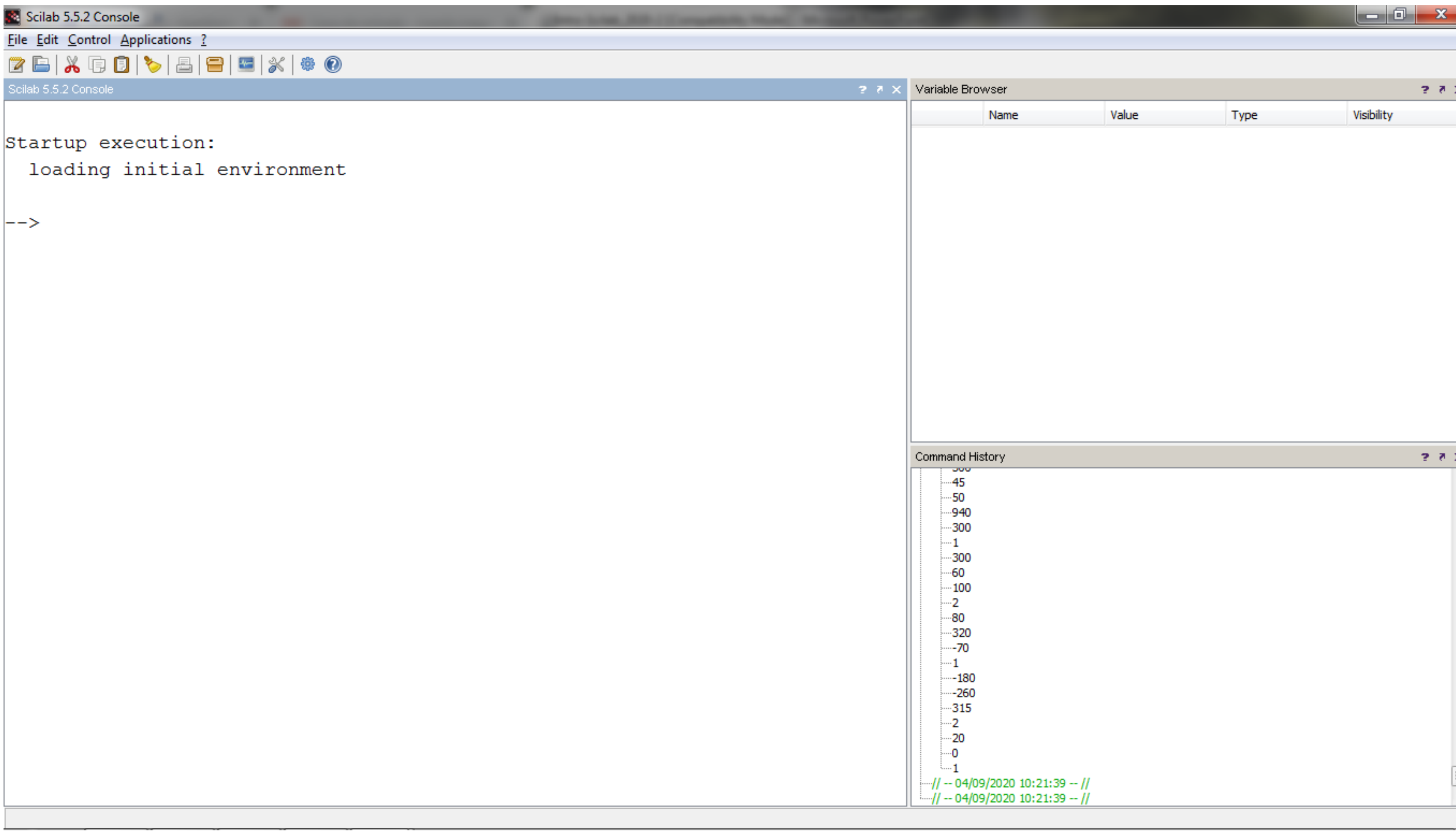
Lecturer: Ricardo Rodrigues Magalhães

ricardorm@ufla.br

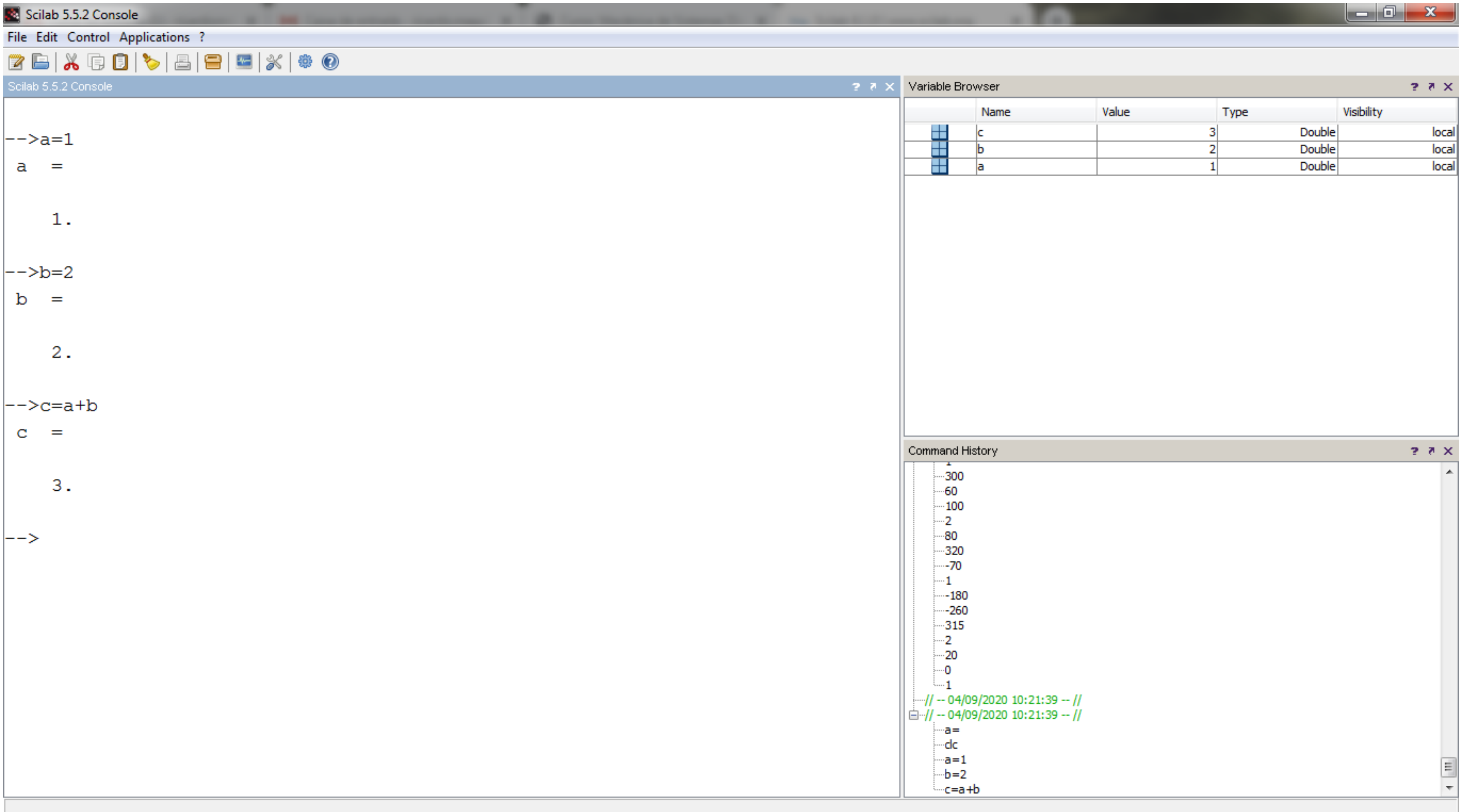
Main objective of the course:
Presenting the Finite Element Method (FEM) and its formulations applied to Scilab.

