

Chapter 5 File and Directory Management

5.1 The MATLAB Workspace

User commands and resulting output in the MATLAB Command window can be logged and saved as an ascii text file in the current folder. The 'diary fname' command keeps a log of the MATLAB session and stores it in the file 'fname' or simply 'diary' if the file name is omitted.

Example 5.1.1

```
diary klee_sept6 % Save MATLAB session in file klee_sept6
x=1;
y=2;
z=x+y
A=5
B=10
C=A*B
diary off

z = 3
A = 5
B = 10
C = 50
```

The entire Command window or a highlighted portion of it can be printed using either **Print** or **Print Selection** from the Command window **File** menu.

All variables and their values present in the MATLAB Workspace at the end of a MATLAB session can be saved using **Save Workspace as ...** from the **File** menu of the Command window. To load the same variables and their stored values in the MATLAB Workspace, the **Load Workspace ...** menu item from the **File** menu is used.

5.2 Saving, Loading, and Deleting Files

Specific variables from the MATLAB Workspace can be saved in a choice of formats using the 'save' command.

Example 5.2.1

```
clear
price =2.5;
volume=2000;
revenue=price*volume;
save % Saves Workspace variables in binary format in file 'matlab.mat'
```

Saving to: matlab.mat

```
clear
rate=5.0;
time=1000;
dist=rate*time;
save trip % Saves Workspace variables in binary format in 'trip.mat'
```

```
clear
rad=3;
Area=pi*rad^2;
circum=2*pi*rad;
save circle Area circum %Saves Workspace variables 'Area' and 'circum'
% in binary format in file 'circle.mat'
```

```
clear
l=8; w=4; h=3;
A=l*w;
V=A*h;
save box l w A V -ascii % Saves Workspace variables 'l', 'w', 'A' and
% 'V' in 8-digit ascii format in file 'box'
type 'box'
```

```
8.0000000e+000
4.0000000e+000
3.2000000e+001
9.6000000e+001
```

Note that the ascii files are not assigned a .mat extension. Selected variables from saved .mat files can be loaded into the MATLAB Workspace using the 'load' command.

Example 5.2.2

```
clear
x=1; y=2; z=x+y;
save % Saves variables 'x','y' and 'z' in binary format in 'matlab.mat'
load % Loads all variables from file 'matlab.mat into MATLAB Workspace
who
```

Saving to: matlab.mat

```
Loading from: matlab.mat
```

```
Your variables are: x          y          z
```

```
clear
theta=0:pi/10:2*pi
y=sin(theta);
z=cos(theta);
save trig theta y z % Saves Workspace variables 'theta','y' and 'z' in
                    % binary format in file 'trig.mat'
```

```
clear
load trig theta y z % Loads variables 'theta','y' and 'z' from file
                    % 'trig.mat' into MATLAB Workspace
```

```
who
```

```
theta =
  Columns 1 through 7
    0    0.3142    0.6283    0.9425    1.2566    1.5708    1.8850
  Columns 8 through 14
    2.1991    2.5133    2.8274    3.1416    3.4558    3.7699    4.0841
  Columns 15 through 21
    4.3982    4.7124    5.0265    5.3407    5.6549    5.9690    6.2832
```

```
Your variables are: theta      y          z
```

Ascii files can be loaded into a text editor or the MATLAB Workspace. In the latter case, the data is saved in an array variable with the same name as the file.

LOAD FNAME.EXT reads ASCII files that contain rows of space separated values. The file can contain MATLAB-style comments that begin with a percent character '%'. The resulting data is placed into a variable with the same name as the file (without the extension). If the file name begins with a digit or an underscore, a X is prepended to the name. All non-alphanumeric characters in the file name are replaced with underscores.

Example 5.2.3

```
clear
V=12; R=2;
i=V/R;
P=(i^2)*R;
save circuit.doc V R i P -ascii % Save 'V','R','i'and 'P' in ascii file
                                % 'circuit.doc'
type 'circuit.doc' % Type contents of ascii file 'circuit.doc'
load circuit.doc V R i P % Loads variables 'V','R','i' and 'P' from
                        % ascii file 'circuit.doc'
circuit % Display array variable 'circuit'
size(circuit) % Check size of 'circuit'
```

```
1.2000000e+001
2.0000000e+000
6.0000000e+000
7.2000000e+001
```

```

circuit =
    12
     2
     6
    72

ans = 4 1

```

To see if a particular data file has been saved in the current directory (or on MATLAB's search path) the 'exist' command is issued and returns a value of 2, otherwise a zero is returned.

