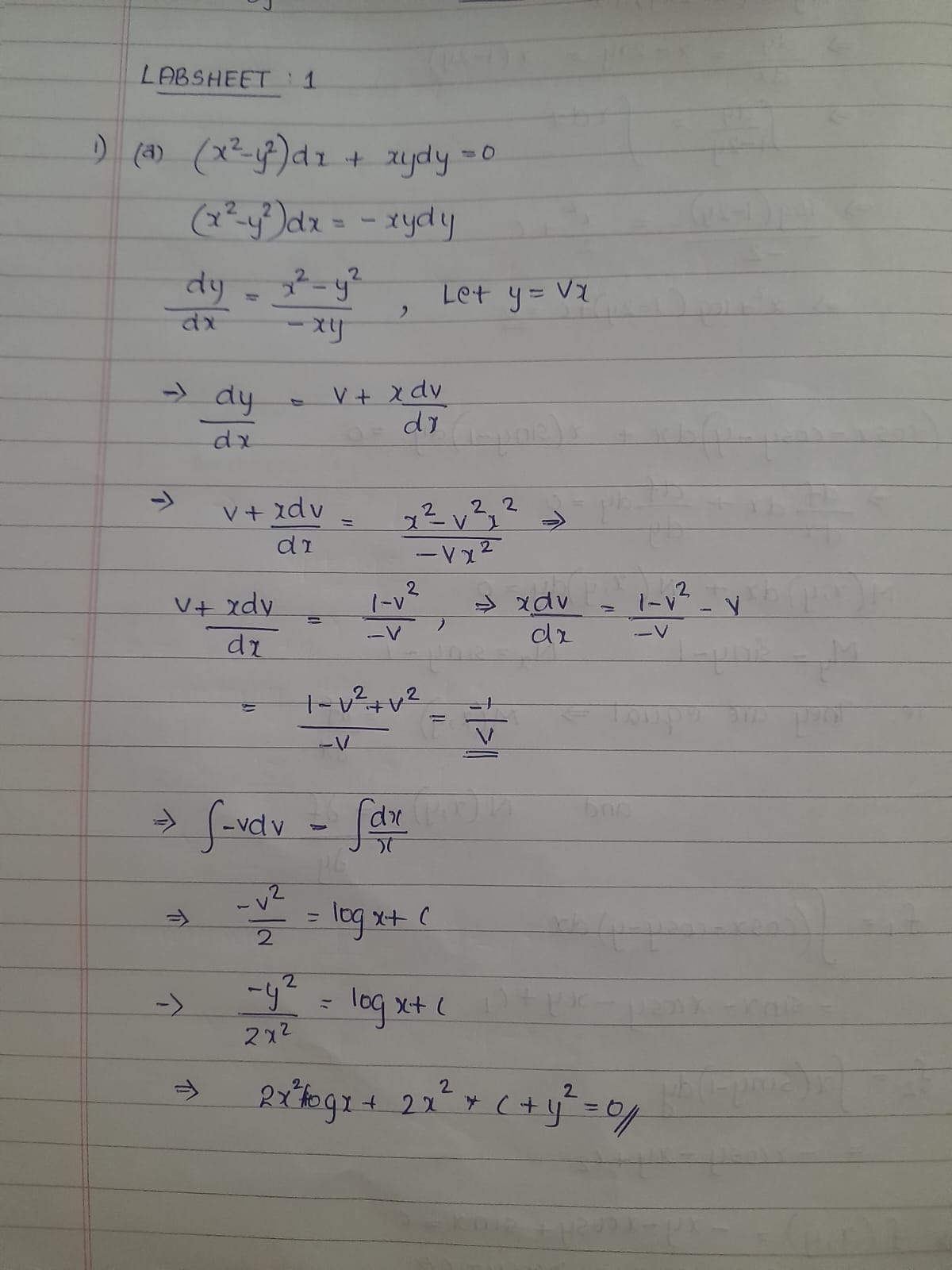