Assessment methods:
Programming proof (40%)
Exercise lists (30%)
Seminar presentation (30%)

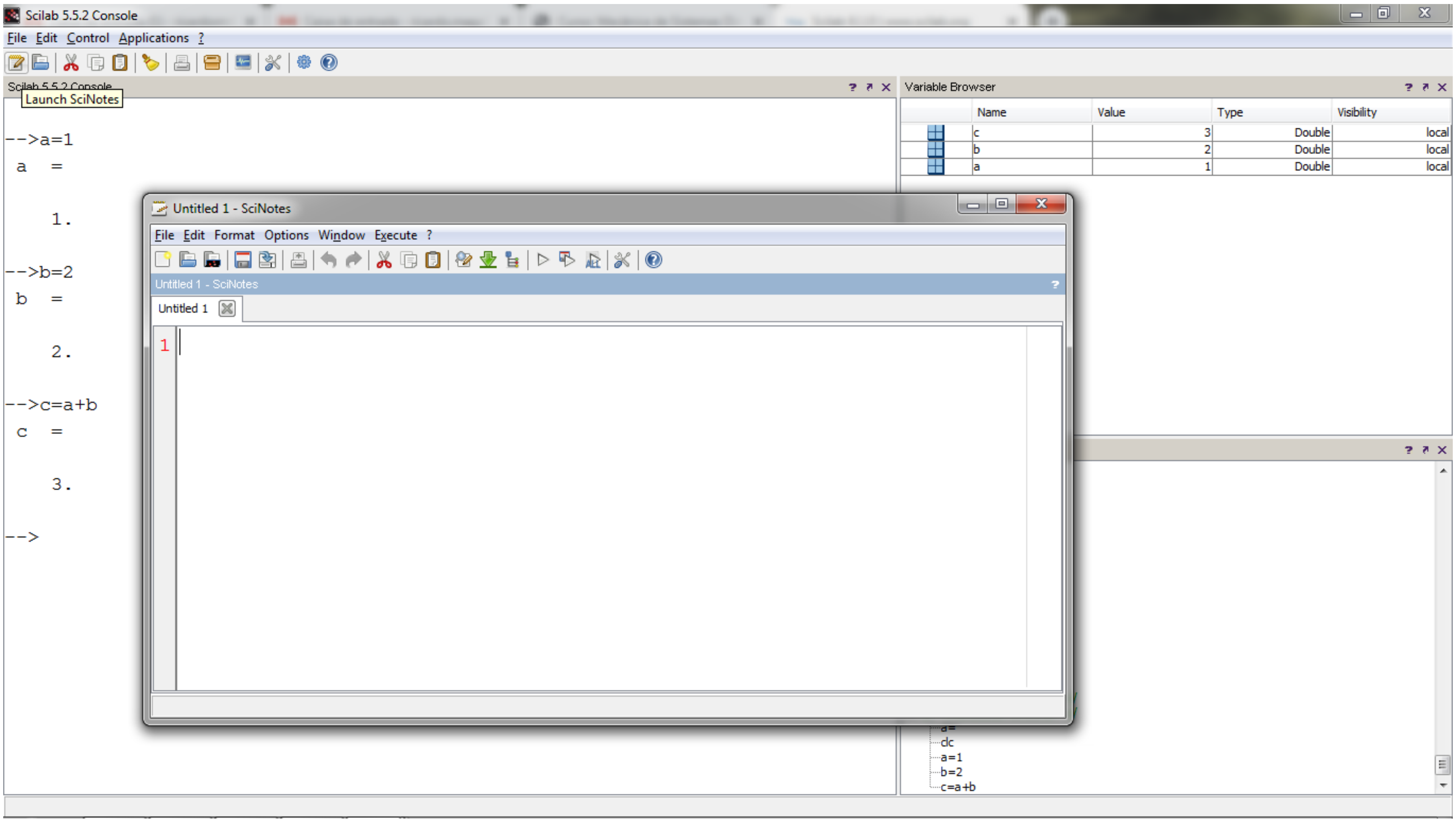
Open source software for numerical computation - Scilab
https://www.scilab.org/download



Assigning values to variables



Files extensions (.sci ou .sce)



Vectors

Declaration:

X = [x1 x2 x3 ...] row vector

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$$

X = [x1;x2;x3;...] column vector

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

Vector transposition: X'

setdiff(A,B): returns values of vector A that are not in B

Exercise:

Given the vectors: A=[2 3 4 6 7] e B = [1 6 3] ;
X = [1;2;3;4;5] e Y = [2;4;6;8;10]

a) Find elements from vector A that are not in B
b) Calculate Z = X + Y;
c) Calculate W1 = A*X and W2 = Y*A;
d) Calculate the transpositions of A and B;
e) Given Z1 = X*Y’ and Z2 = X'*Y, print Z1 and Z2 on screen.

5

Solution

```
clear; clc
A=[2 3 4 6 7];
B = [1 6 3] ;
X = [1;2;3;4;5];
Y = [2;4;6;8;10];
C = setdiff(A,B)
disp(C)
Z = X + Y;
disp(Z)
W1 = A*X; W2 = Y*A;
disp(W1); disp(W2)
At = A' ; Bt = B';
disp(At); disp(Bt)
Z1 = X*Y'; Z2 = X'*Y
disp(Z1); disp(Z2)
```

Vectors declaration

- A = Initial_value: increment: Final_value

Examples:

- A = 1:10;
- B = 1:2:10;
- C = 1:0.2:10;
- D = 10:-1:1;
- E = 1:pi:20;
- F = 20:-2*pi:-10

Vector operations

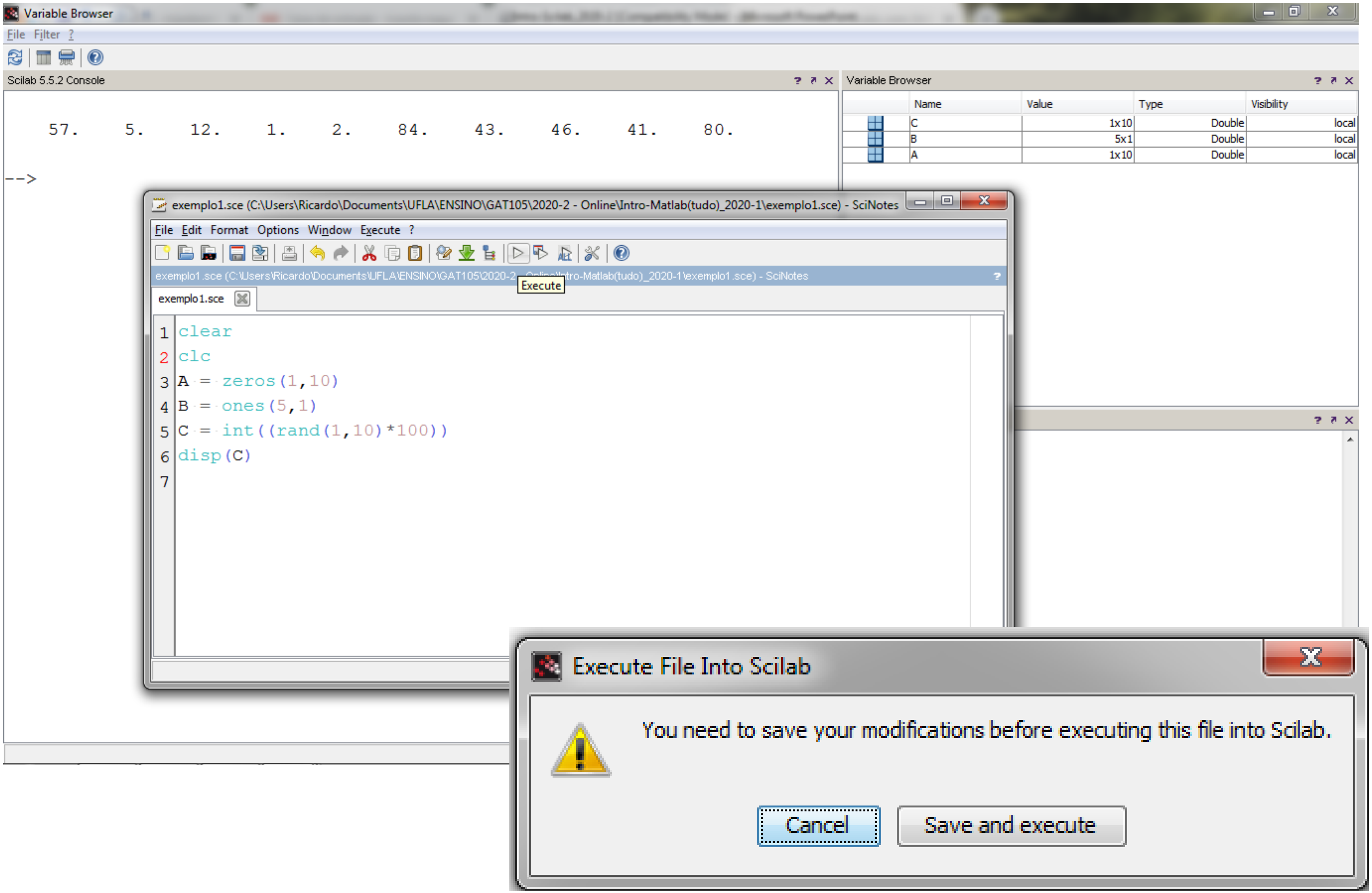
- Dimension: length(x)
- Use “size (A)” to identify the dimensions of the matrix. The largest dimension is given by the command “length(A) ”
- Vectors with all elements equal to one: x = ones(N,1)
- Vectors with all elements equal to zero: x = zeros(N,1)
- Vectors with N random elements between 0 and 1: X = rand(N,1)

  PS: For vectors with N integer elements, use:

  C = int((rand(1,N)*100))

Exercise: Create a row vector (A) with 10 elements equal to zero, a column vector (B) with 5 elements equal to one, and a row vector (C) with 10 random integers between 0 and 100.

