Csc 349a assignment #1

Parm Johal
V00787710

Question 1:

a)

```
Euler.m
function [v] = Euler(m,c,g,t0,v0,tn,n)
% print headings and initial conditions

%fprintf('values of t approximations v(t)\n')
%fprintf('%8.3f',t0),fprintf('%19.4f\n',v0)

% compute step size h
h=(tn-t0)/n;
% set t,v to the initial values
t=t0;
v=v0;
% compute v(t) over n time steps using Euler's method

fprintf('times used in the calculations:\n')
for i=1:n
t=[t+h];
fprintf('%8.3f',t)
end

fprintf('\n')

t=t0;
v=v0;
fprintf('velocities used in the calculations:\n')
for i=1:n
v=[v+(g-c/m*v)*h];
fprintf('%9.4f',v)
end
```

b)

Function call to Euler:

Euler(73.5,13.1,3.71,0,0,16,80)

times used in the calculations:

0.000 0.200 0.400 0.600 0.800 1.000 1.200 1.400 1.600 1.800 2.000 2.200 2.400
2.600 2.800 3.000 3.200 3.400 3.600 3.800 4.000 4.200 4.400 4.600 4.800 5.000
5.200 5.400 5.600 5.800 6.000 6.200 6.400 6.600 6.800 7.000 7.200 7.400 7.600
7.800 8.000 8.200 8.400 8.600 8.800 9.000 9.200 9.400 9.600 9.800 10.000 10.200
10.400 10.600 10.800 11.000 11.200 11.400 11.600 11.800 12.000 12.200 12.400 12.600

12.800 13.000 13.200 13.400 13.600 13.800 14.000 14.200 14.400 14.600 14.800 15.000
15.200 15.400 15.600 15.800 16.000

velocities used in the calculations:

0.0000 0.7420 1.4576 2.1476 2.8130 3.4548 4.0736 4.6704 5.2459 5.8009 6.3361
6.8523 7.3500 7.8300 8.2929 8.7393 9.1698 9.5849 9.9852 10.3713 10.7436 11.1026
11.4489 11.7828 12.1048 12.4153 12.7147 13.0035 13.2819 13.5505 13.8095 14.0592
14.3001 14.5323 14.7563 14.9723 15.1806 15.3814 15.5752 15.7620 15.9421 16.1158
16.2834 16.4449 16.6007 16.7510 16.8959 17.0356 17.1703 17.3003 17.4256 17.5464
17.6630 17.7753 17.8837 17.9882 18.0890 18.1862 18.2799 18.3703 18.4575 18.5416
18.6226 18.7008 18.7762 18.8489 18.9190 18.9866 19.0518 19.1147 19.1753 19.2338
19.2902 19.3445 19.3970 19.4475 19.4963 19.5433 19.5887 19.6324 19.6746

ans =

    19.6746

c)

```
function [v] = Euler_Analytical(m,c,g,t0,v0,tn,n)
%Solves the Euler differential equation analytically
```

```
% compute step size h
h=(tn-t0)/n;
% set t,v to the initial values
t=t0;
v=v0;
% compute v(t) over n time steps using Euler's method

fprintf('times used in the calculations:\n')
fprintf('%8.3f',t)
for i=1:n
t=[t+h];
fprintf('%8.3f',t)
end

fprintf('\n')

t=t0;
v=v0;
fprintf('velocities used in the calculations:\n')
for i=1:n
v=g*m/c*(1-exp(-c*t/m));
fprintf('%9.4f',v)
t=[t+h];
end
```

Euler_Analytical(73.5,13.1,3.71,0,0,16,80)

times used in the calculations:

0.000 0.200 0.400 0.600 0.800 1.000 1.200 1.400 1.600 1.800 2.000 2.200 2.400
2.600 2.800 3.000 3.200 3.400 3.600 3.800 4.000 4.200 4.400 4.600 4.800 5.000
5.200 5.400 5.600 5.800 6.000 6.200 6.400 6.600 6.800 7.000 7.200 7.400 7.600
7.800 8.000 8.200 8.400 8.600 8.800 9.000 9.200 9.400 9.600 9.800 10.000 10.200
10.400 10.600 10.800 11.000 11.200 11.400 11.600 11.800 12.000 12.200 12.400 12.600
12.800 13.000 13.200 13.400 13.600 13.800 14.000 14.200 14.400 14.600 14.800 15.000
15.200 15.400 15.600 15.800 16.000

velocities used in the calculations:

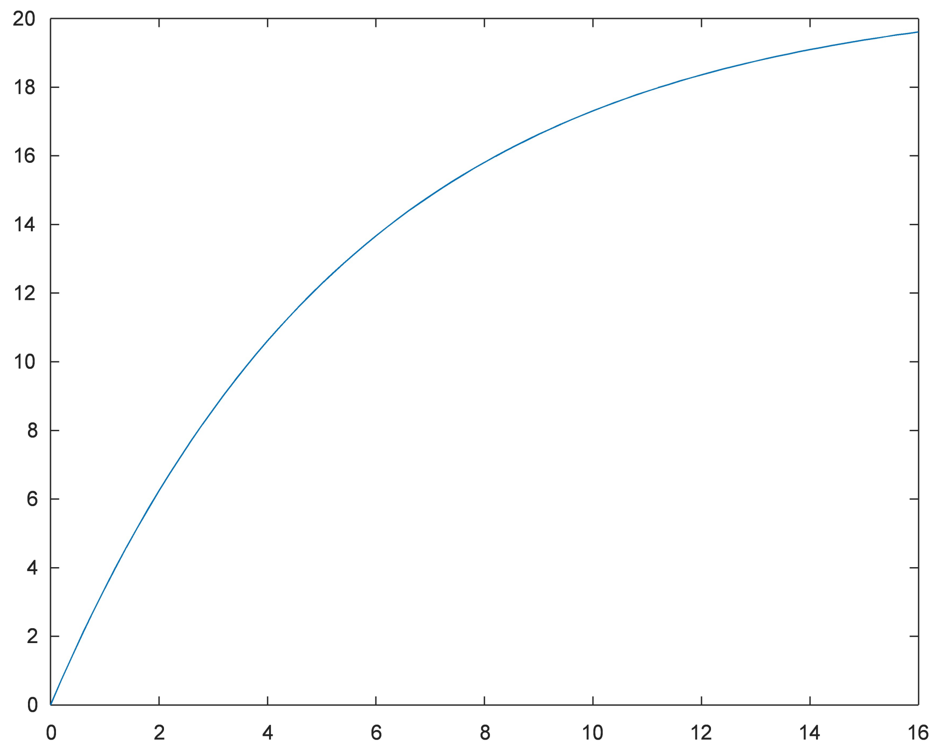
0.0000 0.7289 1.4323 2.1111 2.7661 3.3982 4.0081 4.5967 5.1646 5.7127 6.2416
6.7520 7.2444 7.7197 8.1783 8.6208 9.0479 9.4600 9.8576 10.2414 10.6116 10.9690
11.3138 11.6465 11.9676 12.2775 12.5765 12.8650 13.1434 13.4121 13.6713 13.9215
14.1629 14.3959 14.6207 14.8376 15.0470 15.2490 15.4439 15.6320 15.8136 15.9887

16.1578 16.3209 16.4783 16.6302 16.7767 16.9182 17.0546 17.1864 17.3134 17.4361
17.5544 17.6686 17.7788 17.8852 17.9878 18.0868 18.1824 18.2746 18.3636 18.4495
18.5323 18.6123 18.6894 18.7639 18.8357 18.9051 18.9720 19.0365 19.0988 19.1590
19.2170 19.2730 19.3270 19.3791 19.4294 19.4780 19.5248 19.5700

ans =

   19.5700

d)




```
>> x=0:0.2:16;
>> vt=g*m/c*(1-exp(-c*t/m));
>> plot(x,vt)
```

e)

As t approaches infinity for $v(t) = g*m/c * (1-e^{-c*t/m})$ the exponent of e becomes more negative, resulting in the term dissipating. With 1 left in the bracket the terminal velocity becomes g*m/c which is equal to 20.81564885 ≃ 20.82.

Question 2:

a)

```
function [T] = Euler2(k,Ta,t0,T0,tn,n)

%fprintf('values of t approximations v(t)\n')
%fprintf('%8.3f',t0),fprintf('%19.4f\n',v0)

% compute step size h
h=(tn-t0)/n;
% set t,v to the initial values
t=t0;
T=T0;
% compute v(t) over n time steps using Euler's method

fprintf('times used in the calculations:\n')
for i=1:n
t=[t+h];
fprintf('%8.3f',t)
end

fprintf('\n')

t=t0;
dT=T0;
fprintf('Temperatures used in the calculations:\n')
for i=1:n
dT=[dT+(-k*(T-Ta))*h];
fprintf('%9.4f',dT)
end

end
```

b)

Euler2(0.019,20,0,68,12,96)

times used in the calculations:

0.125 0.250 0.375 0.500 0.625 0.750 0.875 1.000 1.125 1.250 1.375 1.500 1.625
1.750 1.875 2.000 2.125 2.250 2.375 2.500 2.625 2.750 2.875 3.000 3.125 3.250
3.375 3.500 3.625 3.750 3.875 4.000 4.125 4.250 4.375 4.500 4.625 4.750 4.875
5.000 5.125 5.250 5.375 5.500 5.625 5.750 5.875 6.000 6.125 6.250 6.375 6.500

6.625 6.750 6.875 7.000 7.125 7.250 7.375 7.500 7.625 7.750 7.875 8.000 8.125 8.250 8.375 8.500 8.625 8.750 8.875 9.000 9.125 9.250 9.375 9.500 9.625 9.750 9.875 10.000 10.125 10.250 10.375 10.500 10.625 10.750 10.875 11.000 11.125 11.250 11.375 11.500 11.625 11.750 11.875 12.000

Temperatures used in the calculations:

67.8860 67.7720 67.6580 67.5440 67.4300 67.3160 67.2020 67.0880 66.9740 66.8600 66.7460 66.6320 66.5180 66.4040 66.2900 66.1760 66.0620 65.9480 65.8340 65.7200 65.6060 65.4920 65.3780 65.2640 65.1500 65.0360 64.9220 64.8080 64.6940 64.5800 64.4660 64.3520 64.2380 64.1240 64.0100 63.8960 63.7820 63.6680 63.5540 63.4400 63.3260 63.2120 63.0980 62.9840 62.8700 62.7560 62.6420 62.5280 62.4140 62.3000 62.1860 62.0720 61.9580 61.8440 61.7300 61.6160 61.5020 61.3880 61.2740 61.1600 61.0460 60.9320 60.8180 60.7040 60.5900 60.4760 60.3620 60.2480 60.1340 60.0200 59.9060 59.7920 59.6780 59.5640 59.4500 59.3360 59.2220 59.1080 58.9940 58.8800 58.7660 58.6520 58.5380 58.4240 58.3100 58.1960 58.0820 57.9680 57.8540 57.7400 57.6260 57.5120 57.3980 57.2840 57.1700 57.0560

c)

Using the exact analytical solution to compute the temperature at $t = 12$ we get a value of $58.21396447 \simeq 58.2140$. After computing the relative error between this value and the value of 57.0560 from the numerical solution we get a relative error value of $\simeq 0.01989 = 1.989 \times 10^{-2} < 5 \times 10^{-2}$. This tells us that the numerical value is accurate to within 2 significant digits of the true analytical value.

Parm Johal
V00787710

Assignment #1

#3

(A) $e^{-x} \approx 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots \pm \frac{x^n}{n!}$

(B) $e^{-x} \approx \frac{1}{1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}}$

$$\boxed{|\varepsilon_t| = \frac{|e^{-2.5} - p^*|}{|e^{-2.5}|}}$$

approximate $e^{-2.5}$, where true value $p = 0.082084998\dots$

$\boxed{n=1}$

(A) $e^{-2.5} \approx 1 - (2.5) = -1.5 = p^* \Rightarrow |\varepsilon_t| = \frac{|e^{-2.5} - p^*|}{|e^{-2.5}|} = 19.27 \times 10^0 \not< 5 \times 10^0$

(B) $e^{-2.5} \approx \frac{1}{1 + (2.5)} = 0.2857 = p^* \Rightarrow |\varepsilon_t| = 2.481 \times 10^0 \not< 5 \times 10^0$

$\boxed{n=2}$

(A) $e^{-2.5} \approx 1 - (2.5) + \frac{(2.5)^2}{2} = 1.625 = p^* \Rightarrow |\varepsilon_t| = 18.80 \times 10^0 \not< 5 \times 10^0$

(B) $e^{-2.5} \approx \frac{1}{1 + (2.5) + \frac{(2.5)^2}{2}} = 0.1509 = p^* \Rightarrow |\varepsilon_t| = 0.838 \times 10^0 \not< 5 \times 10^{-1}$

$\boxed{n=3}$

(A) $e^{-2.5} \approx -0.97917 = p^* \Rightarrow |\varepsilon_t| = 12.93 \times 10^0 \not< 5 \times 10^0$

(B) $e^{-2.5} \approx 0.108352144 = p^* \Rightarrow |\varepsilon_t| = 3.200 \times 10^{-1} < 5 \times 10^{-1}$

$\boxed{n=4}$

(A) $e^{-2.5} \approx 0.6484375 = p^* \Rightarrow |\varepsilon_t| = 6.900 \times 10^0 \not< 5 \times 10^0$

(B) $e^{-2.5} \approx 0.092108419 = p^* \Rightarrow |\varepsilon_t| = 1.22 \times 10^{-1} < 5 \times 10^{-1}$

$\boxed{n=5}$

(A) $e^{-2.5} \approx -0.165364583 = p^* \Rightarrow |\varepsilon_t| = 3.0146 \times 10^0 < 5 \times 10^0$

(B) $e^{-2.5} \approx 0.085685596 = p^* \Rightarrow |\varepsilon_t| = 4.386 \times 10^{-2} < 5 \times 10^{-2}$

Assignment #1

Parm Johal
V00787710

#3 (continued)

$\boxed{n=6}$

(A) $e^{-2.5} \approx 0.173719618 = p^* \Rightarrow |\epsilon_t| = 1.1163 \times 10^{0} < 5 \times 10^{0}$

(B) $e^{-2.5} \approx 0.083266323 = p^* \Rightarrow |\epsilon_t| = 1.4391 \times 10^{-2} < 5 \times 10^{-2}$

$\boxed{n=7}$

(A) $e^{-2.5} \approx 0.05261817 = p^* \Rightarrow |\epsilon_t| = 3.5898 \times 10^{-1} < 5 \times 10^{-1}$

(B) $e^{-2.5} \approx 0.082435075 = p^* \Rightarrow |\epsilon_t| = 4.2648 \times 10^{-3} < 5 \times 10^{-3}$

In conclusion, the Mclaurin series expansion of equation (B) is a much better approximation as n increases than equation (A) to the true Value of $e^{-2.5}$.