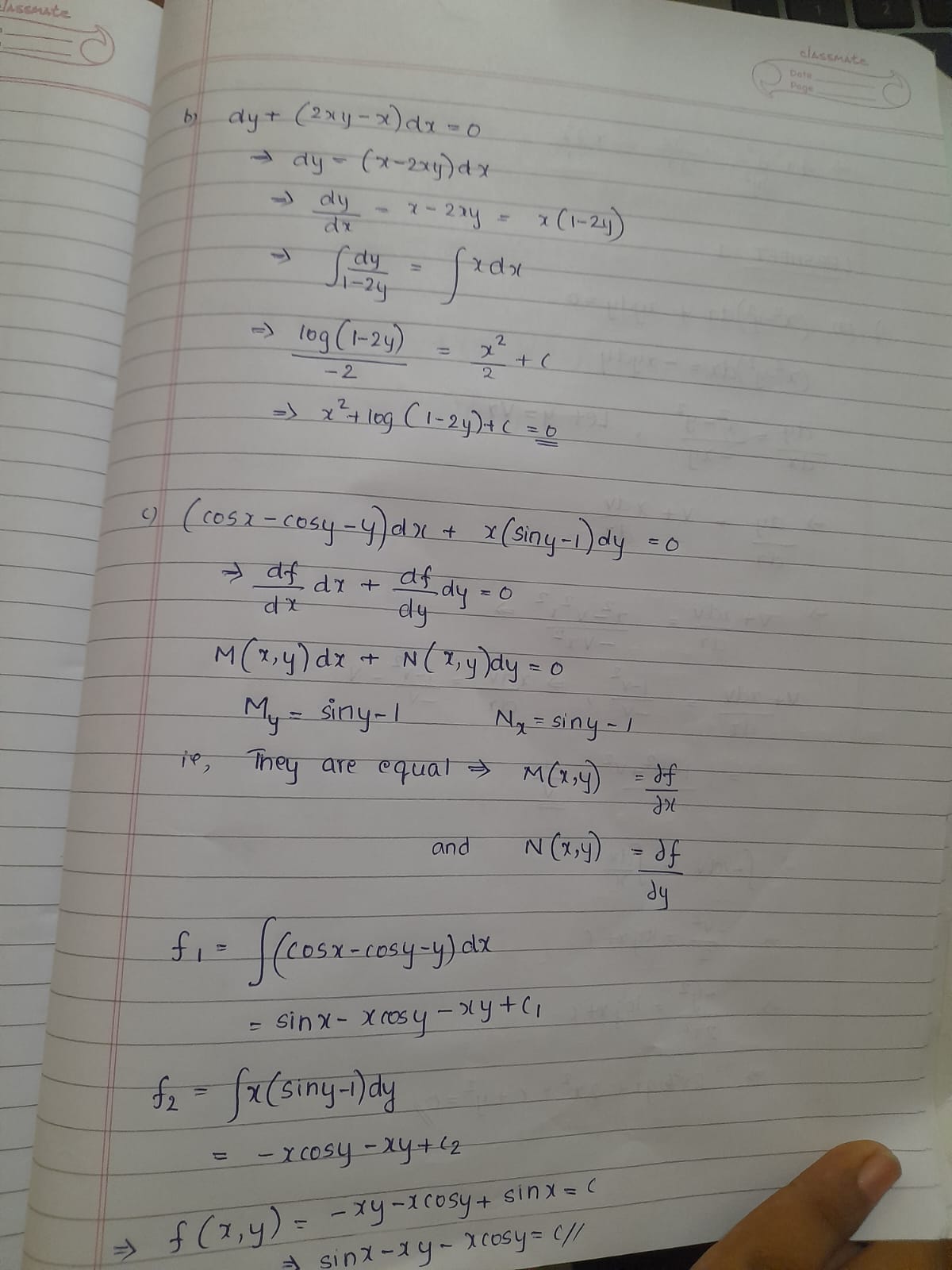
**MATHS ASSIGNMENT**