Solution


Variable Browser
File Filter ?
Scilab 5.5.2 Console
57. 5. 12. 1. 2. 84. 43. 46. 41. 80.
-->
Variable Browser
Name Value Type Visibility
C 1x10 Double local
B 5x1 Double local
A 1x10 Double local
exemplo1.sce (C:\Users\Ricardo\Documents\UFLA\ENSINO\GAT105\2020-2 - Online\Intro-Matlab(tudo)_2020-1\exemplo1.sce) - SciNotes
File Edit Format Options Window Execute ?
Execute
exemplo1.sce
1 clear
2 clc
3 A = zeros(1,10)
4 B = ones(5,1)
5 C = int((rand(1,10)*100))
6 disp(C)
7
Execute File Into Scilab
You need to save your modifications before executing this file into Scilab.
Cancel
Save and execute

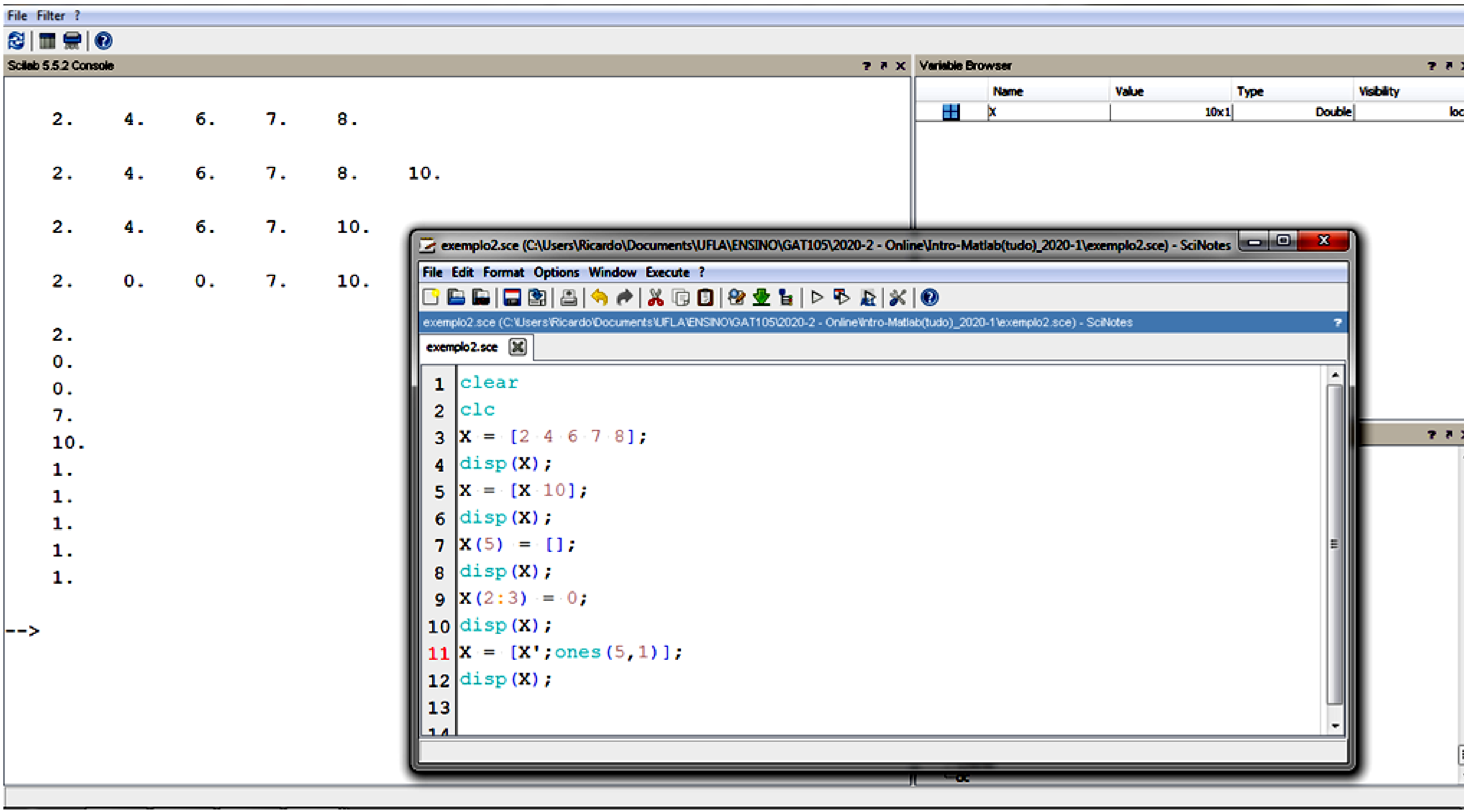
Vector operations

- Delete a given element in the position i: X(i) = []
- Add element i at the end of a vector: X = [X i]
- Access elements between n and m: X(n:m)
- Groups two vectors X and Y: A = [X Y]

Exercise: Given the vector X = [2 4 6 7 8];

- Add the value 10 at the end of a vector X
- Delete the fifth element from the vector X
- Assign number zero to elements between positions 1 and 4
- Group the vectors: X' e C = ones(5,1)

Solution


File Filter ?
Scilab 5.5.2 Console
2. 4. 6. 7. 8.
2. 4. 6. 7. 8. 10.
2. 4. 6. 7. 10.
2. 0. 0. 7. 10.
2.
0.
0.
7.
10.
1.
1.
1.
1.
1.
-->
Variable Browser
Name Value Type Visibility
X 10x1 Double local
exemplo2.sce (C:\Users\Ricardo\Documents\UFLA\ENSINO\GAT105\2020-2 - Online\Intro-Matlab(tudo)_2020-1\exemplo2.sce) - SciNotes
File Edit Format Options Window Execute ?
exemplo2.sce
clear
clc
X = [2 4 6 7 8];
disp(X);
X = [X 10];
disp(X);
X(5) = [];
disp(X);
X(2:3) = 0;
disp(X);
X = [X';ones(5,1)];
disp(X);

Matrices

A general matrix consists of m*n elements arranged in m rows and n columns, according to example below:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Example

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

On Scilab:

M = [1 2 3; 4 5 6; 7 8 9]

To access specific row or column (i):

M(i , :) ou M(: , i)

Operations with matrices

- Arrays with all elements equal to one: A = ones(M,N)
- Arrays with all elements equal to zero: B = zeros(M,N)
- Identity matrix: A = eye(N)

Exercise:

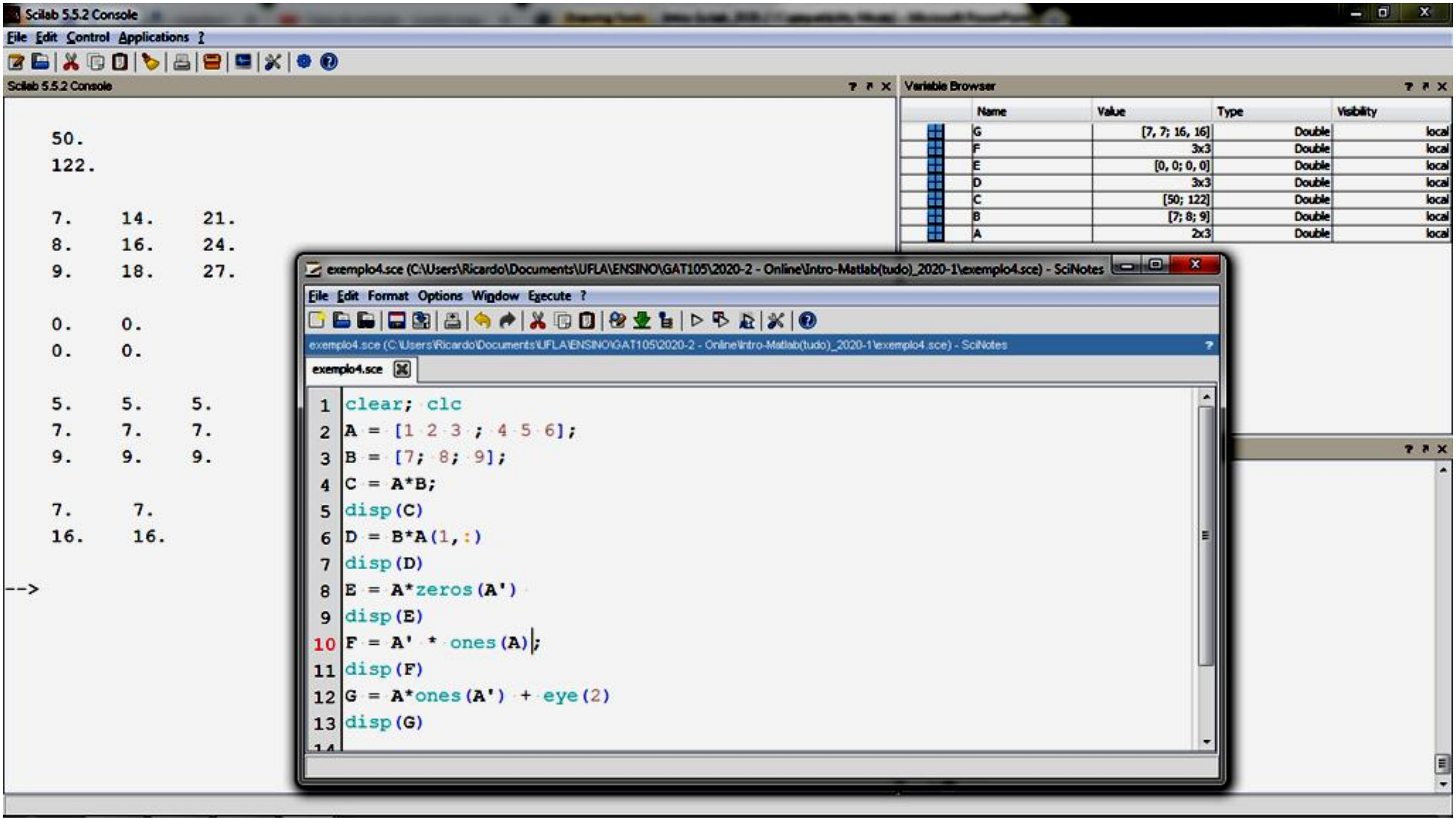
Given the matrices

A = [1 2 3 ; 4 5 6]; B = [7; 8; 9];

Calculate:

- A*B
- B*A(1,:)
- A*zeros(dimension of A')
- A' *ones(dimension of A)
- A*ones(dimension of A') + Identity(2)

Solution


Scilab 5.5.2 Console
File Edit Control Applications ?
Scilab 5.5.2 Console
50.
122.
7. 14. 21.
8. 16. 24.
9. 18. 27.
0. 0.
0. 0.
5. 5. 5.
7. 7. 7.
9. 9. 9.
7. 7.
16. 16.
-->
Variable Browser
Name Value Type Visibility
G [7, 7; 16, 16] Double local
F 3x3 Double local
E [0, 0; 0, 0] Double local
D 3x3 Double local
C [50; 122] Double local
B [7; 8; 9] Double local
A 2x3 Double local
exemplo4.sce (C:\Users\Ricardo\Documents\UFLA\ENSINO\GAT105\2020-2 - Online\Intro-Matlab(tudo)_2020-1\exemplo4.sce) - SciNotes
File Edit Format Options Window Execute ?
exemplo4.sce
clear; clc
A = [1 2 3 ; 4 5 6];
B = [7; 8; 9];
C = A*B;
disp(C)
D = B*A(1,:)
disp(D)
E = A*zeros(A')
disp(E)
F = A' * ones(A);
disp(F)
G = A*ones(A') + eye(2)
disp(G)

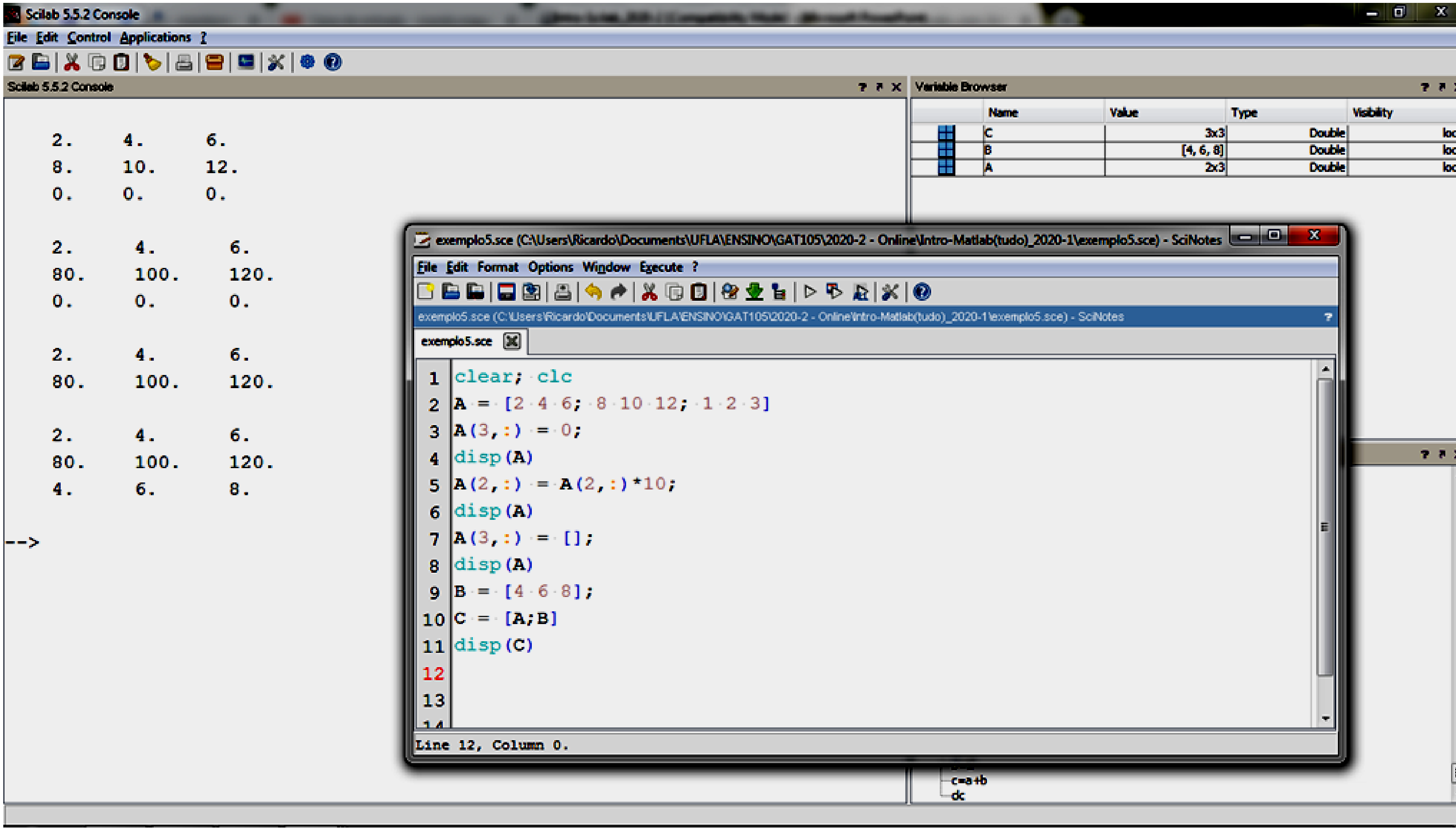
Operations with matrices

- To access specific row i: A(i,:)
- To access specific column j: A(:,j)
- Add a row at the end of the matrix: A = [A; row]
- Add a column at the end of the matrix: A = [A, column]

Exercise: Given the matrix A = [2 4 6; 8 10 12; 1 2 3]

- Assign zero to third line;
- Second row times ten;
- Remove the third row from Matrix A
- Add vector B = [4 6 8] to the last row of matrix A

Solution


Scilab 5.5.2 Console
File Edit Control Applications ?
Scilab 5.5.2 Console
2. 4. 6.
8. 10. 12.
0. 0. 0.
2. 4. 6.
80. 100. 120.
0. 0. 0.
2. 4. 6.
80. 100. 120.
2. 4. 6.
80. 100. 120.
4. 6. 8.
-->
Variable Browser
Name Value Type Visibility
C 3x3 Double local
B [4, 6, 8] Double local
A 2x3 Double local
exemplo5.sce (C:\Users\Ricardo\Documents\UFLA\ENSINO\GAT105\2020-2 - Online\Intro-Matlab(tudo)_2020-1\exemplo5.sce) - SciNotes
File Edit Format Options Window Execute ?
exemplo5.sce
clear; clc
A = [2 4 6; 8 10 12; 1 2 3]
A(3,:) = 0;
disp(A)
A(2,:) = A(2,:)*10;
disp(A)
A(3,:) = [];
disp(A)
B = [4 6 8];
C = [A;B]
disp(C)
Line 12, Column 0.
c=a+b
clc

Operations with Matrices

- Sum: C = A + B
- Multiplication: C = A*B
- Multiplication by a scalar α: B = α*A
- Transpose the matrix: C = A'
- Determinant from a matrix: d = det(A)
- Diagonal matrix: d = diag(A).

