Rashtreeya Sikshana Samithi Trust

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF MATHEMATICS

VECTOR CALCULUS, LAPLACE TRANSFORM AND NUMERICAL METHODS 22MA21A

MANUAL FOR EXPERIENTIAL LEARNING USING MATLAB

II - SEMESTER

Contents

Introduction

In Experiential Learning component of first semester Mathematics course - Fundamentals of Linear Algebra, Calculus and Numerical Methods, an introduction was given to MATLAB software and modules were provided for basic concepts like arithmetic operations, math built-in functions, visualizing 2D and 3D plots, elementary linear algebra, differentiation, integration and interpolation.

On successful completion of learning these modules, the process continues with Experiential Learning component of second semester Mathematics course - Vector Calculus, Laplace Transform and Numerical Methods. A note on scripts and anonymous functions appear which are useful in repeated usage of a set of MATLAB commands. Further, a more rigorous approach is followed in which the modules cover all the basic concepts of higher Engineering Mathematics like statistical data,vector differentiation, Laplace transform and numerical techniques on MATLAB.

It is ensured that after going through the MATLAB course as Experiential Learning of I and II semester Mathematics course, a student is well equipped to code engineering problems successfully.

Scripts and Functions

MATLAB Scripts

To perform a sequence of commands repeatedly or to save for future reference, they are stored in a program file. The simplest type of MATLAB program is a script, which contains a set of commands exactly as it is typed at the command line.

Scripts are the simplest type of program file. They are useful for automating a series of MATLAB commands, such as computations that are performed repeatedly from the command line or series of commands referenced.

The new scripts are created in the following ways:

1. Click the **New Script** button on the **Home** tab.
2. Use the `edit` function by typing it in the MATLAB Command Prompt.

This opens the MATLAB Editor window in which commands can be entered. After entering the commands, save the script with a filename. The default MATLAB Script extension is '.m'.

After saving the script, the code can be run in the following ways:

1. Type the script name on the command line and press **Enter**. For example, to run a script titled `myMATLABProgram.m` script, type `myMATLABProgram`.
2. Click the **Run** button on the **Editor** tab.

Example for a MATLAB Script

The following script calculates the area of a triangle with base 'b' and height 'h'.

```
b = 5;
h = 3;
a = 0.5*(b*h)
```

This can be saved as `triarea.m`, run the script, by typing `triarea` on the MATLAB Command Prompt or click the **Run** button on the **Editor** tab.

Entering Comments in a MATLAB Script

Writing the comment lines in MATLAB script will help the user to understand the code.

Add comments to MATLAB code using the percentage (`%`) symbol. Comment lines can appear anywhere in a program file and comments can be appended at the end of a line of code.

```
%This script calculates the area of a triangle with base 'b' and height 'h'
b = 5;           % Base
h = 3;           % Height
a = 0.5*(b*h)    % Area
```

To obtain more information on Scripts, type the following in Command Window:

```
web(fullfile(docroot, 'matlab/scripts.html'))
```

Live Scripts

To combine code with embedded output, formatted text, equations and images in a single interactive environment, create live scripts.

Live scripts are program files useful for interacting with a series of MATLAB commands and their output. Live scripts contain output and graphics with the code that produced them, together in a single interactive environment called the Live Editor. Add formatted text, images, hyperlinks and equations to live scripts to produce an interactive narrative that can be shared with others. Live scripts are stored as files `.mlx` extension.

To create a new live script, on the **Home** tab, in the **New** drop-down menu, select **Live Script**.

To obtain more information on Live Scripts, type the following in the Command Window:

```
web(fullfile(docroot, 'matlab/matlab_prog/create-live-scripts.html'))
```

Valid script names follow the same rules as variable names. It must start with a letter and can be followed by letters, digits, or underscores.

MATLAB Functions

Both scripts and functions allow to reuse sequences of commands by storing them in program files. Scripts are the simplest type of program, since they store commands exactly as it is typed at the command line.

Functions provide more flexibility, as primarily user can pass input values and return output values. For example, the following function named `fact` computes the factorial of a number (`n`) and returns the result (`f`).

```
function f = fact(n)
    f = prod(1:n);
end
```

This type of function must be defined within a file, not at the command line. The best practice is to use the same name for the function and the file (in this example, `fact.m`), since MATLAB associates the program with the file name.

The function can be called from the command line, using the same syntax rules that apply to functions installed with MATLAB. For instances, calculate the factorial of 5.

```
x = 5;
y = fact(x);
```

Creating MATLAB Functions

Defining Functions

To create a MATLAB function, open the MATLAB Editor by typing `edit` in the Command Window.

The first line of every function is the definition statement, which includes the following elements.

1. The `function` keyword (required)
2. Output arguments (optional)
3. Function Name (required)
4. Input arguments (optional)

Valid function names follow the same rules as variable names. It must start with a letter and can be followed by letters, digits, or underscores.

```
function [x,y] = myFunction(one,two,three)
```

This above function has the following elements:

1. Function Name - `myFunction`
2. Input Arguments - `one, two, three`
3. Output Argument - `x, y`

Contents of Functions and Files

The body of a function can include valid MATLAB expressions, control flow statements, comments, blank lines and nested functions. Note that any variables can be created within a function and are stored within a workspace specific to that function, which is separate from the base MATLAB workspace.

End Statements

Typically, functions end with an `end` statement at the end of the file. Although it is optional, use `end` for better code readability.

Scripts vs Functions

For a comparison on scripts and functions and when to use them, go over the documentation in:

```
web(fullfile(docroot, 'matlab/matlab_prog/scripts-and-functions.html'))
```

Anonymous Functions

What Are Anonymous Functions?

An anonymous function is a function that is *not* stored in a program file. It is associated with a variable whose data type is `function_handle`. Anonymous functions can accept inputs and return outputs, just as standard functions do. However, they can contain only a single executable statement.

For example, create a `function_handle` to an anonymous function that finds the square of a number:

```
sqr = @(x) x.^2;
```

Variable `sqr` is a function handle. The `@` operator creates the handle and the parentheses `()` immediately after the `@` operator include the function input arguments. This anonymous function accepts a single input `x` and implicitly returns a single output, an array the same size as `x` that contains the squared values.

Find the square of a particular value (`5`) by passing the value to the function handle, similar to passing an input argument to a standard function.

```
a = sqr(5)
```

Many MATLAB functions accept function handles as inputs so that functions can be evaluated over a range of values. Handles can be created either for anonymous functions or for functions in program files. The benefit of using anonymous functions is that not to edit and maintain a file for a function that requires only a brief definition.

Example of Anonymous Function with ODE45 solver

The van der Pol equation is a second order ODE

$$y_1'' - \mu\left(1 - y_1^2\right) y_1' + y_1 = 0,$$

where $\mu > 0$ is a scalar parameter. Rewrite this equation as a system of first-order ODEs by making the substitution $y_1' = y_2$. The resulting system of first-order ODEs is

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1. \end{aligned}$$

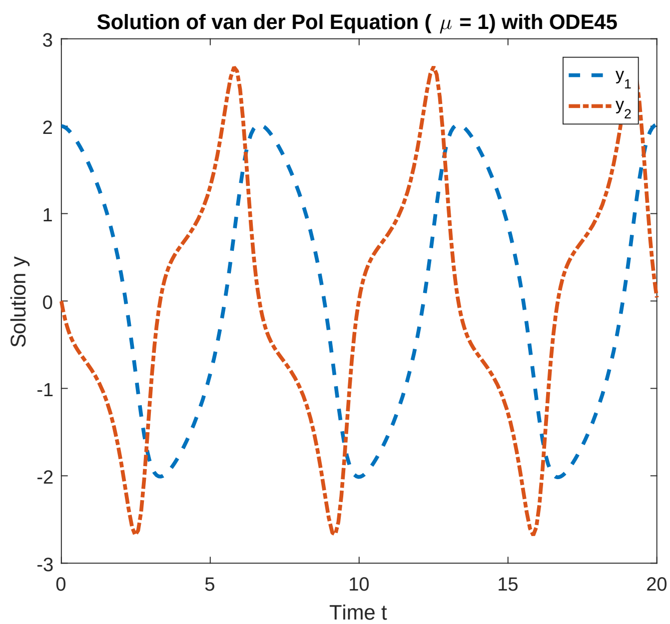
The function `vdpeval` represents the van der Pol equation using $\mu = 1$. The variables y_1 and y_2 are the entries `y(1)` and `y(2)` of a two-element vector, `dydt`.

