MATLAB-Programming

Solving 1st Order differential Equation using Mid-point method.

By Mojahed Nour– Dec 15, 2019

CONTENTS

1st ODE using Mid-point method

01

Solving 1st ODE using Mid-point method.

What Is 1st Order differential Equation?

a differential equation containing one or more functions of one independent variable and the derivatives of those functions.

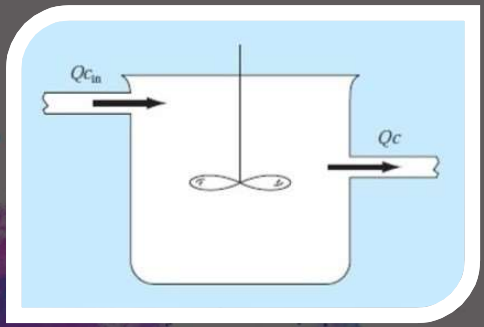
What is the task about ?

Writing a MATLAB function that solves 1st ODE using Mid-point method to solve The accumulation of mass in a reactor(i.e., V(dc/dt)=Qcin-Qc0) .

NOTE: we need to solve ODE .So, we can get the value of c (concentration).Also, this equation is hard to be solved analytically therefore we need to use an approximation method to solve it numerically(e.g., Midpoint method).

02 The technique & code

The mechanism of Solving 1st ODE :



- The accumulation of mass in a reactor is given by the ODE:
  $V(dc/dt)=Qcin-Qc$
  Where V is the **volume**, c is the **concentration** and Q is the **flow rate**.
- It is given that: initial cin = 50 mg/m3, Q = 5 m3/min, V = 100 m3, and c0 = 10 mg/m3 is the initial concentration at t = 0.
- Using the Midpoint Method to find c, up to t = 50 min, with a step size(i.e., h) of 5.
- Plotting the true, approximate solutions and the absolute true error on the same graph .

NOTE: we need to solve ODE .So, we can get the value of c (concentration).Also, this equation is hard to be solved analytically therefore we need to use an approximation method to solve it numerically(e.g., Midpoint method).

02 The code

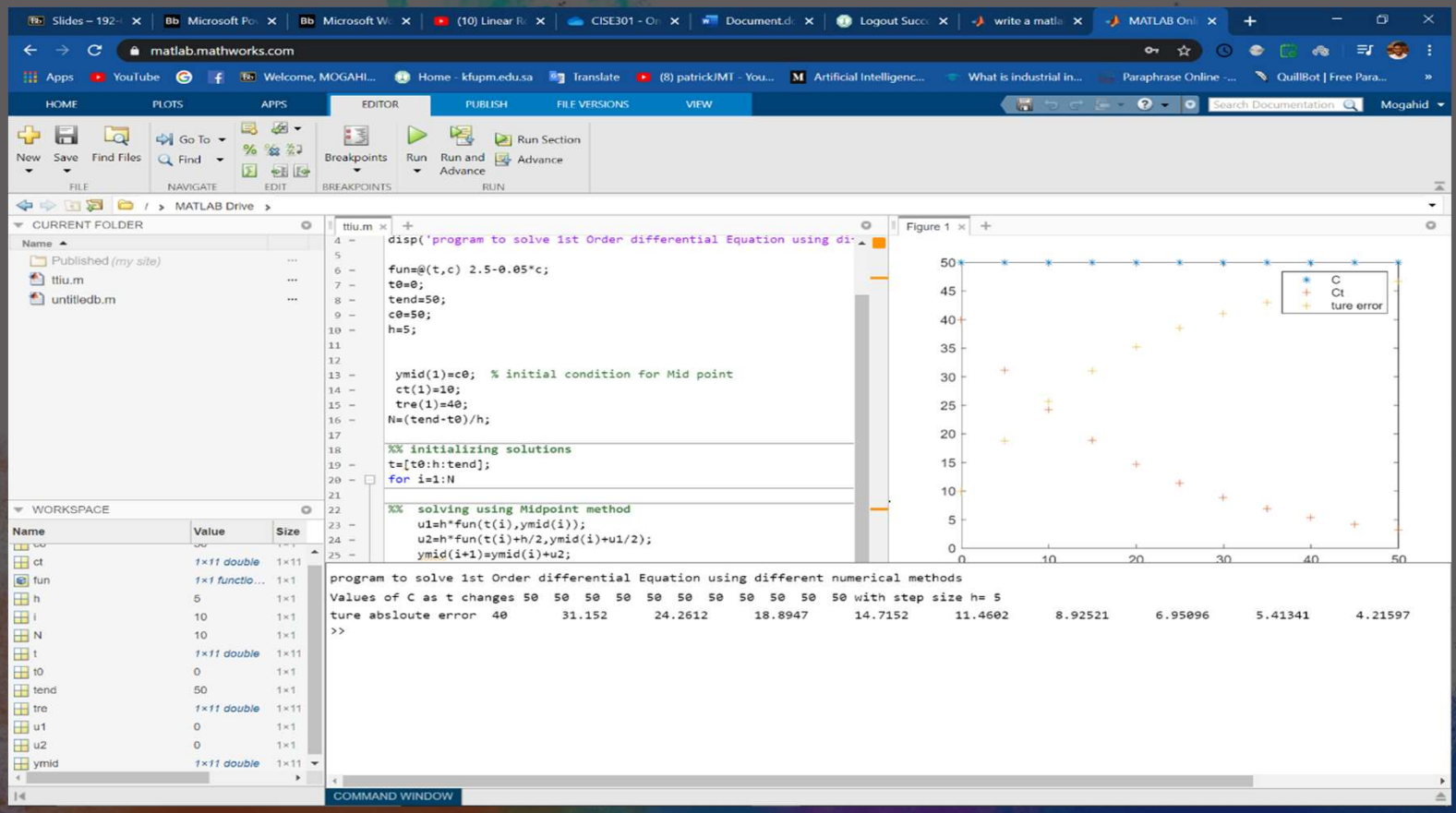
Pseudocode of a MATLAB function that calculates the 1st ODE :

```
clear all
clc
disp('program to solve 1st
Order differential Equation
using different numerical methods    ')


fun=@(t,c) 2.5-0.05*c;
t0=0;
tend=50;
c0=50;
h=5;



ymid(1)=c0;  % initial condition for

Mid point
 ct(1)=10;
 tre(1)=40;
N=(tend-t0)/h;

%% initializing solutions
t=[t0:h:tend];
```

```
for i=1:N

%%  solving using Midpoint method
    u1=h*fun(t(i),ymid(i));
    u2=h*fun(t(i)+h/2,ymid(i)+u1/2);
    ymid(i+1)=ymid(i)+u2;




ct(i+1)=50*(1-exp(-0.05*t(i+1)))+10*exp(-0.05*t(i+1));

    tre(i+1)=abs(ct(i+1)-ymid(i+1));
end



%% plot

    plot(t,ymid(:),'*',t,tre(:),'+',t,ct(:),'+');
   legend('C','Ct','ture error ')
disp(['Values of C as t changes ' num2str(ymid) ' with
step size h= ' num2str(h)])
disp(['ture absloute error  ' num2str(tre) ' with step
size h= ' num2str(h)])
```

03 The output samples

1st ODE output



THANK YOU

You are Welcome To Contact Me

 Mojahed-nour

 MJD_noor

 Mojahed nour

 Mojahednour@email.com