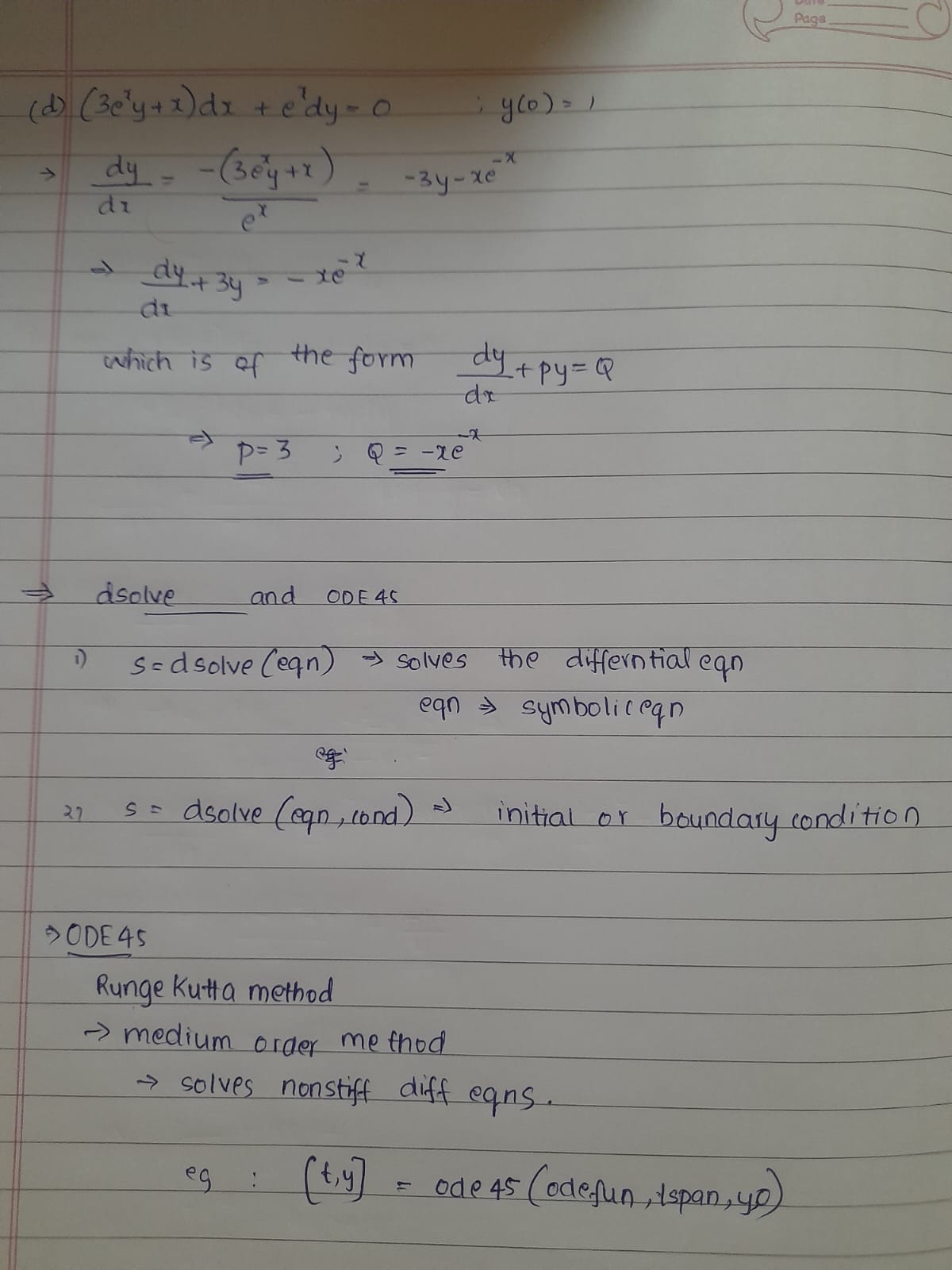
Anuvind MP

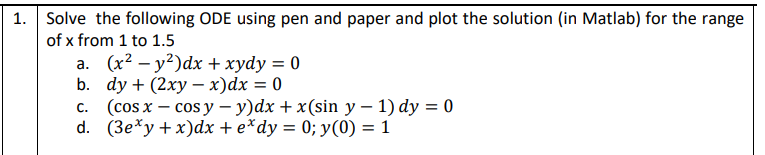
AM.EN.U4AIE22010

**PEN AND PAPER:**













%given de : xydy + (x^2-y^2)dx = 0 , y(1) = 1

%using pen and paper and finding the solution

u = 1.001:0.01:1.5;

v = sqrt(2\*u.^2.\*log(abs(exp(0.5)./u)));

plot(u,v,"\*b")

%analytical solution

syms x y(x);

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

cond = y(1) == 1;

sol = dsolve(ode, cond);

x\_values = 1:0.01:1.5;

y\_values = subs(sol, x, x\_values);

plot(x\_values, y\_values, '\*r');

xlabel('x');

ylabel('y');

%title('Solution of the Differential Equation');

% numerical solution using ode45

xspan = [1 1.5];

y0 = 1;

[x,y] = ode45(@(x, y) y^2 - x^2,xspan,y0);

hold on,plot(x,y,".b")

