Experiment: 1
Evaluation of Multiple Integrals

AIM:

The main objective of this experiment is to

- Understanding the tool to find double integrals with constant limits of integration
- Evaluation of double integrals with variable limits of integration.
- Evaluation of double integrals with change of order of Integration.
- Evaluation of triple integrals using the tool.

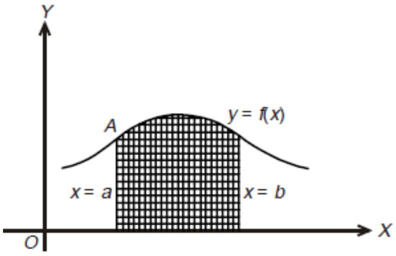
Introduction

MATLAB offers good integral calculus support. Using MATLAB, we can effectively evaluate double and triple integrals with constant limits as well as variable limits of integration.

Mathematical Background

Definite Integral:

An expression of the form $\int_a^b f(x)\,dx$ is called definite integral and it is physically mean the area under a curve $y = f(x)$, the x-axis and the two ordinates $x = a \text{ and } x = b$



MATLAB commands required:

MATLAB Syntax	Description
`int(S,a,b)`	If the expression `S` contains one symbolic variable, definite integration is carried out between the limits `a` and `b`. The limits can be numbers or symbolic variables.
`int(S,var,a,b)`	Definite integration is carried out with respect to the variable `var` between the limits `a` and `b`.

MATLAB code for evaluation for definite integral

```
syms x
fun=input('Enter the function f(x):');
Xlim=input('Enter Xlimits in vector form:');
Sol=int(fun,Xlim(1),Xlim(2))
```

Examples:

Evaluate following definite integrals using MATLAB

1. $\int_0^{\pi} \cos x \, dx$

Input

```
Enter the function f(x):
cos(x)
Enter Xlimits in vector form:
[0 pi]
```

Output

```
Sol =
 0
```

2. $\int_0^{-2} \left(\frac{-2x}{(1+x^2)^2} \right) dx$

Input

```
Enter the function f(x):
-2*x/(1+x^2)^2
Enter Xlimits in vector form:
[0 -2]
```

Output

```
Sol =
 -4/5
```

3. $\int_0^4 (x^4 - x^2 + x) dx$

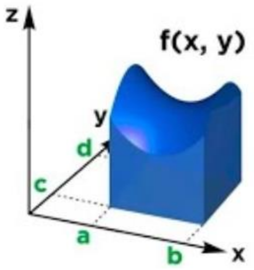
Input

```
Enter the function f(x):
x^4-x^2+x
Enter Xlimits in vector form:
[0 4]
```

Output
```
Sol =
 2872/15
```

Double Integral:

The generalization of definite integral to two dimensions is called double integral and it is denoted by $\iint_R f(x,y)\,dx\,dy = \int_{y=c}^{y=d}\int_{x=a}^{x=b} f(x,y)\,dx\,dy$. This double integral gives volume under the surface $f(x,y)$.



MATLAB commands required for evaluation of double integral with constant limits

MATLAB Syntax	Description
`int(int(f, x, a, b),y,c,d)`	If the expression `f` contains two symbolic variable, inner integration is carried out w.r.t `x` between the limits `a` and `b`. The resultant expression is then integrated w.r.t `y` between the limits `c` and `d`.

MATLAB code for evaluation for definite integral with constant limits

```
syms x y a b
f=input('Enter the function f(x,y):');
Xlim=input('Enter Xlimits in vector form:');
Ylim=input('Enter Ylimits in vector form:');
Sol=int(int(f,x,Xlim(1),Xlim(2)),y,Ylim(1),Ylim(2))
```

Examples:

Evaluate following double integrals using MATLAB

1. $\int_{y=0}^{2}\int_{x=0}^{3} xy\,dx\,dy$

Input
```
Enter the function f(x,y):
x*y
Enter Xlimits in vector form:
[0 3]
Enter Ylimits in vector form:
[0 2]
```

Output
```
Sol =
 9
```

2. $\int_{y=0}^{1}\int_{x=0}^{1}\frac{1}{\sqrt{(1-x^2)(1-y^2)}}\,dx\,dy$

Input

```
Enter the function f(x,y):
1/sqrt((1-x^2)*(1-y^2))
Enter Xlimits in vector form:
[0 1]
Enter Ylimits in vector form:
[0 1]
```

Output

```
Sol =
 pi^2/4
```

3. $\int_{x=0}^{a}\int_{y=0}^{b}(x^2+y^2)\,dx\,dy$

Input

```
Enter the function f(x,y):
x^2+y^2
Enter Xlimits in vector form:
[0 a]
Enter Ylimits in vector form:
[0 b]
```

Output

```
Sol =
 (a*b*(a^2 + b^2))/3
```

4. $\int_{0}^{\infty}\int_{0}^{\infty}e^{-(x^2+y^2)}\,dx\,dy$

Input

```
Enter the function f(x,y):
exp(-(x^2+y^2))
Enter Xlimits in vector form:
[0 inf]
Enter Ylimits in vector form:
[0 inf]
```

Output

```
Sol =
 pi/4
```

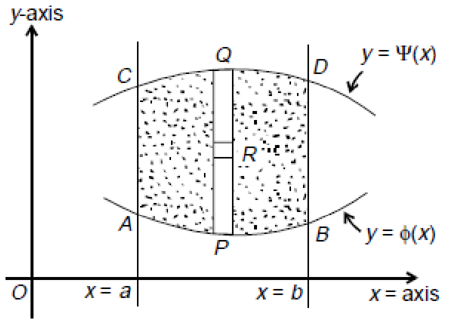
Evaluation of double integrals with variable limits of integration.

Mathematical Background:

Type I:

Evaluation of double integral $\iint_R f(x,y)\,dx\,dy$ when the region R is bounded by two continuous curves $y=\psi(x)$ & $y=\phi(x)$ and the two lines (ordinates) $x=a$ and $x=b$. In such a case, integration is first performed with respect to y keeping x as a constant and then the resulting integral is integrated within the limits $x=a$ and $x=b$ .Mathematically expressed as

$$\iint_R f(x,y)\,dx\,dy=\int_{x=a}^{x=b}\left(\int_{y=\phi(x)}^{y=\psi(x)} f(x,y)\,dy\right)dx$$

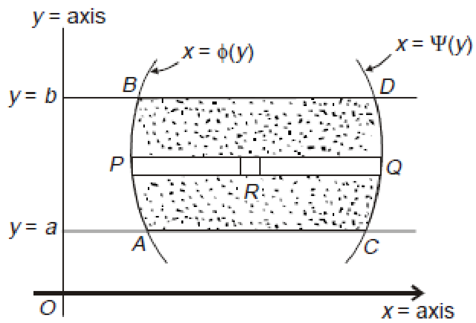


Geometrically, the process is shown in right side figure, where integration is carried out from inner rectangle (i.e., along the one edge of the 'vertical strip PQ' from P to Q) to the outer rectangle.

Type II:

Evaluation of double integral $\iint_R f(x,y)\,dx\,dy$ when the region R is bounded by two continuous curves $x=\phi(y)\,\&\,x=\psi(y)$ and the two lines (ordinates) $y=a$ and $y=b$. In such a case, integration is first performed with respect to x keeping y as a constant and then the resulting integral is integrated within the limits $y=a$ and $y=b$ .Mathematically expressed as

$$\iint_R f(x,y)\,dx\,dy=\int_{y=a}^{y=b}\left(\int_{x=\phi(y)}^{x=\psi(y)} f(x,y)\,dx\right)dy$$



Geometrically, the process is shown in right side figure, where integration is carried out from inner rectangle (i.e., along the one edge of the 'horizontal PQ' from P to Q) to the outer rectangle.

MATLAB commands required for evaluation of double integral with variable limits

MATLAB Syntax	Description
`For Type I Integrals:` `int(int(f,y,phi(x),psi(x)),x,a,b)` where `y` is the inner variable, `x` is the outer variable.	For the expression `f`, inner integration is carried out w.r.t `y` between the limits $y=\phi(x)$ to $y=\psi(x)$. The resultant expression is then integrated w.r.t `x` between the limits `a` and `b`.
`For Type II Integrals:` `int(int(f,x,phi(y),psi(y)),y,a,b)` where `x` is the inner variable, `y` is the outer variable.	For the expression `f`, inner integration is carried out w.r.t `x` between the limits $x=\phi(y)\,\&\,x=\psi(y)$. The resultant expression is then integrated w.r.t `y` between the limits `a` and `b`.

Supporting files required

To visualize the surfaces, two additional m-files namely `viewSolid`, `viewSolidone` are required. These files are to be included in the current working directory before execution.

Syntax for double integral:

For Type I:

```
viewSolid(z,0+0*x+0*y,f,y,phi(x),psi(x),x,a,b)
```

For Type II:

```
viewSolidone (z,0+0*x+0*y,f,x,phi(y),psi(y),y,a,b)
```

MATLAB code for evaluation for definite integral of type I with variable limits

```
syms x y z
f=input('Enter the function f(x,y):');
Xlim=input('Enter Xlimits in vector form:');
Ylim=input('Enter Ylimits in vector form:');
Sol=int(int(f,y,Ylim(1),Ylim(2)),x,Xlim(1),Xlim(2))
viewSolid(z,0+0*x+0*y,f,y,Ylim(1),Ylim(2),x,Xlim(1),Xlim(2))
```

Examples:

Evaluate following double integrals using MATLAB

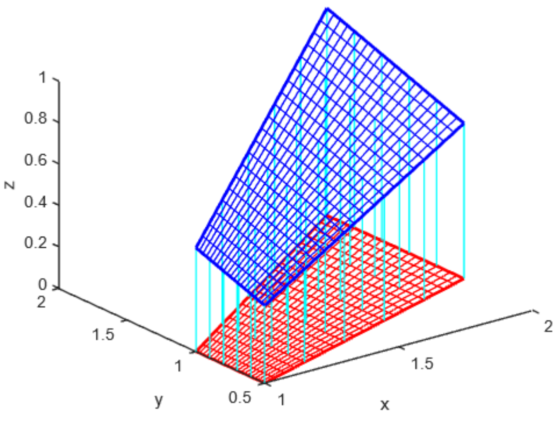
1. $\int_{1}^{2}\int_{\frac{x}{2}}^{x}\left(\frac{x+y}{4}\right)dx\,dy$

Input

```
Enter the function f(x,y):
(x+y)/4
Enter Xlimits in vector form:
[1 2]
Enter Ylimits in vector form:
[x/2 x]
```

Output

```
Sol =
 49/96
```



In this figure the required volume is above the plane $z=0$ (shown in red) and above the surface (shown in green) $z=\frac{x+y}{4}$

2. Evaluate $\iint_R (x^2+y^2)dxdy$ in the positive quadrant for which $x=0, y=0,\ y=1-x$. Hence

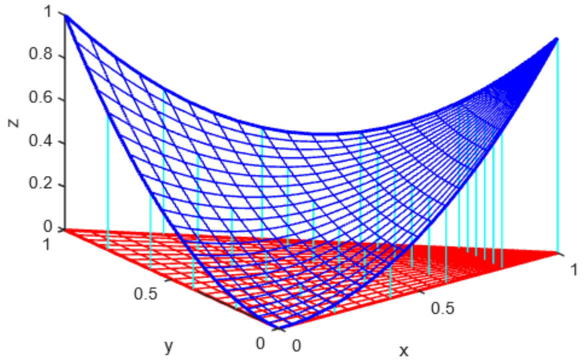
$$\iint_R (x^2+y^2)dxdy = \int_{x=0}^{1}\int_{y=0}^{1-x}(x^2+y^2)dx\,dy$$

Input

```
Enter the function f(x,y):
x^2+y^2
Enter Xlimits in vector form:
[0 1]
Enter Ylimits in vector form:
[0 1-x]
```

Output

```
Sol =
 1/6
```



MATLAB code for evaluation for definite integral of type II with variable limits

```
syms x y z
f=input('Enter the function f(x,y):');
Xlim=input('Enter Xlimits in vector form:');
Ylim=input('Enter Ylimits in vector form:');
Sol=int(int(f,x,Xlim(1),Xlim(2)),y,Ylim(1),Ylim(2))
viewSolidone(z,0+0*x+0*y,f,x,Xlim(1),Xlim(2),y,Ylim(1),Ylim(2))
```

Examples:

1. To find the volume of the prism whose base is the triangle in the $xy-$plane bounded by the $x-$ axis and the lines $y = x$ & $x = 1$ and whose top lies in the plane $z = f(x, y) = 3 - x - y$. The limits of integration here are $y = 0$ & $y = 1$ while $x = y$ & $x = 1$. Hence

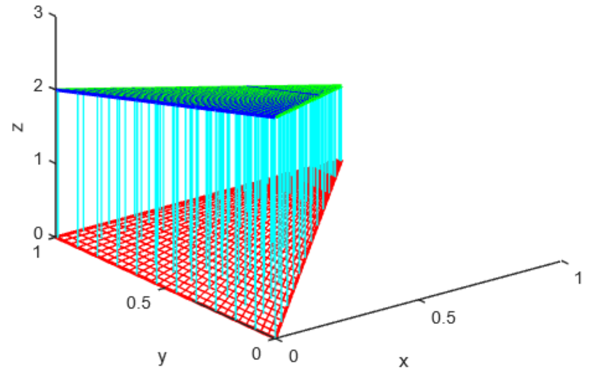
$$\iint_R (3-x-y)dxdy = \int_0^1\int_y^1 (3-x-y)\,dx\,dy$$

Input

```
Enter the function f(x,y):
3-x-y
Enter Xlimits in vector form:
[y 1]
Enter Ylimits in vector form:
[0 1]
```

Output

```
Sol =
 1
```



In this figure the triangular region on the $xy-$plane is shown in red, while the plane surface $z = f(x, y) = 3 - x - y$ above the $xy-$plane is shown in green.