Solve the ODE using the `ode45` function on the time interval `[0 20]` with initial values `[2 0]`. The resulting output is a column vector of time points `t` and a solution array `y`. Each row in `y` corresponds to a time returned in the corresponding row of `t`. The first column of `y` corresponds to y_1 and the second column to y_2.

```
[t,y] = ode45(@vdpeval,[0 20],[2; 0]);
```

Plot the solutions for y_1 and y_2 against `t`.

```
plot(t,y(:,1),'--',t,y(:,2),'-.')
title('Solution of van der Pol Equation (\mu = 1) with ODE45');
xlabel('Time t');
ylabel('Solution y');
legend('y_1','y_2')
```




```
function dydt = vdpeval(t,y)
%VDP1  Evaluate the van der Pol ODEs for mu = 1
%   Jacek Kierzenka and Lawrence F. Shampine
dydt = [y(2); (1-y(1)^2)*y(2)-y(1)];
end
```

Modules

1 Statistics – Measures of Central Tendency and Dispersion

Topic learning outcomes:

Student will be able to:

1. Evaluate mean, median, mode, variance and standard deviation from array.
2. Import data from the excel file to obtain mean, mode, variance and standard deviation.

Definitions:

Mean: It is the average and is computed as the sum of all the observed outcomes from the sample divided by the total number of events.

Median: If the values of a variable are arranged in the ascending order of magnitude, the median is the middle term if the number is odd and is the mean of two middle terms if the number is even.

Mode: It is that value of the variable which occurs most frequently, i.e., the value of the maximum frequency.

Standard deviation: It is the positive square root of the mean of the squares of the deviation from the arithmetic mean.

Variance: It is the square of standard deviation.

Syntax and description:

mean - Average or mean value of array

median - Returns the median value of array.

mode - Most frequent values in array

std - Standard deviation

var - Variance

Example 1.1. *The crushing strength of 10 cement concrete experimental blocks, in metric tonnes per square cm was 4.4, 5.0, 3.8, 9.6, 4.2, 4.7, 4.0, 3.9, 4.6 and 5.0. Find mean, median, mode, standard deviation and variance of the crushing strength of the blocks.*

```
A=[4.4, 5.0, 3.8, 9.6, 4.2, 4.7, 4.0, 3.9, 4.6, 5.0]
mean(A)
mode(A)
median(A)
var(A)
std(A)
```

```
ans =
     4.9200
ans =
     5
ans =
    4.5000
ans =
    2.8884
ans=
    1.6995
```

Import Data from Spreadsheets

You can import data from spreadsheet files into MATLAB interactively, using the Import Tool or programmatically using an import function.

Example 1.2. *Find mean, standard deviation of first and second test marks obtained by 58 students of CS branch.*

Hint: Go to Home. Click import data and open the excel file, select the required columns

Import data from excel file and name the matrix as 'A', select numeric matrix and click import selection. Then write the syntax for finding mean, standard deviation.

```
mean(A)
std(A)
```

```
ans =
     4
ans =
    2.1602
```

Exercise:

1. The tensile strength in megaPascals for 15 samples of tin were determined and found to be: 34.61, 34.57, 34.40, 34.63, 34.63, 34.51, 34.49, 34.61, 34.52, 34.55, 34.58, 34.53, 34.44, 34.48 and 34.40. Calculate the mean and standard deviation for these 15 values, correct to 4 significant figures.

2. The values of capacitances in microfarads of ten capacitors selected at random from a large batch of similar capacitors are: 34.3, 25.0, 30.4, 34.6, 29.6, 28.7, 33.4, 32.7, 29.0 and 31.3. Determine the standard deviation of capacitances for these capacitors, correct to 3 significant figures.

3. The runs scored by 11 members of a cricket team are 25, 39, 53, 18, 65, 72, 0, 46, 31, 08, 34. Find mean, median, mode and standard deviation of score.

4. The age (in years) of ten teachers in a school are 34, 37, 53, 46, 52, 43, 31, 36, 40, 50. Calculate mean, median, mode and variance of age.

2 Plotting Statistical Data

Topic learning outcomes:

Student will be able to:

1. Create bar graphs and pie charts for set of data.
2. Draw the bars with labels, set the relative bar width, specify the style of the bar groups and set the color for all the bars.
3. Draw multiple pie chart with labels.

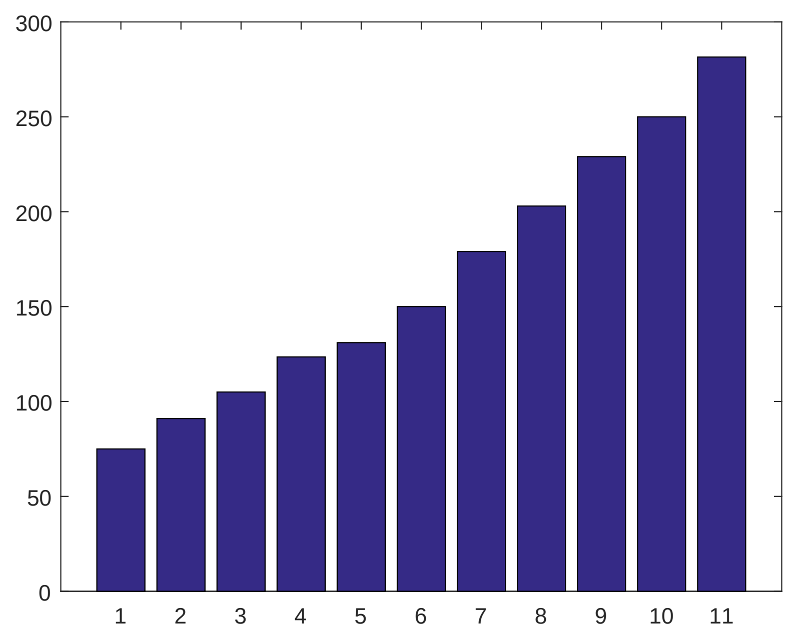
Bar graph

Syntax and description:

- `bar(y)` - creates a bar graph with one bar for each element in `y`. If `y` is a matrix, then `bar` groups the bars according to the rows in `y`.
- `bar(x,y)` - draws the bars at the locations specified by x.
- `bar(_,width)` - sets the relative bar width, which controls the separation of bars within a group. Specify `width` as a scalar value. Use this option with any of the input argument combinations in the previous syntaxes.
- `bar(_,style)` - specifies the style of the bar groups. For example, use 'stacked' to display each group as one multicolored bar.
- `bar(_,color)` - sets the color for all the bars. For example, use `'r'` for red bars.

Example 2.1. *Create bar graph for vector y = [75 91 105 123.5 131 150 179 203 229 250 281.5].*

```
y = [75 91 105 123.5 131 150 179 203 229 250 281.5];
bar(y)
```

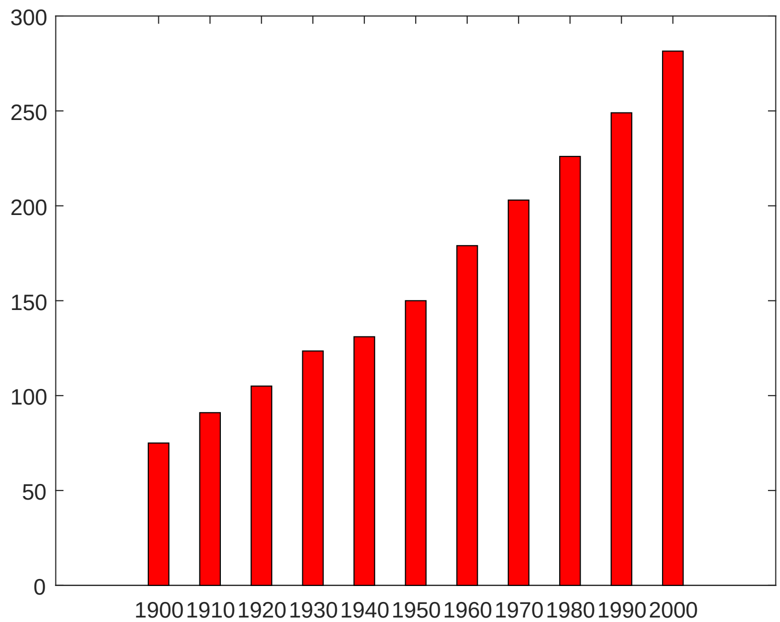


Example 2.2. *The following data shows one country population (in millions) from 1900 to 2000.*

Year of census	*1900*	*1910*	*1920*	*1930*	*1940*	*1950*	*1960*	*1970*	*1980*	*1990*	*2000*
Population	*75*	*91*	*105*	*123.5*	*131*	*150*	*179*	*203*	*226*	*249*	*281.5*

Specify the bar locations along the x-axis and set the width of each bar to 40 percent of the total space available for each bar. Give red color to all bars.

```
x = 1900:10:2000;
y = [75 91 105 123.5 131 150 179 203 226 249 281.5];
bar(x,y,0.4,'r')          %'r' for red color
```

