EGR 103L – Fall 2017

Structured Programming II

Ian Hanus (ih52)
Lab Section 1B, Tuesday, 8:30-11:20 AM
15 October 2017

I understand and have adhered to all the tenets of the Duke Community Standard in completing every part of this assignment. I understand that a violation of any part of the Standard on any part of this assignment can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 Chapra Problem 3.5

Interesting differences in the graphs include the change in relative error and the shape of graphs where the actual values are identical. For example, the sin of 0.7854 and the sin of 7.0686 should represent the same values, but the graphs do not look similar at all. The graph of 7.0686 actually begins with an overestimation as the relative error is positive, whereas the relative error of the approximation for 0.7854 stays negtive across all terms.The actual terms themselves are different too, as the n-term approximation is constantly positive for sin of 0.7854 but sometimes positive and sometimes negative for sin of 7.0686. There are similar discrepancies between the other two graphed values, even though they are actually the same value.

2 Chapra Problem 3.10

The maximum value of displacement of the beam was 1559.9 at a distance of approximately 7.00000 feet from the end of the beam. The minimum value of displacement of the beam was -5265.1 at approximately 7.00001 feet from the end of the beam.

3 Chapra Problem 3.14

This problem relates most closely to the "engineer the tools for scientific discovery" engineering Grand Challenge. This anonymous function allowed us to calculate velocity of a rocket. By calculating the velocity of the rocket, one could determine the path of the rocket and guide it through an exploration of space in pursuit of scientific discovery.

4 Palm Problem 4.44

The a and b values for six gases are

Gas	a (L^2−atm/mol^2)	b (L/mol)
Carbon dioxide, CO_2	3.59	0.0427
Chlorine, Cl_2	6.49	0.0562
Helium, He	0.0341	0.0237
Hydrogen, H_2	0.244	0.0266
Neon, Ne	0.2135	0.01709
Oxygen, O_2	1.36	0.0318

from references [2] and [3]:

5 Data Logger

The diary, data file, and code are in the appropriate appendices.

A Codes

A.1 SinSeries.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
%% Setup function
function [SinApprox, RelError] = SinSeries(x,N)
C = [N];
D = [N];
%% Run loop on funtion
for a = 1:N
    if a > 1
        C(a) = (-1)^(a+1).*x^((2.*a)-1)./(factorial(2*a-1)) + C(a-1);
            D(a) = (sin(x)-C(a))/sin(x);
    else
        C(a) = (-1)^(a+1).*x^((2.*a)-1)./(factorial(2*a-1));
            D(a) = (sin(x)-C(a))/sin(x);
    end
    SinApprox = C;
    RelError = D;
end
```

A.2 SingularityPlot.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
clear; format short e
Singularity = @(x,a,n)...
    (x > a).* (x-a^n) +...
    (x<=0).*0;

x = linspace(0,10,100);
u = -5/6.*(Singularity(x,0,4)-Singularity(x,5,4))+5\2.*Singularity(x,8,3)+325/2.*Singularity(x,7,2)+79/1
plot(x,u,'k--');
title('Displacement vs. Distance Along a Beam (ih52)')
xlabel('Distance(ft)')
ylabel('Displacement')
x = linspace(0,10,10.^6);
u = -5/6.*(Singularity(x,0,4)-Singularity(x,5,4))+5\2.*Singularity(x,8,3)+325/2.*Singularity(x,7,2)+79/1
MaxU = max(u)
MinU = min(u)
LocMaxU = find(MaxU==u)
LocMinU = find(MinU==u)
print -depsc BeamDisplacement
```

A.3 Chapra314.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
clear; format short e

v = @(t) ...
    (t>= 0 & t<8).*(10.*t.^2-5.*t)+...
    (t>=8 & t<16).*(624-3.*t)+...
    (t>=16 & t<=26).*(36.*t+12.*(t-16).^2)+...
    (t>26).*(2136.*exp(-0.1.*(t-26)));

t = linspace(-5,50,1000);
figure(1);clf
plot(t,v(t),'k-')
title('Velocity of a Rocket vs Time (ih52)')
xlabel('Time, s')
ylabel('Velocity, m/s')
grid on
print -depsc Chapra314
```

A.4 VanDerWaals.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
function Pressure = VanDerWaals(Temp,Vol,Gas)

switch Gas
    case {'Helium','He'}
        a = 0.0341;
        b = 0.02237;
    case {'Hydrogen','H2'}
        a = 0.244;
        b = 0.0266;
    case {'Oxygen','O2'}
        a = 1.36;
        b = 0.0318;
    case {'Chlorine','Cl2'}
        a = 6.49;
        b = 0.0562;
    case {'Carbon dioxide','CO2'}
        a = 3.59;
        b = 0.0427;
    case {'Neon','N'}
        a = 0.2135;
        b = 0.01709;
    otherwise
        error('Gas unknown')
end

Pressure = 0.08206.*Temp./(Vol - b) - a./(Vol).^2;
```

