KLS, GOGTE INSTITUTE OF TECHNOLOGY

BELAGAVI

MATLAB(2019-20)

2nd Semester (H divison)

Roll no : 860

USN : GIT19IS064

Name : Prerana Rao

Prerana. N. Rao
860

MATLAB - 1

SOLUTION OF DIFFERENTIAL EQUATION

To solve linear differential equation:

Expression	Matlab code
Create function of y(x)	syms y(x)
Differential equation of first order (dy/dx)	diff (y, x)
To find the solution of differential equation of first order (where ordinary differential equation is differential equation)	solution = dsolve (ode)
To solve differential equation of first order with initial conditions	solution = dsolve (ode, cond)

Examples:

1. $dy = xy$

→
```
% program to solve ordinary differential equation %
syms y(x)
ode = diff (y, x) = x * y
solution = dsolve (ode)
```

Scanned with CamScanner

solution:

>> diff

ode (x) =

diff (y(x), x) = x * y(x)

solution =

c1 * exp (x ^ 2/2)

2. $\frac{dy}{dx}$ = sin x * cos (y)

% program to solve ordnary diffierential equation %

syms y(x)

ode = diff (y, x) = = sin(x) * cos(y)

solution = dslove (ode)

ode (x) =

diff (y(x), x) = = cos (y(x)) * sin(x)

solution =

- log ((1 + exp (c1 - cos (x)) * pi) / (exp (c1 -
  cos (x)) + pi)) pi/2 * pi

3. $2\frac{dy}{dx}\cos x + 4y\sin x = \sin 2x$

```
% program to solve ordinary differential equation %
syms y(x)
ode = 2 * diff (y, x) * cos(x) + 4* y * sin(x) ==
        sin(2* x)
solution = dsolve (ode)

>> diff
ode (x) =
2 * cos(x) * diff (y(x), x) + 4 * sin(x) * y(x) = sin(2* x)
solution =
C1 * cos(x)^2 + cos(x)
```

4. $\frac{dy}{dx} = y\tan x - y^2\sec x$

```
% program to solve differential equation %
syms y(x)
ode = diff (y, x) == y* tan(x) - (y^2 * sec(x))
solution = dsolve (ode)

>> diff 1
ode (x) =
diff (y(x), x) == tan(x) * y(x) - y(x)^2 / cos(x)
solution :
1/ cos(x) * (c1 + tan (x))°
```

5 $xy(1+xy^2)\frac{dy}{dx} = 1$

→ % program to solve differential equation %
```
syms x(y)
ode = diff (x, y) == x * y * (1 + x * y^2)
solution = dsolve (ode)
```

```
>> diff 1
ode (y) =
diff (x(y), y) == y * x(y) * (y^2 * x(y) ^1)
solution =
exp (y^2 / 2)/ (C1 - 2 * exp (y^2/2) * (y^2/2 - 1))°
```

6 $(x+1)\frac{dy}{dx} = y + e^x(x+1)^2$; $y(0)=1$

⇒ % program to solve differential equation %
```
syms y(x)
ode = (x+1) * diff (y, x) == y + exp(x) * (x+1)^2
cond = y(0) == 1
solution = dsolve (ode, cond)
```

```
>> diff 1
ode (x) =
diff (y(x), x) * (x+1) == y(x) + exp (x) * (x+1)^2
cond =
y(0) == 1
solution =
 (exp(x)) * (x+1)
```

7 $\frac{dy}{dx} = -\frac{(x + y\cos x)}{(1 + \sin x)}$; y(0) = pi

→ % program to solve differential ~~effec~~ equation %

```
syms y(x)
ode = diff (y,x) == -(x+y * cos(x))/(1+sin(x))
cond = y(0) == pi
solution = dsolve (ode, cond)
```

```
>> diff 1
ode (x) =
diff (y(x), x) == -((x+ cos x) * y(x))(sin (x) + 1)
cond =
y(0) == pi
solution =
(-x^2+2 * pi) / (2 * (sin (x)+1))
```