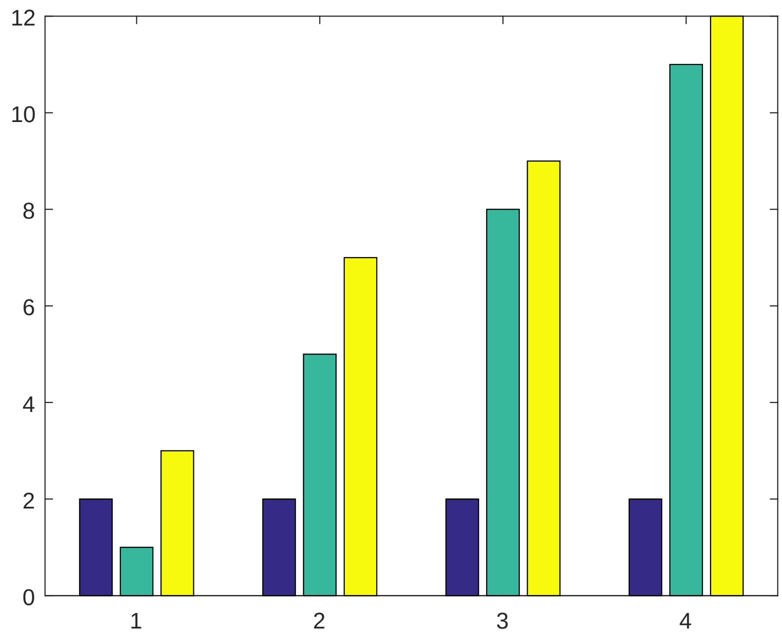


Example 2.3. *The following table shows, three persons visit number of places in four days.*

No. of places visited by	*Day 1*	*Day 2*	*Day 3*	*Day 4*
Person 1	*2*	*2*	*2*	*2*
Person 2	*1*	*5*	*8*	*11*
Person 3	*3*	*7*	*9*	*12*

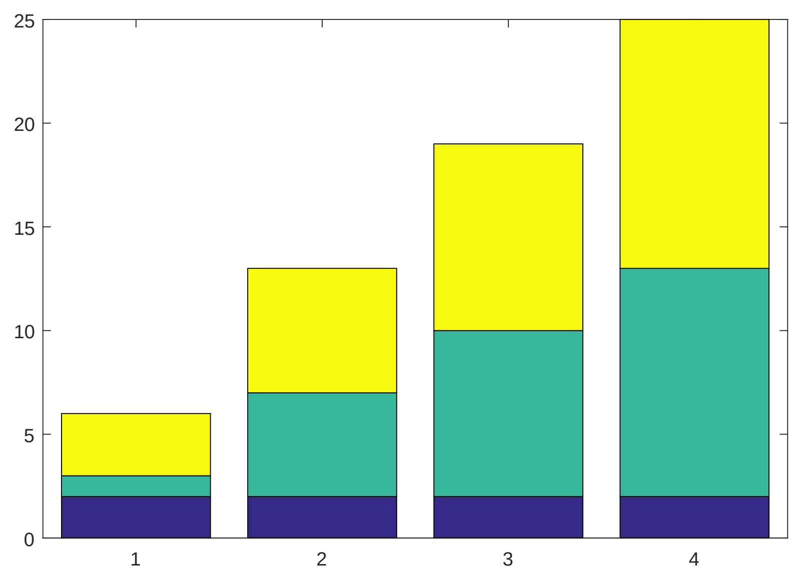
Display four groups of three bars.

```
y = [2 1 3; 2 5 7; 2 8 9; 2 11 12];
bar(y)
```



Example 2.4. *In Example 2.3, display one bar for each day. The height of each bar is the sum of the places visited in one day.*

```
z = [2 1 3; 2 5 6; 2 8 9; 2 11 12];
bar(z,'stacked')
```

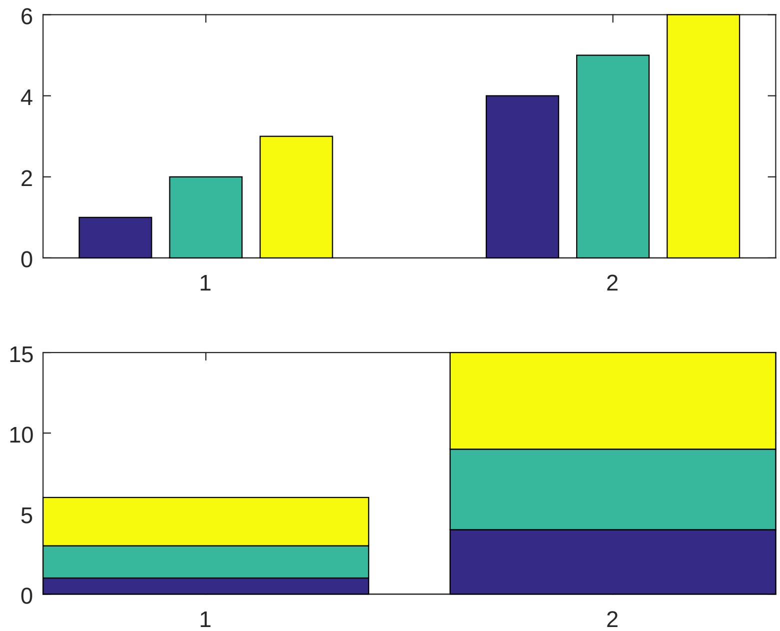


Example 2.5. *Production of two items in three months are given below:*

	Month 1	*Month 2*	*Month 3*
Production of Item 1(no.)	*1*	*2*	*3*
Production of Item 2(no.)	*4*	*5*	*6*

Create a figure with two subplots. In the upper subplot, plot a bar graph. In the lower subplot, plot a stacked bar graph of the same data.

```
y = [1 2 3; 4 5 6];
ax1 = subplot(2,1,1);
bar(ax1,y)
ax2 = subplot(2,1,2);
bar(ax2,y,'stacked')
```



Pie Chart

Syntax and description:

- `pie(X)` draws a pie chart using the data in `X`. Each slice of the pie chart represents an element in `X`.

  1. If sum(X)$\leq$ 1, then the values in X directly specify the areas of the pie slices. `pie` draws only a partial pie if sum(X) $<$ 1

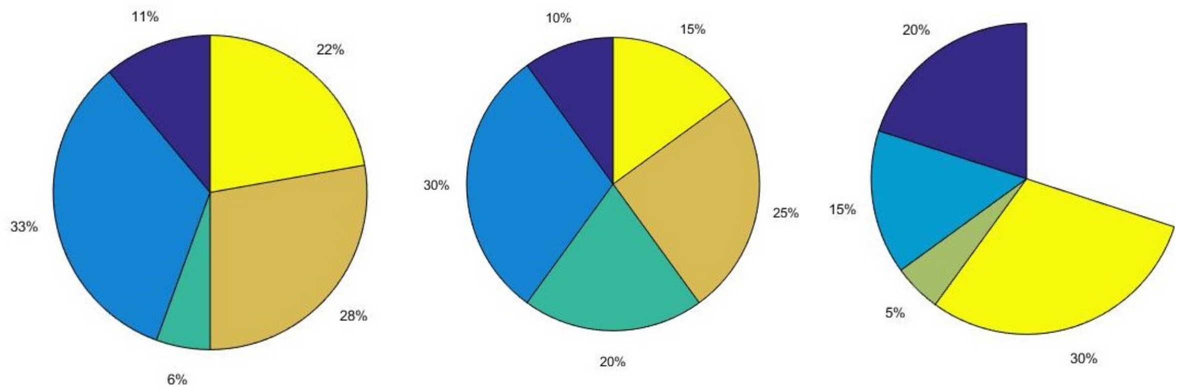
  2. If sum(X) $>$ 1, then pie normalizes the values by X/sum(X) to determine the area of each slice of the pie.

  3. If X is of data type categorical, the slices correspond to categories. The area of each slice is the number of elements in the category divided by the number of elements in X.

- `pie(X,labels)` specifies text labels for the slices. The number of labels must equal the number of slices. `X` must be numeric.
- `pie(ax,_)` plots into the axes specified by `ax` instead of into the current axes (`gca`). The option `ax` can precede any of the input argument combinations in the previous syntaxes.