Example 5.2.4

```

clear
x=0.25
sum=1/(1-x) % Sum of geometric series 1 + x + x^2 + x^3 + ...
save geoseries % Save Workspace variables in file 'geoseries.mat'
exist('geomseries.mat','file') % Check for existence of 'geomseries.mat'
exist('geoseries.mat','file') % Check for existence of 'geoseries.mat'

x = 0.2500
sum = 1.3333
ans = 0
ans = 2

```

The variables stored in a MATLAB data file (.mat extension) can be obtained from the Command window using the 'whos' command along with the file name.

Example 5.2.5

```

whos -file geoseries.mat

Name      Size      Bytes  Class

sum       1x1         8  double array
x         1x1         8  double array

Grand total is 2 elements using 16 bytes

```

Finally, MATLAB data files which are no longer needed can be deleted from within MATLAB with the 'delete' command.

Example 5.2.6

```

delete('geoseries.mat')

```

5.3 Special-Purpose File I/O

The MATLAB functions 'dlmread' and 'dlwrite' are designed to be used with ascii files of numerical values separated by a user specified delimiter.

DLMWRITE Write ASCII delimited file.

`DLMWRITE(FILENAME,M,DLM)` writes matrix `M` into `FILENAME` using the character `DLM` as the delimiter. Specify `'\t'` to produce tab-delimited files.

`DLMWRITE(FILENAME, M, DLM, R, C)` writes matrix `M` starting at offset row `R`, and column `C` in the file. `R` and `C` are zero-based, that is `R=C=0` specifies first number in the file.

NOTE: Any elements whose value is 0 will be omitted. For example, The array `[1 0 2]` will appear in a file like `'1,,2'`.

DLMREAD Read ASCII delimited file.

`M = DLMREAD(FILENAME,DLM)` reads numeric data from the ASCII delimited file `FILENAME` using the delimiter `DLM`. The result is returned in `M`. Use `'\t'` to specify a tab.

`M = DLMREAD(FILENAME,DLM,R,C)` reads data from the DLM-delimited file `FILENAME`. `R` and `C` specify the row `R` and column `C` where the upper-left corner of the data lies in the file. `R` and `C` are zero-based so that `R=0` and `C=0` specifies the first value in the file.

`M = DLMREAD(FILENAME,DLM,RNG)` reads the range specified by `RNG = [R1 C1 R2 C2]` where `(R1,C1)` is the upper-left corner of the data to be read and `(R2,C2)` is the lower-right corner. `RNG` can also be specified using spreadsheet notation as in `RNG = 'A1..B7'`.

`DLMREAD` fills empty delimited fields with zero. Data files where the lines end with a non-space delimiter will produce a result with an extra last column filled with zeros.

Example 5.3.1

```
A=10*rand(5)
dlmwrite('random_num.dat', A, ',') % Create ascii file 'random_num.dat'
    % and store values from array A in it, separated by commas
B=dlmread('random_num.dat', ',') % Open ascii file 'random_num.dat',
    % read comma delimited data and store in array B
```

```
A =
    1.5683    3.9162    9.3979    5.4219    7.8692
    4.1635    2.5278    8.3280    4.5573    6.5598
    0.9403    3.5438    4.6998    8.6309    0.0004
    4.4995    7.4298    6.2987    8.5520    1.3124
    8.6915    6.5083    0.5819    4.7226    4.9487
```

```

B =
    1.5683    3.9162    9.3979    5.4219    7.8692
    4.1635    2.5278    8.3280    4.5573    6.5598
    0.9404    3.5438    4.6998    8.6309    0.0004
    4.4995    7.4298    6.2987    8.5520    1.3124
    8.6915    6.5083    0.5819    4.7226    4.9487

```

Popular spreadsheet programs (Excel, Lotus, etc.) store data in a rectangular grid of rows and columns using **'Lotus WK1 spreadsheet format'**. The command `'wklwrite'` can store MATLAB numeric arrays in WK1 spreadsheet format. A description of `'wklwrite'` is obtained using the `'help'` command.

`help wklwrite`

```

WKLWRITE Write spreadsheet (WK1) file.