8. $dr + (2r\cos\theta + \sin 2\theta)\, d\theta = 0$

→ % program to solve differential equation %

```
syms ~~y(x)~~ r(theta)
ode = diff (r, theta) + 2* r * cot (theta) == -sin(2* theta)
solution = dsolve (ode)
```

```
>> diff 1
ode (theta) =
2 * cot (theta) * r(theta) + diff (r (theta), theta) == -sin
(2 * theta)
solution = cos (2 * theta) / 4 + C1 / sin (theta)^2 - 1/4
```

1. $\frac{d^2y}{dx^2} + 6\frac{dy}{dx} + 8y = 0.$

→ % program to find general solution of I and higher order L.D.E without condition %

```
clc
syms y(x)
ode = input ('enter ode = \n');
ysol(x) = dsolve (ode);
simplify (ysol(x))
```

```
Enter ode :
diff (y,2) + 6 * diff (y,x) + 8 * y == 0
ans =
c2 * exp (-2 * x) + c1 * exp(-4 * x)
```

2. $\frac{d^3y}{dx^3} - 6\frac{d^2y}{dx^2} + 11\frac{dy}{dx} - 6y = 0$

```
Enter ode =
diff (y,x,3) - 6 * diff (y,x,2) + 11 * diff (y,x) - 6 * y
== 0
ans =
exp (x) * (c1 + c2 * exp (x) + c3 * exp (2 * x))
```

3 $\frac{d^2y}{dx^2} + 4\frac{dy}{dx} + 4y = 0$ given that $y(1)=0$, $y'(1)=-1$

→ % program to find solution of second order LDE without condition %

```
clc
syms y(x)
Dy = diff (y);
ode = input ('enter ode = \n');
cond 1 = input ("enter condition1 = \n');
cond 2 = input ("enter condition 2 = \n');
conds = [cond 1, cond 2]
ysol (x) = dsolve (ode , conds);
Simply (y sol (x))

Enter ode :
diff (y, x, 2) + 4 * diff (y, x) + 4 * y == 0
Enter condition 1 :
y (1) = = 0
Enter condition 2 :
D y (1) = = -1
conds =
[ y (1) = = 0 , subs ( diff (y (x, x) , x , 1) = = -1]

ans =
- exp (2 - 2 * x) * (x - 1)
```

4. $\frac{d^3y}{dx^3} - 8y = 0$

Enter ode =

diff (y, x, 3) - 8 * y = 0

ans =

exp (-x) * (c2 * cos (3^(1/2) * x) - c3 * sin (3^(1/2) * x) + c1 * exp (3 * x))

5. $\frac{d^2y}{dx^2} - 4\frac{dy}{dx} + 3y = 20\cos x$

Enter ode =

diff (y, x, 2) - 4 * diff (y, x) + 3 * y == 20 * cos(x)

ans =

2 * 5^(1/2) * cos(x + atan(2)) + c1 * exp(x) + c2 * exp (3 * x

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

Enter ode =

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

ans =

(x ^ 2 * exp (2 * x)) / 2 + C1 * exp (2 * x) + C2 * x * exp (2 * x) + 1

7. $\frac{d^3y}{dx^3} + 8y = x^2$ at $y(0) = 0$, $y'(0) = -1$

Enter ode:

diff (y, x, 2) + 8 * y == x^2

Enter conditions =

y(0) == 0

Enter condition 2 =

Dy(0) == -1

conds =

[y(0) == 0, subs (diff (y(x), x), x, 0) == -1]

ans =

cos (2 * 2^(1/2) * x / 32 - (2^(1/2) * sin(2*2^(1/2)*x)

/4 + x^2/8 - 1/32

	Matlab code.
To find laplace transform of f(t)	laplace (f, t, s)
LT of unit set fnctⁿ or LT of heaviside function	heaviside (t - a)
Inverse LT of F(s)	ilaplace (F, s, t)