Example 2.6. *Create pie charts for vectors X = [1 3 0.5 2.5 2], Y = [0.1 0.3 0.2 0.25 0.15] and Z = [0.2 0.15 0.05 0.3].*

```
X=[1 3 0.5 2.5 2];
Y=[0.1 0.3 0.2 0.25 0.15];
Z=[0.2 0.15 0.05 0.3];
pie(X)
pie(Y)
pie(Z)
```

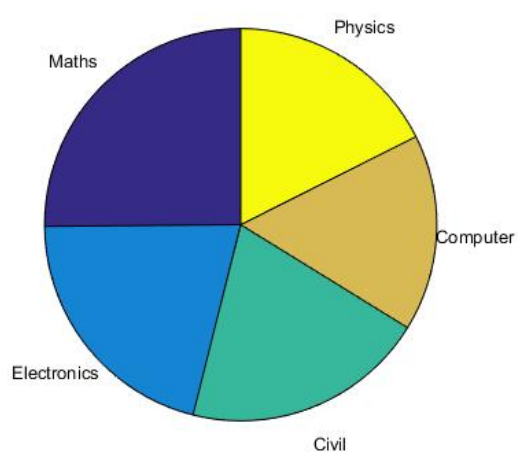


Example 2.7. *The marks obtained by a student in his annual examination are given below:*

Subject	*Maths*	*Electronics*	*Civil*	*Computer*	*Physics*
Marks obtained	*90*	*75*	*72*	*58*	*63*

Draw a pie chart to represent the above data.

```
X=[90 75 72 58 63];
labels = {'Maths','Electronics','Civil','Computer','Physics'};
pie(X,labels)
```

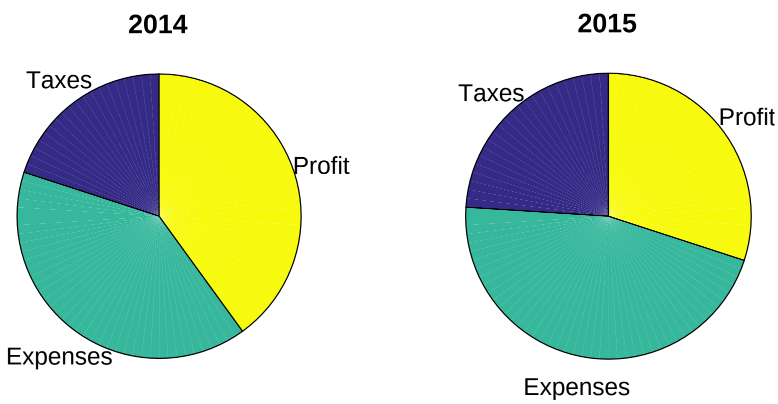


Example 2.8. *Accounts data of a company are given below:*

	Taxes (%)	*Expenses (%)*	*Profit (%)*
2014	*20*	*40*	*40*
2015	*24*	*46*	*30*

Draw multiple pie chart.

```
X = [20 40 40];
labels = {'Taxes','Expenses','Profit'};
ax1 = subplot(1,2,1);
pie(ax1,X,labels)
title(ax1,'2014');

Y = [24 46 30];
ax2 = subplot(1,2,2);
pie(ax2,Y,labels)
title(ax2,'2015')
```



Exercise:

1. The number of cars produced in a factory during five consecutive weeks is given below:

Week	First	Second	Third	Fourth	Fifth
Number of cars produced	800	1300	1060	920	1440

   Create bar graph in green color.

2. The following table shows the export earnings of India (in crores) during five consecutive years:

Year	2010	2011	2012	2013	2014
Spices export	120	125	132	146	150
Handicrafts export	100	107	110	118	130
Books export	102	115	121	132	142

   Display data in five groups of three bars.

3. The number of electric bulbs sold in a shop during a week is given in the following table:

Day	Mon	Tue	Wed	Thu	Fri	Sat
No. of bulbs sold	225	100	200	150	75	120

Create a pie chart to represent the above data.

3 Vector Differentiation

Topic learning outcomes:

Student will be able to:

1. Understand the existence of vector functions, derivatives of vector functions and rules of differentiation and fundamentals of the integration of vector point function.

2. The importance of defining vector differential operator and the operations- Gradient of scalar point functions, Divergence and Curl of vector point functions.

Syntax and description:

- `g = gradient(f,v)` returns the gradient vector of the scalar function f with respect to vector v in Cartesian coordinates. The input f is a function of symbolic scalar variables and the vector v specifies the scalar differentiation variables.
- `g = gradient(f)` returns the gradient vector of the scalar function f with respect to a vector constructed from all symbolic scalar variables found in f. The order of variables in this vector is defined by $symvar$.
- The gradient of a scalar function `f` with respect to the vector `v` is the vector of the first partial derivatives of `f` with respect to each element of `v`.
- `divergence(V,X)` returns the divergence of vector field V with respect to the vector X in Cartesian coordinates. Vectors V and X must have the same length.
- `curl(V,X)` returns the curl of the vector field V with respect to the vector X. The vector field V and the vector X are both three-dimensional.
- `curl(V)` returns the curl of the vector field V with respect to the vector of variables returned by $symvar(V, 3)$.

Example 3.1. *Find the gradient vector of* $2yzsin(x) + 3xsin(z)cos(y)$ *with respect to vector* `[x,y,z]`.

```
syms x y z
f(x,y,z) = 2*y*z*sin(x) + 3*x*sin(z)*cos(y);
gradient(f,[x,y,z])
```

$$ans(x, y, z) = \begin{pmatrix} 3\cos(y)\sin(z) + 2\,y\,z\cos(x) \\ 2\,z\sin(x) - 3\,x\sin(y)\sin(z) \\ 2\,y\sin(x) + 3\,x\cos(y)\cos(z) \end{pmatrix}$$

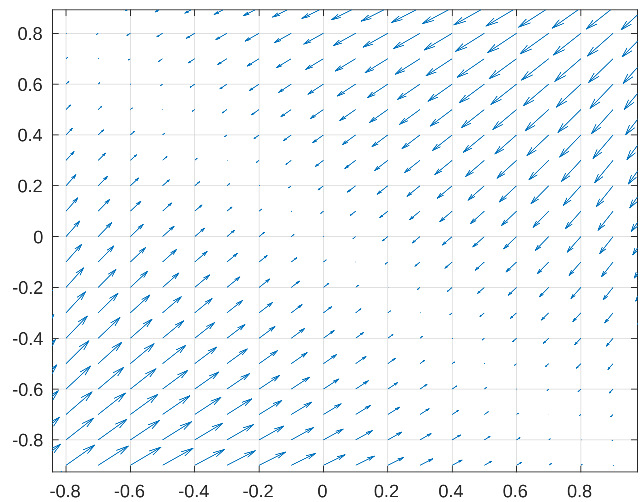
Example 3.2. *Find the gradient of a function* $-(sin(x) + sin(y))^2$ *with respect to vector* `[x,y]`*., and plot it as a quiver (velocity) plot*

```
syms x y
f = -(sin(x) + sin(y))^2;
g = gradient(f,[x,y])
```

$$g = \begin{pmatrix} -2\cos(x)\,(\sin(x) + \sin(y)) \\ -2\cos(y)\,(\sin(x) + \sin(y)) \end{pmatrix}$$

Now plotting the vector field defined by these components. MATLAB provides the `quiver` plotting function for this task. The function does not accept symbolic arguments. First, replace symbolic variables in expressions for components of `g` with numeric values. Then use `quiver`.

```
[X, Y] = meshgrid(-1:.1:1,-1:.1:1);
G1 = subs(g(1),[x y],{X,Y});
G2 = subs(g(2),[x y],{X,Y});
quiver(X,Y,G1,G2)
```



Example 3.3. *Find the divergence of the vector field* $V(x, y, z) = (x, 2y^2, 3z^3)$ *with respect to vector* $X = (x, y, z)$.

```
syms x y z
f = [x 2*y^2 3*z^3];
v = [x y z];
divergence(f,v)
```

$$ans = 9z^2 + 4y + 1$$

Example 3.4. *Find the divergence of the gradient of scalar function* $x^2 + y^2 + z^2$.

```
syms x y z
f = x^2 + y^2 + z^2;
divergence(gradient(f,vars),vars)
```

$$ans = 6$$

Example 3.5. *Compute the curl of the vector field* $V(x, y, z) = (x^3y^2z, y^3z^2x, z^3x^2y)$ *with respect to vector* $X = (x, y, z)$ *in Cartesian coordinates.*

```
syms x y z
V = [x^3*y^2*z, y^3*z^2*x, z^3*x^2*y];
X = [x y z];
curl(V,X)
```

$$ans = (x^2z^3 - 2xy^3z, x^3y^2 - 2xyz^3, -2x^3yz + y^3z^2)$$

Example 3.6. *Compute the curl of the gradient of the scalar function* $x^2 + y^2 + z^2$.

```
syms x y z
f = x^2 + y^2 + z^2;
vars = [x y z];
curl(gradient(f,vars),vars)
```

$$ans = (0, 0, 0)$$

Exercise:

1. Gradient of scalar measure the increase or decrease of a quantity along some direction, Find the gradient vector of $xy^2z^3 - x^3y^2z$ with respect to vector `[x,y,z]`.