```

```

WKLWRITE('FILENAME',M) writes matrix M into a Lotus WK1 spreadsheet
file with the name. '.wk1' is appended to the filename if no
extension is given.

```

```

WKLWRITE('FILENAME',M,R,C) writes matrix M into a Lotus WK1 spreadsheet
file, starting at offset row R, and column C in the file. R and C
are zero-based so that R=C=0 is the first cell in the spreadsheet.

```

Example 5.3.2

```

M=rand(5) % Create 5`5 array of random numbers
wklwrite('C:\My Spreadsheets\klee_data.xls', M, 1,2)% Store MATLAB
           % array M in a WK1 format file 'klee_data.xls' in
           % directory C:My Spreadsheets starting at Row 2, Column C

```

```

M =
    0.4608    0.0056    0.9830    0.7334    0.7939
    0.4574    0.2974    0.5527    0.3759    0.9200
    0.4507    0.0492    0.4001    0.0099    0.8447
    0.4122    0.6932    0.1988    0.4199    0.3678
    0.9016    0.6501    0.6252    0.7537    0.6208

```

The created file `'klee_data.xls'` can be opened in EXCEL for further processing. The command `'wklread'` opens files created by a spreadsheet application and reads the data into a MATLAB variable.

`help wklread`

```

WKLREAD Read spreadsheet (WK1) file.

```

```

A = WKLREAD('FILENAME') reads all the data from a Lotus WK1 spreadsheet
file named FILENAME into matrix A.

```

```

A = WKLREAD('FILENAME',R,C) reads data from a Lotus WK1 spreadsheet
file starting at row R and column C, into the matrix A. R and C are
zero-based so that R=C=0 is the first cell of the spreadsheet.

```

`A = WK1READ('FILENAME',R,C,RNG)` specifies a cell range or named range for selecting data from the spreadsheet. A cell range is specified by `RNG = [R1 C1 R2 C2]` where (R1,C1) is the upper-left corner of the data to be read and (R2,C2) is the lower-right corner. `RNG` can also be specified using spreadsheet notation as in `RNG = 'A1..B7'` or a named range like 'Sales'.

Example 5.3.3

```
RNG=[2 3 5 7]; % Set range of cells to be read
P=wk1read('C:\My Spreadsheets\klee_data.xls',2,3,RNG)% Read spreadsheet
% 'klee_data.xls' in directory C:\My Spreadsheets starting
% in Row 3, Column D ending in Row 5, Column G and save in
% MATLAB variable P
```

```
P =
    0.2974    0.5527    0.3759    0.9200
    0.0492    0.4001    0.0099    0.8447
    0.6932    0.1988    0.4199    0.3678
```

5.5 Disk File Manipulation

MATLAB provides commands for obtaining information about working directories and interrogating directories about the existence of files. Several file manipulation commands are illustrated below.

Example 5.5.1

```
cd % Show present working directory

C:\MATLABR11\work\Matlab_Course

dir % Display files in current directory

.          Sub3.m          cyl_call.m
..         Submarine1.mdl  f1.m
AutoSpeed.mdl  Submarine2.mdl  floan.m
AutoSpeedParam.m  TaylorSeries.m  matlab.mat
AutoSpeedParam1.m  TrafficSignal.m  origin_close.m
Gemini.m       TrafficSignalCall.m  ortho_line.m
Histo.m        campusaerial.jpg  polyadd.m
Loan_Interest.m  circuit          road_median.m
Simple_Statistics.m  circuit.doc      trig.mat
Sub1.m         comet_plot.m     uniform_acc.m
Sub2.m         cyl.m

x=1; y=2; z=x+y;
save xyz
exist('xyz.mat', 'file') % Check existence of file 'xyz'

ans = 2

exist('work/EEL5891', 'dir') % Check existence of directory 'EEL5891'

ans = 7

locate_MATLAB=matlabroot % Return directory path to MATLAB in string
                        % 'locate_MATLAB'

locate_MATLAB = C:\MATLABR11

type Histo % Type the M-file Histo.m in Command window

% Script file Histo.m
% This script file plots a histogram of the data generated
% in script file Simple_Statistics.m which must be run prior
% to Histo.m. Histo.m calls script file 'Simple_Statistics.m'
% and draws a 10 bin histogram of the data in vector x
clear
Simple_Statistics
hist(x) % Draws a 10 bin histogram of the data in vector x
```



```
what % Display MATLAB files in the current directory
```

```
M-files in the current directory C:\MATLABR11\work\Matlab_Course
```

```
AutoSpeedParam      Sub3                floan  
AutoSpeedParam1    TaylorSeries        origin_close  
Gemini              TrafficSignal        ortho_line  
Histo               TrafficSignalCall    polyadd  
Loan_Interest       comet_plot          road_median  
Simple_Statistics   cyl                 uniform_acc  
Sub1                 cyl_call  
Sub2                 f1
```

```
MAT-files in the current directory C:\MATLABR11\work\Matlab_Course
```

```
matlab      trig
```

```
MDL-files in the current directory C:\MATLABR11\work\Matlab_Course
```

```
AutoSpeed      Submarine1      Submarine2
```

```
which cityblock % Display the directory path to cityblock.m
```

```
C:\MATLABR11\work\TrafficSim\cityblock.m
```

