AAE 490F/AT490F
Homework 1
Due Wednesday January 23, 2008

To do this homework you should use MATLAB and use the function atmosphere4.m that can be found on the course web site from Spring 2006

http://roger.ecn.purdue.edu/%7Eandrisan/Courses/AAE490A_S2007/Buffer/

1. Write a MATLAB script to plot the difference between geometric altitude and geopotential altitude versus geopotential altitude over the geopotential altitude range from 0 to 60,000 ft.

2. Write a MATLAB script to plot atmospheric temperature versus geopotential altitude over the geopotential altitude range from 0 to 60,000 ft.

3. Write a MATLAB script to plot atmospheric pressure versus geometric altitude over the geometric altitude range from 0 to 60,000 ft.

Be sure to label the axes of all plots. Always have Matlab put your name somewhere on your plots.

4. Assume that a correctly calibrated altimeter is set to 29.92 inches of mercury in the Kohlsman window. Assume that the altimeter indicates a pressure altitude of 10000 feet. Assume that a corrected temperature gage reads 480degR. What is the density of the air?

Read Chapter 1, 2, and 3 of Kimberlin.

Course Web Site

http://cobweb.ecn.purdue.edu/~andrisan/Courses/AAE490A_S2008/AAE490A_S2008.html

Site where I will put many files useful in this course.

http://roger.ecn.purdue.edu/%7Eandrisan/Courses/AAE490A_S2008/Buffer/

The Standard Atmosphere

1. Temperature is a specified function of altitude. Temperature is either constant with altitude, as in the stratosphere, or varies at a constant rate with altitude. This rate is called the lapse rate. In the troposphere the lapse rate is –3.56616degR/1000ft.
2. The troposphere is the lowest atmospheric region and extends from sea level to 36089.239 geopotential feet.
3. The stratosphere is the next highest atmospheric region and extends from 36,089.239 geopotential feet to 65,616.798 geopotential feet.
4. The tropopause is the dividing line between the troposphere and the stratosphere at an altitude of 36,089.239 geopotential feet.
5. The standard atmosphere ignores the fact that any given day might be hotter or colder than standard or have a higher or lower pressure than standard. In fact, in the standard atmosphere there are no weather variations at all. If we lived in the standard atmosphere at 600 ft elevation (for Lafayette, Indiana) the temperature would always be 56.86 degF, the pressure would always be 29.2769 inHg, and the density would be 0.00233546 slug/ft^3.
6. We will use the 1976 NASA Standard Atmosphere which has 8 atmospheric regions (temperature regions) up to 295,000 geopotential feet.
7. At sea level in the standard atmosphere
   - Pressure=p_o=2116.22 lbf/ft^2=29.92 inches of Hg (mercury)
   - Temperature=T_o=518.67 degR=59 degF [note: T(degR)=T(degF)+459.67]
   - Density=ρ_o=0.00237691 slug/ft^3 [note: when g=g_o a mass of 1 slug will weigh 32.17405 lbf]
   - Gravitational constant g_o=32.17405 ft/sec^2
   - Relative humidity is zero (dry air).
   - R=1716.55 $ft^2/(sec^2 degR)$
8. Geopotential altitude, h, is related to geometric altitude (tapeline altitude), h_g, by
   - $h=(R_E/(R_E+h_g))h_g$ where R_E=Radius of the Earth=20,926,476 ft.
9. The equations for pressure and density in the Standard Atmosphere are based on the Equation of State for Air (i.e., the Perfect Gas Law, p=ρRT (equation 1.5)) and the hydrostatic equation (equation 1.4).
10. In the Troposphere (h<36089.239 geopotential feet)
    - T=518.67-.00356616*h degR
    - p=2116.22*(1-6.87558563248308e-06*h)^(5.25591641274834) lbf/ft^2
    - ρ=0.00237691267925741*(1-6.87558563248308e-06*h)^(4.25591641274834) slug/ft^3
11. In the Stratosphere(36089.239 geopotential feet<h<65616.798239 geopotential feet)
    - T=389.97degR
    - p=472.675801650081*exp(-4.80637968933164e-05*(h-36089.239)) lbf/ft^2
    - ρ=0.000706115448911997*exp(-4.80637968933164e-05*(h-36089.239)) slug/ft^3
12. Commonly we write Standard Atmosphere properties in terms of the ratio of a quantity to the sea level value of that quantity.
    - Pressure ratio $\delta=p/p_o$
    - Density ratio $\sigma=\rho/\rho_o$
    - Temperature ratio $\theta=T/T_o$
    - Equation of State for Air $\delta=\sigma\theta$
13. The speed of sound, a, is given by
    - a=sqrt(γRT)=sqrt(γp/ρ) where γ=1.4 for air.


100 FEET
ALT
RANGE
20,000 FEET
29.9
30.0

MATLAB INTRODUCTION

Below is a MATLAB script introducing you to
MATLAB and giving examples of how to use the standard
atmosphere function called atmosphere4. Put the contents
of this file from below the ****** into a text file called test.m

This file and matlab function atmosphere4 cam be found
http://roger.ecn.purdue.edu/%7Eandrisan/Courses/AAE490A_S2006/Buffer/

Make sure test.m and atmosphere4.m are in the MATLAB search path.

