**ChE 310 Problem Set 10 Due Wed 4/17/19**

Collect all m-files in a single .zip file and upload the .zip file to the course webpage by midnight on Wednesday, April 17, 2019. Please note any collaborations in comments. Each student must upload their own unique copy of the work.

**10\_1 Derivatives and noisy data.**

A jet fighter is landing on an aircraft carrier runway. *Arresting gear* is used to help quickly decelerate the plane on the short runway. However, we need to ensure the deceleration is safe. For the pilots’ comfort, we assume that 5 *g* is a reasonable limit. (*g* = 9.81 m/s2)

Our engineers have assured us that the arresting gear provides a perfectly constant force, and therefore a perfectly constant deceleration. We perform a simple experiment, in which several observers note the position of the jet (x) after a certain time (t). Since we’re doing this by hand, there is a certain amount of error associated with the measurements.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t (s) | 0 | 0.25 | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 |
| x (m) | 0 | 19 | 36 | 53 | 66 | 74 | 82 | 91 | 95 | 96 |

1. Keeping in mind that velocity is equal to $\frac{dx}{dt}$ and acceleration is $\frac{d^{2}x}{dt^{2}}$, numerically calculate the velocity and deceleration experienced by the plane at each time.
2. With this method of calculating derivatives, at what times will the deceleration exceed the 5 *g* limit?
3. If the engineers are correct (and we’ll assume they are ☺), then regression can be used to fit the (t,x) data to a quadratic model. Perform this regression, and determine the constant deceleration that the plane actually experiences according to the quadratic model. Does this actually exceed the 5 *g* safety limit?
4. On a 1x3 subplot, plot: (i) x vs. t, (ii) $\frac{dx}{dt}$ vs. t, and (iii) $\frac{d^{2}x}{dt^{2}}$ vs. t. For each subplot, include both the data calculated in (A) and the corresponding curves from the fitted function calculated in (C). This demonstrates how much the slight measurement errors affect our deceleration calculations, since we expected the deceleration to be perfectly constant!

%PS10\_1

clear; clf

%Input data

t = 0:0.25:2.25;

x = [0 19 36 53 66 74 82 91 95 96];

%Part (A)

%Calculate Derivatives. Method 1: Use the gradient function.

dxdt = gradient(x,0.25);

d2xdt2 = gradient(dxdt,0.25)

%We could also perhaps use the formulas to get the second derivatives,

%without using gradient in succession.

d2xdt2(1) = (x(3)-2\*x(2)+x(1))/(.25^2);

d2xdt2(end) = (x(end-2)-2\*x(end-1)+x(end))/(.25^2);

for ii = 2:length(d2xdt2)-1

 d2xdt2(ii) = (x(ii+1)-2\*x(ii)+x(ii-1))/(.25^2);

end

%Part (B)

decel\_numeric\_g = d2xdt2/9.81

%For this method, the deceleration exceeds 5g at t = 0.75, 1, 1.75.

%Part (C)

p = polyfit(t,x,2);

decel\_fit\_g = p(1)\*2/9.81

%This constant deceleration does NOT exceed the 5g safety limit.

%Part (D)

subplot(1,3,1)

plot(t,x,'ko'); hold on

fplot(@(t) polyval(p,t),xlim)

xlabel('t, s');ylabel('x, m')

subplot(1,3,2)

plot(t,dxdt,'ko'); hold on

fplot(@(t) 2\*p(1)\*t + p(2),xlim)

xlabel('t, s');ylabel('dx/dt, m/s')

subplot(1,3,3)

plot(t,d2xdt2,'ko'); hold on

fplot(@(t) 2\*p(1),xlim)

xlabel('t, s');ylabel('d^2x/dt^2, m/s^2')