1. $\sin(5t) - 4\cos(2t) + e^{-2t}$

→ % program to find laplace transform of function %

```
clc
syms t s
f = input ('enter a function f(t) = ')
F = laplace (f, t, s)

enter function f(t) = sin( 5 * t) + 4* cos (2 * t) +
                      exp (-2 * t)

f =
exp (-2*t) - 4 * cos (2 * t) + sin (5 * t)

F =
   1/(s+2) -(4 * 2) /(s^2 + 4) + 5 /(2^2+ 25)
```

2 $e^{3t}(2t-5)^3$

→
```
% program to find laplace transform %
clc
syms t s
f = input ('enter a function f(t) = ')
F = laplace(f, t, s)

enter a function f(t) = exp(3*t)*(2*t-5)^3
f =
exp(3*t)*(2*t-5)^2
F =
48/(s-3)^4 - 12/(2-3)^3 - (150*((5*s)/6-
7/2))/(s-3)^2
```

3. sin 3t . cos5t / t

⇒
```
% program to find laplace transform %
clc
syms t s
f = input ('enter a function f(t) = ')
F = laplace (f, t, s)
enter a function f(t) = sin(3*t) * cos(5*t)/t
f =
(cos(5* t) * sin(3* t)) / t
F =
atan(8/s)/2 - atan(2/s)/2
```

4. $t^2 e^{-2t} \sin(4t)$

→ % program to find LT %

```
clc
syms ts
 f = input ('enter a function f(t) = ')
 F = laplace (f, t, s)
```

```
Enter a function f(t) = t^2 * exp(-2*t) * sin(4*t)
f =
 t^2 * sin(4* t) * exp(-2*t)
F =
(8 * (2 *s + 4)^2)/((s+2)^2+16)^3 -
8 /((s+2)^2 + 16)^2
```

5. $\frac{\cos\sqrt{t}}{\sqrt{t}}$

% program to find LT %

```
clc
syms ts
f = input ('enter a function f(t) = ')
F = laplace (f, t, s)
enter a function f(t) = cos (sqrt (t)) /sqrt (t)

f =
cos (t^(1/2)) / t^(1/2)
F =
(pi^(1/2)* exp (-1/(4 * s))) /s^(1/2)
```

6. $t^2 u(t-2)$

→ % program to find laplace transform %

```
clc
syms ts
f = input ('enter a function f(t) = ')
F = laplace (f, t, s)

enter a function f(t) = t^2 * heaviside (t-2)
f =
  t^2 * heaviside (t-2)
F =
  (4 * exp (-2 * s)) / s + (4 * exp (-2 * s)) / s^2 +
  (s * exp (-2 * s)) / s^3
```

7. $\frac{\delta(t-2) + 4\,\delta(t-5)}{t}$

→ % program to find laplace transform %

```
clc
syms ts
f = input ('enter a function f(t) = ')
F = laplace (f, t, s)

enter a function f(t) = (dirac (t-2) + 4 * dirac (t-5))/t
f =
(dirac (t-2) + 4 * dirac (t-5)) / t
F =
exp (-2 * s)/2 + (4 * exp (-5 * s)) / 5
```

Inverse laplace:

1. $\dfrac{s+1}{(s-1)(2s+3)(s-4)}$

→ % program to find inverse laplace transform %

```
clc
syms t s
F = input('enter a function F(s) = ')
f = ilaplace (F, s, t)
enter a function F(s) = (s+1)/((s+1)*(2*s+3)*(s-4))
F =
(s+1)/((2*s+3)*(s-1)*(s-4))
f =
(5*exp(4*t))/33 - exp(-(3*t)/2)/55 -
(2*exp(t))/15.
```

2. $\dfrac{s+1}{s^2-7s+15}$

```
Enter a function F(s) = (s+1)/(s^2-7*s+15)
F =
(s+1)/(s^2-7*s+15)
f =
exp((7*t)/2)*(cos((11*(1/2)*t)/2) +
(9*11^(1/2)*sin(11^(1/2)*t)/2))/11)
```

3. $\frac{s}{s^4-1}$

→ % programme to find inverse laplace transform %

```
clc
syms t s
F = input ('enter a function F(s) = ')
f = ilaplace (F, s, t)
enter a function F(s) = s/(s^4-1)
F =
s/(s^4-1)
f =
exp(-t)/4 -cos(t)/2 + exp(t)/4
```