Execute the file test.m from the MATLAB command line
by typing test

```
******
% General Introduction to MATLAB and a
% script to use the MATLAB function called atmosphere4.

disp('Start of the script') %This line displays in the MATLAB
% Command Window the text contained within the two quote marks.
% Comment lines start with a %, like this line.
%  omment lines generate no output in the MATLAB
% Command Window (unless echo in on).
% FILE NAMES
% A MATLAB command is any command that can be executed
% on the MATLAB Command Window.
% A MATLAB script file is a file containing MATLAB commands.
% Script files can have any name as long as it ends in .m
% Function file names (atmosphere4.m) must be the same as the
% function name (atmosphere4) with the .m appended.
% The function name is found on the first line of the
% function file, i.e.,
% function [temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag)
% MATLAB SEARCH PATH (the search path that MATLAB uses to find files)
% The function atmosphere4 must be in the MATLAB search path
% in order for MATLAB to find it. If it is not in the search path
% use the Set Path command in the file menu of the MATLAB Command Window.

% MATLAB COMMANDS (also MATLAB functions)
% The MATLAB function help will type the initial comment lines
% of a file until the first blank or executable line is found.
help atmosphere4

echo on
%  urns echo on so that all lines are printed
```

```
% before being executed.

%type atmosphere4 % will type the entire file.
% lines like help atmosphere4 and type atmosphere4 can also
% be executed from the MATLAB command line.
echo off % Turns echo off so that all lines are not
% printed befor being executed.

% USING FUNCTION atmosphere4 WHEN Hvector is a scalar
% Altitude data (Hvector) can be a scalar, i.e. one number.
Hvector=1000 %When a MATLAB line does not end with a ; the results
% of the line are printed.
[rowsize,columnsize]=size(Hvector) %When a MATLAB line does not end with a ;
%the results of the line are printed. In the case above rowsize and
% colsize are printed.

GeometricFlag=0 % This means that the altitude data is in geopotential feet.
[temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag)

GeometricFlag=1 % This means that the altitude data is in geometric feet.
[temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag)

% USING FUNCTION atmosphere4 WHEN Hvector is a vector
% Altitude data (Hvector) can be a vector, i.e. an array of numbers.
Hvector=0:1000:295000; %When a MATLAB line ends with a ; the results of
% the line is not printed.
% The above line creates a row vector called Hvector.
% it starts at value 0, ends at value 295,000 and values are incremented
% from eachother by 1000. There are to many numbers to print so
% the semicolen prevents printing of the line (296 numbers).

[rowsize,columnsize]=size(Hvector)
GeometricFlag=0 % This means that the altitude data is in geopotential feet.
[temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag);

%PLOTTING
plot(temp,Hgeopvector)
% Try typing help plot at the MATLAB command window.
xlabel('Temperature, degR')
ylabel('Altitude, geopotential feet')
title('1976 NASA Standard Atmosphere')
text(340,55000,'Stratosphere')
text(360,36089,'Tropopause')
text(340,15000,'Troposphere (Beechcraft Baron B58 flys here)')
whos % Lists the sizes of all variables.

% TRANSPOSING VECTOR Hvector
Hvector=Hvector'; % The prime means matrix transpose (i.e. interchange
```

```
% the rows and the columns. This will make Hvector a column vector (296x1)
% instead of a row vector(1x296).
[rowsize,columnsize]=size(Hvector)
GeometricFlag=0 % This means that the altitude data is in geopotential feet.
[temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag);
whos % Lists the sizes of all variables. Notice how the sizes
% of all the matrices have change from the last time we did the whos command.

% HELP COMMANDS
% If you have any questions about what a MATLAB command does
% type help command,i.e.
help text
%help help %Note that these commands have been commented out.
% Remove the % to execute them.
%help
%help general %The many topics you can find from the help command.
%help lang
%help ops
```

```
function [temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag)
%f  tion [temp,press,rho,Hgeopvector]=atmosphere4(Hvector,GeometricFlag)
%    ndard Atmospheric data based on the 1976 NASA Standard Atmoshere.
% Hvector is a vector of altitudes.
% If Hvector is Geometric altitude set GeometricFlag=1.
% If Hvector is Geopotential altitude set GeometricFlag=0.
% Temp, press, and rho are temperature, pressure and density
