NOTES FOR DATA SCIENCE FOR CHEMICAL ENGINEERS (CH 374)

INTRODUCTION TO SCILAB

-Dr. Parag Thakur

Introduction:

In early ages, Calculations were difficult with pen and paper. Thus, the calculators were invented. But, in modern times, calculators are also not useful to solve the engineering problems. Thus, the computers are invented. Various programming languages and software packages are developed for calculation of these problems.

Some programming languages are proprietary. Such as MATLAB, VBScript and Wolfram languages. However, C, C++, C#, Python, Sci-lab, Java, PHP, Kotlin, Ruby. Sci-lab is comparatively more easy for numerical computation providing a powerful computing environment for engineering and scientific applications.

Sci-lab has several windows, including:

- **Console**: For calculations
- **Editor**: For writing programs
- **Graphics windows**: For displaying graphics
- **Browser Window:** For browsing the other files
- **News feed:** For updates
- **Variable windows:** To show saved variables
- **History:** To show recent commands used on console

Mathematical Operations:

- Addition (+), Subtraction (-), Multiplication (*), Division (/), Power (^) can be directly used
- To calculate absolute value abs command can be used

e.g., abs(-6) will give output 6

- To find remainder modulo command can be used

e.g., modulo(10/3) will give output 1

- To calculate square root, sqrt command used

e.g., sqrt(25) will give output 5

To calculate roots, nthroot command can be used

e.g., nthroot(27,3) will give output 3

- To calculate natural logarithm, log command is used

e.g., log(2) will give output 0.69314

- To calculate log with different base, base value should be added after log

e.g., log2(10) will give output 3.32192

- To calculate e^x, exp command is used

e.g., exp(0.1) will give output 1.10517

- **For constant values** % sign should be added

e.g. %pi is 3.14159

%e is 2.71828

- %i is notation for imaginary number
- Sci-lab is Case-Sensitive
- clc command is used to clear the console
- clear command is used to clear the variable
- help command is used to go to help menu
- editor command is used to go to console

Trigonometric operations

Sr. No.	Command	Output
1.	sin(%pi/2)	1
2.	cos(%pi/2)	0 (approximately)
3.	tan(%pi/4)	1
4.	cotg(%pi/4)	1
5.	csc(%pi/4). Cosec	1.4142
6.	sec(%pi/4)	1.4.142
7.	sind(90)	1
8.	cosd(90)	0
9.	tand(90)	Inf
10.	cotd(45)	45
11.	cscd(45)	1.4142
12.	secd(45)	1.4142

NOTE:

put a before command for inverse

- e.g., acos — element wise cosine inverse (radians) acosd — inverse cosine function in degree

3. Arrays and matrices:

An array is a data structure that stores multiple values under a single variable name. Each item in an array is called an element, and each element has a numeric index that allows you to access it. The length of an array is fixed when it is created.

e.g., X=[1 -2 5 8]

Space or comma can be used to separate data points in row

Single inverted comma is used to define string characters

colan: is used to define equidistant points

e.g., X=[9:13] will give answer [9 10 11 12 13]

similarly, X=[9:0.5:13] will give answer [9 9.5 10 10.5 11 11.5 12 12.5 13]

here, first value is start point, second value is step-change in values and last value is endpoint.

Semi-colan: it is used to end current row and to go on next row of matrix

e.g., X=[3 2 1; 4 5 6; 8 7 6] is

$$\begin{matrix} 3 & 2 & 1 \\ 4 & 5 & 6 \\ 8 & 7 & 6 \end{matrix}$$

If order is correct then,

Addition, subtraction, multiplication, division of two matrix can be done directly.

e.g., If A=$\begin{bmatrix} 2 & 3 \\ 4 & -1 \end{bmatrix}$; B=$\begin{bmatrix} 1 & 5 \\ 6 & 2 \end{bmatrix}$