A.5 GraphPressures.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
Vol = linspace(1,2,200);
Temp = 300;
Indices = linspace(1,2,15);
plot(Vol,VanDerWaals(Temp,Vol,'Helium'),'r-')
hold on
plot(Vol,VanDerWaals(Temp,Vol,'Chlorine'),'m-')
plot(Vol,VanDerWaals(Temp,Vol,'Neon'),'c-')
plot(Vol,VanDerWaals(Temp,Vol,'Carbon dioxide'),'k-')
plot(Vol,VanDerWaals(Temp,Vol,'Oxygen'),'b-')
plot(Vol,VanDerWaals(Temp,Vol,'Hydrogen'),'y-')
legend('Chlorine','Helium','Neon','Carbon dioxide','Oxygen','Hydrogen')
PointPlot = plot(Indices,VanDerWaals(Temp,Indices,'Chlorine'),'ks',...
    Indices, VanDerWaals(Temp,Indices,'Helium'),'ro',...
    Indices, VanDerWaals(Temp,Indices,'Neon'),'cp',...
    Indices, VanDerWaals(Temp,Indices,'Carbon dioxide'),'k*',...
    Indices,VanDerWaals(Temp,Indices,'Oxygen'),'bd',...
    Indices,VanDerWaals(Temp,Indices,'Hydrogen'),'y+');
title('Pressures at Differing Volumes')
ylabel('Pressure, atm')
xlabel('Volume, L/mol')
hold off
print -depsc GraphPressures
```

A.6 DataLogger.m

```
 % I have adhered to all the tenets of the
 % Duke Community Standard in creating this code.
 % Signed: [ih52]
%% Initialize workspace
clear; format short e

%% Take user input & setup matrix
TempIn = input('Enter a temperature: ');
counter=1;
    TempMatrix = [counter];
%% Collect and output data until input is negative
while TempIn >= 0;
    TempMatrix(counter) = TempIn;
    counter = counter + 1;
    fprintf('Readings Minimum Average Maximum\n%8.0f  %3.2f  %3.2f  %3.2f\n',...
        (counter-1), min(TempMatrix), mean(TempMatrix), max(TempMatrix));
    TempIn = input('Enter a temperature: ');
end
save MyTemps.txt TempMatrix -ascii
```

B Diary and Data Sets

B.1 MytTemps.txt

```
2.7167269e+02   2.9187220e+02   2.9142828e+02   2.6702781e+02   3.3661465e+02   3.1206778e+02
```

B.2 TempDiary.txt

