

4.3.2 The `fprintf` Command

The `fprintf` command can be used to display output (text and data) on the screen or to save it to a file. With this command (unlike with the `disp` command) the output can be formatted. For example, text and numerical values of variables can be intermixed and displayed in the same line. In addition, the format of the numbers can be controlled.

With many available options, the `fprintf` command can be long and complicated. To avoid confusion, the command is presented gradually. First, this section shows how to use the command to display text messages, then how to mix numerical data and text, next how to format the display of numbers, and finally how to save the output to a file.

Using the `fprintf` command to display text:

To display text, the `fprintf` command has the form:

```
fprintf('text typed in as a string')
```

For example:

```
fprintf('The problem, as entered, has no solution. Please check the  
input data.')
```

If this line is part of a script file, then when the line is executed, the following is displayed in the Command Window:

```
The problem, as entered, has no solution. Please check the input data.
```

With the `fprintf` command it is possible to start a new line in the middle of the string. This is done by inserting `\n` before the character that will start the new line. For example, inserting `\n` after the first sentence in the previous example gives:

```
fprintf('The problem, as entered, has no solution.\nPlease
check the input data.')
```

When this line executes, the display in the Command Window is:

```
The problem, as entered, has no solution.
Please check the input data.
```

The `\n` is called an escape character. It is used to control the display. Other escape characters that can be inserted within the string are:

```
\b    Backspace.
\t    Horizontal tab.
```

When a program has more than one `fprintf` command, the display generated is continuous (the `fprintf` command does not automatically start a new line). This is true even if there are other commands between the `fprintf` commands. An example is the following script file:

```
fprintf('The problem, as entered, has no solution. Please check the
input data.')
```

```
x = 6; d = 19 + 5*x;
```

```
fprintf('Try to run the program later.')
```

```
y = d + x;
```

```
fprintf('Use different input values.')
```

When this file is executed the display in the Command Window is:

```
The problem, as entered, has no solution. Please check the
input data.Try to run the program later.Use different input
values.
```

To start a new line with the `fprintf` command, `\n` must be typed at the start of the string.

Using the `fprintf` command to display a mix of text and numerical data:

To display a mix of text and a number (value of a variable), the `fprintf` command has the form:

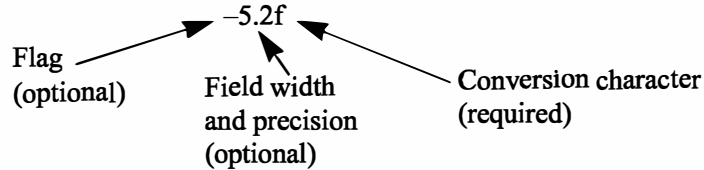
```
fprintf('text as string %-5.2f additional text',
        variable_name)
```

The % sign marks the spot where the number is inserted within the text.

Formatting elements (define the format of the number).

The name of the variable whose value is displayed.

The formatting elements are:



The flag, which is optional, can be one of the following three characters:

<u>Character used for flag</u>	<u>Description</u>
- (minus sign)	Left-justifies the number within the field.
+ (plus sign)	Prints a sign character (+ or -) in front of the number.
0 (zero)	Adds zeros if the number is shorter than the field.

The field width and precision (5.2 in the previous example) are optional. The first number (5 in the example) is the field width, which specifies the minimum number of digits in the display. If the number to be displayed is shorter than the field width, spaces or zeros are added in front of the number. The precision is the second number (2 in the example). It specifies the number of digits to be displayed to the right of the decimal point.

The last element in the formatting elements, which is required, is the conversion character, which specifies the notation in which the number is displayed. Some of the common notations are:

e	Exponential notation using lowercase e (e.g., 1.709098e+001).
E	Exponential notation using uppercase E (e.g., 1.709098E+001).
f	Fixed-point notation (e.g., 17.090980).
g	The shorter of e or f notations.
G	The shorter of E or f notations.
i	Integer.

Information about additional notation is available in the help menu of MATLAB. As an example, the `fprintf` command with a mix of text and a number is used in the script file that calculates the average points scored in three games.

```

% This script file calculates the average points scored in three games.
% The values are assigned to the variables by using the input command.
% The fprintf command is used to display the output.
game(1) = input('Enter the points scored in the first game ');
game(2) = input('Enter the points scored in the second game ');
game(3) = input('Enter the points scored in the third game ');
ave_points = mean(game);
  
```

```
fprintf('An average of %f points was scored in the three games.',ave_points)
```

Notice that, besides using the `fprintf` command, this file differs from the ones shown earlier in the chapter in that the scores are stored in the first three elements of a vector named `game`, and the average of the scores is calculated by using the `mean` function. The Command Window where the script file above (saved as `Chapter4Example6`) was run is shown below.

```
>> Chapter4Example6
Enter the points scored in the first game    75
Enter the points scored in the second game   60
Enter the points scored in the third game    81
An average of 72.000000 points was scored in the three games.
>>
```

With the `fprintf` command it is possible to insert more than one number (value of a variable) within the text. This is done by typing `%g` (or `%` followed by any formatting elements) at the places in the text where the numbers are to be inserted. Then, after the string argument of the command (following the comma), the names of the variables are typed in the order in which they are inserted in the text. In general the command looks like:

```
fprintf('..text...%g...%g...%f...',variable1,variable2,variable3)
```

An example is shown in the following script file:

```
% This program calculates the distance a projectile flies,
% given its initial velocity and the angle at which it is shot.
% the fprintf command is used to display a mix of text and numbers.

v=1584; % Initial velocity (km/h)
theta=30; % Angle (degrees)
vms=v*1000/3600;
t=vms*sind(30)/9.81;
d=vms*cosd(30)*2*t/1000;
```

```
fprintf('A projectile shot at %3.2f degrees with a velocity
of %4.2f km/h will travel a distance of %g km.\n',theta,v,d)
```

When this script file (saved as Chapter4Example7) is executed, the display in the Command Window is:

```
>> Chapter4Example7
A projectile shot at 30.00 degrees with a velocity of
1584.00 km/h will travel a distance of 17.091 km.
>>
```

Additional remarks about the fprintf command:

- To place a single quotation mark in the displayed text, type two single quotation marks in the string inside the command.
- The `fprintf` command is vectorized. This means that when a variable that is a vector or a matrix is included in the command, the command repeats itself until all the elements are displayed. If the variable is a matrix, the data is used column by column.

For example, the script file below creates a 2×5 matrix `T` in which the first row contains the numbers 1 through 5, and the second row shows the corresponding square roots.

```
x=1:5;
y=sqrt(x);
T=[x; y]
fprintf('If the number is: %i, its square root is: %f\n',T)
```

The `fprintf` command displays two numbers from `T` in every line.

When this script file is executed, the display in the Command Window is:

```
T =
    1.0000    2.0000    3.0000    4.0000    5.0000
    1.0000    1.4142    1.7321    2.0000    2.2361
If the number is: 1, its square root is: 1.000000
If the number is: 2, its square root is: 1.414214
If the number is: 3, its square root is: 1.732051
If the number is: 4, its square root is: 2.000000
If the number is: 5, its square root is: 2.236068
```

The 2×5 matrix `T`.

The `fprintf` command repeats five times, using the numbers from the matrix `T` column after column.

Using the `fprintf` command to save output to a file:

In addition to displaying output in the Command Window, the `fprintf` command can be used for writing the output to a file when it is necessary to save the output. The data that is saved can subsequently be displayed or used in MATLAB and in other applications.

Writing output to a file requires three steps:

- a) Opening a file using the `fopen` command.
- b) Writing the output to the open file using the `fprintf` command.
- c) Closing the file using the `fclose` command.