EulerMethod(1, 1, 0.01, 50);

RK4(1, 1, 0.01, 50);

function EulerMethod(x0, y0, h, n)

y = y0;

x = x0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

y = y + h \* f(x, y);

x = x + h;

end

plot(x\_values, y\_values, '-o');

xlabel('x');

ylabel('y');

%title('Euler''s Method');

end

function RK4(x0, y0, h, n)

x = x0;

y = y0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

k1 = h \* f(x, y);

k2 = h \* f(x + 0.5 \* h, y + 0.5 \* k1);

k3 = h \* f(x + 0.5 \* h, y + 0.5 \* k2);

k4 = h \* f(x + h, y + k3);

y = y + (1/6) \* (k1 + 2\*k2 + 2\*k3 + k4);

x = x + h;

end

plot(x\_values, y\_values, '-g');

xlabel('x');

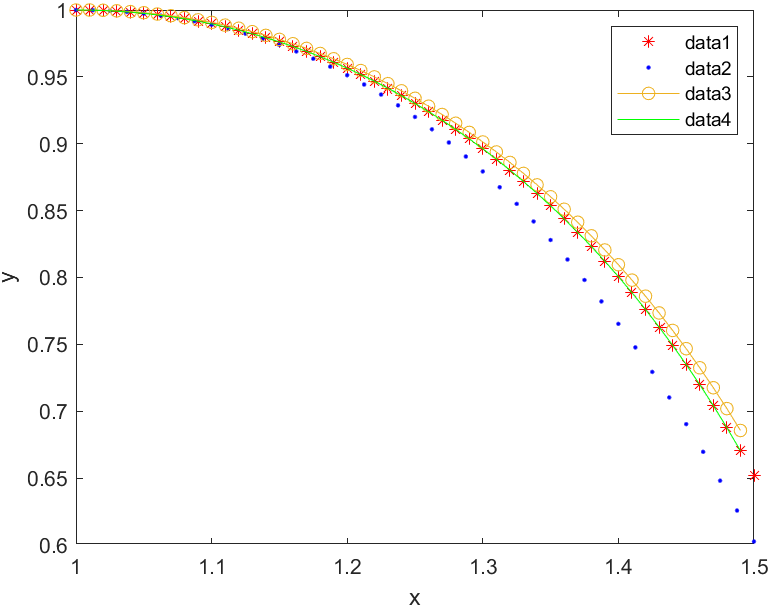
ylabel('y');

end

function dydx = f(x, y)

dydx = (y^2 - x^2) / (x \* y);

end





% analytical solution

syms x y(x);

ode = diff(y(x))+(2\*x\*y(x)-x);

cond = y(1) == 1;

sol = dsolve(ode, cond);

x\_values = 1:0.01:1.5;

y\_values = subs(sol, x, x\_values);

plot(x\_values, y\_values, '\*r');

xlabel('x');

ylabel('y');

%ode45

initial\_cond=1;

[x,y]=ode45(@(x,y) (x\*(1-2\*y)),x\_values,initial\_cond);

hold on; plot(x,y,"-k");

% Rk4

RK4(1,1,0.01,50);

%EulerMethod

EulerMethod(1,1,0.01,50);

function RK4(x0, y0, h, n)

x = x0;

y = y0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

k1 = h \* f(x, y);

k2 = h \* f(x + 0.5 \* h, y + 0.5 \* k1);

k3 = h \* f(x + 0.5 \* h, y + 0.5 \* k2);

k4 = h \* f(x + h, y + k3);

y = y + (1/6) \* (k1 + 2\*k2 + 2\*k3 + k4);

x = x + h;

end

plot(x\_values, y\_values, '-g');

xlabel('x');

ylabel('y');

end

function EulerMethod(x0, y0, h, n)

y = y0;

x = x0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

y = y + h \* f(x, y);

x = x + h;

end

plot(x\_values, y\_values, '-o');

xlabel('x');