5.6 The MATLAB Search Path

MATLAB's M-files (script and function) are stored in numerous directories or folders. The list of all directories where M-files exist is referred to as the MATLAB search path. When MATLAB does not recognize a 'name' as either a variable or built-in function, it looks in the current directory. If its not there, the MATLAB search path is searched in sequential order to locate an M-file called 'name .m'.

At startup, a default search path is defined listing all the directories or folders for MATLAB to search for an M-file. The search path can be viewed and modified by selecting the **Path Browser** button on the Command window tool bar or by entering the 'editpath' command in the Command window.

Example 5.6.1

`editpath`

To simply view the current MATLAB search path, use the 'path' command.

Example 5.6.2

`path`

```
MATLABPATH

C:\MATLABR11\toolbox\matlab\general
C:\MATLABR11\toolbox\matlab\ops
C:\MATLABR11\toolbox\matlab\lang
C:\MATLABR11\toolbox\matlab\elmat
C:\MATLABR11\toolbox\matlab\elfun
C:\MATLABR11\toolbox\matlab\specfun
C:\MATLABR11\toolbox\matlab\matfun
C:\MATLABR11\toolbox\matlab\datafun
C:\MATLABR11\toolbox\matlab\polyfun
C:\MATLABR11\toolbox\matlab\funfun
C:\MATLABR11\toolbox\matlab\sparsfun
C:\MATLABR11\toolbox\matlab\graph2d
C:\MATLABR11\toolbox\matlab\graph3d
C:\MATLABR11\toolbox\matlab\specgraph
C:\MATLABR11\toolbox\matlab\graphics
C:\MATLABR11\toolbox\matlab\uitools
C:\MATLABR11\toolbox\matlab\strfun
C:\MATLABR11\toolbox\matlab\iofun
C:\MATLABR11\toolbox\matlab\timefun
C:\MATLABR11\toolbox\matlab\datatypes
C:\MATLABR11\toolbox\matlab\winfun
C:\MATLABR11\toolbox\matlab\demos
C:\MATLABR11\toolbox\powersys\powerdemo
C:\MATLABR11\toolbox\powersys\powersys
C:\MATLABR11\toolbox\compiler
C:\MATLABR11\toolbox\symbolic
C:\MATLABR11\toolbox\pde
```

```

C:\MATLABR11\toolbox\lmi\lmictrl
C:\MATLABR11\toolbox\lmi\lmiLAB
C:\MATLABR11\toolbox\dspblks\dspblks
C:\MATLABR11\toolbox\dspblks\dspmex
C:\MATLABR11\toolbox\dspblks\dspdemoss
C:\MATLABR11\toolbox\dspblks\dspmasks
C:\MATLABR11\toolbox\mpc\mpccmds
C:\MATLABR11\toolbox\mpc\mpcdemos
C:\MATLABR11\toolbox\fdident\fdident
C:\MATLABR11\toolbox\fdident\fdemos
C:\MATLABR11\toolbox\stats
C:\MATLABR11\toolbox\ncd
C:\MATLABR11\toolbox\signal\signal
C:\MATLABR11\toolbox\signal\siggui
C:\MATLABR11\toolbox\signal\sigdemoss
C:\MATLABR11\toolbox\optim
C:\MATLABR11\toolbox\ident
C:\MATLABR11\toolbox\control
C:\MATLABR11\toolbox\control\ctrlguis
C:\MATLABR11\toolbox\control\obsolete
C:\MATLABR11\toolbox\stateflow\sfdemos
C:\MATLABR11\toolbox\sb2sl
C:\MATLABR11\toolbox\stateflow\stateflow
C:\MATLABR11\toolbox\simulink\simulink
C:\MATLABR11\toolbox\simulink\blocks
C:\MATLABR11\toolbox\simulink\simdemoss
C:\MATLABR11\toolbox\simulink\dee
C:\MATLABR11\toolbox\tour
C:\MATLABR11\work
C:\MATLABR11\toolbox\local
C:\matlab\bin
C:\MATLABR11\work\TrafficSim
C:\MATLABR11\work\EngrgCalc
C:\MATLABR11\work\EGN3420
C:\MATLABR11\work\EEL5891