Step a:

Before data can be written to a file, the file must be opened. This is done with the `fopen` command, which creates a new file or opens an existing file. The `fopen` command has the form:

```
fid = fopen('file_name', 'permission')
```

`fid` is a variable called the file identifier. A scalar value is assigned to `fid` when `fopen` is executed. The file name is written (including its extension) within single quotes as a string. The permission is a code (also written as a string) that tells how the file is opened. Some of the more common permission codes are:


- 'r' Open file for reading (default).
- 'w' Open file for writing. If the file already exists, its content is deleted. If the file does not exist, a new file is created.
- 'a' Same as 'w', except that if the file exists the written data is appended to the end of the file.
- 'r+' Open (do not create) file for reading and writing.
- 'w+' Open file for reading and writing. If the file already exists, its content is deleted. If the file does not exist, a new file is created.
- 'a+' Same as 'w+', except that if the file exists the written data is appended to the end of the file.

If a permission code is not included in the command, the file opens with the default code 'r'. Additional permission codes are described in the help menu.

Step b:

Once the file is open, the `fprintf` command can be used to write output to the file. The `fprintf` command is used in exactly the same way as it is used to display output in the Command Window, except that the variable `fid` is inserted inside the command. The `fprintf` command then has the form:

```
fprintf(fid, 'text %-5.2f additional text', variable_name)
```

 `fid` is added to the `fprintf` command.

Step c:

When the writing of data to the file is complete, the file is closed using the `fclose` command. The `fclose` command has the form:

```
fclose(fid)
```

Additional notes on using the `fprintf` command for saving output to a file:

- The created file is saved in the current directory.
- It is possible to use the `fprintf` command to write to several different files. This is done by first opening the files, assigning a different `fid` to each (e.g. `fid1`, `fid2`, `fid3`, etc.), and then using the `fid` of a specific file in the `fprintf` command to write to that file.

An example of using `fprintf` commands for saving output to two files is shown in the following script file. The program in the file generates two unit conversion tables. One table converts velocity units from miles per hour to kilometers per hour, and the other table converts force units from pounds to newtons. Each conversion table is saved to a different text file (extension `.txt`).

```
% Script file in which fprintf is used to write output to files.
% Two conversion tables are created and saved to two different files.
% One converts mi/h to km/h, the other converts lb to N.

clear all
Vmph=10:10:100;           Creating a vector of velocities in mi/h.
Vkmh=Vmph.*1.609;        Converting mph to km/h.
TBL1=[Vmph; Vkmh];      Creating a table (matrix) with two rows.
Flb=200:200:2000;       Creating a vector of forces in lb.
FN=Flb.*4.448;          Converting lb to N.
TBL2=[Flb; FN];         Creating a table (matrix) with two rows.
fid1=fopen('VmphtoVkm.txt','w'); Open a .txt file named VmphtoVkm.
fid2=fopen('FlbtoFN.txt','w'); Open a .txt file named FlbtoFN.
fprintf(fid1,'Velocity Conversion Table\n \n'); Writing a title and an empty line to the file fid1.
fprintf(fid1,'      mi/h          km/h      \n'); Writing two column headings to the file fid1.
fprintf(fid1,'      %8.2f      %8.2f\n',TBL1); Writing the data from the variable TBL1 to the file fid1.
```

```
fprintf(fid2,'Force Conversion Table\n \n');
fprintf(fid2,'      Pounds      Newtons  \n');
fprintf(fid2,'    %8.2f      %8.2f\n',TBL2);
fclose(fid1);
fclose(fid2);
```

Writing the force conversion table (data in variable TBL2) to the file fid2.

Closing the files fid1 and fid2.

When the script file above is executed two new .txt files, named VmphtoVkm and FlbtoFN, are created and saved in the current directory. These files can be opened with any application that can read .txt files. Figures 4-3 and 4-4 show how the two files appear when they are opened with Microsoft Word.

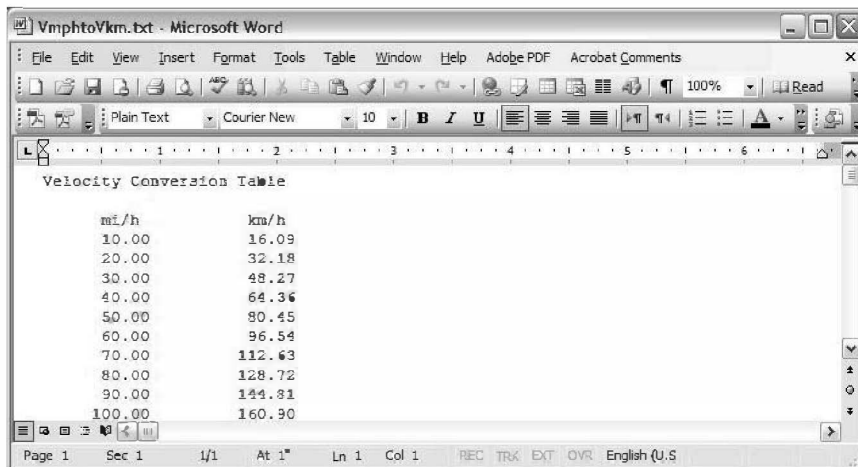


Figure 4-3: The VmphtoVkm.txt file opened in Word.

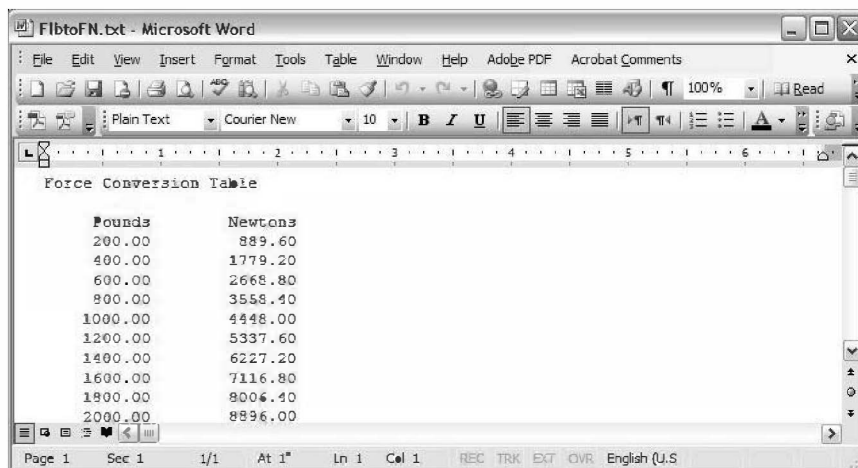


Figure 4-4: The FlbtoFN.txt file opened in Word.

4.4 THE save AND load COMMANDS

The `save` and `load` commands are most useful for saving and retrieving data for use in MATLAB. The `save` command can be used for saving the variables that are currently in the workspace, and the `load` command is used for retrieving variables that have been previously saved, to the workspace. The workspace can be saved when MATLAB is used in one type of platform (e.g., PC), and retrieved for use in MATLAB in another platform (e.g., Mac). The `save` and `load` commands can also be used for exchanging data with applications outside MATLAB. Additional commands that can be used for this purpose are presented in Section 4.5.

4.4.1 The save Command

The `save` command is used for saving the variables (all or some of them) that are stored in the workspace. The two simplest forms of the `save` command are:

```
save file_name
```

and

```
save('file_name')
```

When either one of these commands is executed, all of the variables currently in the workspace are saved in a file named `file_name.mat` that is created in the current directory. In `mat` files, which are written in a binary format, each variable preserves its name, type, size, and value. These files cannot be read by other applications. The `save` command can also be used for saving only some of the variables that are in the workspace. For example, to save two variables named `var1` and `var2`, the command is:

```
save file_name var1 var2
```

or

```
save('file_name', 'var1', 'var2')
```

The `save` command can also be used for saving in ASCII format, which can be read by applications outside MATLAB. Saving in ASCII format is done by adding the argument `-ascii` in the command (for example, `save file_name -ascii`). In the ASCII format the variable's name, type, and size are not preserved. The data is saved as characters separated by spaces but without the variable names. For example, the following shows how two variables (a 1×4 vector and a 2×3 matrix) are defined in the Command Window and then saved in ASCII format to a file named `DatSavAscii`:

```
>> V=[3 16 -4 7.3];
```

```
Create a 1 × 4 vector V.
```

```
>> A=[6 -2.1 15.5; -6.1 8 11];
```

```
Create a 2 × 3 matrix A.
```

```
>> save -ascii DatSavAscii
```

```
Save variables to a file named DatSavAscii.
```

Once saved, the file can be opened by any application that can read ASCII files. For example, Figure 4-5 shows the data when the file is opened with Notepad.