Evaluation of double integrals with changing of variables:

MATLAB Code:

```
syms r theta
x=r*cos(theta);
y=r*sin(theta);
Q=input('Enter the integrand in terms of x and y:')
Q=Q*r;
Rlim=input('Enter the r limits as a vector form:');
thetalim=input('Enter the theta limits as a vector form:');
I1=int(Q,r,Rlim(1),Rlim(2));
Sol=int(I1,theta, thetalim(1), thetalim(2))
```

Examples:

1. Evaluate $\int_0^2 \int_0^{\sqrt{2x-x^2}} (x^2+y^2)dydx$ by changing into polar coordinates.

Input	Output
Enter the integrand in terms of x and y: x^2+y^2 Enter the r limits as a vector form: [0 2*cos(theta)] Enter the theta limits as a vector form: [0 pi/2]	Sol = (3*pi)/4

2. Evaluate $\int_0^\infty \int_0^\infty e^{-(x^2+y^2)}dxdy$ by changing into polar coordinates.

Input	Output
Enter the integrand in terms of x and y: Exp(-x^2-y^2) Enter the r limits as a vector form: [0 inf] Enter the theta limits as a vector form: [0 pi/2]	Sol = pi/4

Evaluation of double integrals with changing of order of integration:

MATLAB Code:

```
syms x y z
f=input('Enter the function f(x,y):');
Xlim=input('Enter Xlimits in vector form:');
Ylim=input('Enter Ylimits in vector form:');
Sol=int(int(f,y,Ylim(1),Ylim(2)),x,Xlim(1),Xlim(2))
viewSolid(z,0+0*x+0*y,f,y,Ylim(1),Ylim(2),x,Xlim(1),Xlim(2))
```

Examples:

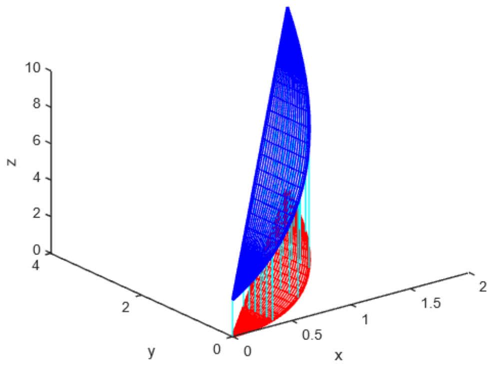
1. Evaluate $\int_0^2 \int_{x^2}^{2x} (4x+2)dydx$ by change of order of integration.

Input

```
Enter the function f(x,y):
4*x+2
Enter Xlimits in vector form:
[0 2]
Enter Ylimits in vector form:
[x^2 2*x]
```

Output

```
Sol =
 8
```



By changing the order of integration, the limits are $y = 0 \ to \ 4 \ \ \& \ x = \frac{y}{2} to \ \sqrt{y}$

MATLAB Code:

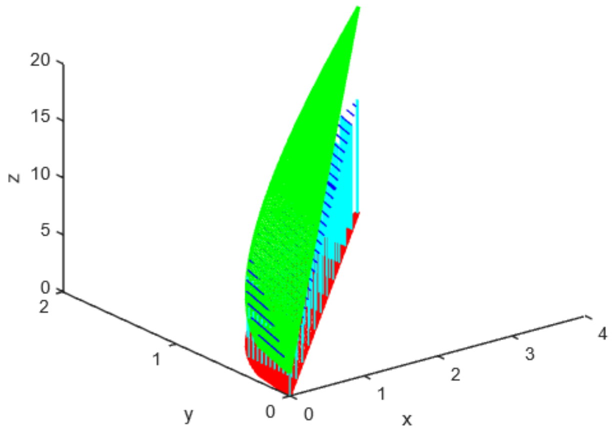
```
syms x y z
f=input('Enter the function f(x,y):');
Xlim=input('Enter Xlimits in vector form:');
Ylim=input('Enter Ylimits in vector form:');
Sol=int(int(f,x,Xlim(1),Xlim(2)),y,Ylim(1),Ylim(2))
viewSolidone(z,0+0*x+0*y,f,x,Xlim(1),Xlim(2),y,Ylim(1),Ylim(2))
```

Input

```
Enter the function f(x,y):
4*x+2
Enter Xlimits in vector form:
[y/2  sqrt(y)]
Enter Ylimits in vector form:
[0  4]
```

Output

```
Sol =
 8
```



Evaluation of triple integrals

Triple integrals enable us to solve more general problems such as to calculate the volumes of three – dimensional shapes, the masses and moments of solids of varying density, and the average value of a function over a three –dimensional region.

The triple integral of $f(x,y,z)$ over the region D is given by

$$\iiint_D f(x,y,z)dV$$

Where the region D is bounded by the surfaces $x=a \;\; to \;\; x=b, y=\psi_1(x)$ to $y=\psi_2(x)$ and $z=\phi_1(x,y)$ to $z=\phi_2(x,y)$

Hence

$$\iiint_D f(x,y,z)dV = \int_a^b \int_{\psi_1(x)}^{\psi_2(x)} \int_{\phi_1(x,y)}^{\phi_2(x,y)} f(x,y,z)\,dz\,dy\,dx. \qquad \text{(Type 1)}$$

Similarly, when the region D is bounded by the surfaces $y=c \;\; to \;\; y=d, \mathrm{x}=\psi_1(\mathrm{y})$ to $x=\psi_2(\mathrm{y})$ and $z=\phi_1(x,y)$ to $z=\phi_2(x,y)$

Hence

$$\iiint_D f(x,y,z)dV = \int_c^d \int_{\psi_1(y)}^{\psi_2(y)} \int_{\phi_1(x,y)}^{\phi_2(x,y)} f(x,y,z)\,dz\,dx\,dy. \qquad \text{(Type 2)}$$

MATLAB commands required for evaluation of triple integral with variable limits

MATLAB Syntax	Description
For Type I Integrals: `int(int(int(f,z,za,zb),y,ya,yb),x,xa,xb)` where `z` is the inner variable, `y` is the next to inner variable and `x` is the outer variable.	For the expression `f`, inner integration is carried out w.r.t `z` between the limits $z=\phi_1(x,y)$ to $z=\phi_2(x,y)$. The resultant expression is then integrated w.r.t `y` between the limits $y=\psi_1(x)$ to $y=\psi_2(x)$ and then final resultant expression is integrated w.r.t `x` between the limits `a` and `b`.
For Type II Integrals: `int(int(int(f,z,za,zb),x,xa,xb),y,ya,yb)`	For the expression `f`, inner integration is carried out w.r.t `z` between the limits

where z is the inner variable, y is the next to inner variable and x is the outer variable.

$z=\phi_1(x,y)$ to $z=\phi_2(x,y)$. The resultant expression is then integrated w.r.t x between the limits $x=\psi_1(y)$ to $x=\psi_2(y)$ and then final resultant expression is integrated w.r.t y between the limits c and d.

Supporting files required

To visualize the surfaces two additional m-files viz., `viewSolid`, `viewSolidone` are required. These files are to be included in the current working directory before execution.

Syntax for double integral:

For Type I:

```
viewSolid(z,za,zb,y,ya,yb,x,xa,xb)
```

For Type II:

```
viewSolidone(z,za,zb,x,xa,xb,y,ya,yb)
```

MATLAB code for evaluation for triple integrals

```
syms x y z
f=input('Enter the function f(x,y,z):');
Xlim=input('Enter Xlimits in vector form:');
Ylim=input('Enter Ylimits in vector form:');
Zlim=input('Enter Zlimits in vector form:');
I=int(int(int(f,z,Zlim(1),Zlim(2)),y,Ylim(1),Ylim(2)),x,Xlim(1),Xlim(2))
viewSolid(z,Zlim(1),Zlim(2),y,Ylim(1),Ylim(2),x,Xlim(1),Xlim(2))
```

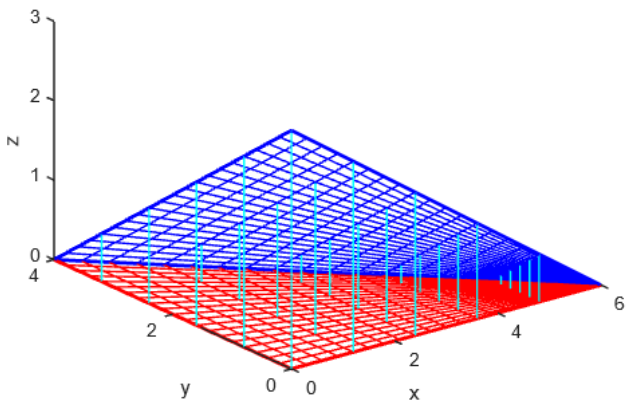
Examples

1. Evaluate $\iiint_V dx\,dy\,dz$ where V is the finite region of space formed by the planes $x=0, y=0, z=0,$ and $2x+3y+4z=12$

Clearly, the limits are $x=0\ \ to\ 6;\ y=0\ to\ \frac{1}{3}(12-2x)\ \&\ z=0\ to\ \frac{1}{4}(12-2x-3y)$

Input

```
Enter the function f(x,y,z):
1
Enter Xlimits in vector form:
[0 6]
Enter Ylimits in vector form:
[0 (1/3)*(12-2*x)]
Enter Zlimits in vector form:
[0 (1/4)*(12-2*x-3*y)]
```



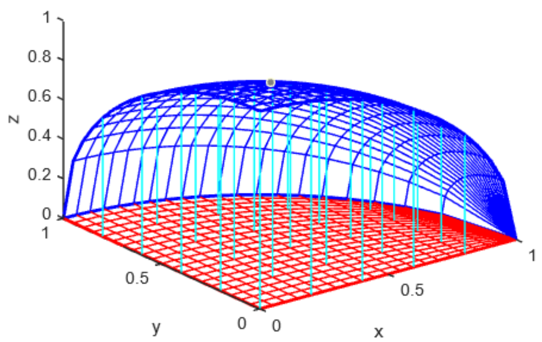
Output

```
I =
 12
```

2. Evaluate $\int_0^1 \int_0^{\sqrt{1-x^2}} \int_0^{\sqrt{1-x^2-y^2}} \frac{1}{\sqrt{1-x^2-y^2-z^2}}\, dz\, dy\, dx$ using MATLAB

Input

```
Enter the function f(x,y,z):
1/(sqrt(1-x^2-y^2-z^2))
Enter Xlimits in vector form:
[0 1]
Enter Ylimits in vector form:
[0 sqrt(1-x^2)]
Enter Zlimits in vector form:
[0 sqrt(1-x^2-y^2)]
```



Output

```
I =
 pi^2/8
```

Exercise Problems

1. Evaluate $\int_0^1 \int_0^1 \frac{1}{\sqrt{(1-x^2)(1-y^2)}}\, dx\, dy$ using MATLAB
2. Evaluate $\int_0^1 \int_x^{\sqrt{x}} (x^2+y^2)\, dx\, dy$ using MATLAB
3. Evaluate $\int_0^1 \int_0^{\sqrt{1-y^2}} y^2\, dy dx$ using MATLAB
4. By changing into polar coordinates, evaluate $\iint \frac{x^2 y^2}{x^2+y^2}\, dx\, dy$ over the annular region b/w the Circles $x^2+y^2=1$ and $x^2+y^2=4$.
5. Evaluate $\int_0^1 \int_0^{\sqrt{1-x^2}} \int_0^{\sqrt{1-x^2-y^2}} \frac{1}{\sqrt{1-x^2-y^2}}\, dz\, dy\, dx$ using MATLAB
6. Using MATLAB, find the volume of the region cut from the cylinder $x^2+y^2=4$ by the planes $z=0$ and $x+z=3$.