2. The steepness of the slope at that point is given by $2x^3y^2z^4$. Find the divergence of the gradient of scalar function.

3. A person on a hang glider is spiraling upward due to rapidly rising air on a path having a vector field $F(x, y, z) = (\frac{x}{x+y}, \frac{y}{x+y})$. Find the divergence of the vector field with respect to vector $X = (x, y)$.

4. Show that the uniform motion of a projectile along a meridian of the rotating earth is given by the vector function $V(x, y, z) = (2xyz, xy - y^2z, x^2 - zx)$ is solenoidal.

5. The velocity vector field for the three-dimensional flow of an ideal fluid around a cylinder is given by $F(x, y, z) = (sinx + z, cosy - z, x - y)$. Show that $F(x, y, z)$ is irrotational.

6. The temperature of all points in a room at a particular time is a scalar field, then compute the curl of the gradient of the scalar function $x^2y + 2xy + z^2$.

4 Laplace Transform

Topic learning outcomes:

Student will be able to:

1. Determine the Laplace transform of elementary functions.
2. Develop the Laplace transform of periodic function, Heaviside (unit step) function and Dirac Delta function.

Syntax and description:

- `laplace(f)` - returns the Laplace transform of function f. By default, the independent variable is t and transformation variable is s.
- `laplace(f,transVar)` - uses the transformation variable `transVar` instead of s.
- `laplace(f,var,transVar)` - uses the specified independent variable var and the transformation variable `transVar` instead of t and s respectively.

Example 4.1. *Compute the Laplace transform of the signal* $f(t) = t + t^{1/4} + t^{9/2} + e^{-bt} + \sin(t)\cos(nt)$.

```
syms s t b w n
laplace(t +t^(1/4)+ t^(9/2) +exp(-b*t) +sin(w*t)-cos(n*t))
```

```
ans =
w/(s^2 + w^2) - s/(n^2 + s^2) + 1/(b + s) + 1/s^2 + (945*pi^(1/2))/(32*s^(11/2))
+ (pi*2^(1/2))/(4*s^(5/4)*gamma(3/4))
```

Example 4.2. *Obtain the Laplace transform of* e^{-at} *by specifying*

1. *the transformation variable as* y.
2. *both the independent and transformation variables as* a *and* y *in the second and third arguments respectively.*

1. ```
   syms a t y
   f = exp(-a*t);
   laplace(f, y)
   ```

   ```
   ans =
   ```

   $\frac{1}{a+y}$

2. ```
   syms a t y
   f = exp(-a*t);
   laplace(f, a, y)
   ```

```
ans =
```

$\frac{1}{t+y}$

Example 4.3. *Compute the Laplace transform of* $te^{-at} + \frac{\cos wt - \cos nt}{t} + t\sin nt$.

```
syms t a w n s
laplace(t*exp(-a*t)+(cos(w*t)-cos(n*t))/t +t*sin(n*t), t, s)


ans =
log(n^2/s^2 + 1)/2 - log(w^2/s^2 + 1)/2 + 1/(a + s)^2 + (2*n*s)/(n^2 + s^2)^2
```

Example 4.4. *Find the Laplace transform of the function* $f(t) = \delta(t-3) + H(t-\pi)$, *where* $\delta\,(t-3)$ *is the Dirac delta function and* $H(t-\pi)$ *is the Heaviside function.*

```
syms t s
laplace(dirac(t-3)+heaviside(t-pi), t, s)


ans =
exp(-3*s) + exp(-pi*s)/s
```

Example 4.5. *Calculate* $L[e^{t-2} + sin(t-2)]H(t-2)]$.

```
syms t s
f=(exp(t-2)+sin(t-2))*(heaviside(t-2))
laplace(f, t, s)


ans =
exp(-2*s)*(1/(s - 1) + 1/(s^2 + 1)
```

Example 4.6. *Obtain the Laplace transform of* $\int_0^t e^{-t}\sin t\cos t dt$.

```
syms t s
I = int(exp(-t)*sin(t)*cos(t),0,t)
simplify(laplace(I, t, s))


ans =
1/(s*(s^2 + 2*s + 5))
```

Example 4.7. *Show that the Laplace transform of the derivative of a function is expressed in terms of the Laplace transform of the function itself.*

```
syms f(t) s
Df = diff(f(t),t);
laplace(Df, t, s)
```

```
ans =
s*laplace(f(t), t, s) - f(0)
```

Exercise:
Compute the followings:

1. $L[\sin t \sin 2t\ \ \sin 3t + \sinh 2t\ \ + te^{-2t}$
2. $L[e^{-4t}t^{-\frac{5}{2}}]$
3. $L\left[\frac{e^{-t}\sin t}{t}\right]$
4. $L[\cosh^2 2t]$
5. $L[\sin(2t) + \int_0^t e^{-t}\cos t dt + 2t^3]$
6. $L[e^{t-2}\ \ + \sin t - 2H(t-2)]$

5 Inverse Laplace Transform

Topic learning outcomes:

Student will be able to:

1. Evaluate the inverse Laplace Transform of the function.
2. Solve ordinary differential equations governed in real world situations.

Syntax and description:

- `ilaplace(F)`-returns the inverse Laplace transform of F. By default, the independent variable is s and the transformation variable is t. If F does not contain s, `ilaplace` uses the function `symvar`.
- `ilaplace(F,transVar)`-uses the transformation variable `transVar` instead of t.
- `ilaplace(F,var,transVar)`-uses the independent variable var and transformation variable `transVar` instead of s and t respectively.

Example 5.1. *Obtain the inverse Laplace transform of* $\frac{1}{s^2}$.

```
syms s
F = 1/s^2;
ilaplace(F)
```

```
ans =t
```

Example 5.2. *Compute the inverse Laplace transform of* $\frac{1}{(s-a)^2}$. *Also specify*

1. *the transformation variable as* x.
2. *both the independent and transformation variables as* a *and* x.

```
syms a s
F = 1/(s-a)^2;
ilaplace(F)
```

```
ans =     t*exp(a*t)
```

1. ```
   syms x
   ilaplace(F, x)
   ```

   ```
   ans =x*exp(a*x)
   ```

2. ```
   syms x
   ilaplace(F,a,x)
   ```

```
ans =x*exp(s*x)
```

Example 5.3. *Determine the time domain signal from the corresponding frequency domain signal* $F(s) = \frac{s}{s^2+4} + \frac{w}{s^2+w^2} + \frac{s}{s^2-n^2}$.

```
syms s w n
a = ilaplace(s/(s^2+4))
b = ilaplace(w/(s^2 + w^2))
c = ilaplace(s/(s^2 - n^2))
answer = a+b+c
```

```
a = cos(2*t)
b = sin(t*w)
c = exp(n*t)/2 + exp(-n*t)/2
answer =
cos(2*t) + exp(n*t)/2 + exp(-n*t)/2 + sin(t*w)
```

Example 5.4. *Compute the following inverse Laplace transform of the function* $1 + \frac{e^{-2s}}{s^2+1} + \tan^{-1}\left(\frac{1}{s}\right)$.

```
syms s t
F = 1+exp(-2*s)/(s^2+1)+atan(1/s);
ilaplace(F, s, t)
```

```
ans =
dirac(t) + sin(t)/t + heaviside(t - 2)*sin(t - 2)
```

Solution of Differential Equations using Laplace Transform:

Example 5.5. *Solve the equation* $y'' - 3y' + 2y = 4e^{2t}$ *where* $y(0) = -3$, $y\prime(0) = 5$.

```
syms s t Y
f = 4 * exp(2 * t)
F = laplace(f,t,s)
Y1 = s * Y + 3
Y2 = s * Y1 - 5
Sol = solve(Y2 - 3 * Y1 + 2 * Y - F, Y)
sol = ilaplace(Sol,s,t)
```

```
f =
4*exp(2*t)
F =
4/(s - 2)
Y1 =
```

```
Y*s + 3
Y2 =
s*(Y*s + 3) - 5
Sol =
(4/(s - 2) - 3*s + 14)/(s^2 - 3*s + 2)
sol =
4*exp(2*t) - 7*exp(t) + 4*t*exp(2*t)
```

Example 5.6. *Use the Laplace transform to determine the output of a system represented by the differential equation $y'' + y = x(t)$ in response to force function $H(t-1)$. Assume the initial conditions $y(0) = 0$, $y'(0) = 1$.*