% output vectors the same size as Hgeomvector.
% Output vector Hgeopvector is a vector of corresponding geopotential altitudes (ft).
% This atmospheric model is good for altitudes up to 295,000 geopotential ft.
% Ref: Intoduction to Flight Test Engineering by Donald T. Ward and Thomas W. Strganac

% index   Lapse rate   Base Temp   Base Geopo Alt      Base Pressure         Base Density
%   i     Ki(degR/ft)  Ti(degR)      Hi(ft)            P, lbf/ft^2           RHO, slug/ft^3
format long g
D= [1     -.00356616    518.67            0                 2116.22         0.00237691267925741
    2        0          389.97      36089.239         472.675801650081      0.000706115448911997
    3      .00054864    389.97      65616.798         114.343050672041      0.000170813471460564
    4      .00153619    411.57     104986.878         18.1283133205764      2.56600341257735e-05
    5        0          487.17     154199.475         2.31620845720195      2.76975106424479e-06
    6     -.00109728    487.17     170603.675         1.23219156244977      1.47347009326248e-06
    7     -.00219456    454.17     200131.234         0.38030066501701      4.87168173794687e-07
    8        0          325.17     259186.352       0.0215739175227548      3.86714900013768e-08];

R=1716.55;      %ft^2/(sec^2degR)
gamma=1.4;
g0=32.17405;    %ft/sec^2
RE=20926476;    % Radius of the Earth, ft
K=D(:,2);       %degR/ft
T=D(:,3);       %degR
H=D(:,4);       %ft
P=D(:,5);       %lbf/ft^2
RHO=D(:,6);     %slug/ft^3

temp=zeros(size(Hvector));
press=zeros(size(Hvector));
r   eros(size(Hvector));
H   vector=zeros(size(Hvector));

% Convert from geometric altitude to geopotental altitude, if necessary.
if GeometricFlag
       Hgeopvector=(RE*Hvector)./(RE+Hvector);
       disp('Convert from geometric altitude to geopotential altitude in feet')
else
   Hgeopvector=Hvector;
   %disp('Input data is geopotential altitude in feet')
end
ih=length(Hgeopvector);
n1=find(Hgeopvector<=H(2));
n2=find(Hgeopvector<=H(3) & Hgeopvector>H(2));
n3=find(Hgeopvector<=H(4) & Hgeopvector>H(3));
n4=find(Hgeopvector<=H(5) & Hgeopvector>H(4));
n5=find(Hgeopvector<=H(6) & Hgeopvector>H(5));
n6=find(Hgeopvector<=H(7) & Hgeopvector>H(6));
n7=find(Hgeopvector<=H(8) & Hgeopvector>H(7));
n8=find(Hgeopvector<=295000 & Hgeopvector>H(8));
icorrect=length(n1)+length(n2)+length(n3)+length(n4)+length(n5)+length(n6)+length(n7)+length(n8);
if icorrect<ih
       disp('One or more altitutes is above the maximum for this atmospheric model')
       icorrect
       ih
end
% Index 1, Troposphere, K1= -.00356616
if length(n1)>0
       i=1;
       h=Hgeopvector(n1);
       TonTi=1+K(i)*(h-H(i))/T(i);
       temp(n1)=TonTi*T(i);
       PonPi=TonTi.^(-g0/(K(i)*R));
       press(n1)=P(i)*PonPi;
       RonRi=TonTi.^(-g0/(K(i)*R)-1);
       rho(n1)=RHO(i)*RonRi;
en
```

```
% Index 2,  K2= 0
if length(n2)>0
        i=2;
        h=Hgeopvector(n2);
        temp(n2)=T(i);
        PonPi=exp(-g0*(h-H(i))/(T(i)*R));
        press(n2)=P(i)*PonPi;
        RonRi=PonPi;
        rho(n2)=RHO(i)*RonRi;
end
% Index 3,  K3= .00054864
if length(n3)>0
        i=3;
        h=Hgeopvector(n3);
        TonTi=1+K(i)*(h-H(i))/T(i);
        temp(n3)=TonTi*T(i);
        PonPi=TonTi.^(-g0/(K(i)*R));
        press(n3)=P(i)*PonPi;
        RonRi=TonTi.^(-g0/(K(i)*R)-1);
        rho(n3)=RHO(i)*RonRi;
end
% Index 4,  K4= .00153619
if length(n4)>0
        i=4;
        h=Hgeopvector(n4);
        TonTi=1+K(i)*(h-H(i))/T(i);
        temp(n4)=TonTi*T(i);
        PonPi=TonTi.^(-g0/(K(i)*R));
        press(n4)=P(i)*PonPi;
        RonRi=TonTi.^(-g0/(K(i)*R)-1);
        rho(n4)=RHO(i)*RonRi;
end
% Index 5,  K5= 0
if length(n5)>0
        i=5;
        h=Hgeopvector(n5);
        temp(n5)=T(i);
        PonPi=exp(-g0*(h-H(i))/(T(i)*R));
        press(n5)=P(i)*PonPi;
        RonRi=PonPi;
        rho(n5)=RHO(i)*RonRi;
end
% Index 6,  K6= -.00109728
if length(n6)>0
        i=6;
        h=Hgeopvector(n6);
        TonTi=1+K(i)*(h-H(i))/T(i);
        temp(n6)=TonTi*T(i);
        PonPi=TonTi.^(-g0/(K(i)*R));
        press(n6)=P(i)*PonPi;
        RonRi=TonTi.^(-g0/(K(i)*R)-1);
        rho(n6)=RHO(i)*RonRi;
end
% Index 7,  K7= -.00219456
if length(n7)>0
        i=7;
        h=Hgeopvector(n7);
        TonTi=1+K(i)*(h-H(i))/T(i);
        temp(n7)=TonTi*T(i);
        PonPi=TonTi.^(-g0/(K(i)*R));
        press(n7)=P(i)*PonPi;
        RonRi=TonTi.^(-g0/(K(i)*R)-1);
        rho(n7)=RHO(i)*RonRi;
end
% Index 8,  K8= 0
if length(n8)>0
        i=8;
        h=Hgeopvector(n8);
        temp(n8)=T(i);
        PonPi=exp(-g0*(h-H(i))/(T(i)*R));
        press(n8)=P(i)*PonPi;
        RonRi=PonPi;
        rho(n8)=RHO(i)*RonRi;
end
```