Experiment: 2
Solving Higher Order Differential Equations

AIM:

The main objective of this experiment is to

- Analytical solution of higher order differential equations
- Numerical solution of higher order differential equations
- Plotting the solutions and interpretation of the same

Introduction

MATLAB offers good differential calculus support. Using MATLAB, we can effectively solve linear and higher order differential equations.

Higher order linear differential equations with constant coefficients

The general form of n^{th} order linear differential equations with constant coefficients is given by

$$a_0 \frac{d^n y}{dx^n} + a_1 \frac{d^{n-1} y}{dx^{n-1}} + a_2 \frac{d^{n-2} y}{dx^{n-2}} + \ldots + a_{n-1} \frac{dy}{dx} + a_n y = f(x)$$

Where a_i's are constants. $f(x)$ may be a function of x or a constant and $a_0 \neq 0$. The general solution of above differential equation is given by

$$\text{Complete solution=Complementary function} + \text{Particular integral}$$
$$\Rightarrow \text{y=C.F+P.I} \Rightarrow \text{y}_c + y_p$$

Eg:

$\frac{d^3 y}{dx^3} + 7\frac{d^2 y}{dx^2} + 8\frac{dy}{dx} - 2y = x$ is a third order linear differential equation with constant coefficients.

MATLAB commands required:

MATLAB Syntax	Description
`syms variable_name variable_name`	Creates multiple symbolic variables
`input(PROMPT)`	Displays the PROMPT string on the screen, waits for input from the keyboard
`diff(y,x)`	Differentiates a symbolic expression *y* with respect to its free variable *x*.
`diff(y,x,n)`	Differentiates symbolic expression *y*, *n* times with respect to its free variable *x*.
`dsolve(ode)`	Provides Symbolic general solution of ordinary differential equation `ode`.
`dsolve(ode,cond)`	Provides particular solution of ordinary differential equation `ode` with initial conditions.
`Simplify(S)`	Returns the expression *S* in simplified form

Analytical Methods: MATLAB code for solving higher order linear differential equations with constant coefficients

```
syms y(x)
ode=input('Enter the differential equation:');
ySol(x) = dsolve(ode);
ySol(x) = simplify(ySol(x))
```

Examples

1. Solve $\frac{d^2y}{dx^2}-8\frac{dy}{dx}+15y=0$

Input

```
Enter the differential equation:

diff(y,x,2)-8*diff(y,x)+15*y == 0
```

Output

```
ySol(x) =

C1*exp(3*x) + C2*exp(5*x)
```

2. Solve $\frac{d^2y}{dx^2}-4\frac{dy}{dx}+4y=x^3e^{2x}$

Input

```
Enter the differential equation:

diff(y,x,2)-4*diff(y,x)+4*y ==
x^3*exp(2*x)
```

Output

```
ySol(x) =

(x^5*exp(2*x))/20 +
C1*exp(2*x) + C2*x*exp(2*x)
```

3. Solve $\frac{d^3y}{dx^3}+3\frac{d^2y}{dx^2}+3\frac{dy}{dx}+y=e^{-x}$

Input

```
Enter the differential equation:

diff(y,x,3)+3*diff(y,x,2)+3*diff(y,x)+y
== exp(-x)
```

Output

```
ySol(x) =

(exp(-x)*(x^3 + 6*C3*x^2 +
6*C2*x + 6*C1))/6
```

4. Solve $\frac{d^3y}{dx^3}-7\frac{d^2y}{dx^2}+10\frac{dy}{dx}=e^{2x}\sin x$

Input

```
Enter the differential equation:

diff(y,x,3)-7*diff(y,x,2)+10*diff(y,x)
== exp(2*x)*sin(x)
```

Output

```
ySol(x) =

(7*exp(2*x)*cos(x))/50 -
C1/10 + (exp(2*x)*sin(x))/50
+ C2*exp(2*x) + C3*exp(5*x)
```

MATLAB code for solving higher order linear differential equations with constant coefficients and initial conditions

1. Solve $\frac{d^2y}{dx^2} = \cos(2x) - y$ with initial conditions $y(0) = 1, \&\ y'(0) = 0$

MATLAB code

```
syms y(x)
ode = input('Enter the differential equation:');
Dy = diff(y);
cond1 = y(0) == 1;
cond2 = Dy(0) == 0;
conds = [cond1 cond2];
ySol(x) = dsolve(ode,conds);
ySol = simplify(ySol)
```

Input

```
Enter the differential equation:

diff(y,x,2)== cos(2*x)-y
```

Output

```
ySol(x) =

1 - (8*sin(x/2)^4)/3
```

2. Solve $\frac{d^3y}{dx^3} - 7\frac{d^2y}{dx^2} + 10\frac{dy}{dx} = e^{2x}\sin x$ with initial conditions $y(0) = 1, y'(0) = -1 \&\ y''(0) = 0$

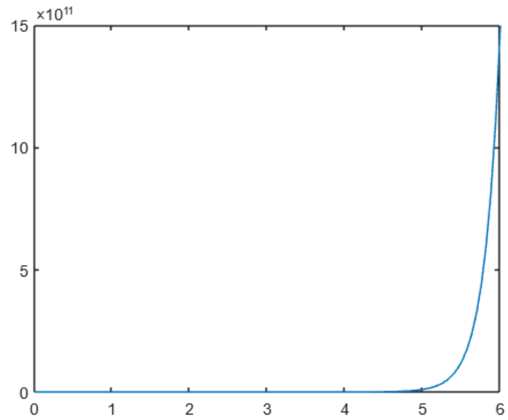
```
syms y(x)
ode = input('Enter the differential equation:');
Dy = diff(y);
DDy = diff(Dy);
cond1 = y(0) == 1;
cond2 = Dy(0) == -1;
cond3 = DDy(0) == 0;
conds = [cond1 cond2 cond3];
ySol(x) = dsolve(ode,conds);
ySol = simplify(ySol)
tt=linspace(0,6,100);
yval=subs(ySol,tt);
plot(tt,yval)
```

Input

```
Enter the differential equation:

diff(y,x,3)-7*diff(y,x,2)+10*diff(y,x)
== exp(2*x)*sin(x)
```

Output

```
ySol(x) =

(7*exp(5*x))/50 - exp(2*x) +
(7*exp(2*x)*cos(x))/50 +
(exp(2*x)*sin(x))/50 + 43/25
```



Numerical Methods: MATLAB code for solving higher order linear differential equations with constant coefficients

MATLAB commands required:

MATLAB Syntax	Description
`odeToVectorField(eqn)`	Converts higher-order differential equation `eqn` to a system of first-order differential equations
`matlabFunction(r)`	Convert the symbolic expression `r` to a MATLAB function with the handle
`ode45(odefun,tspan, conds)` where `tspan = [t0 tf]`	Solves the system of differential equations `y'=f(t,y)` from `t0` to `tf` with initial conditions.
`deval(x,sol)`	Evaluates the solution `sol` of a differential equation at the points contained in `x`

1. Solve $\frac{d^3y}{dx^3} - 7\frac{d^2y}{dx^2} + 10\frac{dy}{dx} = e^{2x}\sin x$ with initial conditions $y(0)=1, y'(0)=-1 \,\&\; y''(0)=0$ numerically.

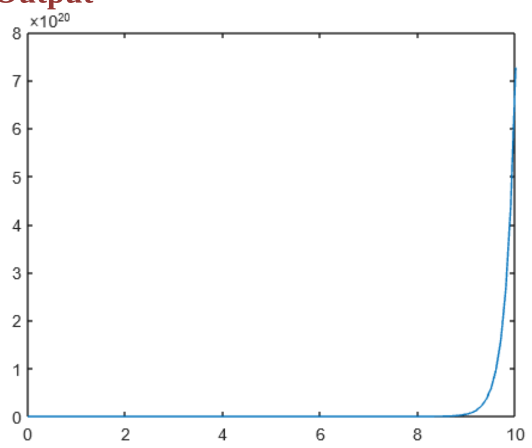
MATLAB code

```
syms y(x)
ode = input('Enter the differential equation:');
conds = [1 -1 0];
V = odeToVectorField(ode);
M = matlabFunction(V,'vars',{'x','Y'});
interval = [0 10];
ySoll = ode45(M,interval,);
tValues=linspace(0,10,100); conds
yValues = deval(ySoll,tValues,1);
plot(tValues,yValues)
```

Input

```
Enter the differential equation:

diff(y,x,3)-7*diff(y,x,2)+10*diff(y,x)
== exp(2*x)*sin(x)
```

Output



2. Solve $\frac{d^3y}{dx^3} - 7\frac{d^2y}{dx^2} + 10\frac{dy}{dx} = e^{2x}\sin x$ with initial conditions $y(0)=1, y'(0)=-1 \,\&\; y''(0)=0$ using analytical and numerical method and compare their solutions graphically.

Note:

To compare the analytical solution and numerical solution graphically, we are required to use two different Matlab codes. First we need to execute program 1, which give us the analytical solution and it's respective graph. It is noted that, the figure window that appear after executing the program 1 should not be closed, as it is required to plot numerical solution too.

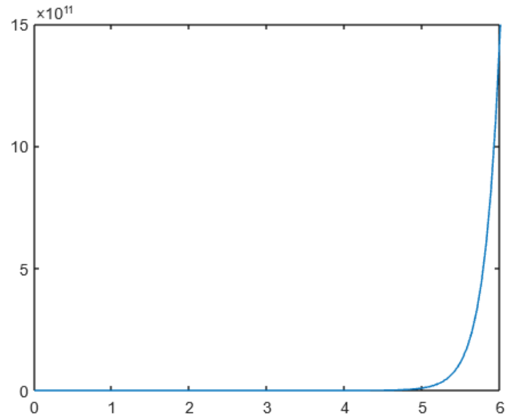
MATLAB code

Program:1

```
syms y(x)
ode = input('Enter the differential equation:');
Dy = diff(y);
DDy = diff(Dy);
cond1 = y(0) == 1;
cond2 = Dy(0) == -1;
cond3 = DDy(0) == 0;
conds = [cond1 cond2 cond3];
ySol(x) = dsolve(ode,conds);
ySol = simplify(ySol)
tt=linspace(0,6,100);
yval=subs(ySol,tt);
plot(tt,yval)
```

Input

```
Enter the differential equation:

diff(y,x,3)-7*diff(y,x,2)+10*diff(y,x)
== exp(2*x)*sin(x)
```



Output

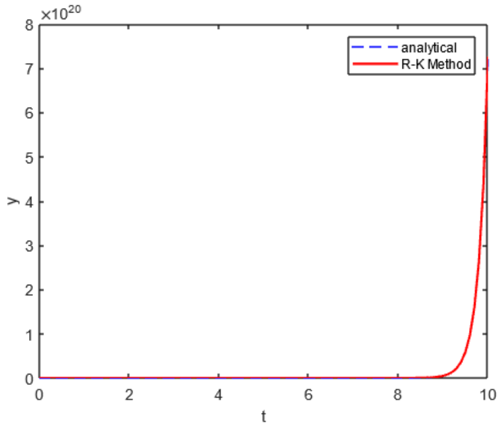
```
ySol(x) =

(7*exp(5*x))/50 - exp(2*x) +
(7*exp(2*x)*cos(x))/50 +
(exp(2*x)*sin(x))/50 + 43/25
```

Program 2

```
syms y(x)
ode = input('Enter the differential equation:');
conds = [1 -1 0];
V = odeToVectorField(ode);
M = matlabFunction(V,'vars',{'x','Y'});
interval = [0 10];
ySoll = ode45(M,interval,conds);
tValues=tt;
yValues = deval(ySoll,tValues,1);
plot(tValues,yValues,'-r','linewidth', 1.5)
legend('analytical','R-K Method')
```

Input

```
Enter the differential equation:

diff(y,x,3)-7*diff(y,x,2)+10*diff(y,x)
== exp(2*x)*sin(x)
```



Interpretation

In the above graph, it is seen that both the graphs of analytical solution and numerical solution overlapping each other. This indicates that the solution obtained by the numerical method is almost similar to the analytical method.