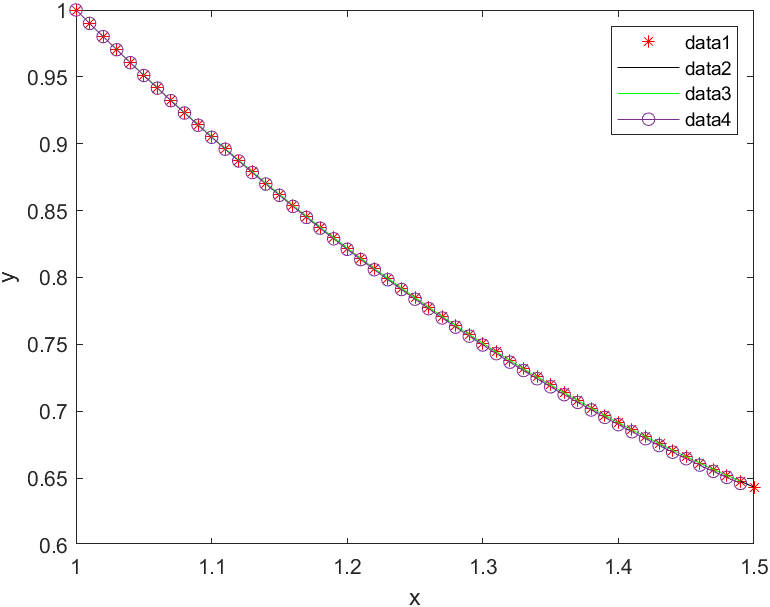
ylabel('y');

end

function dydx = f(x, y)

dydx = x-2\*x\*y;

end





%analytical solution

syms x y(x);

ode = diff(y(x))-(cos(x)-cos(y(x))-y(x))/(x-x\*sin(y(x)))==0;

cond = y(1) == 1;

%sol = dsolve(ode, cond);

x\_values = 1:0.01:1.5;

%y\_values = subs(sol, x, x\_values);

%plot(x\_values, y\_values, '\*r');

xlabel('x');

ylabel('y');

%ode45

initial\_cond = 1;

[x,y] = ode45(@(x,y) (cos(x) - cos(y) - y) / (x - x\*sin(y)),x\_values,initial\_cond);

hold on; plot(x,y,"\*k");

%RK4

RK4(1,1,0.01,50);

%EulersMethod

EulerMethod(1,1,0.01,50);

function RK4(x0, y0, h, n)

x = x0;

y = y0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

k1 = h \* f(x, y);

k2 = h \* f(x + 0.5 \* h, y + 0.5 \* k1);

k3 = h \* f(x + 0.5 \* h, y + 0.5 \* k2);

k4 = h \* f(x + h, y + k3);

y = y + (1/6) \* (k1 + 2\*k2 + 2\*k3 + k4);

x = x + h;

end

plot(x\_values, y\_values, '-g');

xlabel('x');

ylabel('y');

end

function EulerMethod(x0, y0, h, n)

y = y0;

x = x0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

y = y + h \* f(x, y);

x = x + h;

end

plot(x\_values, y\_values, '-o');

xlabel('x');

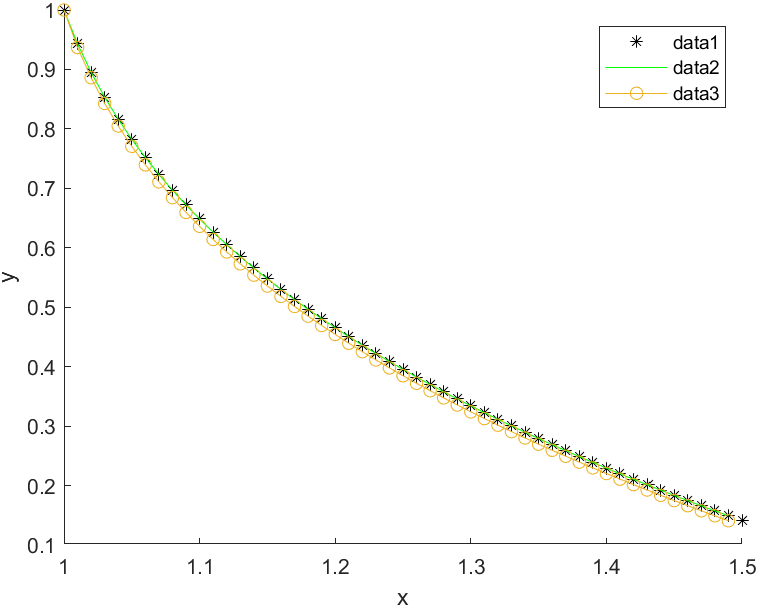
ylabel('y');

end

function dydx = f(x, y)

dydx = (cos(x)-cos(y)-y)/(x-x\*sin(y));

end





% Analytical solution

syms x y(x);

cond = y(0) == 10;

ode = diff(y(x))- (3\*exp(x)\*y+x)/-exp(x)==0;

sol = dsolve(ode, cond);

x\_values = 1:0.01:1.5;

y\_values = subs(sol, x, x\_values);

plot(x\_values, y\_values, '\*r');

xlabel('x');

ylabel('y');

