Collect all m-files in a single .zip file and upload the .zip file to the course webpage by midnight on Wednesday, May 1, 2019. Please note any collaborations in comments. Each student must upload their own unique copy of the work.

12_1. Solve Chapra problem 23.13. How many data points are generated by MATLAB when using ode45 vs. ode23s?

NOTE: There is a typo in the third differential equation in the book (has one extra (-) sign):

$$\frac{dc_3}{dt} = 0.013c_1 - 1000c_1c_3 - 2500c_2c_3$$

Solution:

```
%ps12_1

clear; clf;

%Write ODEs
dc1dt = @(t,c1,c2,c3) -0.013*c1-1000*c1.*c3;
dc2dt = @(t,c1,c2,c3) -2500*c2.*c3;
dc3dt = @(t,c1,c2,c3) 0.013*c1-1000*c1.*c3-2500*c2.*c3;
%Combine
dCdt = @(t,C) [dc1dt(t,C(1),C(2),C(3));
               dc2dt(t,C(1),C(2),C(3));
               dc3dt(t,C(1),C(2),C(3))];
%Solve w/ode45
subplot(1,2,1) %Subplotting is optional
[t, C] = ode45(dCdt,[0 50],[1 1 0]);
c1 = C(:,1); c2 = C(:,2); c3 = C(:,3);
plot(t,c1,'ko-'); hold on;
plot(t,c2,'rd-');
plot(t,c3,'bs-');
xlabel('t')
ylabel('c')
title('ode45')
legend('c1','c2','c3')
num_terms_ode45 = length(t)

%Solve w/ode23s
subplot(1,2,2)
[t, C] = ode23s(dCdt,[0 50],[1 1 0]);
c1 = C(:,1); c2 = C(:,2); c3 = C(:,3);
plot(t,c1,'ko-'); hold on;
plot(t,c2,'rd-');
plot(t,c3,'bs-');
xlabel('t')
ylabel('c')
title('ode23s')
```

```
legend('c1','c2','c3')
num_terms_ode23s = length(t)
```

Output:

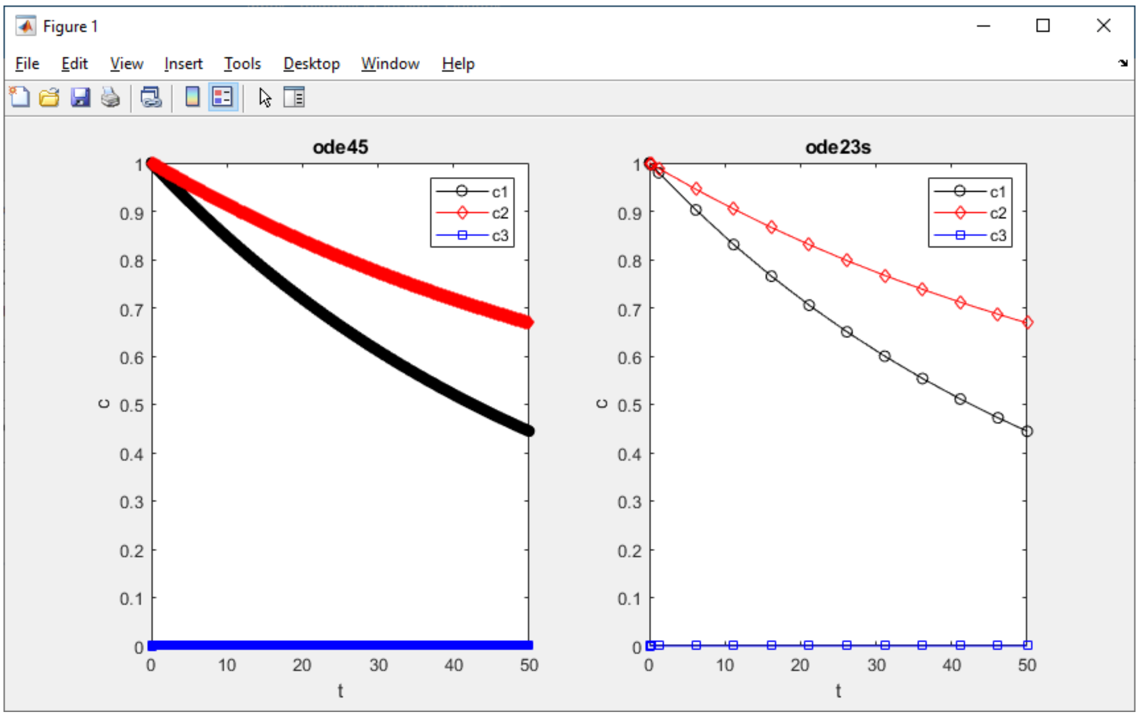
```
>> ps12_1

num_terms_ode45 =

      163881

num_terms_ode23s =

    18
```