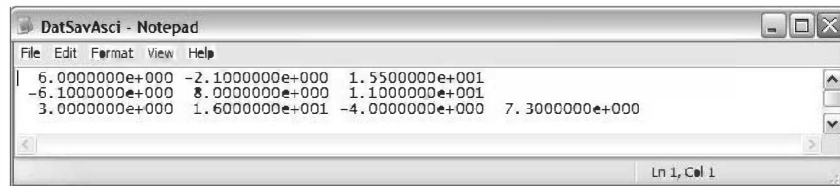


Figure 4-5: Data saved in ASCII format.

Note that the file does not include the names of the variables, just the numerical values of the variables (first A and then V) are listed.

4.4.2 The load Command

The load command can be used for retrieving variables that were saved with the save command back to the workspace, and for importing data that was created with other applications and saved in ASCII format or in text (.txt) files. Variables that were saved with the save command in .mat files can be retrieved with the command:

`load file_name` or `load('file_name')`

When the command is executed, all the variables in the file (with the name, type, size, and values as were saved) are added (loaded back) to the workspace. If the workspace already has a variable with the same name as a variable that is retrieved with the load command, then the variable that is retrieved replaces the existing variable. The load command can also be used for retrieving only some of the variables that are in the saved .mat file. For example, to retrieve two variables named var1 and var2, the command is:

`load file_name var1 var2` or `load('file_name', 'var1', 'var2')`

The load command can also be used to import data that is saved in ASCII or text (.txt) to the workspace. This is possible, however, only if the data in the file is in the form of a variable in MATLAB. Thus, the file can have one number (scalar), a row or a column of numbers (vector), or rows with the same number of numbers in each (matrix). For example, the data shown in Figure 4-5 cannot be loaded with the load command (even though it was saved in ASCII format with the save command), because the number of elements is not the same in all rows. (Recall that this file was created by saving two different variables.)

When data is loaded from an ASCII or text file into the workspace, it has to be assigned to a variable name. Data in ASCII format can be loaded with either of the following two forms of the load command:

`load file_name` or `VarName=load('file_name')`

If the data is in a text file, the extension .txt has to be added to the file name. The form of the load command is then:

`load file_name.txt` or `VarName=load('file_name.txt')`

In the first form of the command the data is assigned to a variable that has the name of the file. In the second form the data is assigned to a variable named VarName.

For example, the data shown in Figure 4-6 (a 3×2 matrix) is typed in Notepad, and then saved as DataFromText.txt.



Figure 4-6: Data saved as .txt file.

Next, two forms of the load command are used to import the data in the text file to the Workspace of MATLAB. In the first command the data is assigned to a variable named DfT. In the second command the data is automatically assigned to a variable named DataFromText, which is the name of the text file where the data was saved.

```
>> DfT=load('DataFromText.txt')
```

```
DfT =
    56.0000    -4.2000
     3.0000     7.5000
    -1.6000    198.0000
```

Load the file DataFromText and assign the loaded data to the variable Dft.

```
>> load DataFromText.txt
```

Use the load command with the file DataFromText.

```
>> DataFromText
DataFromText =
    56.0000    -4.2000
     3.0000     7.5000
    -1.6000    198.0000
```

The data is assigned to a variable named DataFromText.

Importing data to (or exporting from) other applications can also be done, with MATLAB commands that are presented in the next section.

4.5 IMPORTING AND EXPORTING DATA

MATLAB is often used for analyzing data that was recorded in experiments or generated by other computer programs. This can be done by first importing the data into MATLAB. Similarly, data that is produced by MATLAB sometimes needs to be transferred to other computer applications. There are various types of data (numerical, text, audio, graphics, and images). This section describes only how to import and export numerical data, which is probably the most common type of data that needs to be transferred by new users of MATLAB. For other types of data transfer, look in the Help Window under File I/O.

Importing data can be done either by using commands or by using the Import Wizard. Commands are useful when the format of the data being imported is known. MATLAB has several commands that can be used for importing various types of data. Importing commands can also be included in a script file such that the data is imported when the script is executed. The Import Wizard is useful when the format of the data (or the command that is applicable for importing the data) is not known. The Import Wizard determines the format of the data and automatically imports it.

4.5.1 Commands for Importing and Exporting Data

This section describes—in detail—how to transfer data into and out of Excel spreadsheets. Microsoft Excel is commonly used for storing data, and Excel is compatible with many data recording devices and computer applications. Many people are also capable of importing and exporting various data formats into and from Excel. MATLAB also has commands for transferring data directly to and from formats such as csv and ASCII, as well as to the spreadsheet program Lotus 123. Details of these and many other commands can be found in the Help Window under File I/O

Importing and exporting data into and from Excel:

Importing data from Excel is done with the `xlsread` command. When the command is executed, the data from the spreadsheet is assigned as an array to a variable. The simplest form of the `xlsread` command is:

```
variable_name=xlsread('filename')
```

- `'filename'` (typed as a string) is the name of the Excel file. The directory of the Excel file must be either the current directory or listed in the search path.
- If the Excel file has more than one sheet, the data will be imported from the first sheet.

When an Excel file has several sheets, the `xlsread` command can be used to import data from a specified sheet. The form of the command is then:

```
variable_name = xlsread('filename', 'sheet_name')
```

- The name of the sheet is typed as a string.

Another option is to import only a portion of the data that is in the spreadsheet. This is done by typing an additional argument in the command:

```
variable_name = xlsread('filename', 'sheet_name', 'range')
```

- The `'range'` (typed as a string) is a rectangular region of the spreadsheet defined by the addresses (in Excel notation) of the cells at opposite corners of the region. For example, `'C2:E5'` is a 4×3 region of rows 2, 3, 4, and 5 and columns *C*, *D*, and *E*.

Exporting data from MATLAB to an Excel spreadsheet is done by using the `xlswrite` command. The simplest form of the command is:

```
xlswrite('filename', variable_name)
```

- `'filename'` (typed as a string) is the name of the Excel file to which the data is exported. The file must be in the current directory. If the file does not exist, a new Excel file with the specified name will be created.
- `variable_name` is the name of the variable in MATLAB with the assigned data that is being exported.
- The arguments `'sheet_name'` and `'range'` can be added to the `xlswrite` command to export to a specified sheet and to a specified range of cells, respectively.

As an example, the data from the Excel spreadsheet shown in Figure 4-7 is imported into MATLAB by using the `xlsread` command.

	A	B	C	D	E	F	G	H	I	J	K
1	11	2	34	14	-6	0	8				
2	15	6	-20	8	0.56	33	5				
3	0.9	10	3	12	-25	-0.1	4				
4	56	9	1	-0.555	17	6	-30				

Figure 4-7: Excel spreadsheet with data.

The spreadsheet is saved in a file named `TestData1` in a disk in drive A. After the Current Directory is changed to drive A, the data is imported into MATLAB by assigning it to the variable `DATA`:

```
>> DATA = xlsread('TestData1')
DATA =
    11.0000    2.0000   34.0000   14.0000   -6.0000         0    8.0000
    15.0000    6.0000  -20.0000    8.0000    0.5600   33.0000    5.0000
     0.9000   10.0000    3.0000   12.0000  -25.0000   -0.1000    4.0000
    55.0000    9.0000    1.0000   -0.5550   17.0000    6.0000  -30.0000
```

4.5.2 Using the Import Wizard

Using the Import Wizard is probably the easiest way to import data into MATLAB since the user does not have to know, or to specify, the format of the data. The Import Wizard is activated by selecting **Import Data** in the **File** menu of the Command Window. (It can also be started by typing the command `uiimport`.) The Import Wizard starts by displaying a file selection box that shows all the data files recognized by the Wizard. The user then selects the file that contains the data to be imported, and clicks **Open**. The Import Wizard opens the file and displays a portion of the data in a preview box so that the user can verify that the data is the correct choice. The Import Wizard tries to process the data, and if the wizard is successful, it displays the variables it has created with a portion of the data. The user clicks **next** and the wizard shows the Column Separator that was used. If the variable has the correct data, the user can proceed with the wizard (click **next**); otherwise the user can choose a different Column Separator. In the next window the wizard shows the name and size of the variable to be created in MATLAB. (When the data is all numerical, the variable in MATLAB has the same name as the file from which the data was imported.) When the wizard ends (click **finish**), the data is imported to MATLAB.