% Ode45 implementation

ode\_function = @(x, y) (3\*exp(x)\*y+x)/-exp(x);

x\_span = [1 1.5];

initial\_cond = 1;

[x, y] = ode45(ode\_function, x\_span, initial\_cond);

hold on; plot(x,y,"\*k");

%RK4

RK4(1,1,0.01,50);

%EulersMethod

EulerMethod(1,1,0.01,50);

function RK4(x0, y0, h, n)

x = x0;

y = y0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

k1 = h \* f(x, y);

k2 = h \* f(x + 0.5 \* h, y + 0.5 \* k1);

k3 = h \* f(x + 0.5 \* h, y + 0.5 \* k2);

k4 = h \* f(x + h, y + k3);

y = y + (1/6) \* (k1 + 2\*k2 + 2\*k3 + k4);

x = x + h;

end

plot(x\_values, y\_values, '-g');

xlabel('x');

ylabel('y');

end

function EulerMethod(x0, y0, h, n)

y = y0;

x = x0;

x\_values = zeros(1, n);

y\_values = zeros(1, n);

for j = 1:n

x\_values(j) = x;

y\_values(j) = y;

y = y + h \* f(x, y);

x = x + h;

end

plot(x\_values, y\_values, '-o');

xlabel('x');

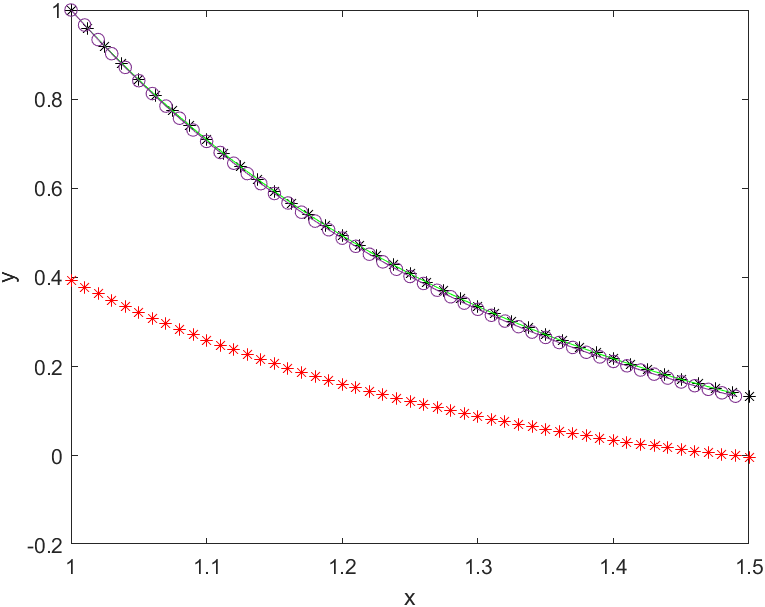
ylabel('y');

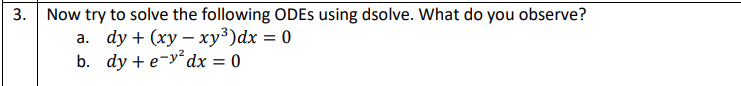
end

function dydx = f(x, y)

dydx = (3\*exp(x)\*y+x)/-exp(x);

end





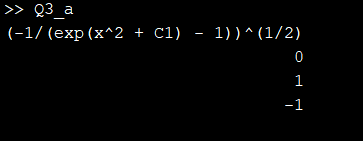
syms x y(x);

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

sol = dsolve(ode);

disp(sol)

Output:





syms x y(x);

ode = diff(y(x)) + exp(-y(x)^2) == 0;

initial\_condition = y(0) == 1;

sol = dsolve(ode,initial\_condition);

disp(sol);

x\_values = 1:0.01:1.5;

y\_values = subs(sol, x, x\_values);

plot(x\_values, y\_values, '\*r');

xlabel('x');

ylabel('y');