Exercise problems

1. Solve $\frac{d^3y}{dx^3}+2\frac{d^2y}{dx^2}-7\frac{dy}{dx}=0$
2. Solve $\frac{d^3y}{dx^3}+2\frac{d^2y}{dx^2}-5\frac{dy}{dx}=x$
3. Solve $\frac{d^2y}{dx^2}+\frac{dy}{dx}+2y=e^{2x}$ with initial conditions $y(0)=1, \& \ y'(0)=0$
4. Solve $4\frac{d^3y}{dx^3}+\frac{d^2y}{dx^2}-5\frac{dy}{dx}=xe^{2x}$ with initial conditions $y(1)=0, y'(1)=4 \ \& \ y''(1)=-1$
5. Solve $\frac{d^3y}{dx^3}+2\frac{d^2y}{dx^2}-5\frac{dy}{dx}+2y=e^x\log x$ with initial conditions $y(0)=1, y'(0)=2 \ \& \ y''(0)=-1$ and compare with numerical solution graphically.

Experiment: 3
Laplace Transforms through MATLAB

AIM:

The main objective of this experiment is to

- Find Laplace transforms of standard elementary functions.
- Find Laplace transforms of periodic, unit step and impulse functions using the tool.
- Evaluate the integrals using Laplace transforms.

Mathematical Background

Laplace Transform

The Laplace transform of a function $f(t)$ is defined as

$$\mathrm{F(s)} = \mathrm{L}\{f(t)\} = \int_0^\infty e^{-st} f(t)\, dt$$

Provided the integral exists. Conventionally, t denotes time and s denotes frequency.

MATLAB Commands required:

Following are new MATLAB commands one need to know for the present experiment.

MATLAB Syntax	Description
`laplace(f)`	To find the laplace transform of `f` with default independent variable `t`. The default return is a function of `s`.
`laplace(f,w)`	Returns the laplace transform of `f` in symbol `w` instead of default `s`.
`laplace(f,x,w)`	Assumes `f` as a function of `x` and returns the laplace transform in terms of `w`.
`heaviside(t-a)`	To input the Heaviside unit step function $u(t-a)$
`dirac(t-a)`	To input the dirac delta function $\delta(t-a)$

MATLAB code for finding Laplace transform

For default variables t and s: The Laplace transform of a function $f(t)$ in terms of s

```
clc
syms t a b   % Note that s is not declared as a symbol here
f=input('Enter the function in terms of t:');
```

```
F=laplace(f)
```

Examples

1. Find the Laplace transform for the following functions

(i) $\sin t$ (ii) $\cos t$ (iii) e^{at} (iv) t^3 (v) $e^{at} \sin bt$

Input

```
Enter the function in terms of t:
sin(t)
```

Output

```
F =
1/(s^2 + 1)
```

Input

```
Enter the function in terms of t:
cos(t)
```

Output

```
F =
s/(s^2 + 1)
```

Input

```
Enter the function in terms of t:
exp(a*t)
```

Output

```
F =
-1/(a-s)
```

Input

```
Enter the function in terms of t:
t^3
```

Output

```
F =
6/s^4
```

Input

```
Enter the function in terms of t:
exp(a*t)*sin(b*t)
```

Output

```
F =
b/(b^2+(a-s)^2)
```

For other variables: The Laplace transform of a function $f(x)$ in terms of w

MATLAB code

```
clc
syms x w
f=input('Enter the function in terms of x:');
F=laplace(f,x,w)
```

Examples

1. Find the Laplace transform for the following functions

(i) $e^x \sin x$ (ii) $\sin x \cos x$ (iii) $x + x^2 + x^3$ (iv) $\sin^3 2x$

Input

```
Enter the function in terms of x:
exp(x)*sin(x)
```

Output

```
F =
1/((w - 1)^2 + 1)
```

Input

```
Enter the function in terms of x:
sin(x)*cos(x)
```

Output

```
F =
1/((w^2 + 4)
```

Input

```
Enter the function in terms of x:
x+x^2+x^3
```

Output

```
F =
1/w^2 +2/w^3 +6/w^4
```

Input

```
Enter the function in terms of x:
sin(2*x)^3
```

Output

```
F =
48/((w^2 + 4)*(w^2 + 36))
```

Unit Step function

The unit step function $u(\mathrm{t})$ is sometimes called a Heaviside function and it is defined as follows:

$$u(\mathrm{t})=\begin{cases}0, & t\le 0\\ 1, & t>0\end{cases} \qquad (or)\ u(\mathrm{t})=\begin{cases}0, & t<0\\ 0.5, & t=0\\ 1, & t>0\end{cases}$$

Examples

Find the Laplace transform of the function $f(x)=\begin{cases}x^2, & 0\le x<1\\ x, & 1\le x<2\\ 0, & x\ge 2\end{cases}$ in terms of w

Input

```
Enter the function in terms of x:
x^2*(heaviside(x)-heaviside(x-1))+x*(heaviside(x-1)-
heaviside(x-2))
```

Output

```
F =

2/w^3 - exp(-w)/w^2 - exp(-2*w)/w^2 - (2*exp(-w))/w^3 -
(2*exp(-2*w))/w
```

2. Find the Laplace transform of the function $f(x)=\begin{cases}x^2, & 0<x<2\\ x-1, & 2<x<3\\ 7, & x>3\end{cases}$ in terms of w

Input

```
Enter the function in terms of x:
x^2*(heaviside(x)-heaviside(x-2))+(x-1)*(heaviside(x-2)-
heaviside(x-3))+7*heaviside(x-3)
```

Output

```
F =

(7*exp(-3*w))/w - (4*exp(-2*w))/w - (4*exp(-2*w))/w^2 -
(2*exp(-2*w))/w^3 + 2/w^3 - (exp(-3*w)*(2*w - exp(w) -
w*exp(w) + 1))/w^2
```

`pretty(simplify(F))` command give us the below simplified solution

```
-exp(-3 w)(w - exp(3 w) 2 + 2 exp(w) + 3 w  exp(w) + 3 w exp(w) - 5 w )
 -----------------------------------------------------------------
                                          3
                                         w
```

Periodic function

Let $f(t)$ be a periodic function with period T, then $L[f(t)]=\dfrac{\int_0^T e^{-st} f(t)\,dt}{1-e^{-sT}}$

Example

Find the Laplace Transform of $f(t)=\left(\frac{2x}{3}\right),\ 0\le x\le 3$ in terms of w.

Input

```
Enter the function in terms of x:
(2*x)/3*(heaviside(x)-heaviside(x-3))
```

Output

```
F =

2/(3*w^2) - (2*exp(-3*w))/(3*w^2) - (2*exp(-3*w))/w
```

Unit impulse function

The unit impulse function is denoted $\delta(\mathrm{t})$ and defined as

$$\delta(\mathrm{t})=\begin{cases}1, & t=0\\ 0, & t\neq 0\end{cases},$$

$$then\ L\{\delta(\mathrm{t})\}=1$$

If there is a delayed impulse function $\delta(\mathrm{t-T})$, then $L\{\delta(\mathrm{t-T})\}=\mathrm{e}^{-Ts}$

Example

Find the Laplace transform of the Dirac delta/impulse function $\delta(x-3)$ in terms of w

Input

```
Enter the function in terms of x:
dirac(x-3)
```

Output

```
F =

exp(-3*w)
```

Evaluation of integrals using Laplace Transforms

We can evaluate number of integrals having lower limit 0 and upper limit ∞ by the help of Laplace transform.

MATLAB code:

```
clc
syms t
f=input('Enter the function in terms of t:');
s=input('Enter the value of s:');
F=laplace(f);
F=subs(F,s)
```

Examples

Evaluate the following integrals using Laplace Transform

(i) $\int_0^\infty t\,e^{-3t}\sin t\,dt$ (ii) $\int_0^\infty \frac{e^{-t}\sin t}{t}\,dt$ (iii) $\int_0^\infty \frac{e^{-2t}\sinh t\,\sin t}{t}\,dt$

Input

```
Enter the function in terms of t:
(t*sin(t))
Enter the value of s:
3
```

Output

```
F =

3/50
```

Input

```
Enter the function in terms of t:
sin(t)/t
Enter the value of s:
1
```

Output

```
F =

pi/4
```

Input

```
Enter the function in terms of t:
(sinh(t)*sin(t))/t
Enter the value of s:
2
```

Output

```
F =

pi/8-atan(1/3)/2
```

Exercise problems

1. Find the Laplace Transform for the following functions

   (i) $t\sin^2 3t$ (ii) $t^3 e^{-5t}$ (iii) $t^2 e^t \sin 4t$ (iv) $\sin 2t \sin 3t$

2. Find the Laplace Transform for the following functions

   (i) $t^{-\frac{1}{2}}$ (ii) $\frac{1-\cos t}{t}$ (iii) $\frac{1}{t}\sin t$

3. Find the Laplace transform of the function $f(x)=\begin{cases} x^3, & 0<x<2 \\ x, & 2<x<3 \\ 7x-2, & x>3 \end{cases}$ in terms of w

4. Find the Laplace transform of the function $f(\mathrm{t})=\begin{cases} t-1, & 1<t<2 \\ 3-t, & 2<t<3 \end{cases}$ in terms of s

5. Find the Laplace transform of the function $f(\mathrm{t})=e^{-4t}\frac{\sin 3t}{t}$ in terms of s

6. Evaluate the following integrals using Laplace Transform

   (i) $\int_0^\infty \frac{e^{-t}-e^{-4t}}{t}\,dt$ (ii) $\int_0^\infty t\,e^{-4t}\sin t\,dt$ (ii) $\int_0^\infty \frac{e^{-2t}\sinh t\,\sin t}{t}\,dt$

Experiment: 4
Inverse Laplace Transforms through MATLAB

AIM:

The main objective of this experiment is to

- Find Inverse Laplace transforms of the standard functions.
- Find the partial fractions and hence finding inverse Laplace transforms
- Solve ordinary differential equations using Laplace Transforms

Mathematical Background

Inverse Laplace Transform

If $L[f(t)] = \mathrm{F}(s)$, then $L^{-1}[F(s)] = f(t)$ where L^{-1} is called the Inverse Laplace transform.

MATLAB Commands required:

Following are new MATLAB commands one need to know for the present experiment.

MATLAB Syntax	Description
`ilaplace(F)`	To find the inverse Laplace transform of `F` with default independent variable `s`. The default return is a function of `t`.
`ilaplace(F,x)`	Returns the inverse Laplace transform of `F` in symbol `x` instead of default `t`.
`ilaplace(F,x,w)`	Assumes `F` as a function of `x` and returns the inverse Laplace transform in terms of `w`.

MATLAB code for finding Inverse Laplace transform of standard functions

We have learnt how MATLAB uses default variable *t* and *s* in the case of Laplace transform. Similar is the case of inverse Laplace transform. In what follows we have used *t* and *s* as the variables. The variables may be changed appropriately, if needed.

