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 number 2***
% EX File Name: Challenge5_Sec1_Group15.m
```

```
%
%
% 1) Zak Reichenbach zare8746@colorado.edu
% 2) Drew Miller anmi4817@colorado.edu
% 3) Kevin Cook keco2766@Colorado.edu
% 4) Nate Sanchez nasa1020@colorado.edu
% 5) Qihan Cai (didnt really talk) qica9589@colorado.edu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%
```

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

Vol_Trap_Check = trapz(x,d);
ResevoirVolumeCheck = Vol_Trap_Check * L;
```

Trapazoid - Calculate Volume

```
%area of one is (1/2)(*(y(i) +y(i+1))*DeltaX
%area of the whole is (DeltaX/2)(y(1) + y(n+1)) + DeltaX(sum(2-n)of
 y(i))
DeltaX = zeros(30,1);

N = length(d)-1;

for i =1:N
 DeltaX(i) = x(i+1)-x(i);

end
DeltaX = sum(DeltaX/length(DeltaX));

SETrap = DeltaX/2*(d(1)+d(end));
Area_Trap =  SETrap;
```

```
for i = 2:(N+1)

 Area_Trap = Area_Trap +(DeltaX * d(i)); %[ft^2]
end
ResevoirVolTrap = Area_Trap * L; %[ft^3]

%
```

Simpson 1/3 - Calculate Volume

```
%area whole is (DeltaX/3)*(y(1)+y(n+1) + 2*sum(i=1-((n/2)-1))
(y(2*i-1)) + 4*sum(i=1-(n/2))y(2i)).

SESimp = (d(1)+d(end));
Area_Simp = SESimp;

for i=2:((N/2))
    vol = 2*(d(2*i-1));

    Area_Simp = Area_Simp + vol;
end

for i=1:(N/2)
    vol2 = 4*(d(2*i));

    Area_Simp = Area_Simp + vol2;
end
Area_Simp = (DeltaX/3)*(Area_Simp); % [ft^2]
ResevoirVolSimp = Area_Simp * L;

%Simpsons estimate will be more accurate, it is more fit to the actual
%data, and will therefore include more true area under the data curve,
%giving us a more accurate estimate.
```

Part 2

Set up

```
%Delta T is 7days, 4 days, 1 day, and 0.5 of a day. Which estimate is
 the
%most accurate?

 del_t = [7,4,1,0.5]; % various delta t values to test [days]
%
%
h0 = 20; % [ft] initial depth

alpha = 1.5*10^6; %[ft^2/day] relating volume out per day to depth
 [ft^2/day]

dV_in = 2*10^7; %[ft^3/day] volume in rate per day
```

Creating vectors

```
t7 =[1 7:del_t(1):28]; % allocate time vector [days]
t4 =[1 4:del_t(2):28];
t1 = [1:del_t(3):28];
tHalf = [1:del_t(4):28];



h7 = zeros(length(t7),1);  % allocate depth vector [ft]
h4 = zeros(length(t4),1);
h1 = zeros(length(t1),1);
hHalf = zeros(55,1);

h7(1)= h0; % set initial value in h vector [ft]
h4(1)= h0;
h1(1)= h0;
hHalf(1)= h0;

%DONT CHANGE
```

Week

```
for i = 1:(length(t7)-1) % Euler method
    dhdt = get_dhdt(h7(i),L,alpha,dV_in); % get dh/dt at this depth
    h7(i+1) = h7(i)+dhdt*del_t(1); %compute next depth value
end
```

4 Days

```
for i = 1:(length(t4)-1) % Euler method
    dhdt = get_dhdt(h4(i),L,alpha,dV_in); % get dh/dt at this depth
    h4(i+1) = h4(i)+dhdt*del_t(2); %compute next depth value
end
```

1 Day

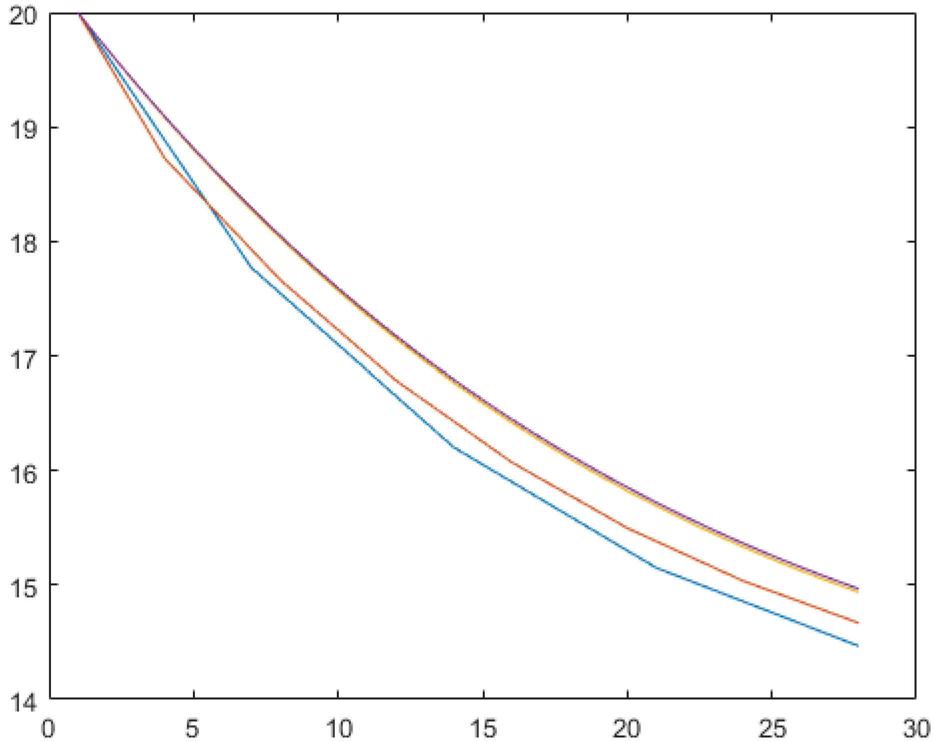
```
for i = 1:(length(t1)-1) % Euler method
    dhdt = get_dhdt(h1(i),L,alpha,dV_in); % get dh/dt at this depth
    h1(i+1) = h1(i)+dhdt*del_t(3); %compute next depth value
end
```