Sr. No.	Command	Output	
1	A+B	$\begin{bmatrix} 3 & 8 \\ 10 & 1 \end{bmatrix}$	Summation of two matrix
2	A-B	$\begin{bmatrix} 1 & -2 \\ -2 & -3 \end{bmatrix}$	Subtraction of two matrix
3	A*B	$\begin{bmatrix} 20 & 16 \\ -2 & 18 \end{bmatrix}$	Multiplication of two matrix
4	A/B	$\begin{bmatrix} 0.5 & 0.25 \\ -0.5 & 0.75 \end{bmatrix}$	Division of Two matrix
5	A.*B	$\begin{bmatrix} 2 & 15 \\ 24 & -2 \end{bmatrix}$	Element wise multiplication of two matrix
6	A./B	$\begin{bmatrix} 2 & 0.6 \\ 0.667 & -0.5 \end{bmatrix}$	Element wise division of two matrix

7.	sum(A)	8	Summation of all elements in the matrix
8.	mean(A)	2	Average of all elements in the matrix
9.	max(A)	4	Maximum value element in the matrix
10.	A’	$\begin{bmatrix} 2 & 4 \\ 3 & -1 \end{bmatrix}$	Transpose of matrix
11.	inv(A)	$\begin{bmatrix} 0.0714 & 0.2142 \\ 0.2857 & -0.1428 \end{bmatrix}$	Inverse of matrix
12.	det(A)	-14	Determinant of matrix
13	diag(A)	$\begin{matrix} 2 \\ -1 \end{matrix}$	To determine diagonal elements of matrix A
14	A(1,1) A(1,2) A(2,1) A(2,2)	2 3 1 5	To call particular element from matrix
15.	A(1,:)	[2,3]	To call all elements in any row
16.	A(:,1)	2 4	To call all elements in any column
17.	zeros(3,3)	0 0 0 0 0 0 0 0 0	To call zero matrix of different dimensions
18.	eye(3,3)	1 0 0 0 1 0 0 0 1	To call identity matrix
19	ones(3,3)	1 1 1 1 1 1 1 1 1	To call matrix of all elements 1

Problem:

Solve following systems of equations

$$3x + 2y - z = 1; 2x - 2y + 4z = -2; \ -x + 0.5y - z = 0$$

Solution:

$$\text{If } A=\begin{bmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{bmatrix} \quad B=\begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

and AX=B, then X=A-1.B

Algorithm:

A=[3 2 -1; 2 -2 4; -1 0.5 -1]

B=[1; -2; 0]

X=inv(A)*B

$$\textbf{Ans: } \begin{bmatrix} 1 \\ -2 \\ -2 \end{bmatrix}$$

Input command:

Input command is used to write the instructions for the user.

Eg. X=input('Enter value value ')

Output will be 'Enter first value'

disp command

disp command is used to display the output on console.

Eg.,

X=input('Enter first value ')

Y=input('Enter second value ')

Z=X+Y

disp(X+Y)

Q. Develop a algorithm to calculate log mean temp difference for co-current heat exchanger

```
//ALGORITHM TO CALCULATE LOG MEAN TEMPERATURE DIFFERENCE OF COCURRENT
HEAT EXCHANGERS

clc; clear;
Hi=input('Enter Hot Fluid Inlet ');
Ho=input('Enter Hot Fluid outlet ');
Ci=input('Enter Cold Fluid inlet ');
Co=input('Enter Cold Fluid Outlet ');
```

```
LMTD=((Hi-Ci)-(Ho-Co))/log((Hi-Ci)/(Ho-Co))
disp(LMTD)
```

Conditional statements:

- & command is used to add the condition

1. if else then loop

This loop can be used for the conditions where the either or situations need to define.