Operations with Matrices

Exercise: Since A and B are two square matrices of order 5 with random elements ranging from 1 to 10, calculate:

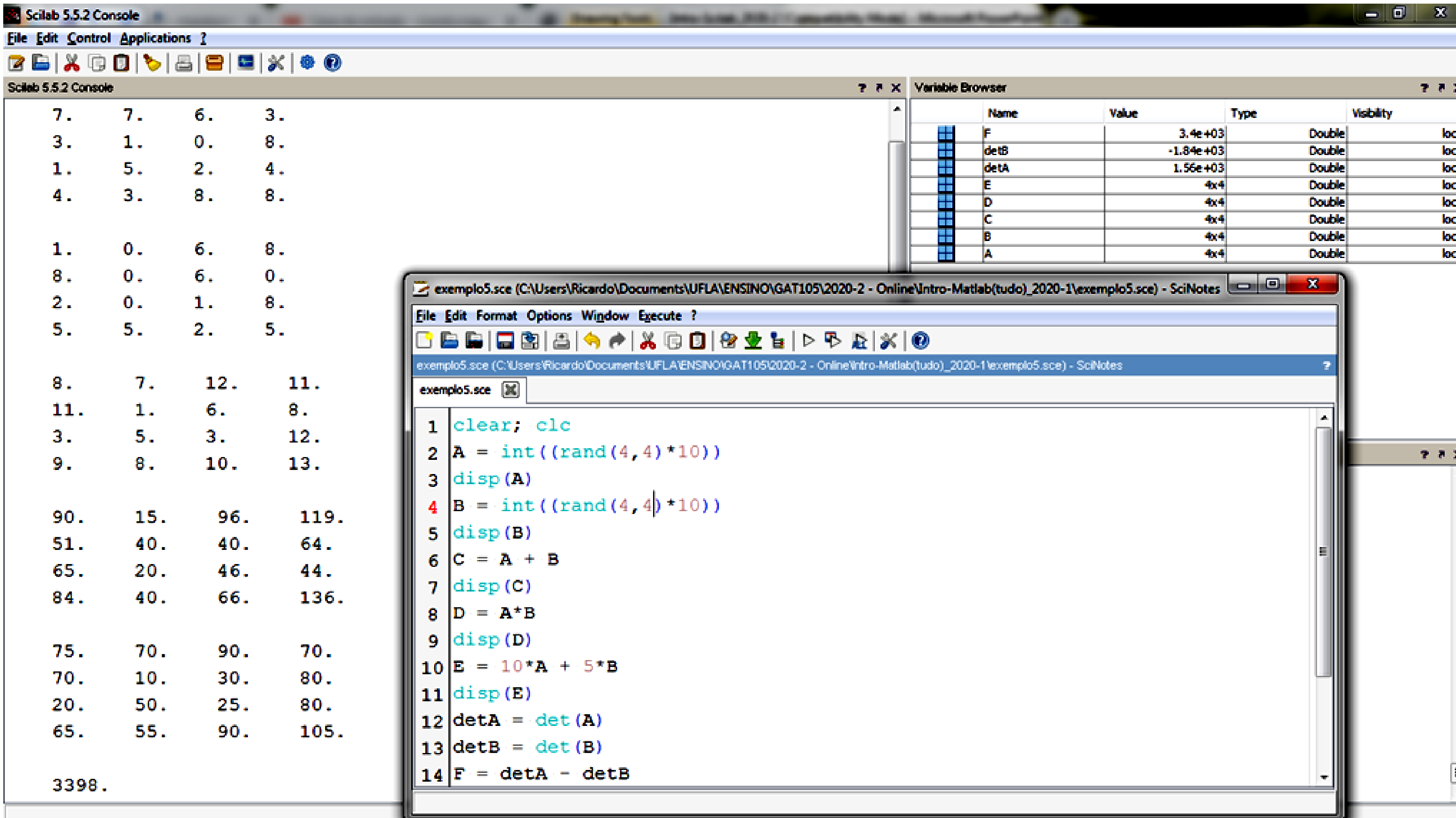
C = A + B

D = A*B’

E = 10*A + 5*B

F = det(A) – det(B)

Solution


Scilab 5.5.2 Console
File Edit Control Applications ?
Scilab 5.5.2 Console
7. 7. 6. 3.
3. 1. 0. 8.
1. 5. 2. 4.
4. 3. 8. 8.
1. 0. 6. 8.
8. 0. 6. 0.
2. 0. 1. 8.
5. 5. 2. 5.
8. 7. 12. 11.
11. 1. 6. 8.
3. 5. 3. 12.
9. 8. 10. 13.
90. 15. 96. 119.
51. 40. 40. 64.
65. 20. 46. 44.
84. 40. 66. 136.
75. 70. 90. 70.
70. 10. 30. 80.
20. 50. 25. 80.
65. 55. 90. 105.
3398.
Variable Browser
Name Value Type Visibility
F 3.4e+03 Double local
detB -1.84e+03 Double local
detA 1.56e+03 Double local
E 4x4 Double local
D 4x4 Double local
C 4x4 Double local
B 4x4 Double local
A 4x4 Double local
exemplo5.sce (C:\Users\Ricardo\Documents\UFLA\ENSINO\GAT105\2020-2 - Online\Intro-Matlab(tudo)_2020-1\exemplo5.sce) - SciNotes
File Edit Format Options Window Execute ?
exemplo5.sce
clear; clc
A = int((rand(4,4)*10))
disp(A)
B = int((rand(4,4)*10))
disp(B)
C = A + B
disp(C)
D = A*B
disp(D)
E = 10*A + 5*B
disp(E)
detA = det(A)
detB = det(B)
F = detA - detB

Matrices: special operators

Considering the linear system equation (Ax = b) in a matrix form, being A the matrix of coefficients, x the vector of the unknowns variables and “b” the vector of the independent terms.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}_{nxn} \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

Operator \

The solution of the system (Ax = b) is equal to $x = A^{-1}b$.
This means that it is just to obtain the inverse matrix from A and multiply the result by vector b as example showed below.

$$\begin{cases} 1x + 3y = 5 \\ 3x + 4y = 2 \end{cases}$$

```
A=[1 3;3 4];
b=[5;2];
x=inv(A)*b
```