4. $\frac{1}{s^2(s^2+1)}$

% program to find inverse laplace transform %

```
clc
syms t s
F = input ('enter a function F(s) = ')
f = ilaplace (F, s, t)
enter a function F(s) = 1/(s^2*(s*2+1))
F =
  1/(s^2 * (s^2+1))
f =
  t - sin(t)
```

5. $\log \frac{(s+3)}{s(s+1)}$ / $\tan^{-1}\left(\frac{1}{s}\right)$

→ % program to find inverse laplace transform %

```
clc
syms t s
F = input ('enter a function F(s) = ')
f = ilaplace (F, s, t)

enter a function F(s) : atan (1/s)
F =
   atan (1/s)
f =
   sin (t)/t
```

MATLAB 4

In this lab we study different ways of evaluating integrals. First we see how to find anti derivative with help of matlab. Certain functions can be symbolically integrated in Matlab with the int command

Example:

Find an anti derivative for the function f(x) = x^2. We can do this in (at least) two different ways

The shortest is:

```
>> int ('x^2')
Ans:
= 1/3* x^3
```

The matlab function for performing symbolic integration is in (f,'t',a,b) where - f is symbolic expression to integrate "t" optionally specifies the variable of integration for case of several different symbols and a and b optionally specify numberical limits of integration

```
>> syms x
>> f = x^2
>> int (f,'x',0,1)
```

Example:

We solve the example $\int_1^2 \int_3^4 (xy + e^y)\,dy\,dx$ using matlab command

```
>> syms x y
>> F = x* y + exp(y)
F =
exp(y) + x*y
>> int(int (F, 'y', 3, 4), 'x', 1, 4)
ans =
3* exp(4) - 3* exp(3) + 105/4
```

Simpibaly we can evaluate triple integrals as follows

-
```
>> syms x y z
>> F = x + y + z
F =
>> I = int(int(int (F, 'z', 0, 1), 'y', 0, 2 ), 'x', 0, 3 )
I =
18
```

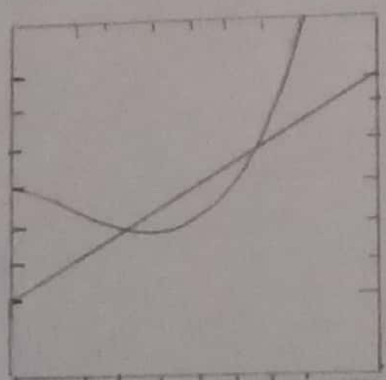
-
```
>> syms x y z
>> F = exp(x + y + z)
F =
exp (x + y + z)
>> I = int (int (int (F, 'z', 0, x+y), 'y', 0, x), 'x', 0, 1)
I =
((exp(1) - 1)^3* (exp (1) + 3)) /8
```

MATLAB - 5

We will address the problem of determining limits for a double integral from a geometric description of the region of integration.

While MATLAB cannot do that for us, it can provide some guidance through its graphics and can also confirm that limits we have choosen define the region we intended. For a first example, we will evaluate $\iint xy(x+y)\,dx\,dy$ with the bounded region between curves $y = x^2$ and $y = x$. We begin by plotting two curves on same axes. You many need to experiment with interval to get a useful plot; it should be large enough to show the region of interest, but small enough so that region of interest occupies most of plot.

```
>> syms x y
>> y1 = x^2;
>> y2 = x;
>> ezplot(y1, [-1,2]); hold on ; ezplot(x, [-1, 2])
```



In oder to limits for x are values for which two functions coincide. We use solve to find them.

```
>> limits >> solve (y1 - y2, x)

limits =
  0
  1
```

Evidently our limits define the right regions. We can now use them to integrate any function we like over the region in question.

```
>> I = int(int (x * y * (x+y), 'y', y1, y2), 'x', 0, 1)

I =
3/56
```

Note:

1) The logic of above MATLAB command is based on particular example.

2) Program and command will change according to requirement of example (syntax remains same)