%cannot be done beacuse dsolve is unable to find a symbolic solution



dydx = @(x, y) -(3\*exp(x\*y)+x)/exp(x);

xspan = [1 1.5];

y0 = 1;

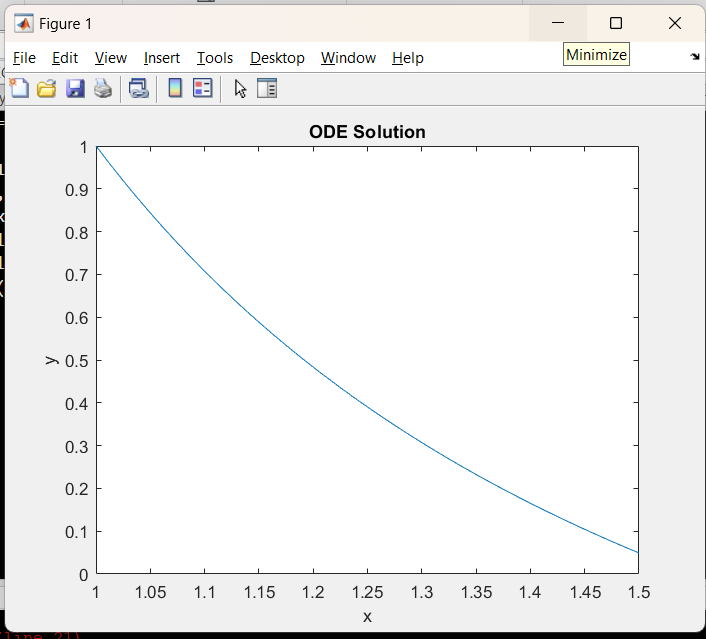
[xsol, ysol] = ode45(dydx, xspan, y0);

plot(xsol, ysol);

xlabel('x');

ylabel('y');

title('ODE Solution');





Euler :

function odeEuler()

plotsoln();

rnge = [1, 1.5];

y0 = 1;

[X, Y] = EulerMethod(@ode1, y0, rnge);

hold on; plot(X, Y, '\*r');

end

function [X, Y] = EulerMethod(dydx, y0, xspan)

X = xspan(1):0.01:xspan(2);

Y = zeros(size(X));

Y(1) = y0;

for i = 1:length(X)-1

Y(i+1) = Y(i) + 0.01 \* dydx(X(i), Y(i));

end

end

function dydx = ode1(x, y)

dydx = (y.^2 - x.^2)/(x\*y);

end

function dydx = ode2(x, y)

dydx = (y+ cos(y) - cos(x))/(x\*(sin(y)-1));

end

function plotsoln()

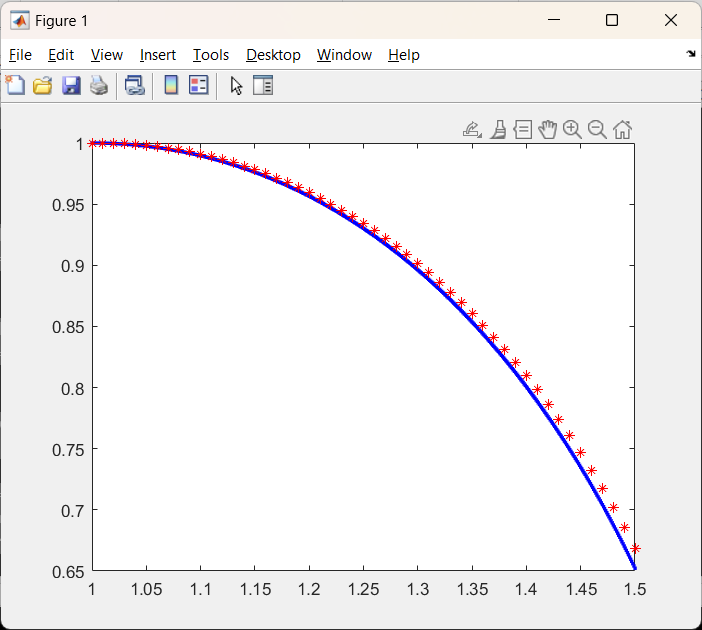
clear all; clc; close all

rnge = 1:0.001:1.5;

y = rnge.\*sqrt(1-2\*log(rnge));

plot(rnge, y, '.b');

end



RK Method :

function odeRK()

plotsoln();

rnge = [1, 1.5];

y0 = 1;