12_2. Use MATLAB to solve example 6.10 in the Fogler text (pp. 352-355 of the attached PDF). NOTE: read the example closely, as it provides all of the needed constants, differential equations, and rate expressions. Create an identical figure to E6-10.1 (bottom of pg. 355 in PDF) as your output. (Note that the labeling positions, axis numbering, fonts, etc. don't have to be *perfectly* identical, but the curves shown should match… and you should still provide sufficient labeling in some form!)

Solution:

```
function PSET12_1
% Coded by Nigel F. Reuel on 12.2.2016
% This code solves problem 1 (Fogler example) with ODE solver
%
y0 = [10 10 0 0 0 0];
span = [0 10];
[tp yp] = ode45(@dydt,span,y0);
plot(tp,yp)
legend('F_A','F_B','F_C','F_D','F_E','F_F')
xlabel('V (dm^3)')
ylabel('F_i (mol/min)')

end

function dy = dydt(t,y)
% FA = y(1) , FB = y(2), FC = y(3), FD = y(4), FE = y(5), FF =
y(6)
% (dependent state variables)
%
% t = V (independent variable)
%
Ft = sum(y);
r1A = -5*8*(y(1)/Ft)*(y(2)/Ft)^2;
r2A = -2*4*(y(1)/Ft)*(y(2)/Ft);
r4C = -5*3.175*(y(3)/Ft)*(y(1)/Ft)^(2/3);
r3B = -10*8*(y(3)/Ft)^2*(y(2)/Ft);
%CA = 2*y(1)/Ft;
rA = r1A+r2A+2*r4C/3;
rB = 1.25*r1A+0.75*r2A+r3B;
rC = -r1A+2*r3B+r4C;
rD = -1.5*r1A-1.5*r2A-r4C;
rE = -0.5*r2A-5*r4C/6;
rF = -2*r3B;
dy = [rA; rB; rC; rD; rE; rF];
end
```

