Table of Contents

EMEC 303 HW4


```
Lance Nichols
Section-002
9/11/2020
```


```
clear; clc;
```

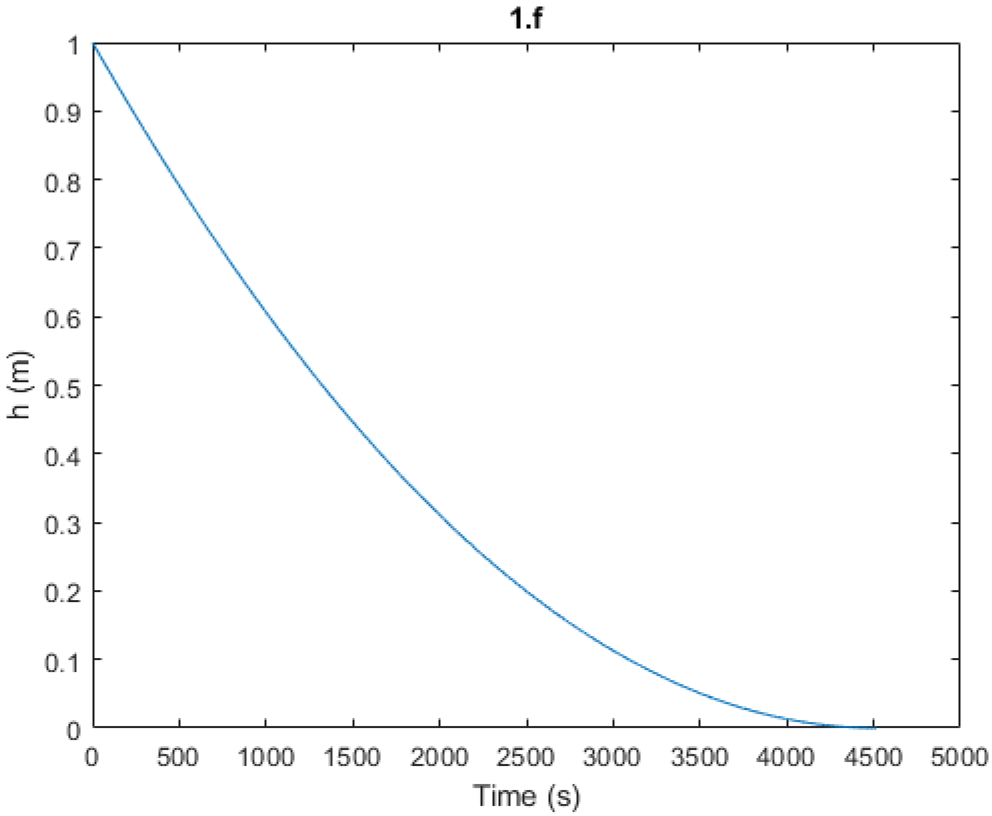
Problem 1: Tank Draining

- (a) This equation is an ODE becauseit only has one dependant varable
- (b) First order, the power of the derivitive is 1
- (c) t
- (d) h
- (e) Initialy all you really need is the hight and it cant go below zero
- (f) See the graph below
- (g) Exponential decay, This makes sense as less fluid is in the container it pushes the other fluid out slower and slower
- (h) 75.2833 minutes
- (i) 0.75 minutes this is a hundreth of the last one witch makes sense as the d_o was reduced by a tenth but its squared

```
% Define Fuction
syms y d_out
dhdt = -(d_out^2/1^2)*sqrt(2*9.8*y);
% Make both matlabFunctions
dhdt = matlabFunction(dhdt);

%Initial Variables
h = 1;
t = 0;
y = 1;
count = 1;
D_O = 0.01;
```

```
while true
    %Save current value
    if isreal(y)
        ylst(count) = y;
    else
        break
    end

    %Advance
    t = t + h;
    count = count + 1;
    %Calculate new y value
    y_star = y + h*dhdt(D_O,y);
    y = y + h*(dhdt(D_O,y)+dhdt(D_O,y_star))/2;

end
disp("1.h: " + num2str(t/60)+ " minutes")

plot(0:h:t-h,ylst);
title("1.f");
xlabel('Time (s)')
ylabel('h (m)')

%Initial Variables
h = 1;
t = 0;
y = 1;
count = 1;
D_O = 0.1;

while true
    %Save current value
    if isreal(y)
        ylst(count) = y;
    else
        break
    end

    %Advance
    t = t + h;
    count = count + 1;
    %Calculate new y value
    y_star = y + h*dhdt(D_O,y);
    y = y + h*(dhdt(D_O,y)+dhdt(D_O,y_star))/2;

end
disp("1.i: " + num2str(t/60)+ " minutes")
```