As an example, the Import Wizard is used to import numerical ASCII data saved in a `.txt` file. The data saved with the file name `TestData2` is shown in Figure 4-8.

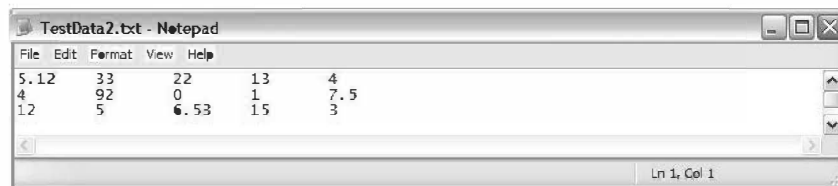


Figure 4-8: Numerical ASCII data.

The display of the Import Wizard during the import process for the TestData2 file is shown in Figures 4-9 and 4-10. Figure 4-10 shows that the name of the variable in MATLAB is `TestData2` and its size is 3×5 .

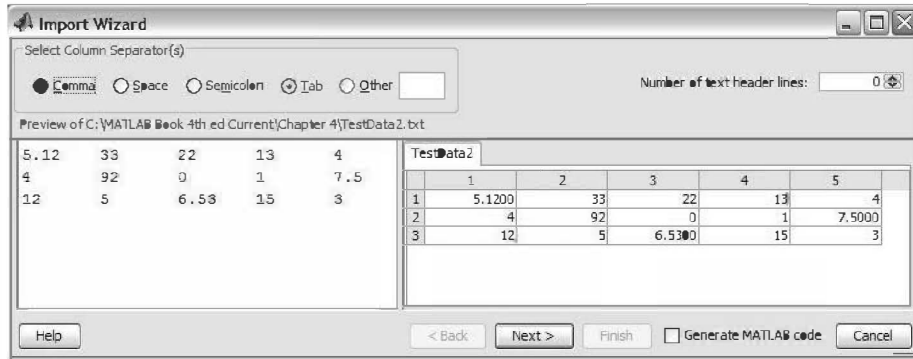


Figure 4-9: Import Wizard, first display.

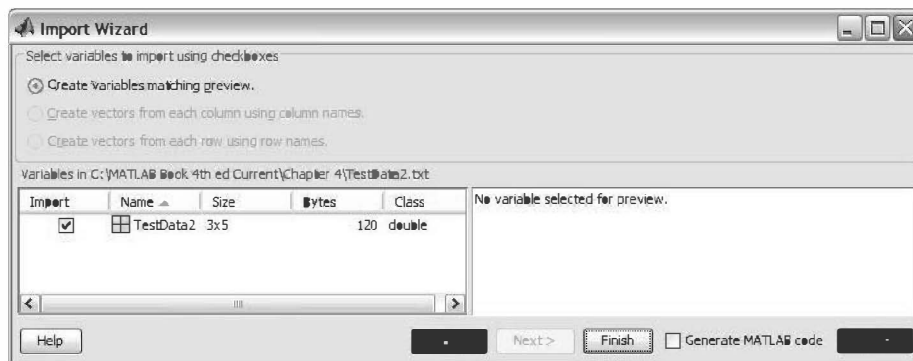


Figure 4-10: Import Wizard, second display.

In the Command Window of MATLAB, the imported data can be displayed by typing the name of the variable.

```
>> TestData2
TestData2 =
    5.1200    33.0000    22.0000    13.0000     4.0000
    4.0000    92.0000         0         1.0000    7.5000
   12.0000     5.0000    6.5300    15.0000     3.0000
```

4.6 EXAMPLES OF MATLAB APPLICATIONS

Sample Problem 4-1: Height and surface area of a silo

A cylindrical silo with radius r has a spherical cap roof with radius R . The height of the cylindrical portion is H . Write a program in a script file that determines the height H for given values of r , R , and the volume V . In addition, the program calculates the surface area of the silo.

Use the program to calculate the height and surface area of a silo with $r = 30$ ft, $R = 45$ ft, and a volume of $200,000$ ft³. Assign values for r , R , and V in the Command Window.

Solution

The total volume of the silo is obtained by adding the volume of the cylindrical part and the volume of the spherical cap. The volume of the cylinder is given by

$$V_{cyl} = \pi r^2 H$$

and the volume of the spherical cap is given by:

$$V_{cap} = \frac{1}{3} \pi h^2 (3R - h)$$

where $h = R - R \cos \theta = R(1 - \cos \theta)$,

and θ is calculated from $\sin \theta = \frac{r}{R}$.

Using the equations above, the height, H , of the cylindrical part can be expressed by

$$H = \frac{V - V_{cap}}{\pi r^2}$$

The surface area of the silo is obtained by adding the surface areas of the cylindrical part and the spherical cap.

$$S = S_{cyl} + S_{cap} = 2\pi r H + 2\pi R h$$

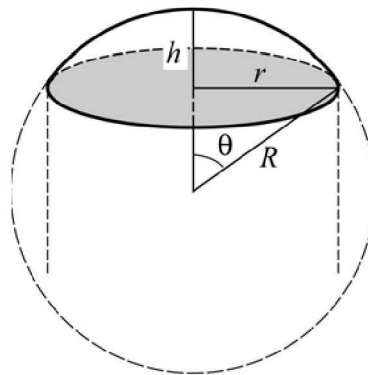
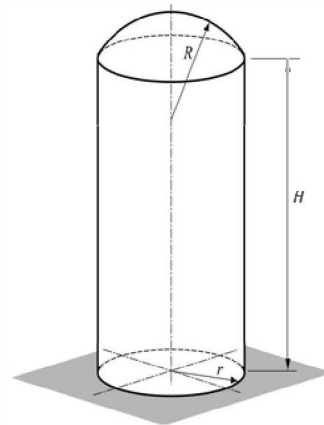
A program in a script file that solves the problem is presented below:

```
theta=asin(r/R);
h=R*(1-cos(theta));
Vcap=pi*h^2*(3*R-h)/3;
```

Calculating θ .

Calculating h .

Calculating the volume of the cap.



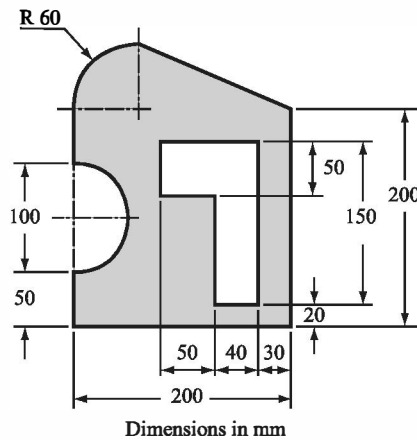

```
H=(V-Vcap)/(pi*r^2);
S=2*pi*(r*H + R*h);
fprintf('The height H is: %f ft.',H)
fprintf('\nThe surface area of the silo is: %f square ft.',S)
```

The Command Window where the script file, named silo, was executed is:

```
>> r=30; R=45; V=200000;
>> silo
The height H is: 64.727400 ft.
The surface area of the silo is: 15440.777753 square ft.
```

Sample Problem 4-2: Centroid of a composite area

Write a program in a script file that calculates the coordinates of the centroid of a composite area. (A composite area can easily be divided into sections whose centroids are known.) The user needs to divide the area into sections and know the coordinates of the centroid (two numbers) and the area of each section (one number). When the script file is executed, it asks the user to enter the three numbers as a row in a matrix. The user enters as many rows as there are sections. A section that represents a hole is taken to have a negative area. For output, the program displays the coordinates of the centroid of the composite area. Use the program to calculate the centroid of the area shown in the figure.

