1. Consistency of a system Non Homogeneous

A=[1,2,3;4,5,7;3,3,4];

B=[14;35;21];

[m,n]=size(A);

Au=[A,B];

rA=rank(A)

rAu=rank(Au)

Number\_of\_Unknown=n

if rA == rAu

if rA == n

disp('The system AX=B has Unique solutions');

x = linsolve(A,B)

else

disp('The system AX=B has Infinitely Many solutions')

end

disp('The number of free variables =')

fv=n-rA

disp('The solution can be obtained from the matrix')

Au=rref(Au)

else

disp('The system AX=B has No solutions(inconsistent)')

end

Consistency of homogeneous system

A=[1,2,3;4,5,7;3,3,4];

rA=rank(A)

Number\_of\_Unknown=n

if rA == n

disp('The system AX=O has Unique trivial solution');

else

disp('The system AX=B has Infinitely Many solutions')

end

2.Gauss Seidal Method

A=[27,6,-1;6,15,2;1,1,54];

B=[85;72;110];

x = 0;

y = 0;

z = 0;

n=10; % number of iterations

for i = 1:n

x = (B(1,1)-A(1,2)\*y-A(1,3)\*z)/A(1,1);

y = (B(2,1)-A(2,1)\*x-A(2,3)\*z)/A(2,2);

z = (B(3,1)-A(3,1)\*x-A(3,2)\*y)/A(3,3);

end

fprintf(&#39;The solution of the given system of linear equation after %d iteration is&#39;,

n)

x

y

z

3.Rayleigh Power Method to find Largest Eigen value and corresponding Eigen Vector

A = [1,1,3;1,5,1;3,1,1]; % Given matrix

X = [1;0;0]; % initial eigen vector

n = 20; % number of iterations

for i=1:n

X = A\*X;

M = max(abs(X));

X=X/M;

end

fprintf(&#39;The Dominant Eigen Value is %f\n&#39;,M)

disp(&#39;The corresponding eigen vector is &#39;)

X

Smallest Eigen value and corresponding Eigen Vector

A = [1,1,3;1,5,1;3,1,1]; % Given matrix

X = [1;0;0]; % initial eigen vector

n = 20; % number of iterations

for i=1:n

X = A\*X;

M = max(abs(X));

X=X/M;

end

B=A-M\*eye(size(A));

Y=[1;0;0];

for i=1:n

Y = B\*Y;

N = -max(abs(Y));

Y=Y/N;

end

small=N+M;

fprintf(&#39;The Smallest Eigen Value is %f\n&#39;,small)

disp(&#39;The corresponding eigen vector is &#39;)

Y

**4. 2D PLOTS for Cartesian and Polar curves**

|  |  |
| --- | --- |
| x = [1 2 3 4 5 6];  y = [3 -1 2 4 5 1];  plot(x,y) | x = 0:pi/100:2\*pi;  y = sin(x);  plot(x,y) |

x = 0:pi/100:2\*pi;

y = sin(x);

plot(x,y)

xlabel(’x = 0:2\pi’)

ylabel(’Sine of x’)

title(’Plot of the Sine function’)

Multiple Plots

|  |  |
| --- | --- |
| x = 0:pi/100:2\*pi;  y1 = 2\*cos(x);  y2 = cos(x); | y3 = 0.5\*cos(x);  plot(x,y1,’--’,x,y2,’-’,x,y3,’:’) ;  xlabel(’0 \leq x \leq 2\pi’) ;  ylabel(’Cosine functions’) ;  legend(’2\*cos(x)’,’cos(x)’,’0.5\*cos(x)’); |

Polar Plot

|  |  |
| --- | --- |
| theta = 0:0.01:2\*pi;  rho = sin(2\*theta).\*cos(2\*theta);  polarplot(theta,rho) | theta = linspace(0,6\*pi);  rho1 = theta/10;  polarplot(theta,rho1)  rho2 = theta/12;  hold on  polarplot(theta,rho2,'--')  hold off |

Implicit Plot

syms x y

fimplicit(y.^2\*(0.1-x)==x.^3)

title 'cissoid'

xlabel 'x'

ylabel 'y'

**5,** Finding angle between polar curves,

syms theta rho1 rho2 phi1 phi2 phi

rho1 = sin(theta)+ cos(theta);

rho2 =2\* sin(theta);

r1 = diff(rho1,theta);

r2 = diff(rho2,theta);

sol\_theta = solve(rho1 == rho2, theta)

tphi1=solve(r1/rho1,sol\_theta)

tphi2=solve(r2/rho2,sol\_theta)

if tphi1 ~= Inf || tphi2 ~= Inf

p = tphi1\*tphi2

if p == -1

fprintf("Curves are orthogonal");

else

phi1=atan(tphi1)

phi2=atan(tphi2)

phi = subs(abs(phi1-phi2))

end

else

phi1=acot(1/tphi1)

phi2=acot(1/tphi1)

phi = subs(abs(phi1-phi2), theta, sol\_theta)

end

6 .Radius of curvature

ROC  Cartesian

syms x y y1 y2

y=sqrt(2\*x);

y1=diff(y,x);

y2=diff(y1,x);

Y1=simplify(y1)

Y2=simplify(y2)

ROC=abs((1+Y1^2)^(1.5)/Y2);

simplify(ROC)

ROC polar

syms theta r r1 r2

r=1-cos(theta);

r1=diff(rho,theta);

r2=diff(r1,theta);

R1=simplify(r1)

R2=simplify(r2)

ROC=abs((r^2+R1^2)^(1.5)/r^2+2\*R1^2-r\*R2);

simplify(ROC)

ROC parametric

syms t x y

x=cos(t)+log(tan(t/2))

y=sin(t)

y1=diff(y,t)/diff(x,t);

s1=simplify(y1);

y2=diff(y1,t)/diff(x,t);

s2=simplify(y2);