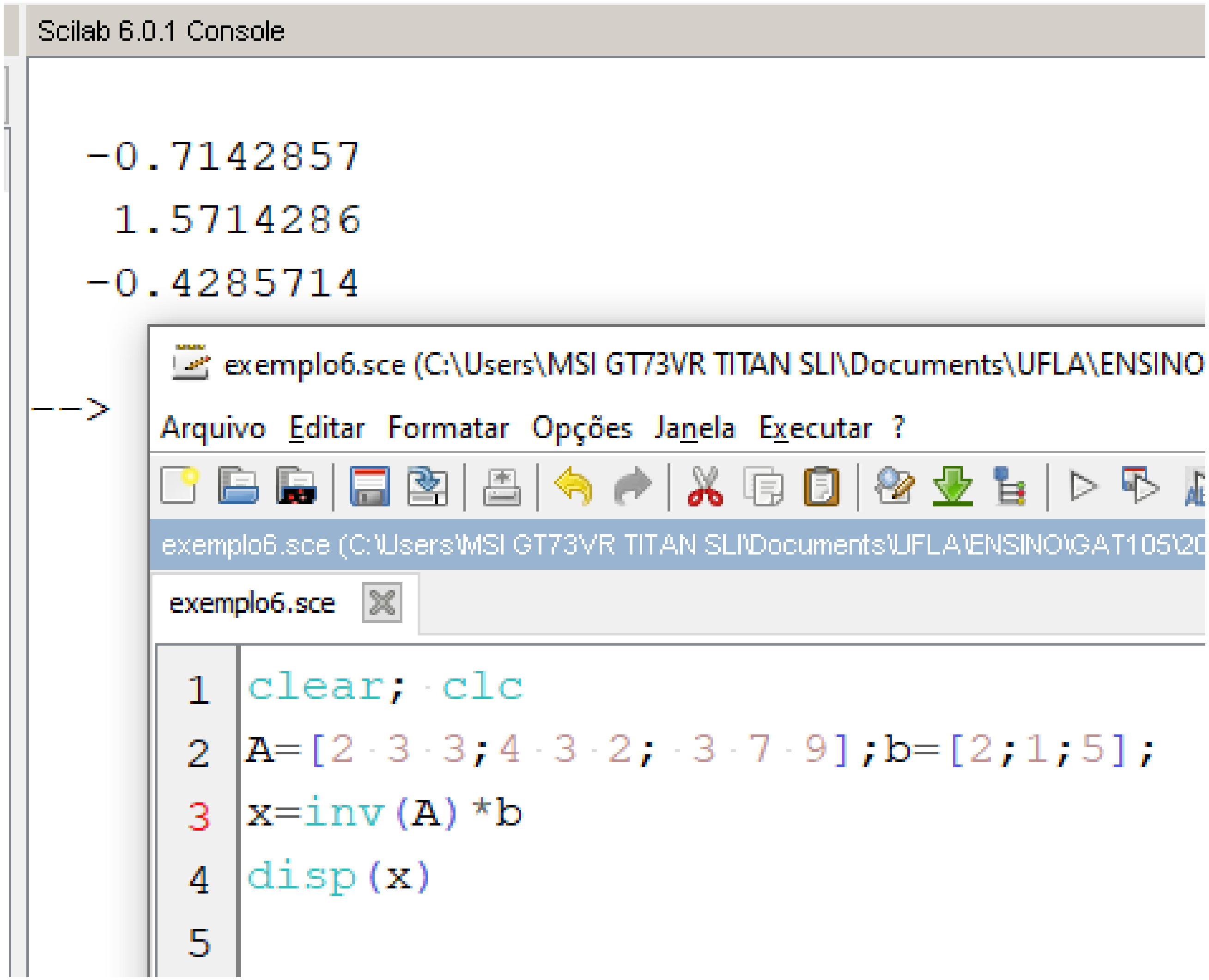
The solution can also be obtained by the “left division” operator whose symbol is \:

x = A\b

Exercise: Solve the linear system:

$$\begin{cases} 2X + 3y + 3z = 2 \\ 4x + 3y + 2z = 1 \\ 3x + 7y + 9z = 5 \end{cases}$$

Solution


Scilab 6.0.1 Console
-0.7142857
1.5714286
-0.4285714
-->
exemplo6.sce (C:\Users\MSI GT73VR TITAN SLI\Documents\UFLA\ENSINO
Arquivo Editar Formatar Opções Janela Executar ?
exemplo6.sce (C:\Users\MSI GT73VR TITAN SLI\Documents\UFLA\ENSINO\GAT105\20
exemplo6.sce
1 clear; clc
2 A=[2 3 3;4 3 2; 3 7 9];b=[2;1;5];
3 x=inv(A)*b
4 disp(x)
5

Sparse matrices

Sparse is a matrix when it presents a lot of elements equal to zero. The sparse matrix is implemented through a set of linked lists that point to non-zero elements.

On Scilab :

```
A = [0 0 1; 2 0 0; 0 3 0]

-->sparse(A)

 ans  =
 (    1,    3)        1.
(    2,    1)        2.
(    3,    2)        3.
```

Operations with sparse matrices

```
A = [0 0 1; 2 0 0; 0 3 0] ;

A= sparse(A);

B = [0 1 0; 2 0 2; 3 0 0];

B = sparse(B);

C = A*B
```

To turn back matrix C in the full form:

```
D= full(C)
```

On scilab:

Scilab 6.0.1 Console

```
(   3,   3) sparse matrix

(   1,   1)          3.
(   2,   2)          2.
(   3,   1)          6.
(   3,   3)          6.

    3.    0.    0.
    0.    2.    0.
    6.    0.    6.

-->
```

exemplo7.sce (C:\Users\MSI GT73VR TITAN SLI\Documents\UFLA\

Arquivo Editar Formatar Opções Janela Executar ?

exemplo7.sce (C:\Users\MSI GT73VR TITAN SLI\Documents\UFLA\ENSINO\GA

exemplo7.sce

```
clear; clc
A = [0 0 1; 2 0 0; 0 3 0] ;
A= sparse(A);
B = [0 1 0; 2 0 2; 3 0 0];
B = sparse(B);
C = A*B
disp(C)
D = full(C)
disp(D)
```

Iteration commands

- The loop for

```
for variable = start : increment : end
    instruction 1
    instruction 2 …
    instruction n
end
```

- Example:

```
a = 0;
for i=1:3
  a = a+i;
end
```

- The loop while

Allowed operators :

- == or = (equals to)
- < (less than)
- > (bigger then)
- <= (less or equal)
- >= (Bigger or equal)
- ~= (different)

- Example:

```
x =1;
while x <= 16;
x = x*2
end
```

if – elseif – else

```
if condition_1
  sequence_1
elseif condition_2
  sequence_2
else
  sequence_3
end
```

- if condition_1 is true, execute sequence_1;
- if condition_1 is false it evaluates condition_2 and so on
- if all conditions are false, execute sequence_3.

Examples:

```
1) x = input('x=');
if x < 0
y = 2*x;
else
   y = x;
end
fprintf('y = %.0f\n',y);
```

```
2) x = input('x=');
if x < 0
  y = -x
elseif x == 1
  y = x
elseif x == 2
  y = 2*x
else
  y = 5*x
end
```

** Results output on the screen (example):

```
// Given the matrix F with values in Newton (N)
F = [ 2 7 -4 0 3 ];
printf('The lowest F value is % .2f N and the highest F value is %.2f N\n ',min(F),
max(F));  //.2f represents the number of decimals place, in this case, two.
```

Functions

```
function[y1,...,yn] = function_name(x1,...,xm)
        instruction_1
        instruction_2
              ...
         instruction_n
endfunction
```

where:

- x1,...,xm are the input arguments,
- y1,...,yn are output arguments and
- instrucão_1, ..., instrucao_n are the instructions performed by the function.

Functions

Example: Define a function that calculates values of x and y from two inputs (a e b).

```
clear; clc

a = input('type a value for the variable a = ');
b = input('type a value for the variable b = ');

function [x, y]=myfct(a, b)
  x=a+b
  y=a-b
endfunction

[x,y]=myfct(a,b)
printf('The value of x is % .0f N and the value of y is %.0f N\n ',x, y)
```

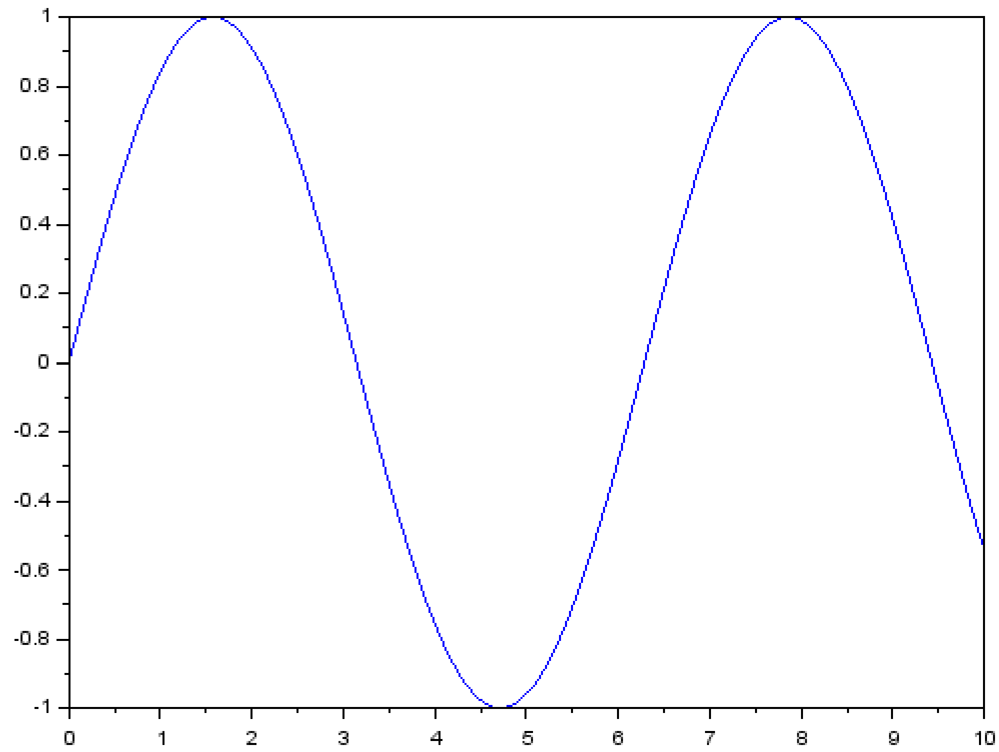
Exercise

Write an algorithm that calculate the roots of a second degree equation using functions. At the main, user must input the coefficients and the function return solutions for delta = 0, delta> 0 and delta <0.