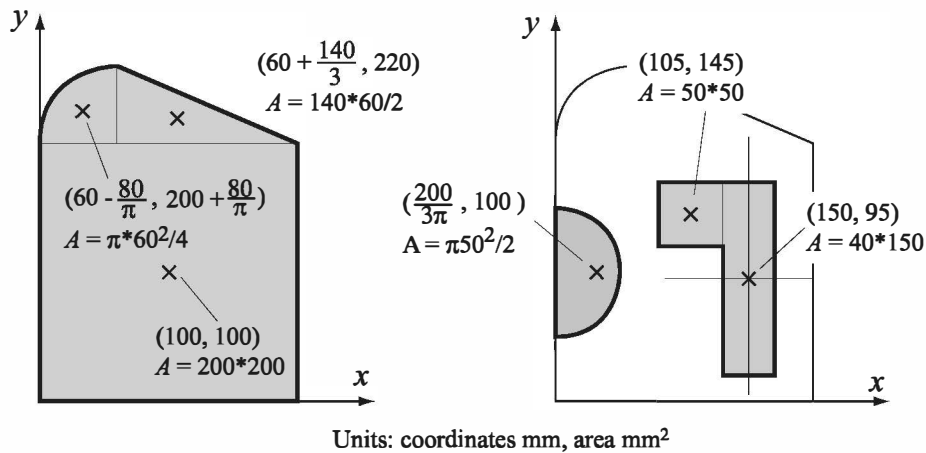


Solution

The area is divided into six sections as shown in the following figure. The total area is calculated by adding the three sections on the left and subtracting the three sections on the right. The location and coordinates of the centroid of each section are marked in the figure, as well as the area of each section.

The coordinates \bar{X} and \bar{Y} of the centroid of the total area are given by $\bar{X} = \frac{\sum A\bar{x}}{\sum A}$ and $\bar{Y} = \frac{\sum A\bar{y}}{\sum A}$, where \bar{x} , \bar{y} , and A are the coordinates of the centroid and area of each section, respectively.

A script file with a program for calculating the coordinates of the centroid of a composite area is provided below.



```

% The program calculates the coordinates of the centroid
% of a composite area.
clear C xs ys As
C=input('Enter a matrix in which each row has three elements.\nIn each row enter the x and y coordinates of the centroid and the area of a section.\n');
xs=C(:,1)';
ys=C(:,2)';
As=C(:,3)';
A=sum(As);
x=sum(As.*xs)/A;
y=sum(As.*ys)/A;
fprintf('The coordinates of the centroid are: ( %f, %f )\n',x,y)

```

Creating a row vector for the x coordinate of each section (first column of C).

Creating a row vector for the y coordinate of each section (second column of C).

Creating a row vector for the area of each section (third column of C).

Calculating the total area.

Calculating the coordinates of the centroid of the composite area.

The script file was saved with the name Centroid. The following shows the Command Window where the script file was executed.

```

>> Centroid
Enter a matrix in which each row has three elements.
In each row enter the x and y coordinates of the centroid
and the area of a section.

```

```
[100 100 200*200
60-80/pi 200+80/pi pi*60^2/4
60+140/3 220 140*60/2
200/(3*pi) 100 -pi*50^2/2
105 145 -50*50
150 95 -40*150]
```

Entering the data for matrix C.
Each row has three elements: the
 x , y , and A of a section.

The coordinates of the centroid are: (85.387547 , 131.211809)

Sample Problem 4-3: Voltage divider

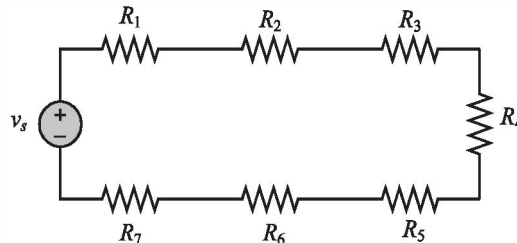
When several resistors are connected in an electrical circuit in series, the voltage across each of them is given by the voltage divider rule:

$$v_n = \frac{R_n}{R_{eq}} v_s$$

where v_n and R_n are the voltage across resistor n and its resistance, respectively, $R_{eq} = \Sigma R_n$ is the equivalent resistance, and v_s is the source voltage. The power dissipated in each resistor is given by:

$$P_n = \frac{R_n}{R_{eq}^2} v_s^2$$

The figure below shows a circuit with seven resistors connected in series.



Write a program in a script file that calculates the voltage across each resistor, and the power dissipated in each resistor, in a circuit that has resistors connected in series. When the script file is executed, it requests that the user first enter the source voltage and then to enter the resistances of the resistors in a vector. The program displays a table with the resistance listed in the first column, the voltage across the resistor in the second column, and the power dissipated in the resistor in the third column. Following the table, the program displays the current in the circuit and the total power.

Execute the file and enter the following data for v_s and the R 's.

$v_s = 24\text{V}$, $R_1 = 20\Omega$, $R_2 = 14\Omega$, $R_3 = 12\Omega$, $R_4 = 18\Omega$, $R_5 = 8\Omega$,
 $R_6 = 15\Omega$, $R_7 = 10\Omega$.

Solution

A script file that solves the problem is shown below.

```
% The program calculates the voltage across each resistor
% in a circuit that has resistors connected in series.
vs=input('Please enter the source voltage ');
Rn=input('Enter the values of the resistors as elements in a
row vector\n');
Req=sum(Rn);
vn=Rn*vs/Req;
Pn=Rn*vs^2/Req^2;
i = vs/Req;
Ptotal = vs*i;
Table = [Rn', vn', Pn'];
disp(' ')
disp(' Resistance Voltage Power')
disp(' (Ohms) (Volts) (Watts)')
disp(' ')
disp(Table)
disp(' ')
fprintf('The current in the circuit is %f Amps.',i)
fprintf('\nThe total power dissipated in the circuit is %f
Watts.',Ptotal)
```

Calculate the equivalent resistance.

Apply the voltage divider rule.

Calculate the power in each resistor.

Calculate the current in the circuit.

Calculate the total power in the circuit.

Create a variable table with the vectors Rn, vn, and Pn as columns.

Display headings for the columns.

Display an empty line.

Display the variable Table.

The Command Window where the script file was executed is:

```
>> VoltageDivider
Please enter the source voltage 24
Enter the value of the resistors as elements in a row vector
[20 14 12 18 8 15 10]
Resistance Voltage Power
(Ohms) (Volts) (Watts)
20.0000 4.9485 1.2244
14.0000 3.4639 0.8571
12.0000 2.9691 0.7346
18.0000 4.4536 1.1019
8.0000 1.9794 0.4897
```

Name of the script file.

Voltage entered by the user.

Resistor values entered as a vector.

```

15.0000    3.7113    0.9183
10.0000    2.4742    0.6122

```

The current in the circuit is 0.247423 Amps.

The total power dissipated in the circuit is 5.938144 Watts.

4.7 PROBLEMS

Solve the following problems by first writing a program in a script file and then executing the program.

1. The Heat Index HI , calculated from the air temperature and relative humidity, is the apparent temperature felt by the body. An equation used by the National Weather Service for calculating the HI is given by:

$$HI = -42.379 + 2.04901523T + 10.14333127R - 0.22475541R - 6.83783 \times 10^{-3}T^2 - 5.481717 \times 10^{-2}R^2 + 1.22874 \times 10^{-3}T^2R + 8.5282 \times 10^{-4}TR^2 - 1.99 \times 10^{-6}T^2R^2$$

where T is the temperature in degrees F and R is the relative humidity in integer percentage. Write a MATLAB program in a script file that calculates HI . For input the program asks the user to enter values for T and R . For output the program displays the message: "The Heat Index temperature is: XX," where XX is the value of the heat index rounded to the nearest integer. Execute the program entering $T = 90^\circ\text{F}$ and $R = 90\%$.