MATLAB Code

```
clc
syms t s a b
F=input('Enter the function in terms of s:');
f=ilaplace(F)
```

Examples

1. Find the inverse Laplace Transform for the following functions

(i) $\frac{1}{s-a}$ (ii) $\frac{1}{s^2-a^2}$ (iii) $\frac{1}{s^2+a^2}$ (iv) $\frac{s}{s^2+a^2}$ (v) $\frac{1}{(s-a)^2+b^2}$ (vi) $\frac{1}{(s-a)^2+b^2}$

Input

```
Enter the function in terms of s:
1/(s-a)
```

Output

```
f =

exp(a*t)
```

Input

```
Enter the function in terms of s:
1/(s^2-a^2)
```

Output

```
f =

exp(a*t)/(2*a) - exp(-a*t)/(2*a)
```

Input

```
Enter the function in terms of s:
1/(s^2+a^2)
```

Output

```
f =

sin(a*t)/a
```

Input

```
Enter the function in terms of s:
1/(s^2+a^2)
```

Output

```
f =

cos(a*t)
```

Input

```
Enter the function in terms of s:
1/((s-a)^2+b^2)
```

Output

```
f =

(exp(a*t)*sin(b*t))/b
```

Input

```
Enter the function in terms of s:
1/((s-a)^2-b^2)
```

Output

```
f =

exp(t*(a + b))/(2*b) - exp(t*(a - b))/(2*b)
```

2. Find the inverse Laplace Transform for the function $\frac{1}{(s-1)^3}$ in terms of t

Input

```
Enter the function in terms of s:
1/(s-1)^3
```

Output

```
f =

(t^2*exp(t))/2
```

3. Find the inverse Laplace Transform for the function $\frac{3e^{-s}}{s^2+4}$ in terms of t

Input

```
Enter the function in terms of s:
3*exp(-s)/(s^2+4)
```

Output

```
f =

(3*heaviside(t - 1)*sin(2*t - 2))/2
```

4. Find the inverse Laplace Transform for the function $\frac{3s-8}{4s^2+25}$ in terms of t

Input

```
Enter the function in terms of s:
(3*s-8)/(4*s^2+25)
```

Output

```
f =

(3*cos((5*t)/2))/4 - (4*sin((5*t)/2))/5
```

5. Find the inverse Laplace Transform for the function $\frac{s^2+3}{s^3+9s}$ in terms of t

Input

```
Enter the function in terms of s:
(s^2+3)/(s^3+9*s)
```

Output

```
f =

(2*cos(3*t))/3 + 1/3
```

Finding the partial fractions and hence finding inverse Laplace transforms through MATLAB

MATLAB Commands required:

MATLAB Syntax	Description
`partfrac(expr,var)`	Finds the partial fraction decomposition of the expression `'expr'` with respect to variable `'var'`

MATLAB code for finding the partial fraction expansions

```
clc
syms  s
F=input('Enter the expression in s:');
Z=partfrac(F,s)
```

Examples

1. Find the partial fraction expansion for the function $F(s)=\frac{3s-1}{s^2+3s+2}$

Input

```
Enter the expression in s:
(3*s-1)/(s^2+3*s+2)
```

Output

```
Z =

7/(s + 2) - 4/(s + 1)
```

2. Find the partial fraction expansion for the function $F(s)=\frac{s+4}{(s^2-s)(s^2+4)}$

Input

```
Enter the expression in s:
(s+4)/((s^2-s)*(s^2+4))
```

Output

```
Z =

1/(s - 1) - 1/(s^2 + 4) - 1/s
```

3. Find the partial fraction expansion for the function $F(s)=\frac{2s^2+3s-1}{(s^2+3s+2)}$

Input

```
Enter the expression in s:
(2*s^2+3*s-1)/(s^2+3*s+2)
```

Output

```
Z =

2 - 1/(s + 2) - 2/(s + 1)
```

MATLAB code for finding inverse Laplace transforms through MATLAB

```
clc
syms t s
F=input('Enter the function in terms of s:');
Z=partfrac(F,s);
f=ilaplace(Z)
```

Examples

1. Find the inverse Laplace transform of the function $\mathrm{F}(s)=\dfrac{3s-1}{s^2+3s+2}$

Input

```
Enter the expression in s:
(3*s-1)/(s^2+3*s+2)
```

Output

```
f =

7*exp(-2*t) - 4*exp(-t)
```

2. Find the inverse Laplace transform of the function $\mathrm{F}(s)=\dfrac{2s^2+3s-1}{(s^2+3s+2)}$

Input

```
Enter the expression in s:
(2*s^2+3*s-1)/(s^2+3*s+2)
```

Output

```
f =

2*dirac(t) - exp(-2*t) - 2*exp(-t)
```

3. Find the inverse Laplace transform of the function $F(s)=\dfrac{7s^3+2s^2+3s-1}{(s^2+3s+2)}$

Input

```
Enter the expression in s:
(7*s^3+2*s^2+3*s-1)/(s^2+3*s+2)
```

Output

```
f =

55*exp(-2*t) - 9*exp(-t) - 19*dirac(t) + 7*dirac(1, t)
```

Solution of ordinary differential equations using Laplace Transforms

MATLAB has some powerful features for solving differential equations of all types. We will explore some of these features using Laplace transforms. The approach here will be that of the Symbolic Math Toolbox. The result will be the form of the function and it may be readily plotted with MATLAB.

MATLAB Code

```
syms s t Y
f = input('Enter RHS of ODE in t');
y0 = input('Enter the intial condition y at t:');
y1 = input('Enter the intial condition y1 at t:');
F = laplace(f,t,s);
Y1 = s * Y - y0; % Using the theorems for the transforms of
derivatives
Y2 = s * Y1 -y1;
LHS=input('Enter LHS of ODE in terms of Y');
Sol = solve(LHS- F, Y);
sol = ilaplace(Sol,s,t)
ezplot(sol,[0,10]);
grid on,
title('GRAPHIC DISPLAY OF SOLUTION OF ODE BY LAPLACE
TRANSFORM')
xlabel('time'),
ylabel('f(t)')
legend('laplace transform')
```

Examples

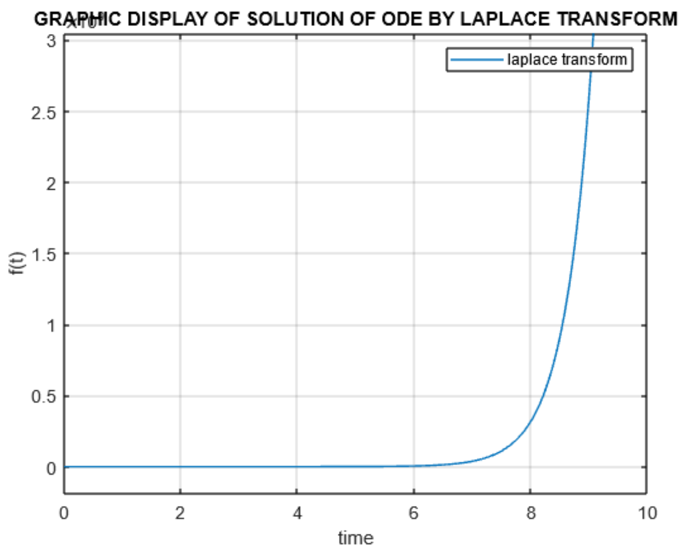
1. Solve the equation $Y'' - 3Y' + 2Y = 4e^{2t}$, $where\ Y(0) = -3,\ Y'(0) = 5,\ 0 \le t \le 10.$

Input

```
Enter RHS of ODE in t
4 * exp(2 * t)
Enter the intial condition y
at t:
-3
Enter the intial condition y1
at t:
5
Enter LHS of ODE in terms of
Y
Y2-3*Y1+2*Y
```

Output

```
sol =

4*exp(2*t) - 7*exp(t) +
4*t*exp(2*t)
```



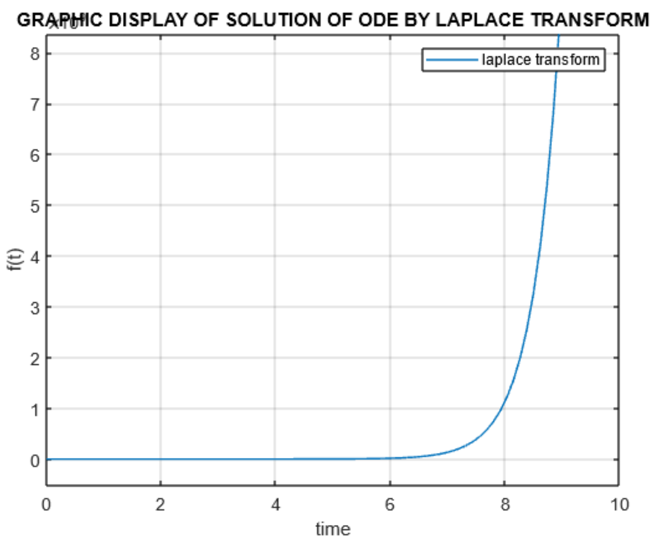
2. Solve the equation $y'' - 4y + 4y = 64\sin 2t$, $where\ y(0) = 0,\ y'(0) = 1, 0 \le t \le 10.$

Input

```
Enter RHS of ODE in t
64*sin(2*t)
Enter the intial condition y
at t:
0
Enter the intial condition y1
at t:
1
Enter LHS of ODE in terms of
Y
Y2-4*Y1+4*Y
```

Output

```
sol =

8*cos(2*t) - 8*exp(2*t) +
17*t*exp(2*t)
```



Exercise problems

1. Find the partial fraction expansion for the function $\mathrm{F}(s)=\frac{7s^3+2s^2+3s-1}{(s^2+3s+2)}$
2. Find the partial fraction expansion for the function $\mathrm{F}(s)=\frac{2s^2+3s-1}{(s^3+5s^2+8s+4)}$
3. Find the inverse Laplace transform of the function $\mathrm{F}(s)=\frac{s}{s^2+6s+25}$
4. Find the inverse Laplace transform of the function $\mathrm{F}(s)=\frac{3s^2-1}{s^2+3s+2}$
5. Find the inverse Laplace transform of the function $\mathrm{F}(s)=\frac{s+2}{(s^2-4s+13)}$
6. Solve the equation $y''+25y=10\cos 5t,\ where\ y(0)=2,\ y'(0)=0.$
7. Solve the equation $y''+2y'+2y=5\sin t,\ where\ y(0)=0,\ y'(0)=0.$

Experiment: 5
Solving Partial Differential Equations

AIM:

The main objective of this experiment is to

- Solve the Partial differential equations.
- Understand the solution using plots
- Understand 3-D plots and its interpretation

MATLAB Commands required:

Following are new MATLAB commands one need to know for the present experiment.

MATLAB Syntax	Description
`SOL =pdepe(M,PDEFUN,ICFUN,BCFUN,XMESH,TSPAN)`	Solves initial-boundary value problems for small systems of parabolic and elliptic PDEs in one space variable x and time t to modest accuracy.

One Dimensional Heat Equation or Diffusion equation

The general form of one-dimensional Heat equation or Diffusion equation is given by

$\frac{\partial^2 u}{\partial x^2}=\frac{1}{c^2}\frac{\partial u}{\partial t}$ (or) $\frac{\partial u}{\partial t}=c^2\frac{\partial^2 u}{\partial x^2}$ where $c^2=\frac{k}{\rho s}$ and c^2 is the diffusivity of the substance, k is thermal conductivity, ρ be the density, and s be the specific heat.