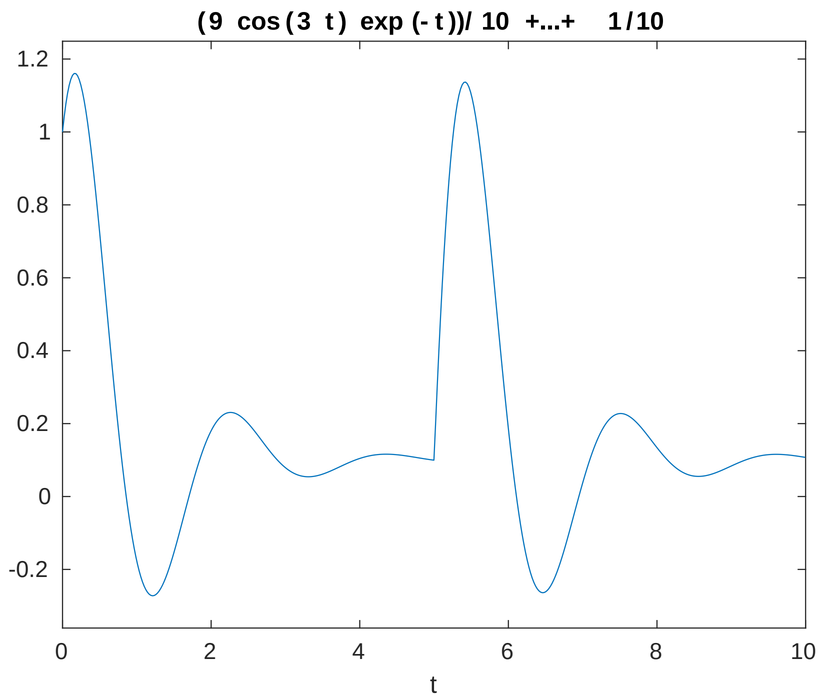
```
syms s t Y
f = heaviside(t-1)
F = laplace(f,t,s)
Y2 = s*s * Y - 1
Sol = solve(Y2 + Y - F, Y)
sol = ilaplace(Sol,s,t)
```

```
f =
heaviside(t-1)

F =
exp(-s)/s

Y2 =
Y*s^2-1

Sol =
(exp(-s)/s+1)/(s^2+1)

sol =
sin(t)-heaviside(t-1)*(cos(t-1)-1)
```

Example 5.7. *Solve the differential equation using Laplace transform $y'' + 2y' + 10y = 1 + 5\delta(t-5)$, $y(0) = 1$, $y'(0) = 2$. Also plot the solution.*

```
syms s t Y
f = 1 + 5*dirac(t-5)
F = laplace(f,t,s)
Y1 = s*Y - 1
Y2 = s*Y1 - 2
Sol = solve(Y2 + 2*Y1 + 10*Y - F, Y)
sol = simplify(ilaplace(Sol,s,t))
ezplot(sol,[0,10])
```

```
f =
5*dirac(t - 5) + 1

F =
5*exp(-5*s) + 1/s

Y1 =
Y*s - 1

Y2 =
s*(Y*s - 1) - 2

Sol =
(s + 5*exp(-5*s) + 1/s + 4)/(s^2 + 2*s + 10)

sol =
(9*cos(3*t)*exp(-t))/10 +(29*sin(3*t)*exp(-t))/30
+(5*heaviside(t - 5)*exp(5 - t)*sin(3*t - 15))/3 + 1/10
```

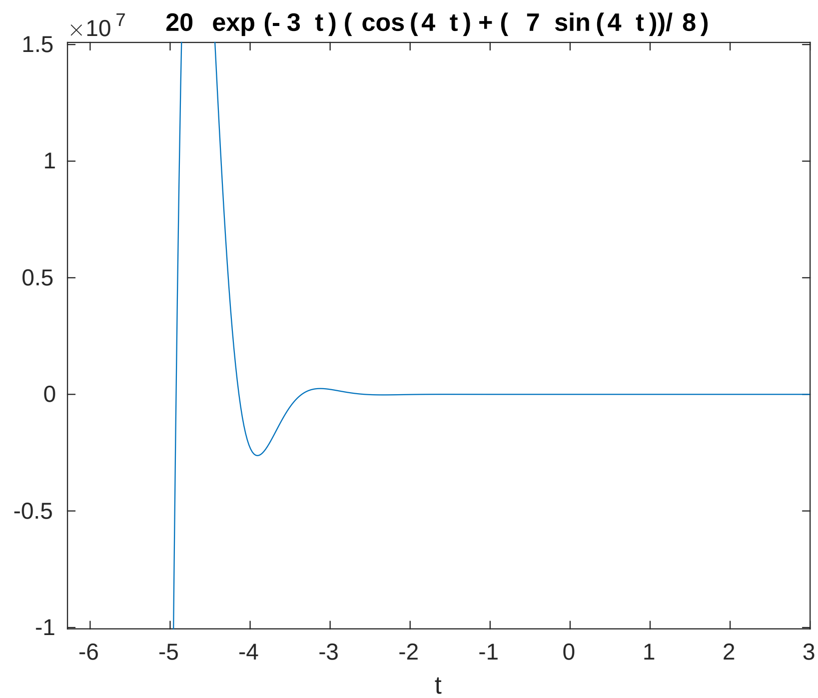


Example 5.8. *A particle is moving along a path satisfying the equation* $\frac{d^2x}{dt^2} + 6\frac{dy}{dx} + 25x = 0$*, where* x *denotes the displacement of the particle at time* t*. If the initial position of the particle is* $x = 20$ *and the initial speed is* 10*, find the displacement of the particle at any time* t *using Laplace transform.*

```
syms s t X
a=20
```

```
b=10
X1=s*X-a
X2 = s*s * X - s*a-b
Sol = solve(X2 +6*X1 +25*X, X)
x(t) = ilaplace(Sol,s,t)
ezplot(x(t))
```

```
a =
20
b =
10
X1 =
X*s - 20
X2 =
X*s^2 - 20*s - 10
Sol =
(20*s + 130)/(s^2 + 6*s + 25)
x(t) =
20*exp(-3*t)*(cos(4*t) + (7*sin(4*t))/8)
```



Example 5.9. *A LCR circuit consists of a resistor R, a capacitor C and an inductor L connected in series together with a voltage source $e(t)$. Prior to closing the switch at time $t = 0$, both the charge on the capacitor and the resulting current in the circuit are zero. Determine the charge $q(t)$ on the capacitor in the circuit at time t given that $R = 160\Omega$, $L = 1H$, $C = 10^{-4}F$ and $e(t) = 20V$.*

Solution: Applying Kirchhoffs second law to the circuit,

$$Ri + L\frac{di}{dt} + \frac{1}{C}\int idt = e(t)$$

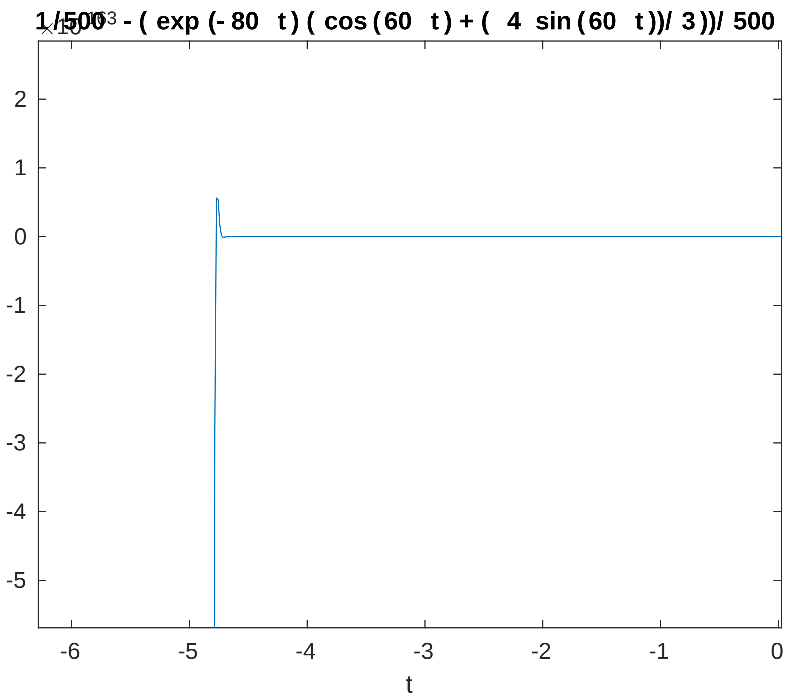
using $i = \frac{dq}{dt}$,

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = e(t)$$

Substituting the given values for L, R, C and $e(t)$ gives $\frac{d^2q}{dt^2} + 160\frac{dq}{dt} + 10^4q = 20$ Solving this differential equation using MATLAB

```
syms s t Q
a=0;
b=0;
f=20
F=laplace(f,t,s)
Q1=s*Q-a
Q2 = s*s * Q - s*a-b
Sol = solve(Q2 +160*Q1 +(10)^(4)*Q-F, Q)
q(t) = ilaplace(Sol,s,t)
ezplot(q(t))
```

```
f =
20
F =
20/s
Q1 =
Q*s
Q2 =
Q*s^2
Sol =
20/(s*(s^2 + 160*s + 10000))
q(t) =
1/500 - (exp(-80*t)*(cos(60*t) + (4*sin(60*t))/3))/500
```