2. The monthly saving P that has to be deposit in a saving account that pays an annual interest rate of r in order to save a total amount of F in N years can be calculated by the formula:

$$P = \frac{F(r/12)}{(1 + r/12)^{12N} - 1}$$

Calculate the monthly saving that has to be deposit in order to save \$100,000 in 5, 6, 7, 8, 9, and 10 years if the annual interest rate is 4.35%. Display the results in a two-column table where the first column is the number of years and the second column is the monthly deposit.

3. The growth of some bacteria populations can be described by

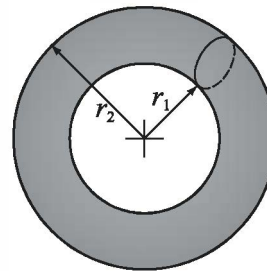
$$N = N_0e^{kt}$$

where N is the number of individuals at time t , N_0 is the number at time $t = 0$, and k is a constant. Assuming the number of bacteria doubles every 40 minutes, determine the number of bacteria every two hours for 24 hours starting from an initial single bacterium.

4. The volume V and the surface area S of a torus-shaped water tube are given by:

$$V = \frac{1}{4}\pi^2(r_1 + r_2)(r_2 - r_1)^2 \quad \text{and} \quad S = \pi^2(r_2^2 - r_1^2)$$

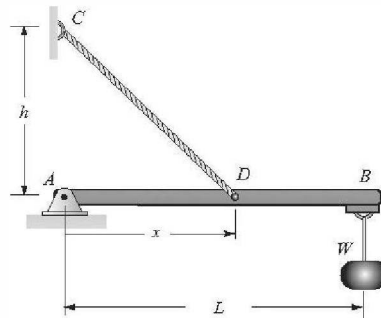
If $r_1 = 0.7r_2$, determine V and S for $r_2 = 12, 16, 20, 24,$ and 28 in. Display the results in a four-column table where the first column is r_2 , the second r_1 , the third V , and the fourth S .



5. A beam with a length L is attached to the wall with a cable as shown. A load $W = 500$ lb is attached to the beam. The tension force, T , in the cable is given by:

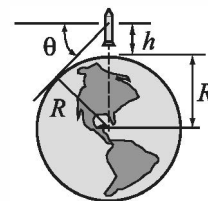
$$T = \frac{WL\sqrt{h^2 + x^2}}{hx}$$

For a beam with $L = 120$ in. and $h = 50$ in., calculate T for $x = 10, 30, 50, 70, 90,$ and 110 in.



6. Write a MATLAB program in a script file that calculate the average, standard deviation, and median of a list of grades as well as the number of grades on the list. The program asks the user (input command) to enter the grades as elements of a vector. The program then calculates the required quantities using MATLAB's built-in functions `length`, `mean`, `std`, and `median`. The results are displayed in the Command Window in the following format:
 "There are XX grades." where XX is the numerical value.
 "The average grade is XX." where XX is the numerical value.
 "The standard deviation is XX." where XX is the numerical value.
 "The median grade is XX." where XX is the numerical value.
 Execute the program and enter the following grades: 92, 74, 53, 61, 100, 42, 80, 66, 71, 78, 91, 85, 79, and 68.

7. A rocket flying straight up measures the angle θ with the horizon at different heights h . Write a MATLAB program in a script file that calculates the radius of the earth R (assuming the earth is a perfect sphere) at each data point and then determines the average of all the values.



h (m)	4	8	12	16	20	24	28	32	36	40
θ (deg)	2.0	2.9	3.5	4.1	4.5	5.0	5.4	5.7	6.1	6.4

8. Decay of radioactive materials can be modeled by the equation $A = A_0 e^{kt}$, where A is the amount at time t , A_0 is the amount at $t = 0$, and k is the decay constant ($k \leq 0$). Iodine-132 is a radioisotope that is used in thyroid function tests. Its half-life time is 13.3 hours. Calculate the relative amount of Iodine-132 (A/A_0) in a patient's body 48 hours after receiving a dose. After determining the value of k , define a vector $t = 0, 4, 8, \dots, 48$ and calculate the corresponding values of A/A_0 .
9. The monthly payment, P , of a N years mortgage of an amount L that with a yearly interest rate of r is given by:

$$P = L \frac{\frac{r}{100 \cdot 12} \left(1 + \frac{r}{100 \cdot 12}\right)^{12N}}{\left(1 + \frac{r}{100 \cdot 12}\right)^{12N} - 1}$$

where r is in % (e.g., 7.5% entered as 7.5). Write a MATLAB program in a script file that calculates P . When the program is executed it asks the user to enter the mortgage amount, the number of years, and the interest rate. The output is displayed in the following format: "The monthly payment of a XX years XXXXXX.XX mortgage with interest rate of XX.XX percent is \$XXXX.XX", where XXX stands for the corresponding quantities. Use the program for determining the monthly payment of a \$250,000 mortgage for 30 years and 4.5% yearly interest rate.

10. The balance of a loan, B , after n monthly payments is given by

$$B = A \left(1 + \frac{r}{1200}\right)^n - \frac{P}{r/1200} \left[\left(1 + \frac{r}{1200}\right)^n - 1\right]$$

where A is the loan amount, P is the amount of a monthly payment, and r is the yearly interest rate entered in % (e.g., 7.5% entered as 7.5). Consider a 5-year, \$20,000 car loan with 6.5% yearly interest that has a monthly payment of \$391.32. Calculate the balance of the loan after every 6 months (i.e., at $n = 6, 12, 18, 24, \dots, 54, 60$). Each time, calculate the percent of the loan that is already paid. Display the results in a three-column table, where the first column displays the month and the second and third columns display the corresponding value of B and percentage of the loan that is already paid, respectively.

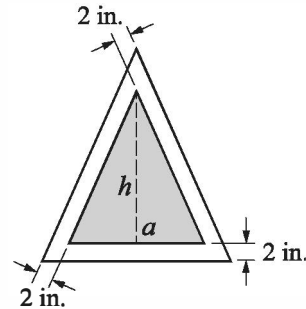
11. Early explorers often estimated altitude by measuring the temperature of boiling water. Use the following two equations to make a table that modern-day hikers could use for the same purpose.

$$p = 29.921(1 - 6.8753 \times 10^{-6}h), \quad T_b = 49.161 \ln p + 44.932$$

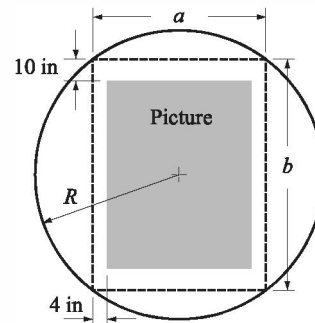
where p is atmospheric pressure in inches of mercury, T_b is boiling temperature in °F, and h is altitude in feet. The table should have two columns, the

first altitude and the second boiling temperature. The altitude should range between -500 ft and $10,000$ ft at increments of 500 ft.

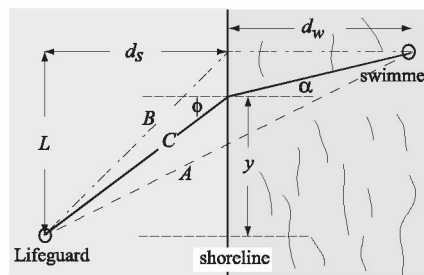
12. An isosceles triangle sign is designed to have a triangular printed area of 600 in.^2 (shaded area with a base length of a and height of h in the figure). As shown in the figure, there is a 2 in. gap between the sides of the triangles. Write a MATLAB program that determine the dimensions a and h such that the overall area of the sign will be as small as possible. In the program define a vector a with values ranging from 10 to 120 with increments of 0.1 . Use this vector for calculating the corresponding values of h and the overall area of the sign. Then use MATLAB's built-in function `min` to find the dimensions of the smallest sign.



13. A round billboard with radius $R = 55 \text{ in.}$ is designed to have a rectangular picture placed inside a rectangle with sides a and b . The margins between the rectangle and the picture are 10 in. at the top and bottom and 4 in. at each side. Write a MATLAB program that determines the dimensions a and b such that the overall area of the picture will be as large as possible. In the program define a vector a with values ranging from 5 to 100 with increments of 0.25 . Use this vector for calculating the corresponding values of b and the overall area of the picture. Then use MATLAB's built-in function `max` to find the dimensions of the largest rectangle.