Half Day

```
for i = 1:(length(tHalf)-1) % Euler method
    dhdt = get_dhdt(hHalf(i),L,alpha,dV_in); % get dh/dt at this depth
    hHalf(i+1) = hHalf(i)+dhdt*del_t(4); %compute next depth value
end
%
```

plot results

```
 figure(1) % create figure
plot(t7,h7)
hold on
plot(t4,h4)
plot(t1,h1)
plot(tHalf,hHalf)
hold off
```



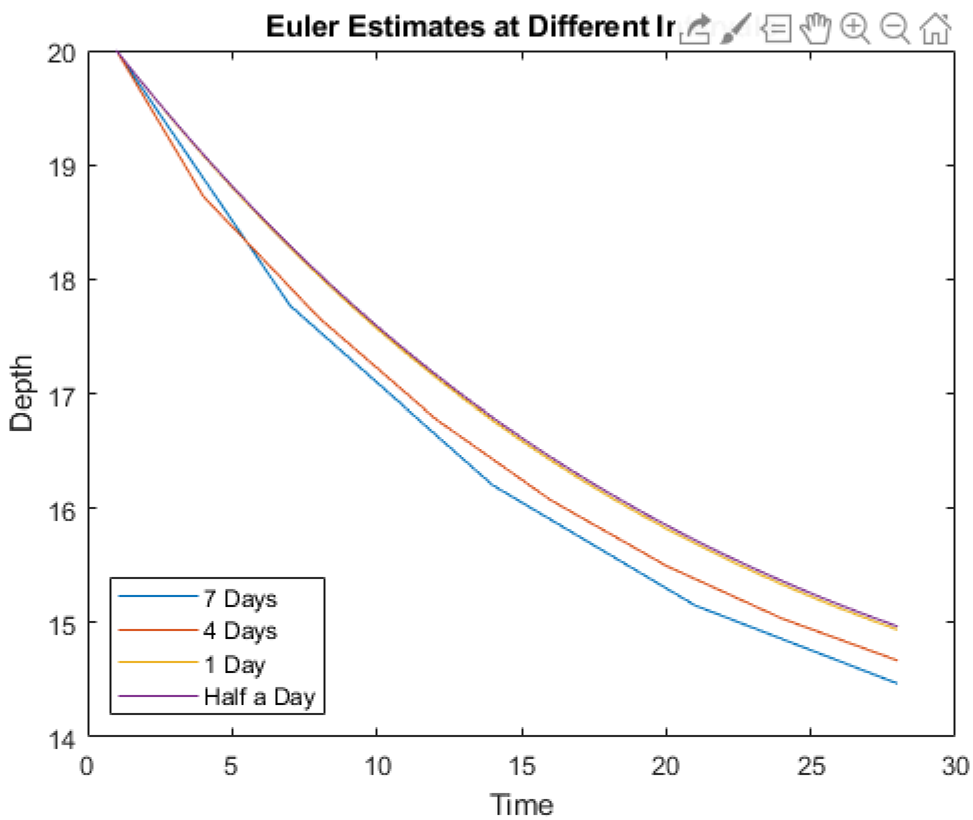
labels for plot

```
title('Euler Estimates at Different Intervals')
xlabel('Time')
ylabel('Depth')
legend('7 Days', '4 Days', '1 Day', 'Half a
 Day','Location','southwest');

% We can clearly see that the most accurate model is the one that
 takes the
% most step, which is the half day interval estimate. This would make
 clear
% sense, as more steps give us more analysis of the over all trend,
% creating a much clearer picture.
%
```

```
%


% Function is put here just to show that we did it.
% function [dhdt] = get_dhdt(h,L,alpha,dV_in)
%
% dV_out = alpha*h; % calculate dV_out
% dVdt = dV_in-dV_out; % calculate net dV/dt
% [~,dVdh] = get_Volume(h,L); % get current dV/dh
% dhdt = dVdt/dVdh; % convert dV/dt to dh/dt
% end
```



Published with MATLAB® R2019a