Exercise:

1. Write the MATLAB code to solve the following:
   (a) $L[\sin t \sin 2t\ \sin 3t + \sinh 2t\ + te^{-2t}$
   (b) $L[e^{-4t}t^{-\frac{5}{2}}]$
   (c) $L\left[\frac{e^{-t}\sin t}{t}\right]$
   (d) $L[\cosh^2 2t]$
   (e) $L[\sin(2t) + \int_0^t e^{-t}\cos t dt + 2t^3]$
2. Write the MATLAB code to solve the following:
   (a) $L^{-1}[\frac{4s+5}{(s-1)^2(s+2)}] + L^{-1}[\tan^{-1}\frac{a}{s}]$
   (b) $L^{-1}[\frac{1}{(s^2+16)^2}]$
   (c) $L^{-1}[\frac{s}{s^4+s^2+1}]$
   (d) $L^{-1}[\frac{2s^2+5s-4}{s^3+s^2-2s}]$

6 Solution to Algebraic and Transcendental Equations

Topic learning outcomes:

Student will be able to:

1. Find the roots of algebraic and transcendental equations numerically.
2. Solve wide variety of application problems.

Syntax and description:

- `S = vpasolve(eqn)` - numerically solves the equation `eqn` for the variable determined by `symvar`.
- `S = vpasolve(eqn,var)` - numerically solves the equation `eqn` for the variable specified by `var`.
- `S = vpasolve(eqn,var,init_guess)` - numerically solves the equation `eqn` for the variable specified by `var` using the starting point in `init_guess`. If you do not specify `var`, `vpasolve` solves for variables determined by `symvar`.
- `S = vpasolve(eqn,var,[a,b])` - numerically solves the equation `eqn` for the variable specified by `var` using the search range specified in $[a, b]$, where a and b are real numbers. If you do not specify `var`, `vpasolve` solves for variables determined by `symvar`.

Example 6.1. *Find all the roots of the equation* $x^4 + 2x^2 - 16x + 5 = 0$.

```
syms x
vpasolve(x^4+2*x^2-16*x+5 == 0, x)
```

```
ans =

                                             0.32653906234460422084857323401358
                                              2.1111516462471155473107246160362
 - 1.2188453542958598840796489250249 - 2.4015366984564279025134418606114i
 - 1.2188453542958598840796489250249 + 2.4015366984564279025134418606114i
```

Example 6.2. *Determine the root of the equation* $x^2 - \log x = 12$ *in (3, 4).*

```
vpasolve(x^2-log(x) == 12, x,[3,4])
```

```
ans =

3.6460448399001736218717213181561
```

Example 6.3. *The volume V of liquid in a spherical tank of radius r is related to the depth h of the liquid by* $V = \frac{\pi h^2(3r-h)}{3}$*. Obtain the roots of equation which give height h to which the dipstick immersed is wet with liquid, given* $r = 1m$ *and* $V = 0.5\ m^3$*.*

```
syms h
vpasolve(h^3-3*h^2+0.4775 == 0, h)
```

```
ans =

 -0.37608006988252444956702991370319
  0.43113799780627533182088418262644
   2.9449420720762491177461457310768
```

Here radius of spherical tank is 1m, hence diameter = 2m. Therefore $h = 0.4311$m is the only possibility for the given physical situation.

Example 6.4. *The bacteria concentration* c *in a reservoir varies as* $t^4 - t = c$*. Calculate the time* t *required for the bacteria concentration to be* $c = 10$*, given that initial time is* $t = 2s$*.*

```
syms t
vpasolve(t^4-t-10 == 0,t,[1,2.5])
```

```
ans =

1.8555845286409378637602505648888
```

Exercise:

1. Obtain all the roots of the equation $x^6 - x^4 - x^3 + 1 = 0$.
2. Determine the root of the equation $\tan x + \tanh x = 0$ in (2, 3).
3. The current i in an electric circuit is given by $i = 10e^{-t}\sin(2\pi t)$, where t is in seconds. Find the value of t for $i = 2$ amp.
4. Water is falling in a trapezoidal channel at a rate of $Q = 20m^3/s$. The critical depth y for such a channel must satisfy the equation $1 - \frac{Q^2 B}{gA^3} = 0$ where $g = 9.81\text{m/s}^2$, A is the cross sectional area in m^2 and B is the width of the channel at surface in m. For this case width and cross-sectional area can be related to depth y by $B = 3 + y$ and $A = 3y + \frac{y^2}{2}$. Solve it for critical depth.

Published with MATLAB® R2016b

7 Numerical Solution to Ordinary Differential Equations

Topic learning outcomes:

Topic learning outcomes:

Student will be able to:

1. Find the value of function (dependent variable) at different values of independent variable by numerically solving the differential equation.

2. Write script and function files to solve application problems.

Syntax and description:

- `[t,y] = ode45(odefun, tspan,y0)`, where `tspan = [t0 tf]` - integrates the system of differential equations $y' = f(t, y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`. (Solves nonstiff differential equations medium order method).
- `[t,y] = ode23(odefun, tspan,y0)`, where `tspan = [t0 tf]` - integrates the system of differential equations $y' = f(t, y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`. (Solves nonstiff differential equations low order method).
- `[t,y] = ode113(odefun, tspan,y0)`, where `tspan = [t0 tf]` - integrates the system of differential equations $y' = f(t, y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`. (Solve nonstiff differential equations variable order method).
- `[t,y] = ode15s(odefun, tspan, y0)`, where `tspan = [t0 tf]` - integrates the system of differential equations $y' = f(t, y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`. (Solve stiff differential equations and differential-algebraic equations (DAEs) variable order method).

Note: Stiff differential equation - A differential equation of the form $y' = f(t, y)$ is said to be stiff if its exact solution $y(t)$ includes a term that decays exponentially to zero as t increases, but whose derivatives are much greater in magnitude than the term itself. An example of such a term is e^{-ct}, where c is a large, positive constant, because its k^{th} derivative is $c^k e^{-ct}$. Because of the factor of c^k, this derivative decays to zero much more slowly than e^{-ct} as t increases. Because the error includes a term of this form, evaluated at a time less than t, the error can be quite large if h is not chosen sufficiently small to offset this large derivative. Furthermore, the larger c is, the smaller h must be to maintain accuracy.

Example 7.1. *Solve the initial value problem* $\frac{dy}{dx} = 2y + 3e^x, y(0) = 0$ *for* x *= 0.1, 0.2, 0.3.*

```
xspan = [0:0.1:0.3];
y0 = 0;
[x,y] = ode45(@(x,y) 2*y+3*exp(x), xspan, y0)
```

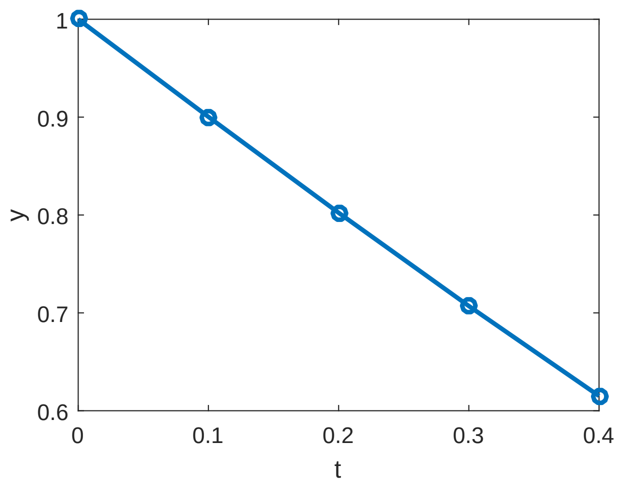
```
x =

0
```

```
  0.1000
  0.2000
  0.3000

  y =

  0
  0.3487
  0.8113
  1.4168
```

Example 7.2. *Solve the initial value problem* $\frac{dy}{dt} = t^2y - 1, y(0) = 1$ *for t = 0.1 to 0.4.*

```
tspan = [0:0.1:0.4];
y0 = 1;
[t,y] = ode23(@(t,y) t^2*y-1, tspan, y0)
plot(t,y,'-o')
```

```
 t =

 0
 0.1000
 0.2000
 0.3000
 0.4000
 y =

   1.0000
 0.9003
 0.8023
 0.7070
 0.6151
```