Output

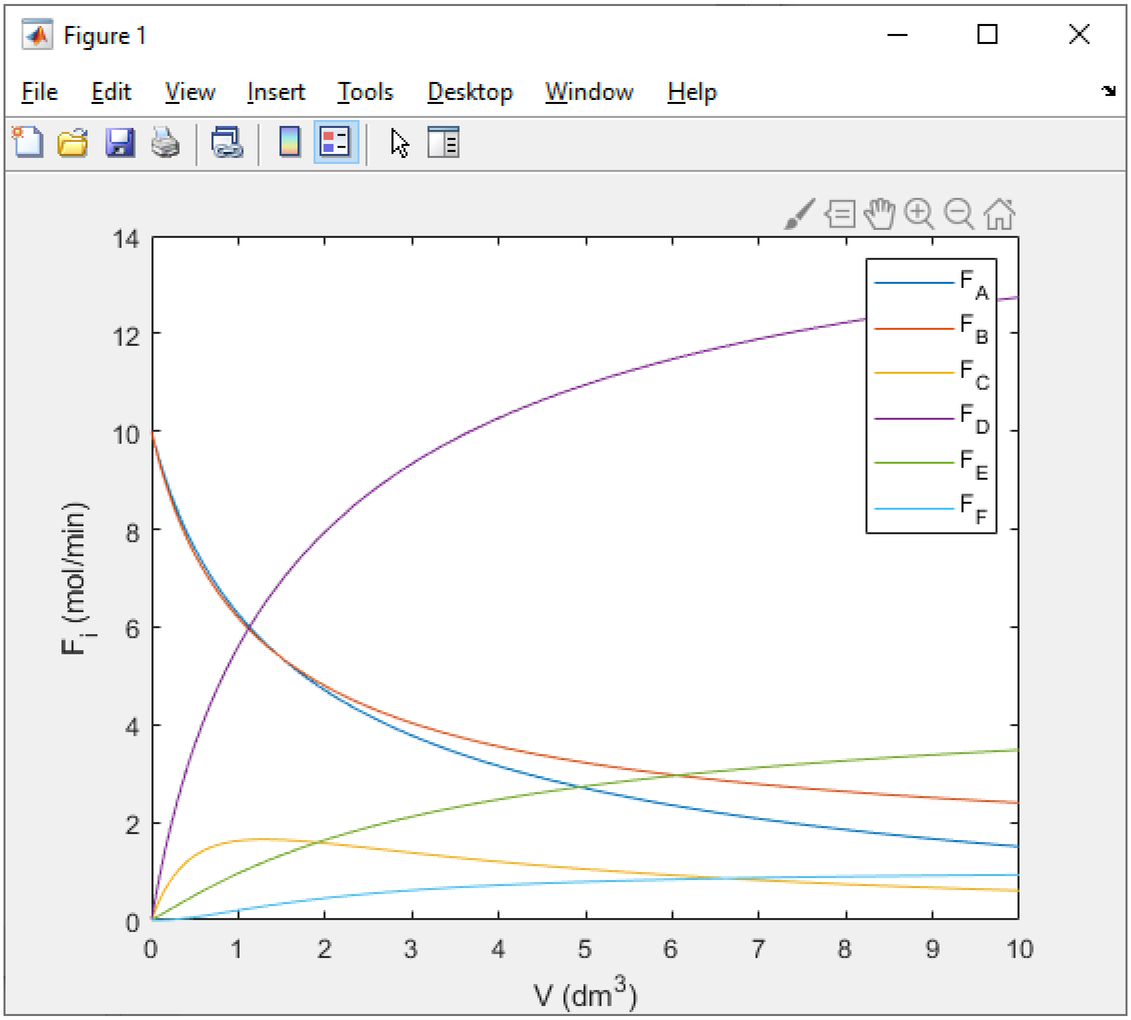


Figure 1
File Edit View Insert Tools Desktop Window Help
F_A
F_B
F_C
F_D
F_E
F_F
14
12
10
8
6
4
2
0
0 1 2 3 4 5 6 7 8 9 10
F_i (mol/min)
V (dm^3)

12_3. Solve Chapra problem 24.11. Plot the concentration of *A* as a function of distance.

Solution:

```
% PS12_3.m
% Coded by NFR on 12.5.2016
% This code solves problem 24.11 from the textbook
%
% Use shooting method, two shots because the problem is linear.
% 2017 comment - or you can use fzero or fminsearch to find the
unknown
% transformed variable.
A0 = 0.1/1000; %mol/cm^3
zg1 = -.001; %First guess for z(0)
y0 = [A0 zg1];
span = [0 4];
[tp,yp] = ode45(@dydt,span,y0);
hend1 = yp(end,1);
zg2 = .001; %Second guess for z(0)
y0 = [A0 zg2];
span = [0 4];
[tp,yp] = ode45(@dydt,span,y0);
hend2 = yp(end,1);
% Interpolate between the two to find the correct z(0) that
gives us the
% desired end concentration of 0 mol/cm^3
EndDesired = 0;
Slope = (zg2-zg1)/(hend2-hend1);
zcorrect = Slope*(EndDesired-hend1)+zg1;
% Use the correct z value to solve and plot h vs. x:
y0 = [A0 zcorrect];
[tp,yp] = ode45(@dydt,span,y0);
plot(tp,yp(:,1)*1000) %NOTE: converted units back to Molar
xlabel('Length (cm)')
ylabel('A concentration (M)')

function dy = dydt(t,y)
% t = x, y(1) = A, y(2) = z;
k = 5*10^-6; %-s
D = 1.5*10^-6; %cm^2/s
dy = [y(2);
    k*y(1)/D];
end
```