AAE 490A
MATLAB Tutorial

Including the find function, writing functions, local variables in functions, array operations, avoiding for loops using array operations, and plotting.

By Professor Dominick Andrisani

WHAT FOLLOWS IS A TUTORIAL MATLAB SCRIPT CONTAINING SOME ASPECTS OF MATLAB PROGRAMMING THAT I WOULD LIKE ALL STUDENTS IN AERONAUTICS AND ASTRONAUTICS TO UNDERSTAND. THE SCRIPT CAN BE FOUND ON THE AAE490A COURSE WEB SITE.

```
% A&AE490A Students:
%
% Below is a MATLAB script introducing you to
% MATLAB and giving examples of how to use a function called rhofun. Put the
contents
% of this file from below the ****** into a text file called test.m
%
% Put the contents of the second e-mail in a text file rhofun.m
%
% Make sure test.m and rhofun.m are in the MATLAB search path.
%
% Execute the file test.m from the MATLAB command line
% by typing test
%
%
% ******
% General Introduction to MATLAB and a
% script to use the MATLAB function called density.
%
disp(' ') % display a blank line
disp('******* Start of the script *****') %This line displays in the MATLAB
% Command Window the text contained within the two quote marks.
% Comment lines start with a %, like this line.
% Comment lines generate no output in the MATLAB
% Command Window (unless echo is on).
%
% A MATLAB command is any command that can be executed
% on the MATLAB Command Window.
%
% A MATLAB script file is a file containing MATLAB commands.
% Script files can have any name as long as it ends in .m
echo on
% These examples show use of vectors and the find function.
help find

% Example 1
% This example shows
% 1. definition of a vector called x,
% 2. use of the find function,
% 3. and use of the isempty function.
x=[1,2,3,4,4,4,5,5,10,1,2,3,4,12,13,2,2,2]
```

```
j=find(x>15)
isempty(j)
x(j)=100;
x % This command prints the vector x in the command window.
% There are no elements of x>15 so j is empty and
% isempty returns the value 1 indicating isempty is 'true'.
% MATLAB uses value 1 to indicate the logical result called 'true'.
% In this case it is true that j is empty.

% Example 2
j=find(x>5)
isempty(j)
x(j)=100;
x
% There are 3 elements of x>5 so j has three elements in it and
% isempty returns the value 0 indicating isempty is 'false'.
% MATLAB uses value 0 to indicate the logical result called 'false'.
% In this case it is false that j is empty.
% j contains the elements 9,14,15 indicating that these elements
% of x satisfy the logical test 'Is x>5?'

% Example 3
% This example shows the use of the logical function and (&)
x=[1,2,3,4,4,4,5,5,10,1,2,3,4,12,13,2,2,2]
j=find(x>5 & x<11)
isempty(j)
x(j)=45;
x
% There is only one element of x (the 9th one) which is
% greater then 5 and less then 11.

% Example 4 Use of a for loop to compute atmospheric
% density as a function of altitude. The function [rho]=rhofun(h)
% expects its input argument (h) to be a scalar.
echo off
x=0:100:230000;
rhomat=zeros(size(x));
echo on
rho=1111 % rho before multiple calls to rhofun
h=2222   % h   before multiple calls to rhofun
echo off
t0=clock; % start a clock to time the computation of density
for j=1:length(x)
   rhomat(j)=rhofun(x(j));
end
format long;
```

```
echo on
ElapsedTime=etime(clock,t0)% the amount of time to find density
% This is the amount of time it takes to comput the density by the
% for loop method. The array method shown in Example 6 is much faster.
format
rho % rho after multiple calls to rhofun
h   % h   after multiple calls to rhofun
% Note that the local variables rho and h used in function
% rhofun do not change the values of variables rho ahd h defined
% here in the MATLAB workspace.
%
% In MATLAB it is desirable from a computational speed point of view
% to eliminate for loops. Whenever possible you should try
% to find a way to program using the array aperations discussed above.
% Example 5 below computes the same density data but uses
% array operations instead of the for loop.
echo off
plot(x,rhomat)
% The problem with this figure is that it is unlabeled.
% NEVER produce a plot with unlabeled axes! Be sure to
% include the variable name (e.g. altitude) and the units (e.g. ft).

% Example 5 Use array operations
echo on
y=[2,2,2,3,3,3,3] %define an array
z0=y.^3  % element by element raising to a power
% The above operation is used in the function rhofun2.
z1=y+1   % adding a scalar to each element of a vector
%        multiplying a vector by a constant
z2=2*y+1 % and adding 1 to each element of the vector

z3=y+z0  %adding two vectors