Q. A chemical reaction need to be maintain between 40- 60 ^{0}C and volume of 40-60 litres. write a Sci-lab algorithm for the conditions to guide the user.

```
//Algorithm for if else then loop

clear; clc;
Temperature=input('Enter Temperature Value ')
volume=input('Enter Volume of reactant ')
if Temperature>=60 & volume>=40 then
   disp('Temperature and volume should be reduced')
elseif Temperature >=60 & volume<=30
   disp('Temperature should be decreased and volume should be increased')
elseif Temperature >=60 & volume >=30 & volume <=40
   disp('Temperature should be decreased and volume should be maintained')
elseif Temperature <=60 & Temperature >=40 & volume >=30 & volume <=4
   disp('Temperature & volume should be maintained')
elseif Temperature <=60 & Temperature >=40 & volume >=40
   disp('Temperature should be maintained and volume should be decreased')
elseif Temperature <=60 & Temperature >=40 & volume <=30
   disp('Temperature should be maintained and volume should be increased')
elseif Temperature <=40 & volume >=40
   disp('Temperature should be increased and volume should be decreased')
elseif Temperature <=40 & volume >=30 & volume <=40
   disp('Temperature should be increased and volume should be maintained')
else
   disp('Temperature should be increased and volume should be increased')
end
```

2. While loop

When we need to run the loop for certain condition only. Then while loop is used.

Q. Write scilab algorithm to calculate addition of odd numbers between two numbers.

```
//Algorithm for addition of even numbers between two numbers
```

```
clc;clear;
a=input('Enter first number ')
B=input('Enter second number ')
if (modulo(a,2)==0) then
   s=0
   while a<=B
      s=s+a
      a=a+2
   end
else
   a=a+1
   s=0
   while a<=B
      s=s+a
      a=a+2
   end
end
disp(s)
```

3. For loop

When we need to run the loop for specific times only then for loop can be used

Q. Develop Scilab algorithm for fabonacci series

```
//Algorithm of fabonacci series
clc;clear;
n=15;
f(1)=0
f(2)=1

for i=3:n
   f(i)=f(i-1)+f(i-2)
end
disp(f)
```

- **functions**

If we need to develop the functions within the scilab. then function command can be used

```
//Algorithm for Quadratic equations
clc;clear
function [p, q]=d(a, b, c)
   p=(-b+sqrt(b^2-4*a*c))/(2*a)
   q=(-b-sqrt(b^2-4*a*c))/(2*a)
endfunction
```

```
//Develop a function to solve three equations

clear; clc;
function [y, q, p]=f(x, d, g)
   y=(3*x)^2+(2*x)+3
   q=3*d*g
   p=x/sin(d)
endfunction
x=input('Enter the value ')
d=input('Enter 2nd value ')
g=input('Enter 3rd value ')
[y,q,p]=f(x,d,g)
disp(y)
disp(q)
disp(p)
```

- **Plotting of data**

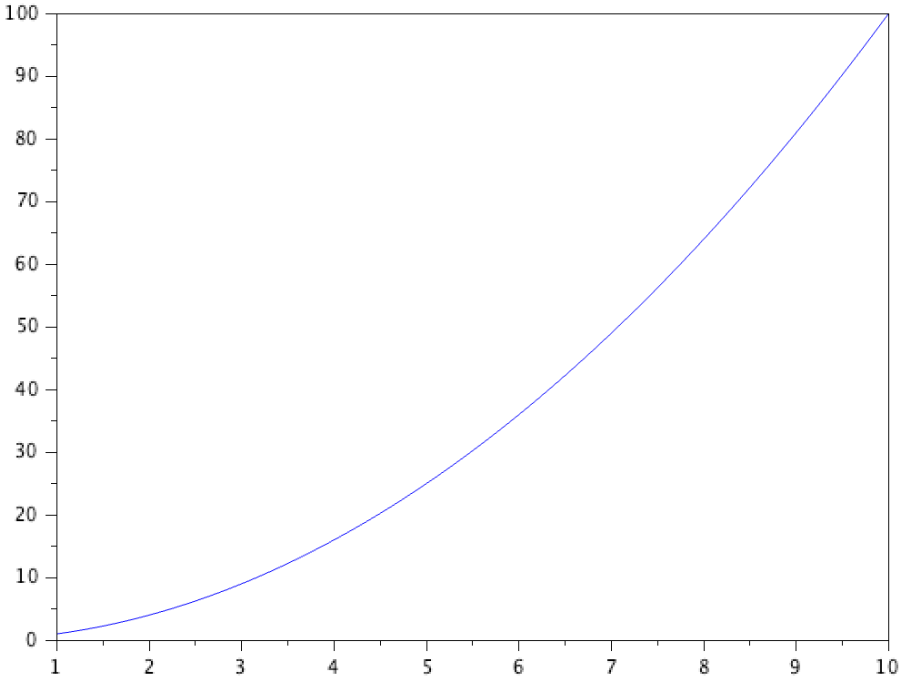
plot command is used to plot the 2D graph. clf() command is used to clear the graphic window.

e.g.