ROC=abs((1+s1^2)^(1.5)/s2);

simplify(ROC)

ROC Pedal

syms r p

diff(solve(r^3==p^2,r),p)

ROC=r\*diff(solve(r^3==p^2,r),p)

**7. Solution of first order differential equation and plotting the graphs**

Solve the differential equation 2x2y’’(x)+3xy’(x)-y(x)=0.

syms y(x)

ode = 2\*x^2\*diff(y,x,2)+3\*x\*diff(y,x)-y = = 0;

ySol(x) = dsolve(ode)

Solve this nonlinear differential equation with an initial condition. (y’+y)2 = 1; with the condition y(0)=0.

syms y(t)

ode = (diff(y, t)+y)^2 = = 1;

cond = y(0) = = 0;

ysol(t) = dsolve(ode, cond)

t=linspace(-2,2);

plot(t,ysol(t))

8. Finding G.C.D using Euclid’s algorithm

syms M

A=input('What is A');

B=input('what is B');

C=[A B];

D = sym(C);

[G,M] = gcd(D);

while A~=0 & B~=0

if A>B

A=A-B;

else

B=B-A;

end

GCD=A+B;

end

fprintf('GCD of A and B is %d \n',GCD)

fprintf('LC of A and B is:')

disp(M)

9. **Taylors and Maclaurins series with plotting**

Maclaurin Series

|  |  |
| --- | --- |
| taylor([f](file:///C:\Program%20Files\MATLAB\R2016b\help\symbolic\taylor.html?searchHighlight=taylors#inputarg_f)) | Approximates f with the [Taylor series expansion](file:///C:\Program%20Files\MATLAB\R2016b\help\symbolic\taylor.html#busozb7-8) of f up to the fifth order at the point x=0. |
| taylor(f, x, 'ExpansionPoint', a) | Approximates f with the [Taylor series expansion](file:///C:\Program%20Files\MATLAB\R2016b\help\symbolic\taylor.html#busozb7-8) of f up to the fifth order at the point x=a. |
| taylor(f, 'ExpansionPoint', a, 'Order', n) | Approximates f with the [Taylor series expansion](file:///C:\Program%20Files\MATLAB\R2016b\help\symbolic\taylor.html#busozb7-8) of f up to the nth order at the point x=a. |

|  |  |
| --- | --- |
| >> syms x  >> taylor(exp(x))  ans =  x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1  >> taylor(sin(x))  ans =  x^5/120 - x^3/6 + x  >> taylor(sin(x))  ans =  x^4/24 - x^2/2 + 1 | >>syms x  >>taylor(log(x), x, 'ExpansionPoint', 1)  ans =  x - (x - 1)^2/2 + (x - 1)^3/3 - (x - 1)^4/4 + (x - 1)^5/5 - 1  >>syms x  >>f = sin(x)/x  >>taylor(f, x, 'Order', 8)  ans =  - x^6/5040 + x^4/120 - x^2/6 + 1 |
| %Matlab Program for Maclaurins series expansion of y = f(x),  %about a point x=a.  syms x  F = exp(x);  T1 = taylor(F, 'ExpansionPoint', 2, 'Order', 1);  T2 = taylor(F, 'ExpansionPoint', 2, 'Order', 2);  T5 = taylor(F, 'ExpansionPoint', 2, 'Order', 5);    fplot([T1 T2 T5 F])  xlim([-5 5])  ylim([-20 20])  grid on  legend('approximation of exp(x) up to O(x^1)',...  'approximation of exp(x) up to O(x^2)',...  'approximation of exp(x) up to O(x^{5})',...  'exp(x)','Location','Best')  title('McLaurins Series Expansion about =2') | %Matlab Program for Maclaurins series expansion of y = f(x),  %about a point x=a.  syms x  F = sin(x);  T1 = taylor(F, 'ExpansionPoint', 2, 'Order', 1);  T2 = taylor(F, 'ExpansionPoint', 2, 'Order', 2);  T5 = taylor(F, 'ExpansionPoint', 2, 'Order',56);    fplot([T1 T2 T5 F])  xlim([-5 5])  ylim([-20 20])  grid on    legend('approximation of exp(x) up to O(x^1)',...  'approximation of exp(x) up to O(x^2)',...  'approximation of exp(x) up to O(x^{5})',...  'sin(x)','Location','Best')  title('McLaurins Series Expansion about =2') |

10. Finding partial derivatives, Jacobian and plotting the graph.

syms x y

f=x^2+2\*y^2-22

P=diff(f,x)

Output

f =x^2 + 2\*y^2 - 22

P =2\*x

% Here, I have calculated the (partial) differentiation of function &quot;f&quot; w.r.t &#39;x&#39;

% Now, I want to know the value of &#39;P&#39; at certain point (say x=1.5, y=2.0)

syms x y

f=x^2+2\*y^2-22

P=diff(f,x)

subs(P,{x,y},{1.5,2})

output

f =x^2 + 2\*y^2 - 22

P =2\*x

ans =3

For second order derivatives

syms x y

f=x^2+2\*y^2-22

P=diff(f,x)

P = 2\*x

p1=diff(diff(f,x))

p1 =2

For mixed order derivatives

syms x y

f=x^2+2\*y^2-22

p2=diff(diff(f,x),y)

f =x^2 + 2\*y^2 - 22

p2 =0

Jacobian

syms x y z

A=jacobian ([x\*y\*z, y^2, x + z], [x, y, z])

A =[y\*z, x\*z, x\*y]

[ 0, 2\*y, 0]

[ 1, 0, 1]

det(A)

Output ans =2\*y^2\*z - 2\*x\*y^2

**10.**