z4=z0.*y % element by element multiplication of two vectors
z5=z0./y % element by element division of two vectors
echo off

% Example 6 Use array operations to compute atmospheric
% density as a function of altitude. The function [rho]=rhofun2(h)
% expects its input argument (h) to be a vector (or scalar).
format long % print out more decimal places
t0=clock; % start clock again
echo on
rhomat2=rhofun2(x);  % compute vector or air densities
ElapsedTime2=etime(clock,t0) % the amount of time to do this computation
% MATLAB has to compile the function rhofun2 to execute
% the above function call. The first time you run this code
```

```
% ElapsedTime2 will have the compile time included in it.
% If you run the code a second time, MATLAB will use the already
% compiled function and ElapsedTime 2 will produce the
% time measure we want.
format % print out normal number of decimal places
% Notice how much less computer time this takes compared to the for loop method.
%
% Check out the figures generated by this script
echo off
factor=ElapsedTime/ElapsedTime2
text1=num2str(factor); % convert the number to a string
% assemble a longer string
text2=['It takes ',text1,' times as long to use the for loop method.'];
disp(text2) % display the ratio of computer time for the two methods
figure(2) % open a second figure window and plot density vs altitude
plot(x,rhomat2)
% The following lines label the plot
xlabel('altitude, feet')
ylabel('density, slugs/ft^3')
title('Density versus altitude computed using function rhofun2')
text(50000,.001,'NEVER produce a plot with unlabeled axes!')

% Additional Plotting Examples

% Plotting Example 1: Overplots on the same figure
% including added horizontal lines and text to indicate
% the atmospheric regions called troposphere and stratosphere.
figure(3) % open a third window to make overplots
plot(x,rhomat,'-',x,rhomat2,':')
legend('Using rhofun','Using rhofun2')
xlabel('altitude, feet')
ylabel('density, slugs/ft^3')
title('Density versus altitude computed using rhofun and rhofun2')
text(1000,.00225,'troposphere')
text(36089,.00185,'stratosphere')
hold on % plot the following two horizontal lines
% to indicate the regions of the troposphere and stratosphere
plot([0,36089],[.0022,.0022]) % Draw line
plot([36089,65617],[.0018,.0018])
hold off % if you turn hold on be sure to turn it off

% Plotting Example 2: Plot two figures on the same page
figure(4) % open a forth window to generate two plots per page
% plot a 2x1 array of figures on one page and do the first
% plot now.
subplot(211); plot(x,rhomat)
```

```
xlabel('altitude, feet')
ylabel('density, slugs/ft^3')
title('Density versus altitude computed using function rhofun')
text(1000,.0023,'troposphere')
text(36089,.0019,'stratosphere')
hold on % plot the following two horizontal lines
% to indicate the regions of the troposphere and stratosphere
plot([0,36089],[.0022,.0022]) % Draw line
plot([36089,65617],[.0018,.0018])
hold off % if you turn hold on be sure to turn it off
% continue with the 2x1 array of figures on one page and do the
% second plot now.
subplot(212); plot(x,rhomat2)
xlabel('altitude, feet')
ylabel('density, slugs/ft^3')
title('Density versus altitude computed using function rhofun2')
text(1000,.0023,'troposphere')
text(36089,.0019,'stratosphere')
text(36089,.0015,'The U.S. Air Force considers you to be an astronaut if')
text(36089,.0013,'you fly to an altitude greater than 50 miles (264,000 feet).')
text(36089,.0011,'Low earth orbit is about 100 miles up (528,000 feet).')
text(36089,.0009,'There is not much air left above 100,000 feet.')
hold on % plot the following two horizontal lines
% to indicate the regions of the troposphere and stratosphere
plot([0,36089],[.0022,.0022]) % Draw line
plot([36089,65617],[.0018,.0018])
hold off % if you turn hold on be sure to turn it off
echo on
%
%
% Try executing the folowing help commands from the MATLAB
%     command window to get more information.
% help subplot % tells how to arrange many subplots on the same page
% help plot % tells how to change plot symbols.
% help title % how to put a title on a plot or subplot
% help xlabel % how to put a label on the x-axis
% help ylabel % how to put a label on the y-axis
% help legend % how to put legends on a plot with more than one line on it.
% help text % how to put text on a plot at a specified location.
% help ops % tells all the MATLAB operations including
%          logical operations like & and array operations like .*
%
% END OF SCRIPT
%
echo off
```