Examples

1. Solve the Heat equation $\pi^2\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)$ defined for $0\le x\le 1$ and for times $t\ge 0$ with the initial condition $u(x,0)=\sin \pi x$ and boundary conditions $u(0,t)=0,\ \pi e^{-t}+\frac{\partial u}{\partial x}(1,t)=0$

Note:

Before we code the equation, you need to rewrite it in a form that the `pdepe` solver expects. The standard form that `pdepe` expects is

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right)$$

The standard form for the boundary conditions expected by the solver is

$$p(x,t,u)+q(x,t,u)\,f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

MATLAB Code for equation:

The PDE is already in the form $\pi^2\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)$

So we can read off the relevant terms

- $m=0$
- $c\left(x,t,u,\frac{\partial u}{\partial x}\right)=\pi^2$
- $f\left(x,t,u,\frac{\partial u}{\partial x}\right)=\frac{\partial u}{\partial x}$
- $s\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$

Now we can create a function to code the equation. The function should have the signature

`[c,f,s] = pdex1pde (x,t,u,dudx):`

The outputs `c`, `f`, and `s` correspond to coefficients in the standard PDE equation form expected by `pdepe`. These coefficients are coded in terms of the input variables `x`, `t`, `u`, and `dudx`.

```
function [c,f,s] = pdex1pde(x,t,u,DuDx)
c = pi^2;
f = DuDx;
s = 0;
```

MATLAB Code initial condition:

Next, we write a function that returns the initial condition. The initial condition is applied at the first time value `tspan(1)`. The function should have the signature `u0 = pdex1ic(x).`

The corresponding function for $u(x,0)=\sin\pi x$ is

```
function u0 = pdex1ic(x)
u0 = sin(pi*x);
```

MATLAB Code boundary conditions:

Now, we write a function that evaluates the boundary conditions.

$$u(0,t)=0,\ \pi e^{-t}+\frac{\partial u}{\partial x}(1,t)=0$$

The standard form for the boundary conditions expected by the solver is

$$p(x,t,u)+q(x,t,u)\, f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

Written in this form, the boundary conditions for this problem become

$$u(0,t)-0+0.\, f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0,$$

$$\pi e^{-t}+1.\, f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

- The outputs `pl` and `ql` correspond to $p_L(x,t,u)$ and $q_L(x,t)$ for the left boundary (x = 0 for this problem).
- The outputs `pr` and `qr` correspond to $p_R(x,t,u)$ and $q_R(x,t)$ for the right boundary (x = 1 for this problem).

The boundary conditions in this example are represented by the function:

```
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul;
ql = 0;
pr = pi * exp(-t);
qr = 1;
```

MATLAB Code to solve the equation:

Finally, solve the equation using the symmetry `m`, the PDE equation, the initial conditions, the boundary conditions, and the meshes for `x` and `t`.

```
m = 0;
x = linspace(0,1,20);
t = linspace(0,2,5);
sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
u = sol(:,:,1);
```

Complete MATLAB code for this problem

```
m = 0;
x = linspace(0,1,20);
t = linspace(0,2,5);

sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
% Extract the first solution component as u.
u = sol(:,:,1);

% A surface plot is often a good way to study a solution.
surf(x,t,u)
title('Numerical solution computed with 20 mesh points.')
```

```
xlabel('Distance x')
ylabel('Time t')

% A solution profile can also be illuminating.
figure
plot(x,u(end,:))
title('Solution at t = 2')
xlabel('Distance x')
ylabel('u(x,2)')
```

Note: This function file should be saved with the file name `pdex1pde.m`

```
function [c,f,s] = pdex1pde(x,t,u,DuDx)
c = pi^2;
f = DuDx;
s = 0;
```

Note: This function file should be saved with the file name `pdex1ic.m`

```
function u0 = pdex1ic(x)
u0 = sin(pi*x);
```

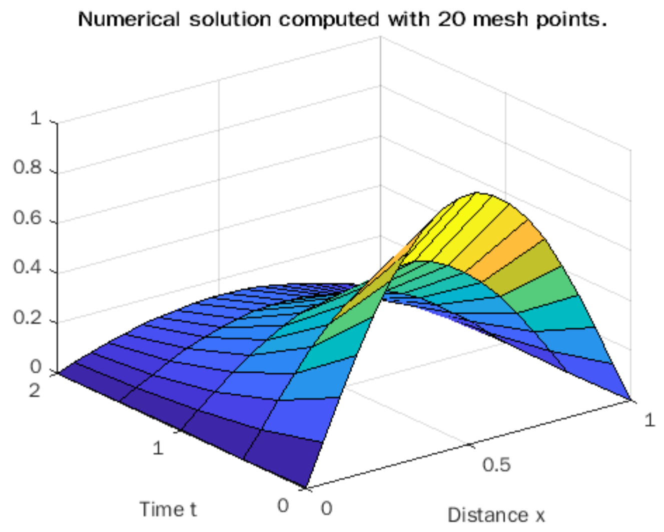
Note: This function file should be saved with the file name `pdex1bc.m`

```
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul;
ql = 0;
pr = pi * exp(-t);
qr = 1;
```

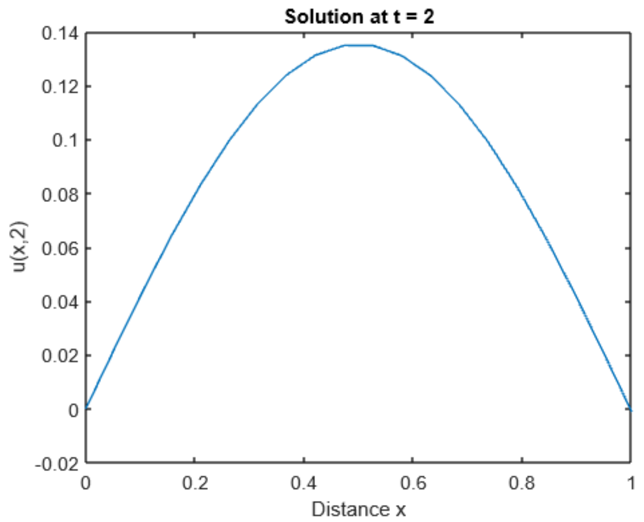
Note: All the function files should be saved locally along with main script file.

OUTPUT

The surface plot shows the behaviour of the solution.



The following plot shows the solution profile at the final value of `t` (i.e., `t = 2`).



2. Solve the partial differential equation $\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)$ defined for $0 \le x \le 1$ *and* $t \ge 0$. The initial condition is $u(0,x) = \frac{2x}{1+x^2}$ and the boundary conditions are $u(t,0)=0,\ u(t,1)=1$

MATLAB Code for equation:

The PDE is already in the form $\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)$

So we can read off the relevant terms

- $m = 0$
- $c\left(x,t,u,\frac{\partial u}{\partial x}\right) = 1$
- $f\left(x,t,u,\frac{\partial u}{\partial x}\right) = \frac{\partial u}{\partial x}$
- $s\left(x,t,u,\frac{\partial u}{\partial x}\right) = 0$

Now we can create a function to code the equation. The function should have the signature

```
[c,f,s] = pdex1pde (x,t,u,dudx):
```

The outputs `c`, `f`, and `s` correspond to coefficients in the standard PDE equation form expected by `pdepe`. These coefficients are coded in terms of the input variables `x`, `t`, `u`, and `dudx`.

```
function [c,f,s] = pdex1pde(x,t,u,dudx)
c = 1;
f = dudx;
s = 0;
```

MATLAB Code initial condition:

Next, we write a function that returns the initial condition. The initial condition is applied at the first time value `tspan(1)`. The function should have the signature `u0 = pdex1ic(x)`.

The corresponding function for $u(0,x)=\dfrac{2x}{1+x^2}$ is

```
function u0 = pdex1ic(x)
u0 = 2*x/(1+x^2);
```

MATLAB Code boundary conditions:

Now, we write a function that evaluates the boundary conditions.

$u(t,0)=0,\ \ u(t,1)=1$

The standard form for the boundary conditions expected by the solver is

$$p(x,t,u)+q(x,t,u)\,f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

Written in this form, the boundary conditions for this problem become

$$u(t,0)-0+0.\,f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0,$$

$$u(t,1)-1+0.\,f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

- The outputs `pl` and `ql` correspond to $p_L(x,t,u)$ and $q_L(x,t)$ for the left boundary ($x = 0$ for this problem).
- The outputs `pr` and `qr` correspond to $p_R(x,t,u)$ and $q_R(x,t)$ for the right boundary ($x = 1$ for this problem).

The boundary conditions in this example are represented by the function:

```
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul-0;
ql = 0;
```

```
pr = ur - 1;
qr = 0;
```

Complete MATLAB code for this problem

```
m = 0;
x = linspace(0,1,20);
t = linspace(0,2,5);

sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
% Extract the first solution component as u.
u = sol(:,:,1);

% A surface plot is often a good way to study a solution.
surf(x,t,u)
title('Numerical solution computed with 20 mesh points.')
xlabel('Distance x')
ylabel('Time t')

% A solution profile can also be illuminating.
figure
plot(x,u(end,:))
title('Solution at t = 2')
xlabel('Distance x')
ylabel('u(x,2)')
```

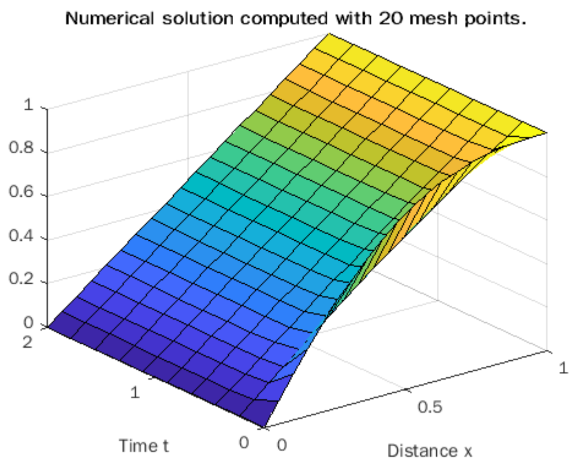
```
function [c,f,s] = pdex1pde(x,t,u,dudx)
c = 1;
f = dudx;
s = 0;
```

```
function u0 = pdex1ic(x)
u0 = 2*x/(1+x^2);
```

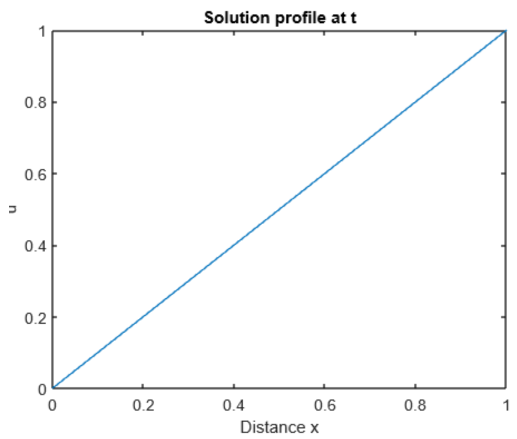
```
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul-0;
ql = 0;
pr = ur - 1;
qr = 0;
```

OUTPUT

The surface plot shows the behaviour of the solution.



The following plot shows the solution profile at the final value of `t` (i.e., `t = 1`).



Exercise Problems

1. Solve the partial differential equation $\pi \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ defined for $0 \le x \le 4$ *and* $t \ge 0$. The initial condition is $u(0, x) = \frac{2}{1+x^2}$ and the boundary conditions are $u(t,0) = 0,\ u(t,4) = 2$
2. Solve the partial differential equation $\pi^2 \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ defined for $0 \le x \le 4$ *and* $t \ge 0$. The initial condition is $u(0, x) = 2x$ and the boundary conditions are $u(t,0) = 0,\ u(t,4) = 2$
3. Solve the partial differential equation $\pi^2 \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ defined for $0 \le x \le 10$ *and* $t \ge 0$. The initial condition is $u(0, x) = \frac{2x^3}{(1+x)}$ and the boundary conditions are $u(t,0) = 0,\ u(t,10) = 4$

Experiment: 6
Application Problems through MATLAB

AIM:

The main objective of this experiment is to

- Understanding the application problems such as, mass spring oscillations, ELECTRIC circuits (LCR) and bending problems.

Applications

Mathematical Model of a Mass Spring System

The motion of an object with mass *m* at the end of a spring that is either vertical (Figure 1) or horizontal (Figure 2) on a level surface:

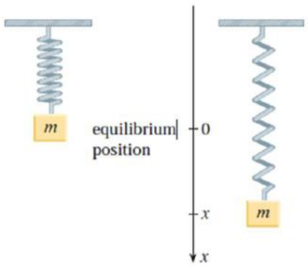


Figure 1

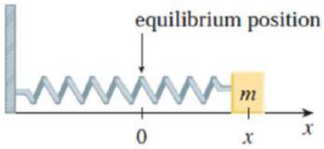


Figure 2

If the string is stretched *x* units from its natural length, then it exerts force (restoring force) that is proportional to *x;*

Restoring force $=-kx$, where k is a positive constant (called the spring constant).