```
ih52
out =
   2.7167e+02
Readings Minimum Average Maximum
       1  271.67  271.67  271.67
out =
   2.9187e+02
Readings Minimum Average Maximum
       2  271.67  281.77  291.87
out =
   2.9143e+02
Readings Minimum Average Maximum
       3  271.67  284.99  291.87
out =
   2.6703e+02
Readings Minimum Average Maximum
       4  267.03  280.50  291.87
out =
   3.3661e+02
Readings Minimum Average Maximum
       5  267.03  291.72  336.61
out =
   3.1207e+02
Readings Minimum Average Maximum
       6  267.03  295.11  336.61
```

C Figures

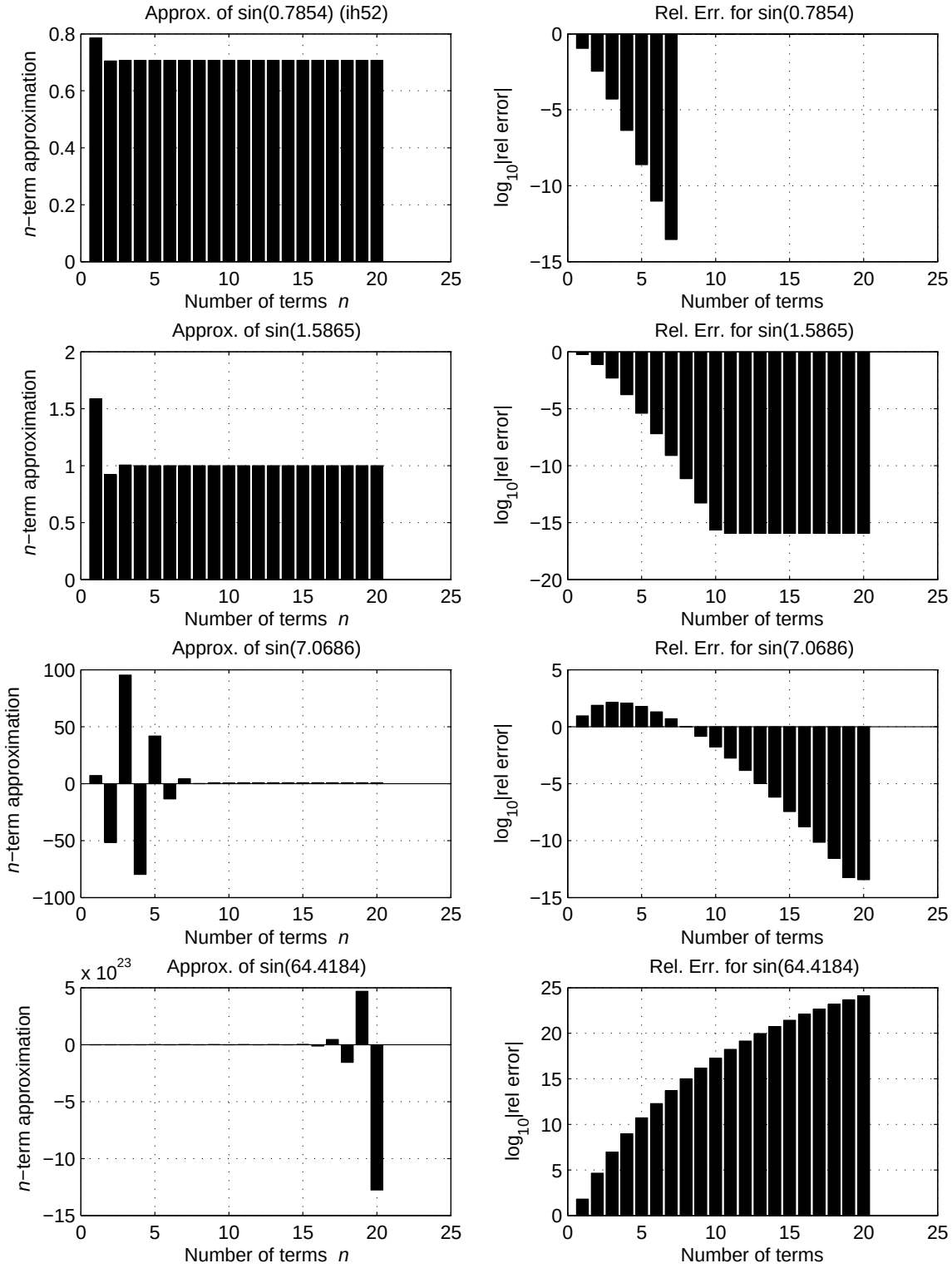

Figure 1: Output of `SinSeriesChecker.m` for Chapra 3.5