14. A student has a summer job as a lifeguard at the beach. After spotting a swimmer in trouble, he tries to deduce the path by which he can reach the swimmer in the shortest time. The path of shortest distance (path A) is obviously not the best since it maximizes the time spent swimming (he can run faster than he can swim).



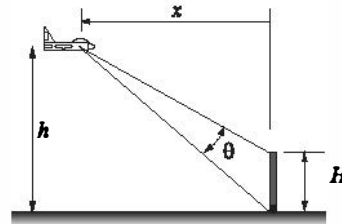
Path B minimizes the time spent swimming but is probably not the best, since it is the longest (reasonable) path. Clearly the optimal path is somewhere in between paths A and B .

Consider an intermediate path C and determine the time required to reach

the swimmer in terms of the running speed $v_{run} = 3$ m/s the swimming speed $v_{swim} = 1$ m/s; the distances $L = 48$ m, $d_s = 30$ m, and $d_w = 42$ m; and the lateral distance y at which the lifeguard enters the water. Create a vector y that ranges between path A and path B ($y = 20, 21, 22, \dots, 48$ m) and compute a time t for each y . Use MATLAB built-in function `min` to find the minimum time t_{min} and the entry point y for which it occurs. Determine the angles that correspond to the calculated value of y and investigate whether your result satisfies Snell's law of refraction:

$$\frac{\sin \phi}{\sin \alpha} = \frac{v_{run}}{v_{swim}}$$

15. An airplane is flying at a height of $h = 900$ ft while watching a target that is 70 ft tall ($H = 70$ ft), as shown in the figure. The best view of the target is when θ is maximum. Write a MATLAB program that determines the distance x at which θ is maximum. Define a vector x with elements ranging from 50 to 1500 with spacing of 0.5. Use this vector to calculate the corresponding values of θ . Then use MATLAB's built-in function `max` to find the value of x that corresponds to the largest value of θ .



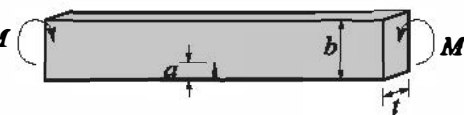
16. The stress intensity factor K at a crack in a beam exposed to pure bending M is given by:

$$K = C\sigma\sqrt{\pi a}$$

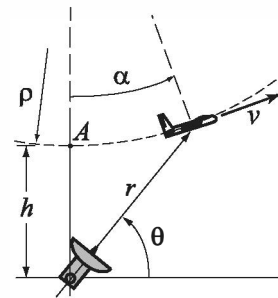
where $\sigma = \frac{6M}{tb^2}$. a is the crack length, b is the width, t is the thickness, and C is a parameter that depends on the geometry of the specimen and crack. For the case of pure bending,

$$C = \sqrt{\frac{\tan \beta}{\beta} \left[\frac{0.923 + 0.199(1 - \sin \beta)^2}{\cos \beta} \right]} \quad \text{where } \alpha = a/b \text{ and } \beta = (\pi\alpha)/2$$

Write a program in a script file that calculates the stress intensity factor K . The program should read the values of M , b , t , and a from an ascii text file using the `load` command. The output should be in the form of a paragraph combining text and numbers,— i.e., something like: "The stress intensity factor for a beam that is 0.25 m wide and 0.01 m thick with an edge crack of 0.05 m and an applied moment of 20 N-m is XX Pa-sqrt(m)." where XX stands for the value of K . Use the program to calculate K when $M = 20$ N-m, $b = 0.25$ m, $t = 0.01$ m, and $a = 0.25$ m.



17. The airplane shown is flying at a constant speed of $v = 50$ m/s in a circular path of radius $\rho = 2000$ m and is being tracked by a radar station positioned a distance $h = 500$ m below the bottom of the plane path (point A). The airplane is at point A at $t = 0$, and the angle α as a function of time is given (in radians) by $\alpha = \frac{v}{\rho}t$. Write a MATLAB program that calculates θ and r as functions of time. The program should first determine the time at which $\alpha = 90^\circ$. Then construct a vector t having 15 elements over the interval $0 \leq t \leq t_{90^\circ}$, and calculate θ and r at each time. The program should print the values of ρ , h , and v , followed by a 15×3 table where the first column is t , the second is the angle θ in degrees, and the third is the corresponding value of r .



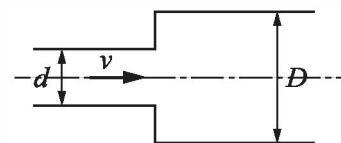
18. The intrinsic electrical conductivity σ of a semiconductor can be approximated by:

$$\sigma = e^{(C - \frac{E_g}{2kT})}$$

where σ is measured in $(\Omega \cdot m)^{-1}$, E_g is the band gap energy, k is Boltzmann's constant (8.62×10^{-5} eV/K), and T is temperature in kelvins. For Germanium, $C = 13.83$ and $E_g = 0.67$ eV. Write a program in a script file that calculates the intrinsic electrical conductivity for Germanium for various temperatures. The values of the temperature should be read from an xls spreadsheet using the `xlsread` command. The output should be presented as a table where the first column is the temperature and the second column is the intrinsic electrical conductivity. Use the following values for temperature: 400, 435, 475, 500, 520, and 545 K.

19. The pressure drop Δp in Pa for a fluid flowing in a pipe with a sudden increase in diameter is given by:

$$\Delta p = \frac{1}{2} \left[1 - \left(\frac{d}{D} \right)^2 \right]^2 \rho v^2$$



where ρ is the density of the fluid, v , the velocity of the flow, and d and D are defined in the figure. Write a program in a script file that calculates the pressure drop Δp . When the script file is executed it request the user to input the density in kg/m^3 , the velocity in m/s, and values of the non-dimensional ratio d/D as a vector. The program displays the inputted values of ρ and v followed by a table with the values of d/D in the first column and the corre-

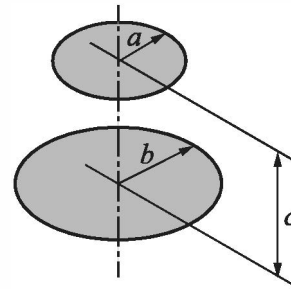
spending values of Δp in the second column.

Execute the program assuming flow of gasoline ($\rho = 737 \text{ kg/m}^3$) at $v = 5 \text{ m/s}$ and the following ratios of diameters $d/D = 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.2$.

20. The net heat exchange by radiation from plate 1 with radius b to plate 2 with radius a that are separated by a distance c is given by:

$$q = \sigma \pi b^2 F_{1-2} (T_1^4 - T_2^4)$$

Where T_1 and T_2 are the absolute temperatures of the plates, $\sigma = 5.669 \times 10^{-8} \text{ W/(m}^2\text{-K}^4)$ is the Stefan-Boltzmann constant, and F_{1-2} is a shape factor which, for the arrangement in the figure, is given by:



$$F_{1-2} = \frac{1}{2} [Z - \sqrt{Z^2 - 4X^2Y^2}]$$

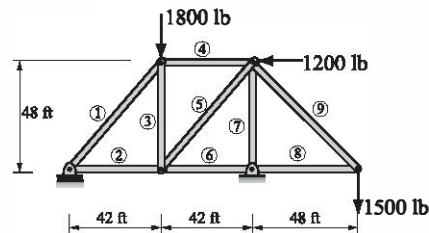
Where $X = a/c$, $Y = c/b$, and $Z = 1 + (1 + X^2)Y^2$. Write a script file that calculates the heat exchange q . For input the program asks the user to enter values for T_1 , T_2 , a , b , and c . For output the program prints a summary of the geometry and temperatures and then print the value of q . Use the script to calculate the results for $T_1 = 400 \text{ K}$, $T_2 = 600 \text{ K}$, $a = 1 \text{ m}$, $b = 2 \text{ m}$, and $c = 0.1, 1, \text{ and } 10 \text{ m}$.