WHAT FOLLOWS IS THE OUPUT OF THE MATLAB SCRIPT

```
******* Start of the script *****
% These examples show use of vectors and the find function.
help find

 FIND   Find indices of nonzero elements.
    I = FIND(X) returns the indices of the vector X that are
    non-zero. For example, I = FIND(A>100), returns the indices
    of A where A is greater than 100. See RELOP.

    [I,J] = FIND(X) returns the row and column indices of
    the nonzero entries in the matrix X.  This is often used
    with sparse matrices.

    [I,J,V] = FIND(X) also returns a vector containing the
    nonzero entries in X.  Note that find(X) and find(X~=0)
    will produce the same I and J, but the latter will produce
    a V with all 1's.

    See also SPARSE, IND2SUB.


% Example 1
% This example shows
% 1. definition of a vector called x,
% 2. use of the find function,
% 3. and use of the isempty function.
x=[1,2,3,4,4,4,5,5,10,1,2,3,4,12,13,2,2,2]

x =

  Columns 1 through 12

     1     2     3     4     4     4     5     5    10     1     2     3

  Columns 13 through 18

     4    12    13     2     2     2

j=find(x>15)

j =

     []
```

```
isempty(j)

ans =

     1

x(j)=100;
x % This command prints the vector x in the command window.

x =

  Columns 1 through 12

     1     2     3     4     4     4     5     5    10     1     2     3

  Columns 13 through 18

     4    12    13     2     2     2

% There are no elements of x>15 so j is empty and
% isempty returns the value 1 indicating isempty is 'true'.
% MATLAB uses value 1 to indicate the logical result called 'true'.
% In this case it is true that j is empty.


% Example 2
j=find(x>5)

j =

     9    14    15

isempty(j)

ans =

     0

x(j)=100;
x

x =

  Columns 1 through 12

     1     2     3     4     4     4     5     5   100     1     2     3
```

```
  Columns 13 through 18

    4   100   100     2     2     2

% There are 3 elements of x>5 so j has three elements in it and
% isempty returns the value 0 indicating isempty is 'false'.
% MATLAB uses value 0 to indicate the logical result called 'false'.
% In this case it is false that j is empty.
% j contains the elements 9,14,15 indicating that these elements
% of x satisfy the logical test 'Is x>5?'


% Example 3
% This example shows the use of the logical function and (&)
x=[1,2,3,4,4,4,5,5,10,1,2,3,4,12,13,2,2,2]

x =

  Columns 1 through 12

    1     2     3     4     4     4     5     5    10     1     2     3

  Columns 13 through 18

    4    12    13     2     2     2

j=find(x>5 & x<11)

j =

    9

isempty(j)

ans =

    0

x(j)=45;
x

x =

  Columns 1 through 12
```

```
    1     2     3     4     4     4     5     5    45     1     2     3

  Columns 13 through 18

    4    12    13     2     2     2

% There is only one element of x (the 9th one) which is
% greater then 5 and less then 11.


% Example 4 Use of a for loop to compute atmospheric
% density as a function of altitude. The function [rho]=rhofun(h)
% expects its input argument (h) to be a scalar.
echo off
rho=1111 % rho before multiple calls to rhofun

rho =

        1111

h=2222   % h   before multiple calls to rhofun

h =

        2222

echo off
ElapsedTime=etime(clock,t0)% the amount of time to find density

ElapsedTime =

   0.38162600000000

% This is the amount of time it takes to comput the density by the
% for loop method. The array method shown in Example 6 is much faster.
format
rho % rho after multiple calls to rhofun

rho =

        1111

h   % h   after multiple calls to rhofun

h =
```

```
    2222

% Note that the local variables rho and h used in function
% rhofun do not change the values of variables rho ahd h defined
% here in the MATLAB workspace.
%
% In MATLAB it is desirable from a computational speed point of view
% to eliminate for loops. Whenever possible you should try
% to find a way to program using the array aperations discussed above.
% Example 5 below computes the same density data but uses
% array operations instead of the for loop.
echo off
y=[2,2,2,3,3,3,3] %define an array

y =

   2   2   2   3   3   3   3

z0=y.^3  % element by element raising to a power

z0 =

   8   8   8  27  27  27  27

% The above operation is used in the function rhofun2.
z1=y+1   % adding a scalar to each element of a vector

z1 =

   3   3   3   4   4   4   4

%        multiplying a vector by a constant
z2=2*y+1 % and adding 1 to each element of the vector

z2 =

   5   5   5   7   7   7   7


z3=y+z0  %adding two vectors

z3 =
10   10   10   30   30   30   30
z4=z0.*y % element by element multiplication of two vectors
z4 =
16   16   16   81   81   81   81
```