Algorithm

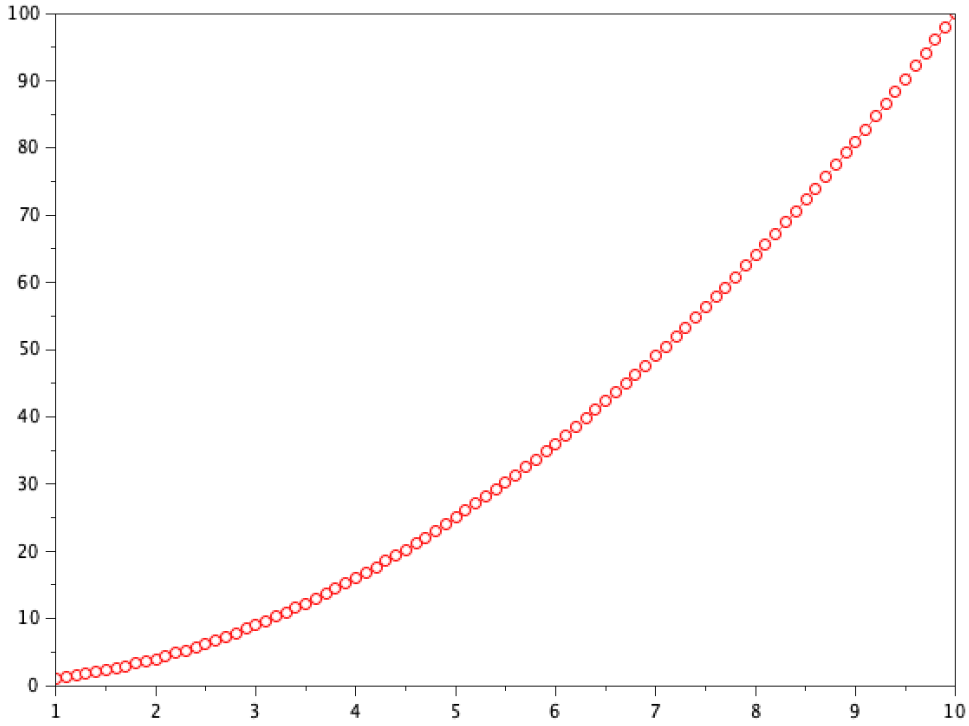
```
clear;clc;clf()
x=[1:0.1:10]
y=[x.^2]
plot(x,y)
```

Output



To format the plot, specific commands can be used for the in plot command.