21. Given the coordinates of three points (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) it is possible to find the coordinates of the center of the circle (C_x, C_y) that passes through the three points by solving the following simultaneous equations:

$$2 \begin{bmatrix} (x_1 - x_2) & (y_1 - y_2) \\ (x_2 - x_3) & (y_2 - y_3) \end{bmatrix} \begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} (x_1^2 + y_1^2) - (x_2^2 + y_2^2) \\ (x_2^2 + y_2^2) - (x_3^2 + y_3^2) \end{bmatrix}$$

Write a program in a script file that calculates the coordinates of the center and the radius of a circle that passes through three given points. When executed the program asks the user to enter the coordinates of the three points. The program then calculates the center and radius and displays the results in the following format: "The coordinates of the center are (xx.x, xx.x) and the radius is xx.x.", where xx.x stands for the calculated quantities rounded to the nearest tenth. Execute the program entering the following three points: $(10.5, 4)$, $(2, 8.6)$, and $(-4, -7)$.

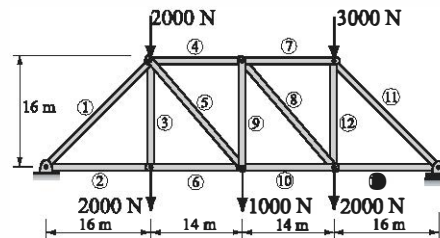
22. A truss is a structure made of members joined at their ends. For the truss shown in the figure, the forces in the nine members are determined by solving the following system of nine equations:



$$\begin{aligned} F_2 + \cos(48.81^\circ)F_1 &= 0 \\ F_6 + \cos(48.81^\circ)F_5 - F_2 &= 0, \\ \sin(48.81^\circ)F_5 + F_3 &= 0 \quad -\cos(48.81^\circ)F_1 + F_4 = 0 \\ -\sin(48.81^\circ)F_1 + F_3 &= 1800, \quad -F_4 - \cos(48.81^\circ)F_5 = 1200, \\ -F_7 - \sin(48.81^\circ)F_5 - \sin(45^\circ)F_9 &= 0, \quad \sin(45^\circ)F_9 = 1500, \\ -\cos(45^\circ)F_9 - F_8 &= 0 \end{aligned}$$

Write the equations in matrix form and use MATLAB to determine the forces in the members. A positive force means tensile force and a negative force means compressive force. Display the results in a table where the first column displays the member number and the second column displays the corresponding force.

23. A truss is a structure made of members joined at their ends. For the truss shown in the figure, the forces in the 13 members are determined by solving the following system of 13 equations.



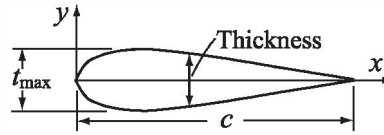
$$\begin{aligned} F_2 + 0.7071F_1 &= 0, \quad -F_2 + F_6 = 0 \\ F_3 - 2000 &= 0, \\ F_4 + 0.6585F_5 - 0.7071F_1 &= 0 \\ 0.7071F_1 + F_3 + 0.7526F_5 + 2000 &= 0, \quad F_7 + 0.6585F_8 - F_4 = 0 \\ 0.7526F_8 + F_9 &= 0, \quad F_{10} - 0.6585F_5 - F_6 = 0, \quad F_9 + 0.7526F_5 - 1000 = 0 \\ 0.7071F_{11} - F_7 &= 0, \quad 0.7071F_{11} + F_{12} + 3000 = 0, \\ F_{12} + 0.7526F_8 - 2000 &= 0, \quad F_{13} + 0.7071F_{11} = 0 \end{aligned}$$

Write the equations in matrix form and use MATLAB to determine the forces in the members. A positive force means tensile force and a negative force means compressive force. Display the results in a table where the first column displays the member number and the second column displays the corresponding force.

24. The graph of the function $f(x) = ax^3 + bx^2 + cx + d$ passes through the points $(-2.6, -68)$, $(0.5, 5.7)$, $(1.5, 4.9)$, and $(3.5, 88)$. Determine the constants a , b , c , and d . (Write a system of four equations with four unknowns, and use MATLAB to solve the equations.)

25. The surface of many airfoils can be described with an equation of the form

$$y = \mp \frac{tc}{0.2} [a_0 \sqrt{x/c} + a_1(x/c) + a_2(x/c)^2 + a_3(x/c)^3 + a_4(x/c)^4]$$



where t is the maximum thickness as a fraction of the chord length c (e.g., $t_{\max} = ct$). Given that $c = 1$ m and $t = 0.2$ m, the following values for y have been measured for a particular airfoil:

x (m)	0.15	0.35	0.5	0.7	0.85
y (m)	0.08909	0.09914	0.08823	0.06107	0.03421

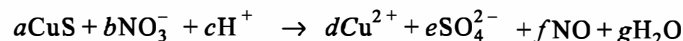
Determine the constants $a_0, a_1, a_2, a_3,$ and a_4 . (Write a system of five equations and five unknowns, and use MATLAB to solve the equations.)

26. During a golf match, a certain number of points are awarded for each eagle and a different number for each birdie. No points are awarded for par, and a certain number of points are deducted for each bogey and a different number deducted for each double bogey (or worse). The newspaper report of an important match neglected to mention what these point values were, but did provide the following table of the results:

Golfer	Eagles	Birdies	Pars	Bogeys	Doubles	Points
A	1	2	10	1	1	5
B	2	3	11	0	1	12
C	1	4	10	1	10	11
D	1	3	10	12	0	8

From the information in the table write four equations in terms of four unknowns. Solve the equations for the unknown points awarded for eagles and birdies and points deducted for bogeys and double bogeys.

27. The dissolution of copper sulfide in aqueous nitric acid is described by the following chemical equation:



where the coefficients $a, b, c, d, e, f,$ and g are the numbers of the various molecule participating in the reaction and are unknown. The unknown coefficients are determined by balancing each atom on left and right and then balancing the ionic charge. The resulting equations are:

$$a = d, \quad a = e, \quad b = f, \quad 3b = 4e + f + g, \quad c = 2g, \quad -b + c = 2d - 2e$$

There are seven unknowns and only six equations. A solution can still be obtained, however, by taking advantage of the fact that all the coefficients must be positive integers. Add a seventh equation by guessing $a = 1$ and solve the system of equations. The solution is valid if all the coefficients are positive integers. If this is not the case, take $a = 2$ and repeat the solution. Continue the process until all the coefficients in the solution are positive integers.

28. The wind chill temperature, T_{wc} , is the air temperature felt on exposed skin due to wind. In U.S. customary units it is calculated by:

$$T_{wc} = 35.74 + 0.6215T - 35.75v^{0.16} + 0.4275Tv^{0.16}$$

where T is the temperature in degrees F, and v is the wind speed in mi/h. Write a MATLAB program in a script file that displays the following chart of wind chill temperature for given air temperature and wind speed in the Command Window:

	Temperature (F)								
Speed (mi/h)	40	30	20	10	0	-10	-20	-30	-40
10	34	21	9	-4	-16	-28	-41	-53	-66
20	30	17	4	-9	-22	-35	-48	-61	-74
30	28	15	1	-12	-26	-39	-53	-67	-80
40	27	13	-1	-15	-29	-43	-57	-71	-84
50	26	12	-3	-17	-31	-45	-60	-74	-88
60	25	10	-4	-19	-33	-48	-62	-76	-91

29. The stress intensity factor K at a crack is given by $K = C\sigma\sqrt{\pi a}$ where σ is the far-field stress, a is the crack length, and C is a parameter that depends on the geometry of the specimen and crack. For the case of the edge crack shown in the figure, C is given by:

$$C = 0.265\left(1 - \frac{a}{b}\right) + \frac{0.857 + 0.265\frac{a}{b}}{\left(1 - \frac{a}{b}\right)^{3/2}}$$

Write a script file that will print out a table of values with the ratio a/b in the first column and the corresponding parameter C in the second column. let a/b range between 0 and 0.95 with increments of 0.05.