Output:

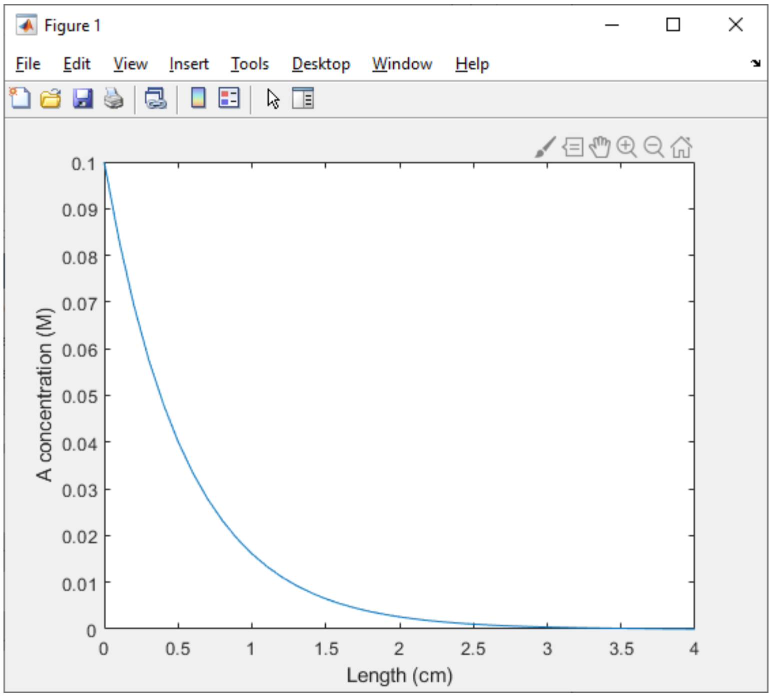


Figure 1
File Edit View Insert Tools Desktop Window Help
A concentration (M)
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 0.5 1 1.5 2 2.5 3 3.5 4
Length (cm)

12_4. Solve Chapra problem 24.18.

```
% PS12_4.m
% Coded by Nigel F. Reuel on 12.2.2016
% This solves problem 24.18 in the book
%
% Part (A)
% This equation is linear, so we can do two shots and then
interpolate.
zg1 = -1; %First guess for z(0)
y0 = [10 zg1];
span = [0 1000];
[tp,yp] = ode45(@dydt,span,y0);
hend1 = yp(end,1);
zg2 = 1; %Second guess for z(0)
y0 = [10 zg2];
span = [0 1000];
[tp,yp] = ode45(@dydt,span,y0);
hend2 = yp(end,1);
% Interpolate between the two to find the correct z(0) that
gives us the
% desired end height of h(1000) = 5m;
EndDesired = 5;
Slope = (zg2-zg1)/(hend2-hend1);
zcorrect = Slope*(EndDesired-hend1)+zg1;
% Use the correct z value to solve and plot h vs. x:
y0 = [10 zcorrect];
[tp,yp] = ode45(@dydt,span,y0);
plot(tp,yp(:,1))
% Part B - see paper work for derivation of tridiagonal matrix
e = ones(1,9);
e(1) = 0;
g = ones(1,9);
g(9) = 0;
f = ones(1,9)*-2;
r = ones(1,9)*-0.1333;
r(1) = -10.1333;
r(9) = -5.13333;
hvec = Tridiag(e,f,g,r);
% Add the end values ot your internal points:
hvec = [10 hvec 5];
xvec = 0:100:1000;
hold on
plot(xvec,hvec,'o')
xlabel('X distance (m)')
ylabel('Water table height (m)')
```

```
function x = Tridiag(e,f,g,r)
% Tridiag: Tridiagonal equation solver banded system
% x = Tridiag(e,f,g,r): Tridiagonal system solver.
% input:
% e = subdiagonal vector
% f = diagonal vector
% g = superdiagonal vector
% r = right hand side vector
% output:
% x = solution vector
n=length(f);
% forward elimination
for k = 2:n
factor = e(k)/f(k-1);
f(k) = f(k) - factor*g(k-1);
r(k) = r(k) - factor*r(k-1);
end
% back substitution
x(n) = r(n)/f(n);
for k = n-1:-1:1
x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
end

function dy = dydt(t,y)
% t = x, y(1) = h, y(2) = z
K = 1;
N = 0.0001;
hbar = 7.5; % Average of the boundary conditions;
dy = [y(2);
    -N/(K*hbar)];
end
```