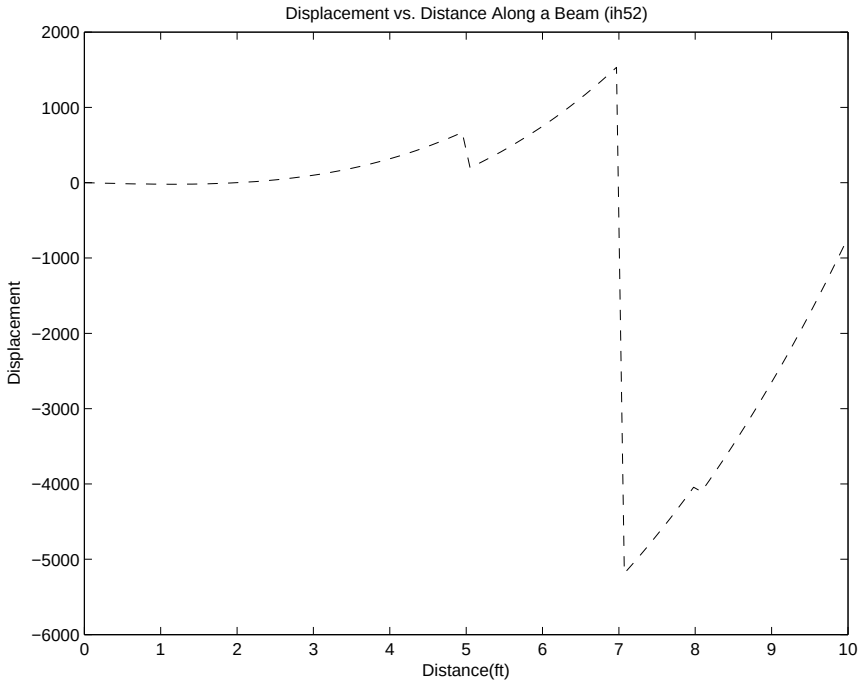

Figure 2: Beam Displacement versus Distance for Chapra 3.10

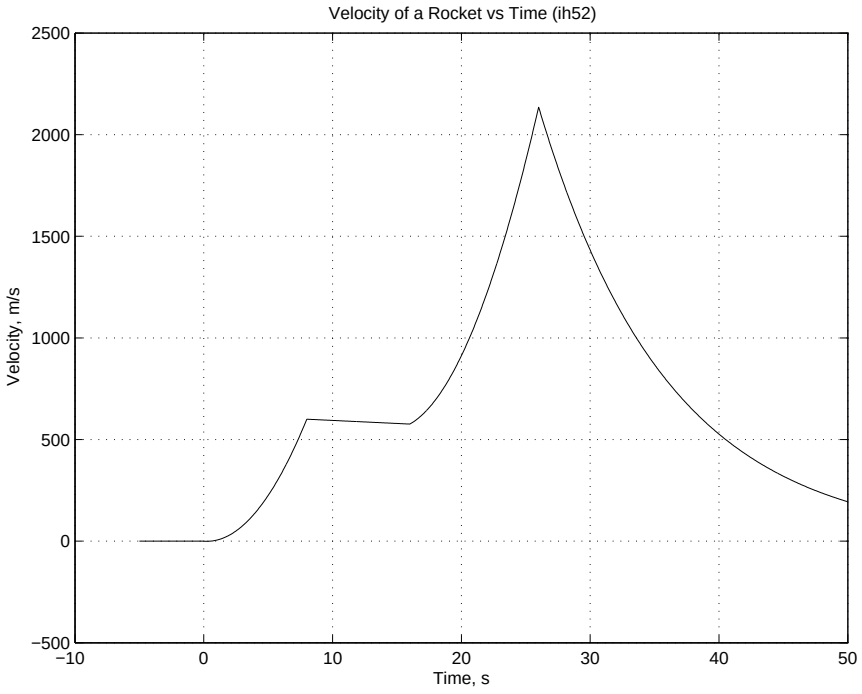

Figure 3: Velocity of a Rocket vs. Time

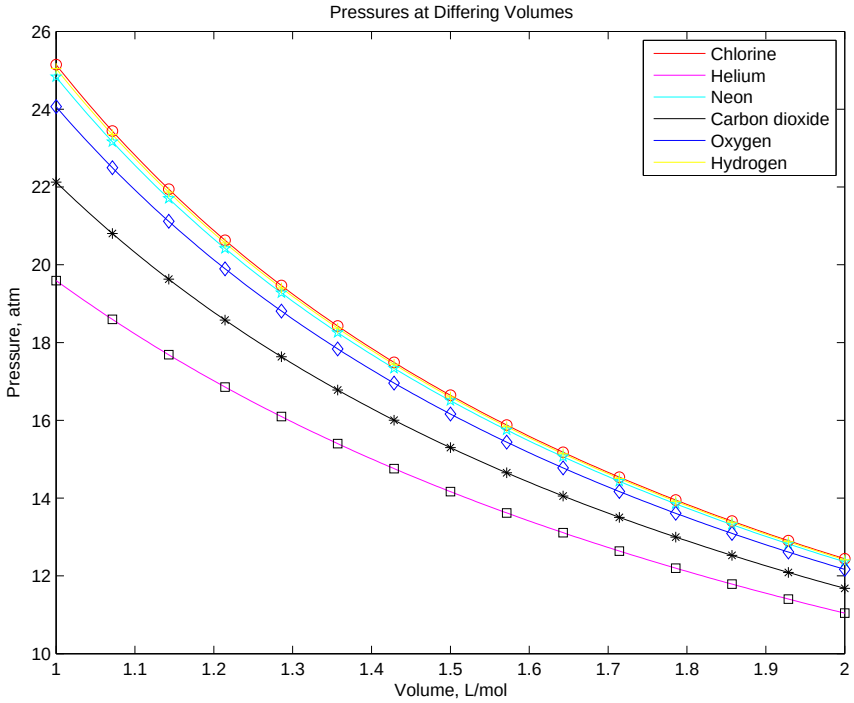

Figure 4: Pressures of Gases at Differing Volumes

References

[1] Chapra, Steven C., *Applied Numerical Methods with MATLAB for Engineering and Scientists*. McGraw-Hill, New York, 3rd Edition, 2012.

[2] Palm, William J., *Introduction to MATLAB for Engineers*. McGraw-Hill, New York, 3rd Edition, 2011.

[3] Weast, R.C. *Van Der Waals Constants (Data Page)* Wikipedia, Wikimedia Foundation 23 Sept. 2017