PS: When delta < 0, insert two random values for x_1 and x_2 and omit from the main program screen in order to avoid output errors for this specific case.

Graphics

```
t=0:0.01:10;
y=sin(t);
plot(t,y)
```



Lines and colors examples

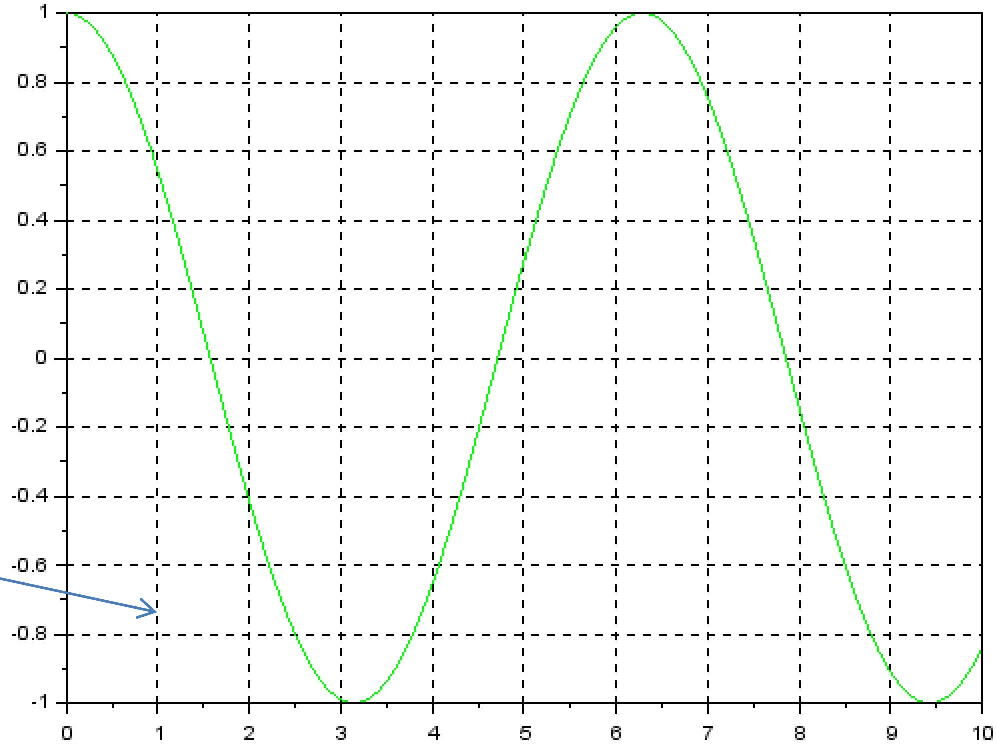
Lines type	
-	____________
--	-------------------
-.	-.-.-.-.-.-.-.-.-.-.
.	....

Dots type	
.	....
*	* * * * * * * *
o	o o oo o o o o o o
+	++++++++++
x	xx x x x x x x

Colors	
y	yellow
m	majenta
c	light blue
r	red
g	green
b	dark blue
w	white
k	black

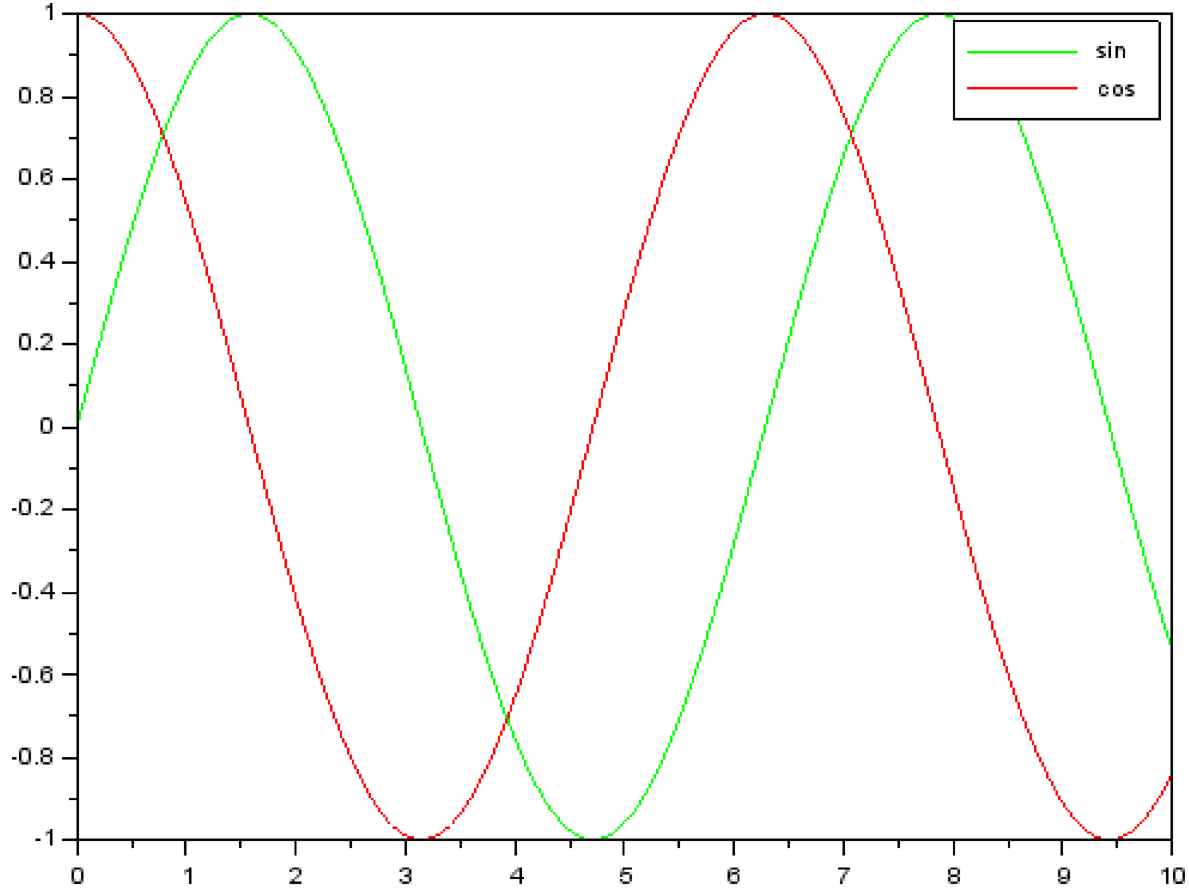
Graphics

```
t=0:0.01:10;
z=cos(t);
plot(t,z,'g-')
xgrid
```