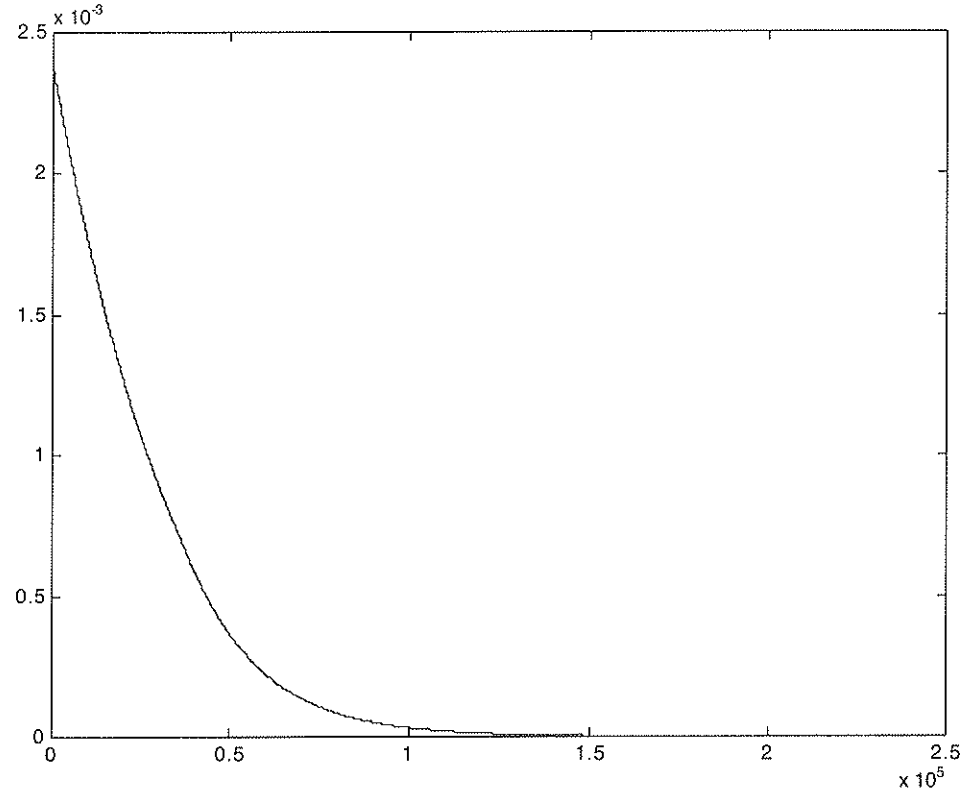
```
z5=z0./y % element by element division of two vectors

z5 =

    4    4    4    9    9    9    9
echo off
rhomat2=rhofun2(x);  % compute vector or air densities
ElapsedTime2=etime(clock,t0) % the amount of time to do this computation
ElapsedTime2 =

   0.01889800000000

% MATLAB has to compile the function rhofun2 to execute
% the above function call. The first time you run this code
% ElapsedTime2 will have the compile time included in it.
% If you run the code a second time, MATLAB will use the already
% compiled function and ElapsedTime 2 will produce the
% time measure we want.
format % print out normal number of decimal places
% Notice how much less computer time this takes compared to the for loop method.
%
% Check out the figures generated by this script
echo off

factor =

   20.1940

It takes 20.194 times as long to use the for loop method.
%
% Try executing the folowing help commands from the MATLAB
%      command window to get more information.
% help subplot % tells how to arrange many subplots on the same page
% help plot % tells how to change plot symbols.
% help title % how to put a title on a plot or subplot
% help xlabel % how to put a label on the x-axis
% help ylabel % how to put a label on the y-axis
% help legend % how to put legends on a plot with more than one line on it.
% help text % how to put text on a plot at a specified location.
% help ops % tells all the MATLAB operations including
%           logical operations like & and array operations like .*
% END OF SCRIPT
echo off
»
```