e.g., plot(x,y,'—or') will give dashed line with circular markers with red color



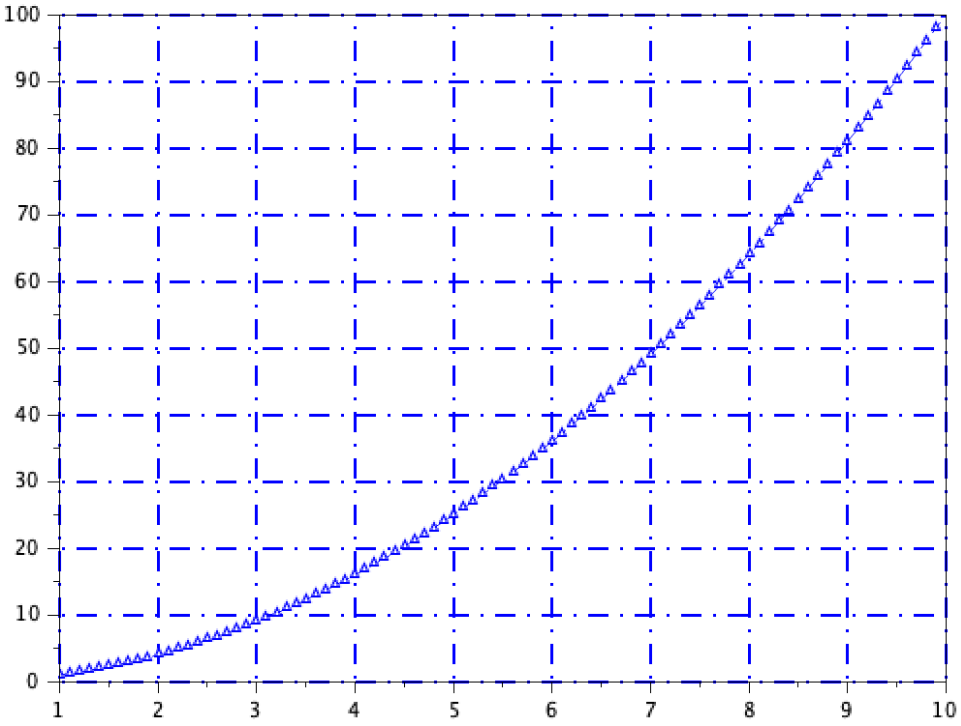
blue line is default line in the scilab,

Sr. No.	command	Output
• **Line type**		
1.	**-**	Solid line (Default)
2.	**--**	Dashed line
3.	**:**	Dotted line
4.	**-.**	Dash-dotted line
• **Line color**		
5.	k	Black
6.	b	Blue (Default)
7.	r	Red
8.	y	Yellow
9.	g	Green
10.	m	Magenta
11.	c	Cyan
12.	w	White
• **Marker Type**		
13.	+	Plus type marker
14.	o	circle
15.	*	asterisk
16.	.	point
17.	X	cross
18.	‘square’ or ‘s’	square
19.	‘diamond’ or ‘d’	diamond
20.	^	upward pointing
21.	>	rightpointing
22.	<	left pointing
23.	pentagram or ‘p’	five pointed star
24.	|	vertical line
25.	‘minus’ or ‘m’	horizontal line

xgrid command is used to generate the grid,

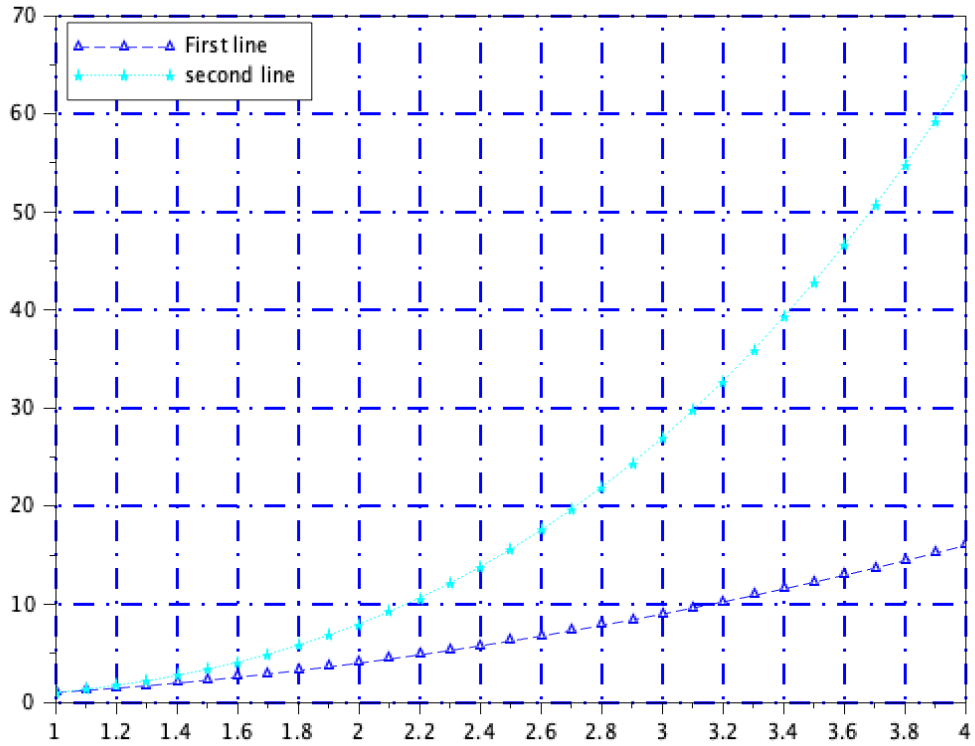
color, thickness and style of grid can be formatted through specifiying the numbers in the brackets