Simultaneous graphics

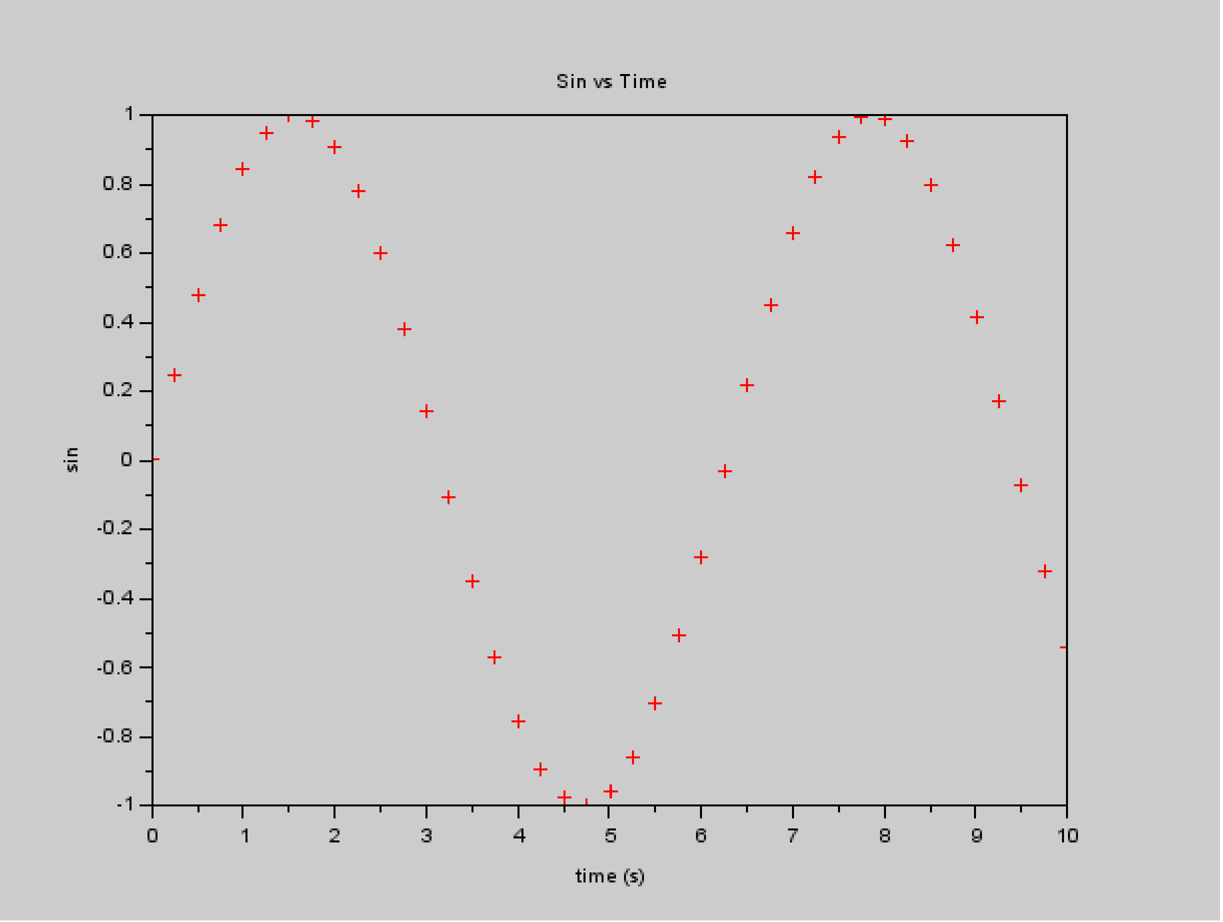
```
t=0:0.01:10;
y=sin(t);
z=cos(t);
plot(t,y,'g-',t,z,'r-')
legend('sin','cos')
```



Graphics

Assigning title and name to the axes

```
t=0:0.25:10;
y=sin(t);
plot(t,y,'r+')
title('Sin vs Time')
xlabel('time (s)')
ylabel('sin')
```

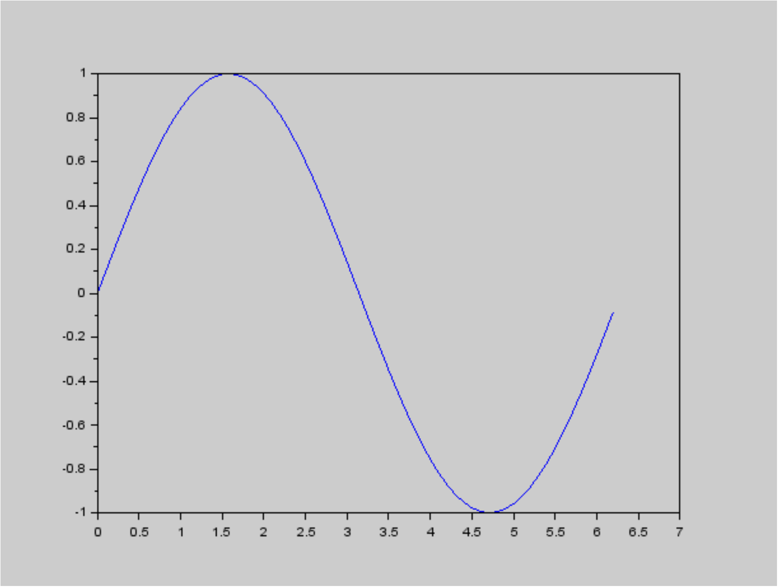


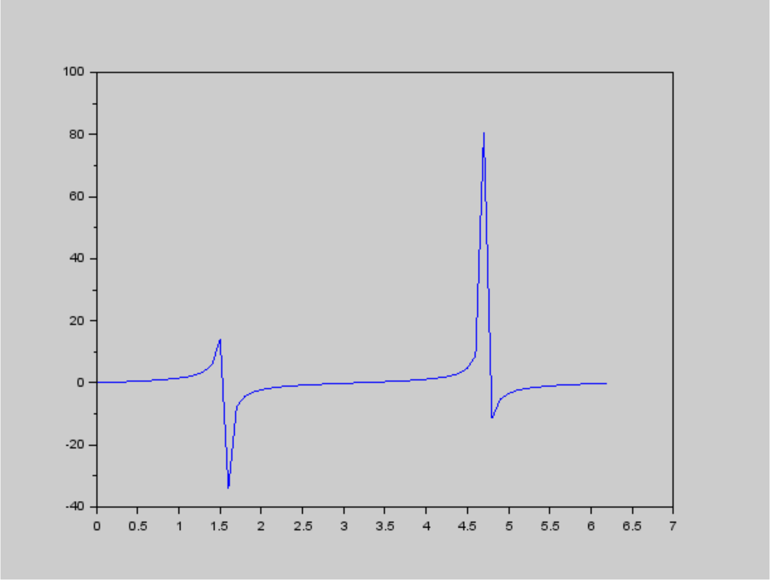
Graphics on different screens

Figure (i) - adding new screens.

Example:

```
x = [0:0.1:2*%pi];
y = sin(x);
z = tan(x);
figure(1);
plot (x,y);
figure(2);
plot (x,z);
```



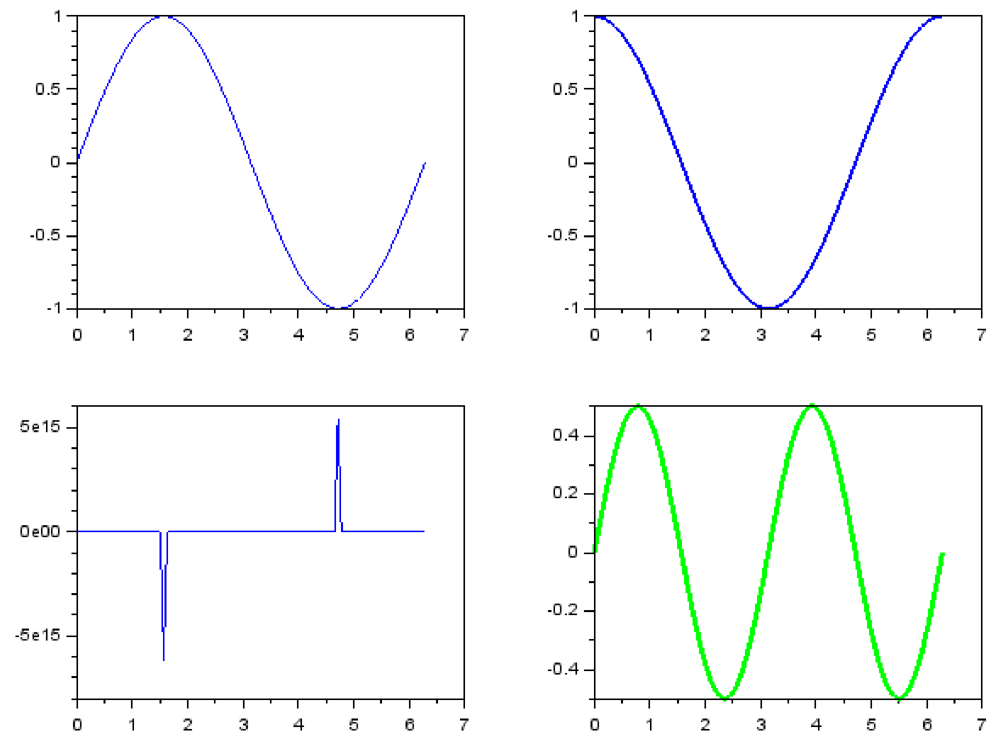


Subplot command

- divides the screen into different graphics.

Example:

x=0:(2*%pi)/100:2*%pi;
subplot(2,2,1)
plot(x,sin(x))
subplot(2,2,2)
plot(x,cos(x),'LineWidth',2)
subplot(2,2,3)
plot(x,tan(x),'LineWidth',1)
subplot(2,2,4)
plot(x,sin(x).*cos(x) , 'g-','LineWidth',3)



3D Graphics

mesh: generates graphics in 3 dimensions

Example:

```
[X,Y]=meshgrid(-5:1:5,-4:1:4);
Z=X.^2-Y.^2;
mesh(X,Y,Z);
```