Output:

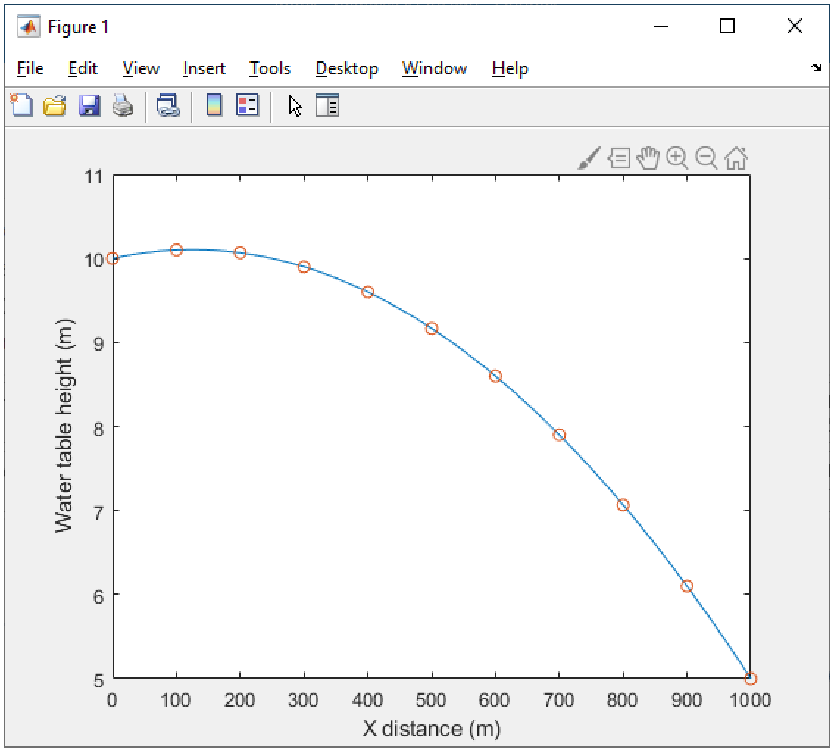
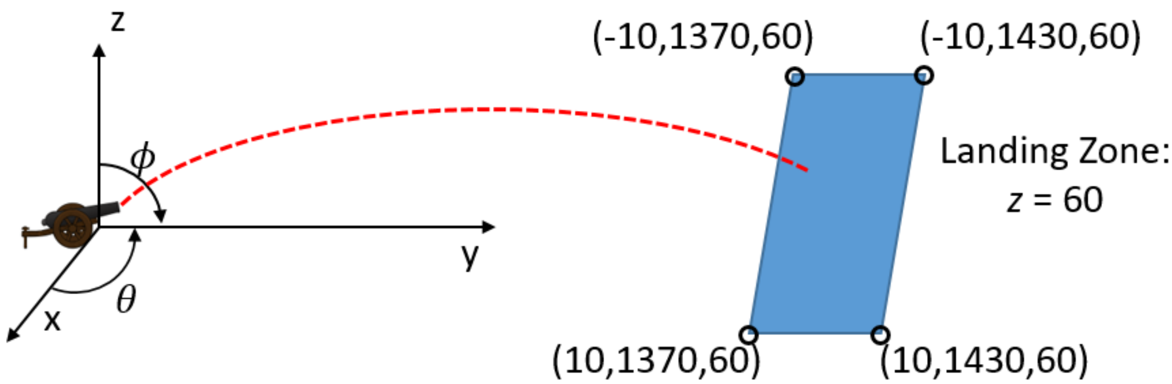