Example 7.3. *A 1500 gallon tank initially contains 600 gallons of water with 5 lbs of salt dissolved in it. Water enters the tank at a rate of 9 gallon per hour and the water entering the tank has a salt concentration of* $\frac{1}{5}(1+\cos t)$ *lbs/gal. If a well mixed solution leaves the tank at a rate of 6 gal/hr. If amount of salt at any time* t *is* $Q(t)$ *then the IVP for this situation is* $\frac{dQ}{dt}+\frac{2Q}{200+t}=\frac{9}{5}(1+\cos t)$, $Q(0)=5$ *and it overflows at t = 300 hrs. How much salt is in the tank when it overflows?*

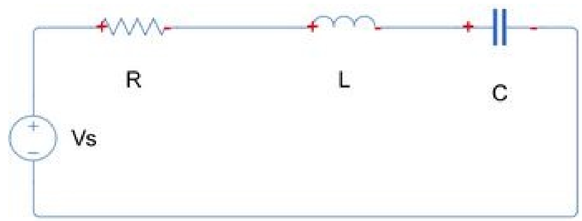
```
tspan = [0:150:300];
Q0 = 5;
[t,Q] = ode23(@(t,Q) (9*(1+cos(t))/5)-2*Q/(200+t), tspan, Q0)
```

```
t =

0
150
300

Q =

5.0000
171.1931
279.8302
```

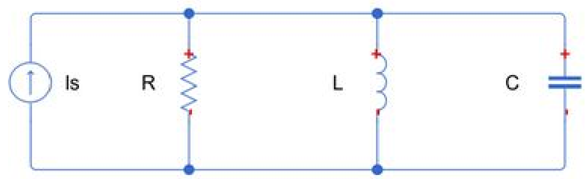
Higher Order Differential Equations - Application Examples

RLC Circuit

Consider the series RLC circuit below:



This circuit can also be transformed into a parallel RLC circuit with a current source I_S.



The voltage across the inductor and the capacitor are equal.

$$V_L(t) = V_C(t)$$
$$L\frac{dI_L}{dt} = V_C(t)$$
$$\frac{dI_L}{dt} = \frac{1}{L}V_C(t)$$

By Kirchoff's current law:

$$I_S = I_R + I_L + I_C$$

Rearranging:

$$I_C + I_L + I_R - I_S = 0$$
$$C\frac{dV_C}{dt} + I_L + \frac{V_C}{R} - I_S = 0$$
$$\frac{dV_C}{dt} = \frac{I_S}{C} - \frac{I_L}{C} - \frac{V_C}{RC}$$

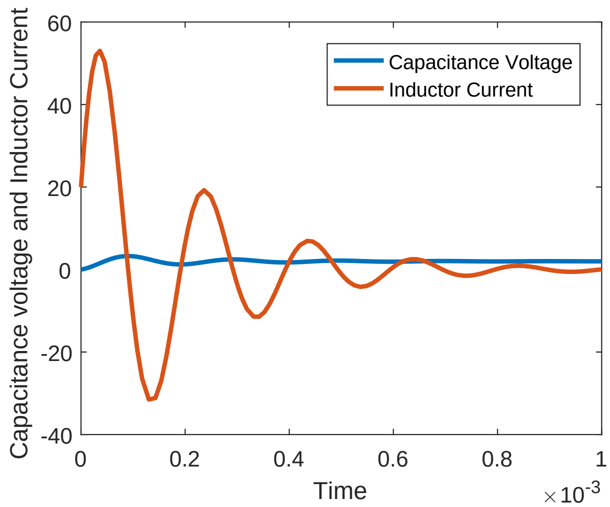
Assuming the initial conditions:

$$x_1(t) = I_L(t), x_2(t) = V_C(t)$$
$$\frac{dx_1}{dt} = \frac{1}{L}V_C(t) = \frac{1}{L}x_2(t)$$
$$\frac{dx_2}{dt} = \frac{dV_C}{dt} = \frac{I_S}{C} - \frac{x_1}{C} - \frac{x_2}{RC}$$

Implementing this in a function RLC (found below) which is used in the following script:

```
t0 = 0;             %Start Time
tf = 0.001;         %Stop Time
x0 = [0 20];        %Initial Conditions for Capacitance Voltage and Inductor Curr

%Using ODE23 to solve the set of equations
[t,x] = ode23(@RLC,[t0,tf],x0);
plot(t,x);
```

```
legend('Capacitance Voltage','Inductor Current');
xlabel('Time')
ylabel('Capacitance voltage and Inductor Current');
function xdot = RLC(t,x)
```



This function provides the xdot parameter for a second order RLC circuit

t is the time x is the initial condition (Capacitance voltage and inductor current) x is a 2 element vector - $x(1)$ is the capacitor voltage and $x(2)$ is the inductor current

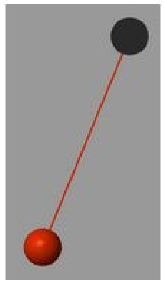
```
% Source Current
Is = 2;

%RLC values
R = 100;
L = 1e-3;
C = 1e-6;

%Calculating dx1/dt and dx2/dt and putting them into a column vector
xdot(1) = 1/L * x(2);
xdot(2) = 1/C * Is - 1/C*x(1) - 1/(R*C)*x(2);
xdot = xdot';

end
```

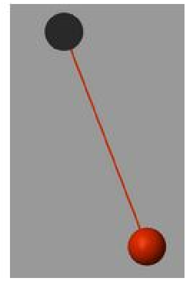
Simple Pendulum

The equation of oscillation of a simple pendulum is given by:

$$\frac{d^2\theta}{dt^2} = \frac{-g}{L}\sin\theta$$

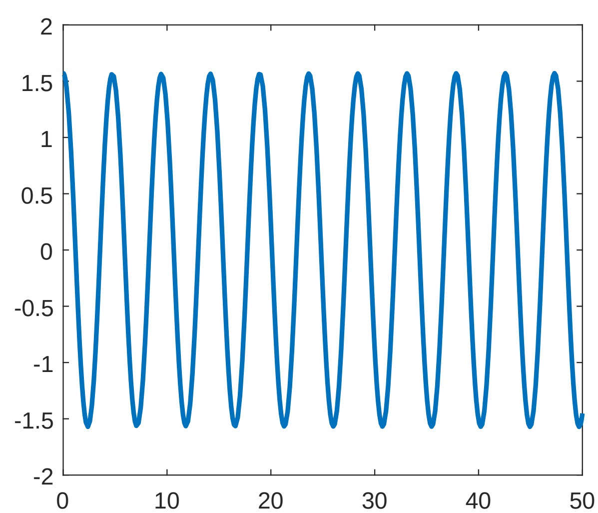
where g is the gravitation constant and L is the length of the pendulum.





We will use the built-in function `ode45` to solve the equation after setting up the differential equation in the above form.

```
%========= Sets initial parameters for pendulum ===========================
g = 9.81;  % Gravity (ms-2)
L = 4;  % pendulum length (m)
initialangle1 = pi/2;  % Initial angle 1
initialangle2 = 0;   % Initial angle 2
runtime = 50;  % Runs simulations for this time

%============== Solves simple pendulum differential equations =============
deq1=@(t,x) [x(2); -g/L * sin(x(1))]; % Pendulum equations uncoupled
[t,sol] = ode45(deq1,[0 runtime],[initialangle1 initialangle2]);
% uses a numerical ode solver
plot(t,sol(:,1));
```



Exercise:

1. Solve $\frac{dy}{dt} = \frac{t^3-2y}{t}, 1 \le t \le 3$ with $y = 4.2$ at $t = 1$.

2. An aeroplane uses a parachute and other means of braking as it slows down on the runway after landing. Its acceleration is given by $a = -0.0045v^2 - 3\text{m/s}^2$. Since $a = \frac{dv}{dt}$, the rate of change of

velocity is given by: $\frac{dv}{dt} = -0.0035v^2 - 3$. Consider an aeroplane with a velocity of 300 km/h that opens its parachute and starts decelerating at $t = 0$s. By solving the differential equation, determine and plot the velocity as a function of time from $t = 0$s until the aeroplane stops.

3. Solve the initial value problem $\frac{dy}{dx} = \frac{y^2-x^2}{y^2+x^2}$ with $y(0) = 1$ at $x = 0.2, 0.4$.

4. The number N of bacteria in a culture grew at a rate proportional to N. The value of N was initially 100 and increased to 332 in 1 hr. The governing equation is $\frac{dN}{dt} = log(3.32)N, N(0) = 100$. What is the value of N at time t = 1.5 hr.

RV Educational Institutions ®
RV College of Engineering ®
Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi

Go, change the world

DEPARTMENT OF MATHEMATICS

VISION

Impart outcome based education, focusing on quality teaching with an emphasis on Mathematical modeling and sustainable interdisciplinary research

MISSION

- *To create infrastructure and resources for quality teaching*
- *To train graduates with mathematical background for solving complex engineering problems*
- *Enable to acquire latest skills of experiential learning for problem analysis and develop solutions*
- *Create an ambience for sustainable interdisciplinary research*