***Output***

decel\_numeric\_g =

 -3.2620 -3.2620 0 -6.5240 -8.1549 0 1.6310 -8.1549 -4.8930 -4.8930

decel\_fit\_g =

 -3.5585

****

**10\_2 2-D Integration**

Consider the following temperatures measured on a two-dimensional plate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | $$x=0$$ | $$x=2$$ | $$x=4$$ | $$x=6$$ | $$x=8$$ |
| $$y=0$$ | 100.00 | 90.00 | 80.00 | 70.00 | 60.00 |
| $$y=2$$ | 85.00 | 64.49 | 53.50 | 48.15 | 50.00 |
| $$y=4$$ | 70.00 | 48.90 | 38.43 | 35.03 | 40.00 |
| $$y=6$$ | 55.00 | 38.78 | 30.39 | 27.07 | 30.00 |
| $$y=8$$ | 40.00 | 35.00 | 30.00 | 25.00 | 20.00 |

Use an integration method of your choice to determine the average plate temperature.

%ps10\_2.m

clear

%import temperatures

x = 0:2:8; y = x;

T = [100.00 90.00 80.00 70.00 60.00

85.00 64.49 53.50 48.15 50.00

70.00 48.90 38.43 35.03 40.00

55.00 38.78 30.39 27.07 30.00

40.00 35.00 30.00 25.00 20.00];

%h is constant for all x and y data (h=2)

h = x(2)-x(1);

%We can integrate across rows, and then over the columns.

%Don't forget to divide by the x- and y-lengths in order to get an average...

%Simplest way: trapezoid method

T\_int\_x=zeros(5,1);

for ii = 1:5

 T\_int\_x(ii) = trapz(x,T(ii,:))/8;

end

%Now, we can integrate over the column.

T\_int\_trap = trapz(y,T\_int\_x)/8;

fprintf('The average temperature (Trapezoid method) is %4.4g K\n',T\_int\_trap)

%We could also do this in a single line

T\_int\_trap = trapz(x,trapz(y,T))/64;

%We can try other methods as well...

%Composite Simpson's 1/3

for ii = 1:5

 T\_int\_x(ii) = (h/3\*(T(ii,1)+T(ii,end))+4\*h/3\*sum(T(ii,2:2:end-1)) + ...

 2\*h/3\*sum(T(ii,3:2:end-2)))/8;

end

T\_int\_Simpson13 = (h/3\*(T\_int\_x(1)+T\_int\_x(end))+...

 4\*h/3\*sum(T\_int\_x(2:2:end-1))+2\*h/3\*sum(T\_int\_x(3:2:end-2)))/8;

fprintf('The average temperature (Simpson 1/3 method) is %4.4g K\n',T\_int\_Simpson13)

%how about Boole's rule?

T\_int\_x=zeros(5,1);

for ii = 1:5

 T\_int\_x(ii) = 2\*h/45\*(7\*T(ii,1)+32\*T(ii,2)+12\*T(ii,3)+32\*T(ii,4)+7\*T(ii,5))/8;

end

T\_int\_Boole = 2\*h/45\*(7\*T\_int\_x(1)+32\*T\_int\_x(2)+12\*T\_int\_x(3)...

 +32\*T\_int\_x(4)+7\*T\_int\_x(5))/8;

fprintf('The average temperature (Boole method) is %4.4g K\n',T\_int\_Boole)

***Output***

The average temperature (Trapezoid method) is 48.11 K

The average temperature (Simpson 1/3 method) is 47.03 K

The average temperature (Boole method) is 46.98 K

**10\_3** Solve problem 19.22 from the text.

%PS10\_3

clear; clf;

% Coded by Nigel F. Reuel on 11.13.2017, updated LTR 4.15.2019

% See paper work for notes on spherical coordinate derivation

r = [0 1100 1500 2450 3400 3630 4500 5380 6060 6280 6380].\*10^5; % Convert units to cm

rho = [13 12.4 12 11.2 9.7 5.7 5.2 4.7 3.6 3.4 3]; %g/cm^3

%Triple integral: need to integrate rho\*r^2\*sin(phi) dtheta dphi dr

rho\_rsqr\_vec = rho.\*r.^2;

%First integral: dtheta over 0 to 2pi is just 2pi

integrand\_aftertheta = rho\_rsqr\_vec\*2\*pi;