Figure 1
File Edit View Insert Tools Desktop Window Help
Water table height (m)
X distance (m)
11
10
9
8
7
6
5
0 100 200 300 400 500 600 700 800 900 1000

Challenge 12_5 (4 points extra credit)

We're firing a cannon to land a projectile within a target zone. We define a coordinate system in which *y* is the direction toward the target zone (positive direction moves from us toward the target), *x* is the "left/right" direction when facing the target zone (positive direction to the right), and *z* is the direction normal to the ground (positive direction upward).



The movement of the projectile is governed by the following equations of motion:

$$\frac{dx}{dt} = v_x \qquad \frac{dy}{dt} = v_y \qquad \frac{dz}{dt} = v_z$$

$$\frac{dv_x}{dt} = a_x \qquad \frac{dv_y}{dt} = a_y \qquad \frac{dz}{dt} = a_z$$

The accelerations can be defined by the following equations:

$$a_x = [sign(v_{wind,x} - v_x)]\frac{C_d}{2}\rho_{air}(v_{wind,x} - v_x)^2 A_{ref}/m_{ball}$$

$$a_y = [sign(v_{wind,y} - v_y)]\frac{C_d}{2}\rho_{air}(v_{wind,y} - v_y)^2 A_{ref}/m_{ball}$$

$$a_z = [sign(v_{wind,z} - v_z)]\frac{C_d}{2}\rho_{air}(v_{wind,z} - v_z)^2 A_{ref}/m_{ball} - g$$

Note the "sign" function, which can be used in MATLAB. This returns a (+1) or a (-1); this comes from air resistance opposing the direction of movement.

The appropriate A_{ref} value for a sphere's surface area is calculated as:

$$A_{ref} = \frac{\pi}{4}D_{ball}^2$$

For this problem, we consider the following physical constants:

$$C_d = 0.47 \qquad \rho_{air} = 0.737\frac{kg}{m^3} \qquad D_{ball} = 0.254\,m \qquad m_{ball} = 45\,kg \qquad g = 9.81\frac{m}{s^2}$$

We also consider the following initial position of the projectile:

$$x(t = 0) = 0 \qquad y(0) = 0 \qquad z(0) = 0$$

The initial velocities are determined by the following equations, as determined by the launch angles of the cannon:

$$v_x(0) = v_0 \sin(\phi)\cos(\theta) \qquad v_y(0) = v_0 \sin(\phi)\sin(\theta) \qquad v_z(0) = v_0 \cos(\phi)$$

For this problem, v_0 is fixed at 142 m/s.

The target zone is located in a 10x10 square landing area defined by coordinates:

$$(-10,1370,60); (10,1370,60); (-10,1430,60); (10,1430,60)$$

(A) First consider the case in which $v_{wind,x} = 0$ and $\theta = 90°$. (This corresponds to the case in which all x-components are 0.) Assume $v_{wind,z} = 0$. For each of the following cases, find a value of ϕ for which the projectile will land in the target zone.

a. $v_{wind,y} = 0 \frac{m}{s}$

b. $v_{wind,y} = -20 \frac{m}{s}$ (wind in your face)

c. $v_{wind,y} = 30 \frac{m}{s}$

(B) There's now a strong cross wind: $v_{wind,x} = -30$, $v_{wind,y} = -10$, $v_{wind,z} = 0$. Find a pair of θ and ϕ such that your cannon will hit its target zone.