1.h: 75.2833 minutes
1.i: 0.75 minutes



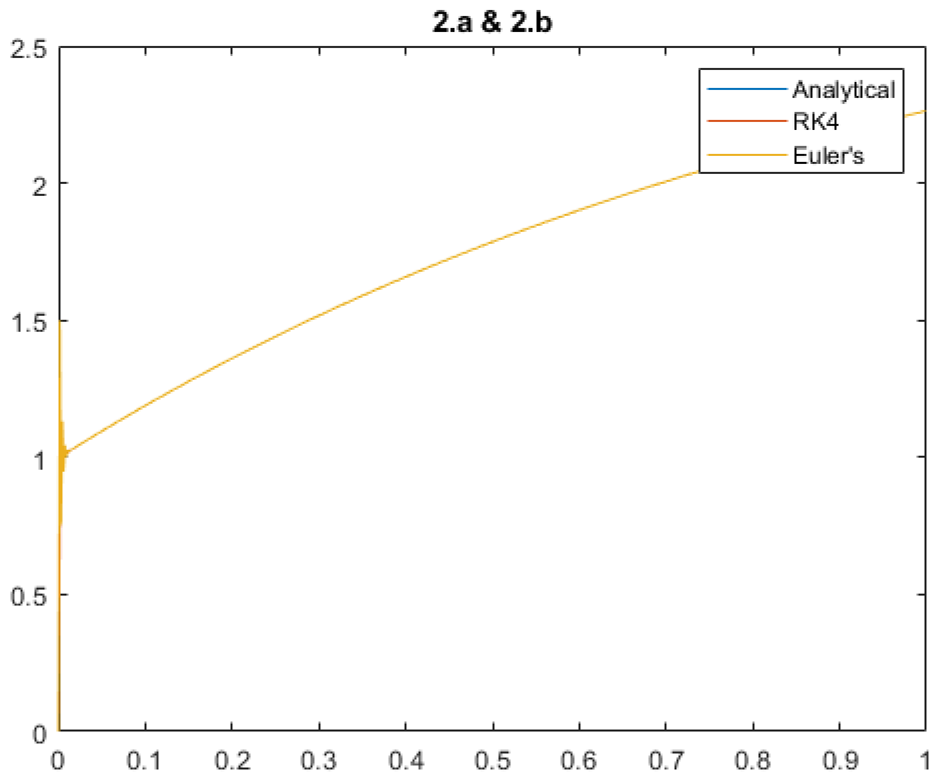
Problem 2: 4th Order Runge-Kutta Method

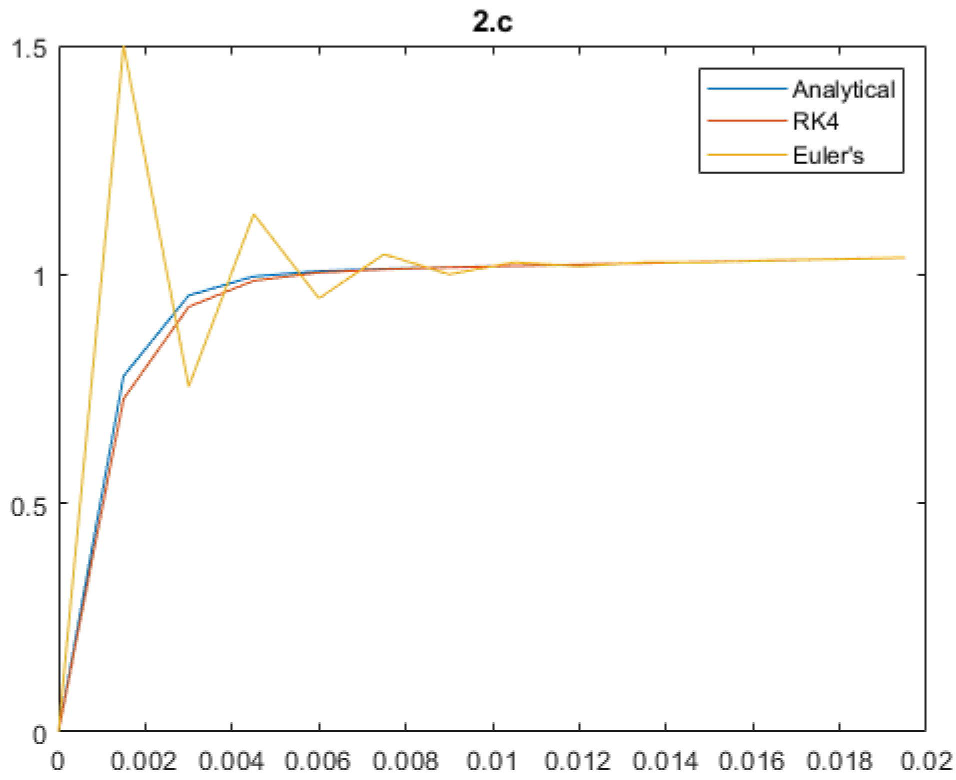
- (a) They are perfectly matched
- (b) Euler's method is qute "noisy" in the short term but works in the long run
- (c) Eluer's method really struggles at this time frame but improves with smaller step size but overall shows its ineffecenies (see figure 3 "2.c")

```
% Define Fuction
syms x y
dydx = -2000*exp(-x)-1000*y+3000;
% Make both matlabFunctions
dydx = matlabFunction(dydx);

h = 0.0015;
x = 0;
y = 0;
ye = 0;
xend = 1;

count = 1;

t = x:h:xend;
ylst = zeros(1,length(t));
yeul = zeros(1,length(t));
```

```
alx = 3-0.998*exp(-1000*t)-2.002*exp(-t);

for i = t
    ylst(count) = y;
    yeul(count) = ye;
    %Runge-Kutta
    count = count + 1;
    k1 = dydx(i,y);
    k2 = dydx(i+0.5*h,y+0.5*k1*h);
    k3 = dydx(i+0.5*h,y+0.5*k2*h);
