

AAE 190

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 AAE190 COURSE WEB SITE.

```
% A&AE190 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 and 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 operations 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 fourth 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 following 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 than 5 and less than 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 compute 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 and 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 operations 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 following 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

```



