Table of Contents


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
% CODE CHALLENGE 5 - Template Script
%
% The purpose of this challenge is to predict whether or not the
 Boulder
% Reservior will have to close due to a major leak.
%
% To complete the challenge, execute the following steps:
% Part 1:
% 1) Read in the data file
% 2) Set values to any constants
% 3) Perform a trapazoid integration on the data w/r.t. x
% 4) Perform a simpson's 1/3 integration on the data w/r.t. x
% 5) Display which volume measurement is more accurate and why
%
% Part 2:
% 1) Define which delta t will be used in the Euler integration
% 2) Set values to any constants and initial conditions
% 3) Propagate h with t using Euler integration
% 4) Repeat steps 1-4 with different delta t values
% 5) Display which delta t gives a more accurate result and why.
%
%
% NOTE: DO NOT change any variable names already present in the code.
%
% Upload your team's script to Gradescope to complete the challenge.
%
% NAME YOUR FILE AS Challenge5_Sec{section number}_Group{group
 breakout #}.m
% ***Section numbers are 1 or 2***
% EX File Name: Challenge5_Sec1_Group15.m
%
%
% 1) Andrew Pankey
% 2) Cate Billings
% 3) Matthew Aycock
% 4) Andrew Logue
% 5) Sabastian Boysen
```

%%

Housekeeping

don't "clear variables", it makes things easier to grade

```
close all;    % Close all open figure windows
clc;          % Clear the command window
```

Part 1

Set up

```
data = table2array(readtable('depth_data.csv')); % read in .csv
x = data(:,1); % [ft]
d = data(:,2); % [ft]
L = 4836; % length of reservior [ft]
```

Trapazoid - Calculate Volume

```
delta_x = x(2);
n = 30;
Vol_trap = L*((delta_x/2)*(d(1)+d(n+1)) + delta_x*((sum(d)-d(1)))); %
 [ft^3]
fprintf('The Volume of the Resevoir using the Trapezoid method is %.2e
 ft^3', Vol_trap);
```

The Volume of the Resevoir using the Trapezoid method is 4.61e+08 ft^3

Simpson 1/3 - Calculate Volume

```
evens = 0;
y = 0;
for i = 2:((n/2)-1)
    y = d(2*i-1);
    evens = evens + y;

end
odds = 0;
p = 0;
for i = 1:(n/2)
    p = d(2*i);
    odds = odds +p;
end
V_simp = (delta_x / 3)*L*(d(n+1)+d(1)+2*evens + 4 * odds); % [ft^3]
fprintf('The Volume of the Resevoir using the Simpson method is %.2e
 ft^3', V_simp);
```

The Volume of the Resevoir using the Simpson method is 4.59e+08 ft^3

Part 2

Set up

```
del_t = 4; % various delta t values to test [days]
how_long = 20;


L = 4836;
h0 = 20; % initial depth
alpha = 1500000*h0; % relating volume out per day to depth [ft^2/day]
dV_in = 20000000; % volume in rate per day
t4 = 0:del_t:how_long; % allocate time vector [days]
h4 = zeros(size(t4)); % allocate depth vector [ft]
h4(1)= h0; % set initial value in h vector

for i = 1:(length(t4)-1) % Euler method
    dhdt = get_dhdt(h4(i),4836,1500000,20000000); % get dh/dt at this
 depth
    h4(i+1) = h4(i)+h4(i)*dhdt; %compute next depth value
end


del_t = 7; % various delta t values to test [days]
how_long = 20;


L = 4836;
h0 = 20; % initial depth
alpha = 1500000*h0; % relating volume out per day to depth [ft^2/day]
dV_in = 20000000; % volume in rate per day
t7 = 0:del_t:how_long; % allocate time vector [days]
h7 = zeros(size(t7)); % allocate depth vector [ft]
h7(1)= h0; % set initial value in h vector

for i = 1:(length(t7)-1) % Euler method
    dhdt = get_dhdt(h7(i),4836,1500000,20000000); % get dh/dt at this
 depth
    h7(i+1) = h7(i)+h7(i)*dhdt; %compute next depth value
end



del_t = 1; % various delta t values to test [days]
how_long = 20;


L = 4836;
h0 = 20; % initial depth
alpha = 1500000*h0; % relating volume out per day to depth [ft^2/day]
dV_in = 20000000; % volume in rate per day
t1 = 0:del_t:how_long; % allocate time vector [days]
```

```
h1 = zeros(size(t1)); % allocate depth vector [ft]
h1(1)= h0; % set initial value in h vector

for i = 1:(length(t1)-1) % Euler method
    dhdt = get_dhdt(h1(i),4836,1500000,20000000); % get dh/dt at this
 depth
    h1(i+1) = h1(i)+h1(i)*dhdt; %compute next depth value
end
```

```
del_t = .5; % various delta t values to test [days]
how_long = 20;


L = 4836;
h0 = 20; % initial depth
alpha = 1500000*h0; % relating volume out per day to depth [ft^2/day]
dV_in = 20000000; % volume in rate per day
t5 = 0:del_t:how_long; % allocate time vector [days]
h5 = zeros(size(t5)); % allocate depth vector [ft]
h5(1)= h0; % set initial value in h vector

for i = 1:(length(t5)-1) % Euler method
    dhdt = get_dhdt(h5(i),4836,1500000,20000000); % get dh/dt at this
 depth
    h5(i+1) = h5(i)+h5(i)*dhdt; %compute next depth value
end

% plot results
figure(1);
hold on
plot(t4,h4);
plot(t7,h7);
plot(t1,h1);
plot(t5,h5);
% labels for plot
xlabel('Time(days)')
ylabel('Depth (ft)')
title('Depth vs. Time')

legend('step size: 4 days','step size: 7 days','step size: 1
 days','step size: 0.5 days');
hold off


fptinf('The most accurate step size for estimating volume at any given
 time is 0.5 days.')
```

Undefined function 'fptinf' for input arguments of type 'char'.

```
Error in Coding_Challenge_5_Template (line 172)
fptinf('The most accurate step size for estimating volume at any given
 time is 0.5 days.')
```

Published with MATLAB® R2020a