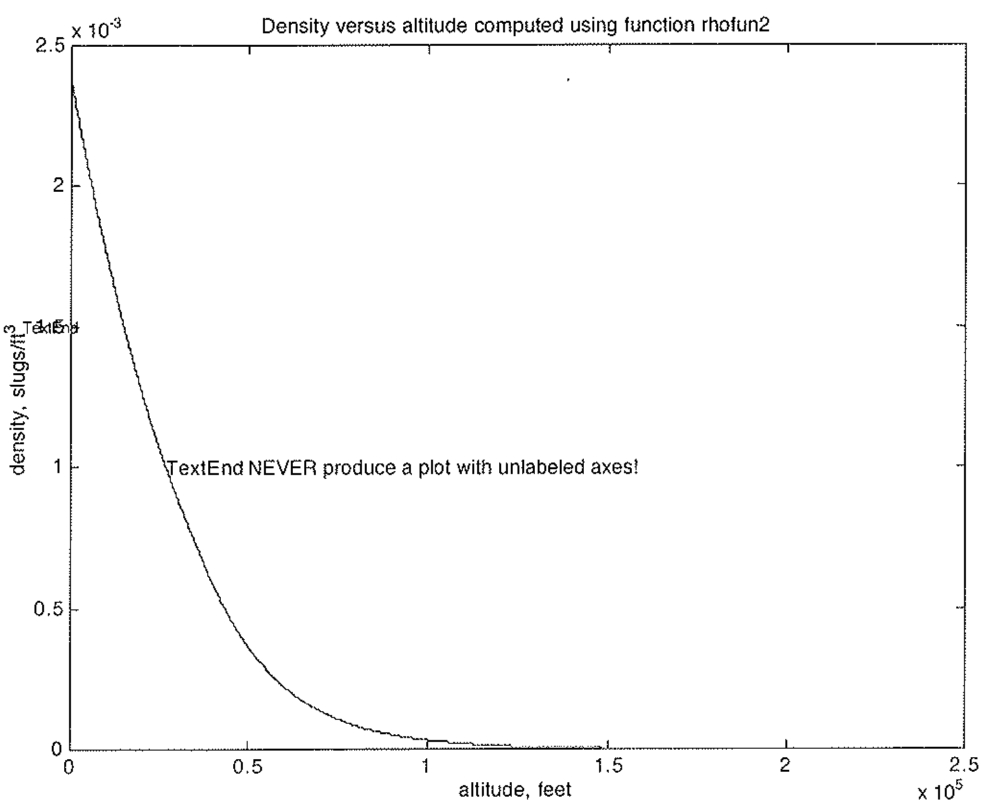
SCRIPT OF THE FUNCTION RHOFUN USED TO COMPUTE THE ATMOSPHERIC DENSITY (RHO) IN THE STANDARD ATMOSPHERE (h IS A SCALAR)

```
function [rho]=rhofun(h)
%  function [rho]=rhofun(h)
%  Standard Atmosphere Computations in English Units
%  for the 1976 standard atmosphere up to 230,000 ft.
%  Author: Ilan Kroo (kroo@leland.stanford.edu)  31 Dec 95
%  Converted to MATLAB by D. Andrisani, 2 Nov 99
%  Scalar input h is geometric altitude in feet
%
%  Output   Units
%  rho      slug/ft^3     (density)
%
%  Because h must be a scalar, the arithmetic operations
%  used below (e.g., ^)are the normal arithmetic
operations.

RHOSL = 0.00237689;   % slug/ft^3
saSigma = 1.0;

   if  h<232940 & h>=167323
      saSigma = ( 0.79899-h/606330)^11.20114;
   end


   if h<167323 & h>=154199
      saSigma = 0.00116533*exp((h-154200)/-25992);
   end

   if h<154199 & h>=104987
      saSigma = (0.857003+h/190115)^-13.20114;
   end

   if h<104987 & h>=65617
      saSigma = (0.978261+h/659515)^-35.16319;
   end

  if h<65617 & h>=36089
      % This is the stratosphere.
      saSigma = 0.297076 *exp((36089-h)/20806);
   end

  if h<36089
      % This is the troposphere.
      saSigma = (1.0-h/145442)^4.255876;
   end
rho = RHOSL * saSigma;   % slug/ft^3
```

SCRIPT OF THE FUNCTION RHOFUN2 USED TO COMPUTE THE ATMOSPHERIC DENSITY (RHO) IN THE STANDARD ATMOSPHERE USING ARRAY OPERATIONS (h IS A VECTOR)

```
function [rho]=rhofun2(h)
%  function [rho]=rhofun2(h)
%  Standard Atmosphere Computations in English Units
%  for the 1976 standard atmosphere up to 230,000 ft.
%  Author: Ilan Kroo (kroo@leland.stanford.edu)  31 Dec 95.
%  Converted to MATLAB by D. Andrisani, 2 Nov 99.
%  Vector input h is geometric altitude in feet.
%  Vector output rho is the same size as vector input h.
%
%  Output   Units
%  rho      slug/ft^3     (density)
%
%  Because h is a vector, the arithmetic operations
%  used below (e.g., .^)are the array operations.
%  The find command is used to determine which region
%  in the atmosphere the altitude data is in, and then
%  the correct equation can be used for each atmospheric region.
RHOSL = 0.00237689;   % slug/ft^3
saSigma =NaN*zeros(size(h)) ; % Initialize saSigma to the
% correct array size and fill the array with elements
% that are not numbers. NaN is the indicator that MATLAB
% uses for 'not a number'. In this application we fill this
% array initially with NaN so that the output
% array rho will contain NaN corresponding to the
% altitudes which are out of range (h<232940). MATLAB will not
% allow computations to be done on NaN array elements.
   j1=find(h<232940 & h>=167323);
   if  isempty(j1)
   else
      saSigma(j1) = ( 0.79899-h(j1)/606330).^11.20114;
   end
  j2=find(h<167323 & h>=154199);
   if   isempty(j2)
   else saSigma(j2) = 0.00116533*exp((h(j2)-154200)/-25992);
   end
  j3=find(h<154199 & h>=104987);
   if isempty(j3)
   else   saSigma(j3) = (0.857003+h(j3)/190115).^-13.20114;
   end
  j4=find(h<104987 & h>=65617);
   if isempty(j4)
   else  saSigma(j4) = (0.978261+h(j4)/659515).^-35.16319;
   end
  j5=find(h<65617 & h>=36089);
%  This is the stratosphere.
   if isempty(j5)
   else   saSigma(j5) = 0.297076 *exp((36089-h(j5))/20806);
   end
  j6=find(h<36089);
%  This is the troposphere.
   if isempty(j6)
   else   saSigma(j6) = (1.0-h(j6)/145442).^4.255876;
   end
 rho = RHOSL * saSigma;  % slug/ft^3
```


x 10^-3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
x 10^5


Density versus altitude computed using function rhofun2
x 10^-3
2.5
2
1.5
1
0.5
0
density, slugs/ft^3
TextEnd
TextEnd NEVER produce a plot with unlabeled axes!
0
0.5
1
1.5
2
2.5
altitude, feet
x 10^5