If we ignore any external resisting forces (due to air resistance of friction), then by Newton's second law, we have $m\dfrac{d^2x}{dt^2}=-kx \ \ (or) \ \ m\dfrac{d^2x}{dt^2}+kx=0$ (1)

Its auxiliary equation is $mr^2+k=0$ with roots $r=\pm wi$, where $w=\sqrt{k/m}$

Thus, the solution is

$$x(t)=c_1\cos wt+c_2\sin wt \ \ (or) \ \ x(t)=A\cos(wt+\delta)$$

Where $w=\sqrt{k/m}$ (frequency) & $A=\sqrt{c_1^{\,2}+c_2^{\,2}}$ (Amplitude)

$$\cos\delta=\frac{c_1}{A};\sin\delta=-\frac{c_2}{A} \ \ (\delta \text{ is the phase angle})$$

This type of motion is called simple harmonic motion.

Damped Vibrations

The motion of a spring that is subject to a frictional force (in the case of horizontal spring of figure 2) or a damping force (in the case of vertical spring moves through a fluid as in figure 3). An example is the damping force supplied by a shock absorber in a car or a bicycle (figure 4).

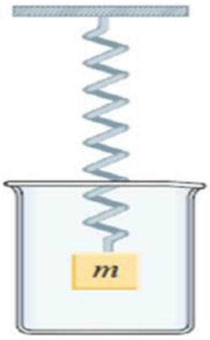


Figure 3

Figure 4

The damping force is proportional to the velocity of the mass and acts in the direction opposite to the motion.

Damping force $= -c\dfrac{dx}{dt}$, where c is a positive constant, called the damping constant.

In this case, Newton's second law gives

$$m\frac{d^2x}{dt^2} = restoring\ \ force + damping\ \ force = -kx - c\frac{dx}{dt}$$

(or)

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0$$

Its auxiliary equation is $mr^2 + cr + k = 0$ with roots $r_1 = \dfrac{-c+\sqrt{c^2-4mk}}{2m}; r_2 = \dfrac{-c-\sqrt{c^2-4mk}}{2m}$

There are three cases shown below. The solution curves of the differential equations are going to be visualized by solving the problems.

Case 1: $c^2 - 4mk > 0$ (Over damping)

Case 2: $c^2 - 4mk = 0$ (Critical damping)

Case 3: $c^2 - 4mk < 0$ (Under damping)

MATLAB Code:

```
clear all
clc
syms x(t)
a=input('Enter the coefficient of D2x:');
b=input('Enter the coefficient of Dx:');
c=input('Enter the coefficient of x:');
A=[a b c];
r=roots(A);
if imag(r)~=0
```

```
    disp('under damping:')
elseif r(1)==r(2)
    disp('critical damping:')
else
    disp('Over damping:')
end
ode = input('Enter the differential equation:');
Dx = diff(x);
cond1 = x(0) == 0;
cond2 = Dx(0) == 0.6;
conds = [cond1 cond2];
xSol(t) = dsolve(ode,conds);
xSol = simplify(xSol)
tt=linspace(0,2,100);
xval=subs(xSol,tt);
plot(tt,xval)
xlabel('t');
ylabel('x')
```

Examples:

1. Suppose that the spring with a mass of 2 kg has natural length 0.5 m. A force of 25.6 N is required to maintain it stretched to a length of 0.7 m. The spring is immersed in a fluid with damping constant c=40. Find the position of the mass at any time t if it starts from the equilibrium position and is given a push to start with an initial velocity of 0.6 m/s.

Solution:

Given that

Mass $(m)=$ 2 kg, String stretched from its natural length (x) units $=0.7-0.5=0.2$

& spring constant $k(0.2)=25.6\Rightarrow k=128$

Therefore, the differential equation is $2\frac{d^2x}{dt^2}+40\frac{dx}{dt}+128x=0$ with $x(0)=0;\ x'(0)=0.6$

So, the differential equation becomes $\frac{d^2x}{dt^2}+20\frac{dx}{dt}+64x=0$

The auxiliary equation is $r^2+20r+64=0\Rightarrow r=-4,-16$

Since $mr^2+cr+k=0,\ c^2-4mk>0$ (Over damping)

So the motion is over damped. $x(t)=c_1e^{-4t}+c_2e^{-16t}$ (1)

We are given that $x(0)=0;\ x'(0)=0.6$

From (1), we have

$0=c_1+c_2\Rightarrow c_2=-c_1$

Differentiating (1), we get $x'(t)=-4c_1e^{-4t}-16c_2e^{-16t}$ (2)

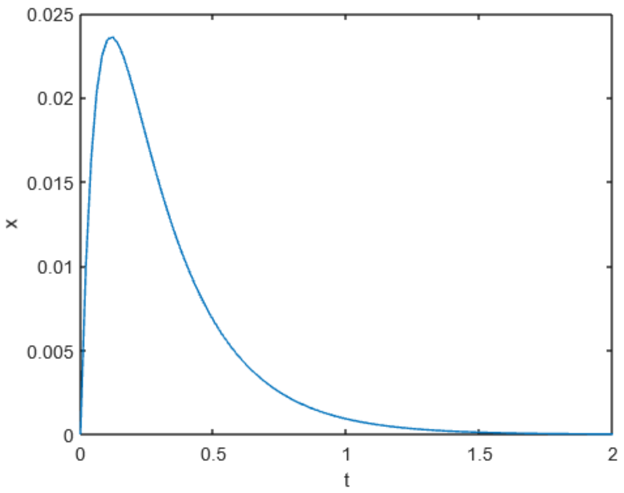
From (2), we have

$x'(0)=-4c_1-16c_2=0.6\Rightarrow c_1=0.05$

The particular solution is $x(t)=0.05e^{-4t}-0.05e^{-16t}=0.05(e^{-4t}-e^{-16t})$

Input

```
Enter the coefficient of D2x:
2
Enter the coefficient of Dx:
40
Enter the coefficient of x:
128
Over damping:

Enter the differential
equation:
2*diff(x,t,2)+40*diff(x,t)+128*
x == 0
```



Output

```
xSol(t) =

(exp(-16*t)*(exp(12*t) - 1))/20
```

Inference:

Solution x for $2\frac{d^2x}{dt^2}+40\frac{dx}{dt}+128x=0$ with $x(0)=0;\ x'(0)=0.6$ in the interval [0 2]. Here $m=2,\ c=40,\ k=128;\ c^2-4mk=576>0$. It is a case of over damping, therefore no oscillations occur.

2. Suppose that the spring with a mass of 2 kg has natural length 0.5m. A force of 25.6N is required to maintain it stretched to a length of 0.7 m. The spring is immersed in a fluid with damping constant c=14. Find the position of the mass at any time t if it starts from the equilibrium position and is given a push to start with an initial velocity of 0.6 m/s.

Solution:

Given that

Mass $(m)=$ 2 kg, String stretched from its natural length (x) units $=0.7-0.5=0.2$

& spring constant $k(0.2)=25.6 \Rightarrow k=128$

Therefore, the differential equation is $2\frac{d^2x}{dt^2}+14\frac{dx}{dt}+128x=0$ with $x(0)=0;\ x'(0)=0.6$

So, the differential equation becomes $\frac{d^2x}{dt^2}+7\frac{dx}{dt}+64x=0$

The auxiliary equation is $r^2+7r+64=0 \Rightarrow r=\frac{-7+3i\sqrt{23}}{2}, \frac{-7-3i\sqrt{23}}{2}$

Since $mr^2+cr+k=0,\ c^2-4mk<0$ (Under damping)

So the motion is under damped.

$$x(t)=c_1e^{-\frac{7}{2}t}\cos\left(\frac{3\sqrt{23}}{2}t\right)+c_2e^{-\frac{7}{2}t}\sin\left(\frac{3\sqrt{23}}{2}t\right) \qquad (1)$$

We are given that $x(0)=0$, so we have $c_1=0$

Differentiating (1), we get

$$x'(t)=c_1e^{-\frac{7}{2}t}\left(-\sin\left(\frac{3\sqrt{23}}{2}t\right)\right)*\frac{3\sqrt{23}}{2}+c_1\left(\frac{-7}{2}\right)e^{-\frac{7}{2}t}*\left(\cos\left(\frac{3\sqrt{23}}{2}t\right)\right)$$

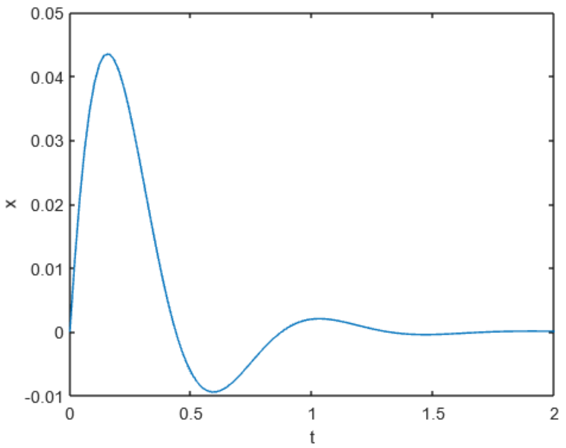
$$+c_2e^{-\frac{7}{2}t}*\left(\cos\left(\frac{3\sqrt{23}}{2}t\right)\right)*\frac{3\sqrt{23}}{2}+c_2\left(\sin\left(\frac{3\sqrt{23}}{2}t\right)\right)*\left(\frac{-7}{2}\right)e^{-\frac{7}{2}t}$$

$$x'(0)=c_1*\left(\frac{-7}{2}\right)+c_2*\frac{3\sqrt{23}}{2} \Rightarrow c_2=\frac{0.4}{\sqrt{23}}$$

The particular solution is $x(t)=\frac{0.4}{\sqrt{23}}e^{-\frac{7}{2}t}\left(-\sin\left(\frac{3\sqrt{23}}{2}t\right)\right)$

Input

```
Enter the coefficient of D2x:
2
Enter the coefficient of Dx:   14
Enter the coefficient of x:
128
Under damping:

Enter the differential equation:
2*diff(x,t,2)+140*diff(x,t)+128*x
== 0
```



Output

```
xSol(t) =

(2*23^(1/2)*exp(-
(7*t)/2)*sin((3*23^(1/2)*t)/2))/11
5
```

Inference:

Solution x for $2\frac{d^2x}{dt^2}+14\frac{dx}{dt}+128x=0$ with $x(0)=0;\ x'(0)=0.6$ in the interval [0 2]. Here $m=2,\ c=14,\ k=128;\ c^2-4mk<0$. It is a case of under damping, therefore some oscillations occur.

Forced vibrations

In addition to the restoring force and the damping force, the motion of the spring affected by the external forces $F(t)$.Then Newton's second law gives

$$m\frac{d^2x}{dt^2}=restoring\ \ force+damping\ \ force+external\ \ force=-kx-c\frac{dx}{dt}+F(t)$$

The motion of the spring is now governed by the following non-homogeneous differential equation

$$m\frac{d^2x}{dt^2}+c\frac{dx}{dt}+kx=F(t)$$

Note

The MATLAB code for solving non-homogeneous differential equation is as same as the code given above for homogeneous differential equation.

Mathematical Model of a LCR/ RLC Circuit:

LCR circuit contains an electromotive force E (supplied by a battery or generator), a resistor R , an inductor L, and a capacitor C , in series. If the charge on the capacitor at time t is $Q=Q(t)$, then the current is the rate of change of Q with respect to t : $I=\frac{dQ}{dt}$. It is known from physics that the voltage drops across the resistor, inductor, and capacitor are $RI,\ L\frac{dI}{dt},\ \frac{Q}{C}$ respectively. Kirchhoff's voltage law says that the sum of these voltage drops is equal to the supplied voltage:

$$L\frac{dI}{dt}+RI+\frac{Q}{C}=E(t) \qquad (1)$$

Since $I=\frac{dQ}{dt}$, equation (1) becomes

$$L\frac{d^2Q}{dt^2}+R\frac{dQ}{dt}+\frac{1}{C}Q=E(t) \qquad (2)$$

which is a second-order linear differential equation with constant coefficients. If the charge Q_0 and the current I_0 are known at time 0, then we have the initial conditions

$Q(0)=Q_0$ & $Q'(0)=I(0)=I_0$

and this initial-value problem can be solved by various methods.