    k4 = dydx(i+h,y+k3*h);
    y = y + (k1 + k2*2 + k3*2 + k4)*h/6;
    %Euler
    ye = ye + dydx(i,ye)*h;
end


figure(2);
plot(t,alx);
hold on
plot(t,ylst);
plot(t,yeul);
hold off
title("2.a & 2.b");
legend("Analytical","RK4","Euler's");

% 2.c

h = 0.0015;
x = 0;
y = 0;
ye = 0;
xend = .02;

count = 1;

t = x:h:xend;
ylst = zeros(1,length(t));
yeul = zeros(1,length(t));
alx = 3-0.998*exp(-1000*t)-2.002*exp(-t);

for i = t
    ylst(count) = y;
    yeul(count) = ye;
    %Runge-Kutta
    count = count + 1;
    k1 = dydx(i,y);
    k2 = dydx(i+0.5*h,y+0.5*k1*h);
    k3 = dydx(i+0.5*h,y+0.5*k2*h);
    k4 = dydx(i+h,y+k3*h);
    y = y + (k1 + k2*2 + k3*2 + k4)*h/6;
    %Euler
    ye = ye + dydx(i,ye)*h;
end
```

```
figure(3);
plot(t,alx);
hold on
plot(t,ylst);
plot(t,yeul);
hold off
title("2.c");
legend("Analytical","RK4","Euler's");
```





Published with MATLAB® R2020a