e.g. xgrid(2,3,4) will give following output



legend can specified by legend command

legend([‘First line’,’second line’],2)



xtitle command is used to specify the title of graph and axis.

Xtitle('Title','X-axis','Y-axis')

Subplot command is used to separate the plot in the graphic window.

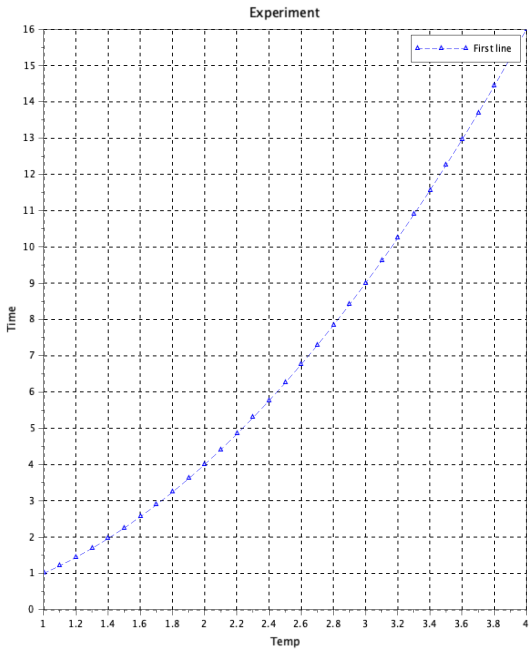
e.g. subplot(1,2,1) will divide graphic window in 1 row, 2 column system and will position the graph in 1st position.

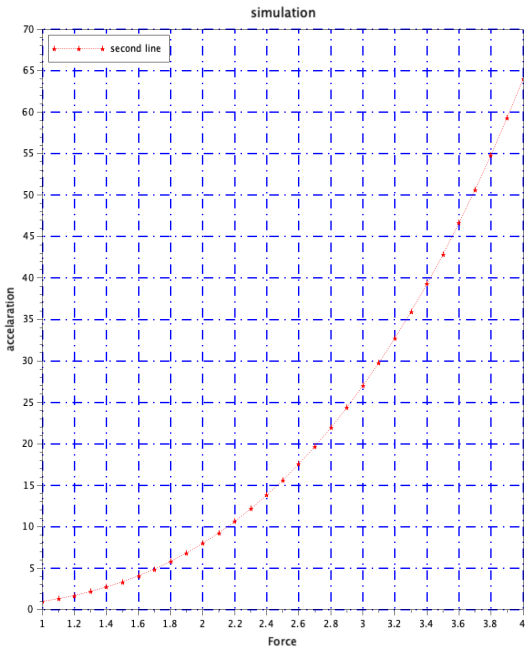
e.g.,

algorithm

```
clear;clc;clf()
x=[1:0.1:4]
y=[x.^2]
z=[x.^3]
subplot(1,2,1)
plot(x,y,'o--^')
xgrid(1,2,3)
xtitle('Experiment','Temp','Time')
legend(['First line'],1)
subplot(1,2,2)
plot(x,z,'p:r')
xgrid(2,3,4)
xtitle('simulation','Force','accelaration')
legend(['second line'],2)
```

output





- comet command can be used to animate the 2D plot
- plot2D and fplot are also useful commands

3d plots can be plotted by plot3d command

mesh can be generated by the meshgrid command

surf command can be used in the 3D plots to analyze the surface of geometry