%second integral: integrating sin(phi) over 0 to pi is 2

integrand\_afterphi = integrand\_aftertheta \* 2;

%final integral: integrate over r. can just use trapz.

disp('Mass of Earth in Metric Tonnes:')

mass = trapz(r,integrand\_afterphi)\*1e-6 %convert to metric tonnes

%To get average density, divide mass by total volume

VolSphere = 4/3\*pi\*(r(end))^3; % Volume of sphere in cm^3

disp('Average Density of Earth in g/cm^3:')

AvgRho = mass\*1e6/VolSphere

% Vertical stacked plots

subplot(2,1,1)

plot(r,rho)

xlabel('Radius (km)')

ylabel('Density (g/cm^3)')

subplot(2,1,2)

Mass\_Vec = 4\*pi\*cumtrapz(r,rho\_rsqr\_vec)/10^6;

plot(r,Mass\_Vec)

xlabel('Radius (km)')

ylabel('Mass (metric tonnes)')

***Output***

Mass of Earth in Metric Tonnes:

mass =

 6.1087e+21

Average Density of Earth in g/cm^3:

AvgRho =

 5.6156

****

**10\_4** Solve problem 21.25 from the text.

%ps10\_4.m

%Chapra Problem 21.25

clear; clf; close all;

%Input data

z = 0:4:16; %m

V = [9.8175 5.1051 1.9635 0.3927 0.0000]; %10^6 m^3

c = [10.2 8.5 7.4 5.2 4.1]; %g/m^3

h = z(2)-z(1); %uniform step size, m

%First, calculate derivatives. Let's stick with 1st order.

A = gradient(V,h); % uniform step size of 4; 10^6 m^2

cA = c.\*A ; %10^6 g/m

%Composite trapezoid method

%Calculate numerator and denominator integrals

num\_int\_trap = trapz(z,cA); %10^6 g

den\_int\_trap = trapz(z,A); %10^6 m^3

c\_avg\_trap = num\_int\_trap / den\_int\_trap; %g/m^3

fprintf('The average concentration integrated using the trapezoid rule is %4.4g g/m^3.\n',c\_avg\_trap)

%Simpson's 1/3 method

num\_int\_Simp13 = h/3\*(cA(1)+cA(end))+ 4\*h/3\*sum(cA(2:2:end-1))...

 +2\*h/3\*sum(cA(3:2:end-2));

den\_int\_Simp13 = h/3\*(A(1)+A(end))+ 4\*h/3\*sum(A(2:2:end-1))...

 +2\*h/3\*sum(A(3:2:end-2));

c\_avg\_Simp13 = num\_int\_Simp13 / den\_int\_Simp13; %g/m^3

fprintf('The average concentration integrated using Simpson 1/3 rule is %4.4g g/m^3.\n',c\_avg\_Simp13)

***Output:***

The average concentration integrated using the trapezoid rule is 8.226 g/m^3.

The average concentration integrated using Simpson 1/3 rule is 8.097 g/m^3.

**10\_5** Solve problem 21.29 from the text.

%PS10\_5

%Chapra Problem 21.29

clear;

%First, import data.

V = [220 250 282.5;

 4.1 4.7 5.23;

 2.2 2.5 2.7;

 1.35 1.49 1.55;

 1.1 1.2 1.24;

 0.90 0.99 1.03;

 0.68 0.75 0.78;

 0.61 0.675 0.7;

 0.54 0.6 0.62]; %L/mol

P = [0.1 5 10 20 25 30 40 45 50]; %atm

%calculate derivatives at T = 400, using 1st order centered.

dVdT = (V(:,3)-V(:,1))/(100) ;

%set up the integrand

integrand = V(:,2)-400\*dVdT;

%integrate

int\_trap = trapz(P,integrand); %L\*atm/mol

%optional, convert to kJ/mol

int\_trap = int\_trap \* 101.325/1000;

fprintf('The enthalpy using trapz is %4.4g kJ/mol.\n',int\_trap)

***Output***

The enthalpy using trapz is 2.173 kJ/mol.