[X, Y] = RKMethod(@ode1, y0, rnge);

hold on; plot(X, Y, '\*r');

end

function [X, Y] = RKMethod(dydx, y0, xspan)

X = xspan(1):0.01:xspan(2);

Y = zeros(size(X));

Y(1) = y0;

for i = 1:length(X)-1

% Update y using Runge-Kutta method

k1 = 0.01 \* dydx(X(i), Y(i));

k2 = 0.01 \* dydx(X(i) + 0.01/2, Y(i) + k1/2);

k3 = 0.01 \* dydx(X(i) + 0.01/2, Y(i) + k2/2);

k4 = 0.01 \* dydx(X(i) + 0.01, Y(i) + k3);

Y(i+1) = Y(i) + (k1 + 2\*k2 + 2\*k3 + k4)/6;

end

end

function dydx = ode1(x, y)

dydx = (y.^2 - x.^2)/(x\*y);

end

function plotsoln()

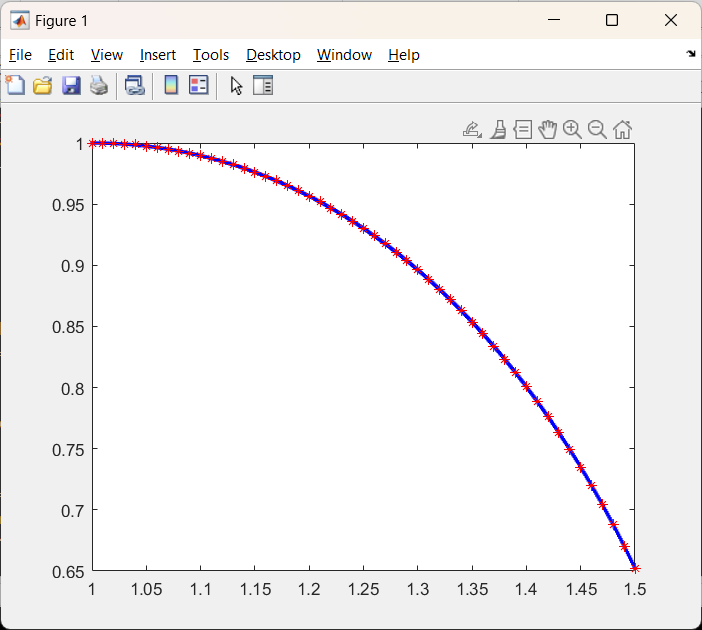
clear all; clc; close all

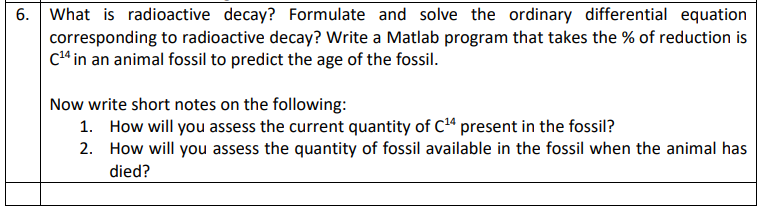
rnge = 1:0.001:1.5;

y = rnge.\*sqrt(1-2\*log(rnge));

plot(rnge, y, '.b');

end





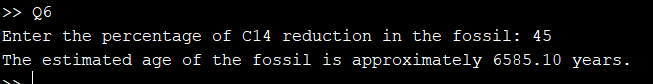
decay\_constant\_C14 = 0.0001212598;

percentage\_reduction = input('Enter the percentage of C14 reduction in the fossil: ');

reduction\_factor = percentage\_reduction / 100;

age = -log(reduction\_factor) / decay\_constant\_C14;

fprintf('The estimated age of the fossil is approximately %.2f years.\n', age);



write a short note on How will you assess the current quantity of C14 present in the fossil?

**Ans:**

Fossils reveal their age through C14, a radioactive timer. Sensitive machines directly measure its remaining whisper, or stable isotopes hint at its past abundance. But beware, contamination and environment can muddle the message. Experts decipher these clues, unlocking ancient secrets one decay at a time.

How will you assess the quantity of fossil available in the fossil when the animal has died?

**Ans**:

Assessing fossil quantity involves field excavation, estimating volume, utilizing advanced imaging like photogrammetry, conducting laboratory analysis, and employing remote sensing techniques such as LiDAR. These methods help determine the extent, density, and composition of fossil material, aiding in understanding ancient ecosystems and population dynamics.