Similarly, a differential equation for the current can be obtained by differentiating Equation (2) with respect to and remembering that $I=\frac{dQ}{dt}$:

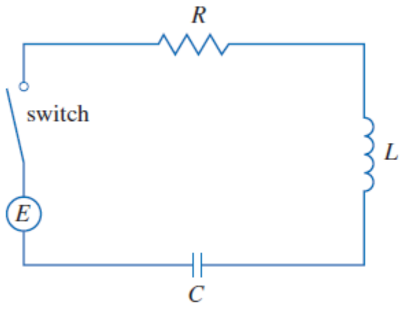


Fig. 5

$$L\frac{d^2I}{dt^2}+R\frac{dI}{dt}+\frac{1}{C}I=E'(t) \qquad (3)$$

Examples

1. Find the charge and current at time t in the circuit of Figure 5, if $R=40\Omega, L=1H, C=16\times10^{-4}\,F, E(t)=100\cos10t,$ and the initial charge and current are both 0.

Solution

With the given values of $R,\ L,\ C,\&\ E(t)$, equation (2) becomes

$$\frac{d^2Q}{dt^2}+40\frac{dQ}{dt}+625Q=100\cos10t \qquad (1)$$

The auxiliary equation is $r^2+40r+625=0\ \ with\ \ roots\ r=-20\pm15i$

So, the complementary solution is

$$Q_c(t)=e^{-20t}(c_1\cos15t+c_2\sin15t) \qquad (2)$$

Similarly, the particular integral is

$$Q_p(t)=\frac{1}{697}(84\cos10t+64\sin10t) \qquad (3)$$

The general solution is

$$Q(t)=Q_c(t)+Q_p(t)=e^{-20t}(c_1\cos15t+c_2\sin15t)+\frac{1}{697}(84\cos10t+64\sin10t)$$

Imposing the initial condition $Q(0)=0$, we get

$$Q(0)=c_1+\frac{84}{697}=0\Rightarrow c_1=-\frac{84}{697}$$

To impose other initial condition, we first differentiate $Q(\mathrm{t})$ to find the current

$$I=\frac{dQ}{dt}=e^{-20t}\left[(-20\,\mathrm{c}_1+15\,\mathrm{c}_2)\cos15\,\mathrm{t}+(-15\,\mathrm{c}_1-20c_2)\sin15t\right]$$

$$+\frac{40}{697}\left(-21\sin10t+16\cos10t\right)$$

$$I(0)=-20c_1+15c_2+\frac{640}{697}=0\Rightarrow c_2=-\frac{494}{2091}$$

Thus, the expression for the charge is

$$Q(t)=\frac{4}{697}\left[\frac{e^{-20t}}{3}\left(-63\cos15t-116\sin15t\right)+\left(21\cos10t+16\sin10t\right)\right]$$

and the expression for the current is

$$Q(t)=\frac{1}{2091}\left[e^{-20t}\left(-1920\cos15t+13,060\sin15t\right)+120\left(-21\sin10t+16\cos10t\right)\right]$$

MATLAB Code:

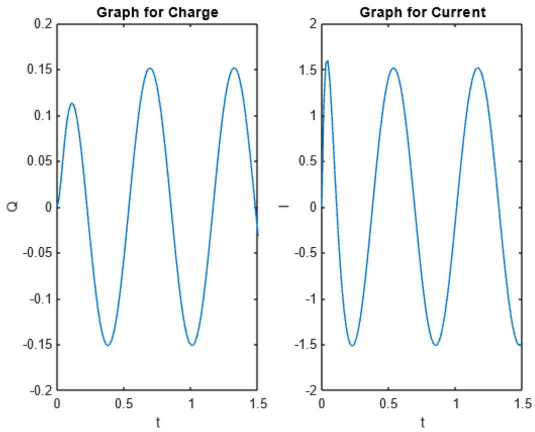
```
clear all
clc
syms Q(t) I(t)
ode = input('Enter the differential equation:');
DQ = diff(Q); DI = diff(I);
cond1 = Q(0) == 0;
cond2 = DQ(0) == 0;
conds= [cond1 cond2];
QSol(t) = dsolve(ode,conds);
QSol = simplify(QSol)
ISol=diff(QSol)
tt=linspace(0,1.5,100);
Qval=subs(QSol,tt);
Ival=subs(ISol,tt);
subplot(1,2,1),plot(tt,Qval)
title('Graph for Charge')
xlabel('t')
ylabel('Q')
subplot(1,2,2),plot(tt,Ival)
title('Graph for Current')
xlabel('t')
ylabel('I')
```

Input

```
Enter the differential equation:

diff(Q,t,2)+40*diff(Q,t)+625*Q
== 100*cos(10*t)
```

Output

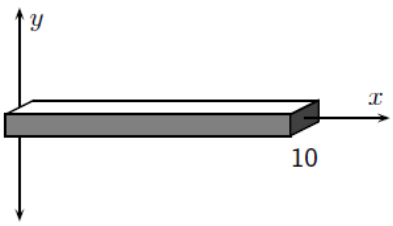
```
QSol(t) =

(84*cos(10*t))/697 +
(64*sin(10*t))/697 -
(84*cos(15*t)*exp(-20*t))/697 -
(464*sin(15*t)*exp(-20*t))/2091


ISol(t) =

(640*cos(10*t))/697 -
(840*sin(10*t))/697 -
(640*cos(15*t)*exp(-20*t))/697 +
(13060*sin(15*t)*exp(-
20*t))/2091
```

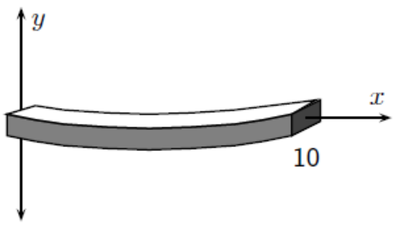


Bending (deflection) of a horizontal beam:

Mathematical Background:

Here we will take a look at the differential equations associated with beams that are suspended horizontally in some way. The beams themselves will not be horizontal over their entire lengths, because the force of gravity will cause some bending. The first thing to understand is the mathematical setup. Suppose that we have a beam of length 10 feet. We put the cross-sectional center of its left end at the origin of an xy coordinate plane, and the cross-sectional center of its right end at the point (10, 0). The longitudinal axis of symmetry of the beam then runs along the x-axis from $x=0$ to $x=10$; see the figure below and to the left.





Now the beam will deflect in some way, due to any weight it is supporting, including its own weight. The shape it takes will depend on the manner in which it is supported, but one possibility is shown in the figure above and to the right. The points along what was the original axis of symmetry of the beam now follow the graph of a function, which we will call $y(x)$. Note that the domain of the function is just the interval [0, 10]. Now, the **governing equation for deflection of horizontal beam** is

$$EI\frac{d^4y}{dx^4}=w(x) \qquad (1)$$

. The differential equation (1) itself is fourth order.

Here E is Young's modulus of elasticity for the material from which the beam is made, I is the moment of inertia of a cross-section of the beam and $w(x)$ is the load per unit length of the beam. If the beam has uniform cross-section and the only weight that it is supporting is its own weight, then $w(x)$ is a constant. We will consider only that situation.

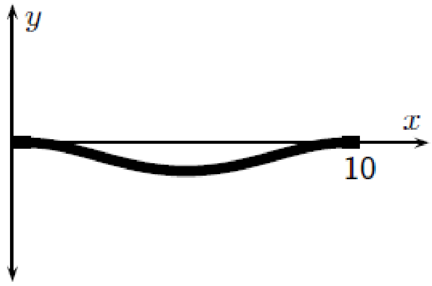


Fig. 6

Let's consider the situation shown in Figure 6, where both ends are embedded. Because the ODE (1) is fourth order, we will need four boundary conditions to determine all of the constants that will arise in solving it. We first recognize that because the two ends are supported, there will be no deflection at either end. Therefore $y(0) = y(10) = 0$. This will be the case for any horizontal beam that is supported at both ends. Next, we consider the fact that the ends of the beam are embedded horizontally into a wall. The embedding causes both ends to be horizontal right at the points where they leave the walls they are embedded in, so the slope of the beam is zero at those points. Mathematically we express this by $y'(0) = y'(10) = 0$. When we put the ODE together with these boundary conditions we get a boundary value problem. Suppose that for our ten foot beam beam $E = 10$, $I = 5$ & $w(x) = 100$.

The boundary value problem (BVP) that we have is then

$$50\frac{d^4y}{dx^4} = 100, \quad y(0) = 0, y'(0) = 0, \ y(10) = 0, y'(10) = 0 \qquad (2)$$

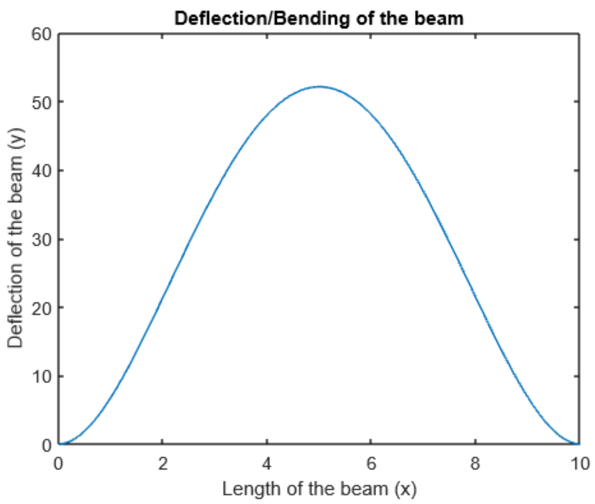
The MATLAB code for solving this BVP is as follows:

MATLAB Code:

```
clear all
clc
syms y(x)
ode = input('Enter the differential equation:');
X=input('Enter length of the beam:');
Dy = diff(y);
cond1 = y(0) == 0;
cond2 = Dy(0) == 0;
cond3 = y(10) == 0;
cond4 = Dy(10) == 0;
conds = [cond1 cond2 cond3 cond4];
ySol(x) = dsolve(ode,conds);
ySol = simplify(ySol)
xx=linspace(0,X,100);
yval=subs(ySol,xx);
plot(xx,yval)
title('Deflection/Bending of the beam')
xlabel('Length of the beam (x)')
ylabel('Deflection of the beam (y)')
```

Input

```
Enter the differential equation:
50*diff(y,x,4)==100
Enter length of the beam:
10
```



Output

```
ySol(x) =

(x^2*(x - 10)^2)/12
```

Exercise Problems

1. Suppose that the spring with a mass of 2 kg has natural length 0.5 m. A force of 25.6 N is required to maintain it stretched to a length of 0.7m. The spring is stretched to a length of 0.7 m and then released with initial velocity 0 m/s. The spring is immersed in a fluid with damping constant c=32. Find the position of the mass at any time t if it starts from the equilibrium position and is given a push to start with an initial velocity of 0.6 m/s.
2. A spring with mass of 2kg has damping constant 14, and a force of 6N is required to keep the spring stretched 0.5 m beyond its natural length. The spring is stretched 1m beyond its natural length and then released with zero velocity. Find the position of the mass at any time t.
3. Solve the initial value problem $\frac{d^2x}{dt^2}+w^2x=\sin\lambda t$ with $x(0)=0, x'(0)=0$ for the following cases:

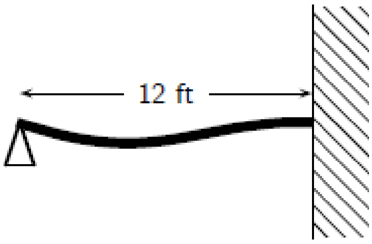
   (i) $w=1$ & $\lambda=2$

   (ii) $w=\lambda=2$

   Interpret the solution in terms of mass spring system.
4. Find the current $I(t)$ in an RLC circuit with $R=11\Omega, L=0.1H, C=10^{-2}F$, which is connected to a source of voltage $E(t)=100\sin 400t$. Assume that the current and the charge are zero when $t=0$.
5. A series circuit consists of a resistor with $R=20\Omega$, an inductor with $L=1H$, a capacitor with $C=0.002F$, and a 12-V battery. If the initial current and charge are both 0, find the current and charge at time t.

6. Find the deflection function $y = y(x)$ for each beam pictured below that is embedded at both ends, carrying a constant load of $w(x) = 150$ pounds per foot. Suppose also that $E = 30$ & $I = 80$, in the appropriate units.

(a)



(b)