```

The current MATLAB search path can be saved as a string as well.

Example 5.6.3

P=path

```

P =
C:\MATLABR11\toolbox\matlab\general;C:\MATLABR11\toolbox\matlab\ops;C:\M
ATLABR11\toolbox\matlab\lang;C:\MATLABR11\toolbox\matlab\elmat;C:\MATLAB
R11\toolbox\matlab\elfun;C:\MATLABR11\toolbox\matlab\specfun;C:\MATLABR1
1\toolbox\matlab\matfun;C:\MATLABR11\toolbox\matlab\datafun;C:\MATLABR11
\toolbox\matlab\polyfun;C:\MATLABR11\toolbox\matlab\funfun;C:\MATLABR11\
toolbox\matlab\sparfun;C:\MATLABR11\toolbox\matlab\graph2d;C:\MATLABR11\
toolbox\matlab\graph3d;C:\MATLABR11\toolbox\matlab\specgraph;C:\MATLABR1
1\toolbox\matlab\graphics;C:\MATLABR11\toolbox\matlab\uitools;C:\MATLABR
11\toolbox\matlab\strfun;C:\MATLABR11\toolbox\matlab\iofun;C:\MATLABR11\
toolbox\matlab\timefun;C:\MATLABR11\toolbox\matlab\datatypes;C:\MATLABR1
1\toolbox\matlab\winfun;C:\MATLABR11\toolbox\matlab\demos;C:\MATLABR11\t
oolbox\powersys\powerdemo;C:\MATLABR11\toolbox\powersys\powersys;C:\MATL
ABR11\toolbox\compiler;C:\MATLABR11\toolbox\symbolic;C:\MATLABR11\toolbo

```

```
x\pde;C:\MATLABR11\toolbox\lmi\lmictrl;C:\MATLABR11\toolbox\lmi\lmilab;C
:\MATLABR11\toolbox\dspblks\dspblks;C:\MATLABR11\toolbox\dspblks\dspmex;
C:\MATLABR11\toolbox\dspblks\dspdem;C:\MATLABR11\toolbox\dspblks\dspma
sks;C:\MATLABR11\toolbox\mpc\mpccmds;C:\MATLABR11\toolbox\mpc\mpcdemos;C
:\MATLABR11\toolbox\fdident\fdident;C:\MATLABR11\toolbox\fdident\fdemos
;C:\MATLABR11\toolbox\stats;C:\MATLABR11\toolbox\ncd;C:\MATLABR11\toolbo
x\signal\signal;C:\MATLABR11\toolbox\signal\siggui;C:\MATLABR11\toolbox\
signal\sigdem;C:\MATLABR11\toolbox\optim;C:\MATLABR11\toolbox\ident;C:
\MATLABR11\toolbox\control;C:\MATLABR11\toolbox\control\ctrlguis;C:\MATL
ABR11\toolbox\control\obsolete;C:\MATLABR11\toolbox\stateflow\sfdem;C:
\MATLABR11\toolbox\sb2sl;C:\MATLABR11\toolbox\stateflow\stateflow;C:\MAT
LABR11\toolbox\simulink\simulink;C:\MATLABR11\toolbox\simulink\blocks;C:
\MATLABR11\toolbox\simulink\simdem;C:\MATLABR11\toolbox\simulink\dee;C:
\MATLABR11\toolbox\tour;C:\MATLABR11\work;C:\MATLABR11\toolbox\local;C:
\matlab\bin;C:\MATLABR11\work\TrafficSim;C:\MATLABR11\work\EngrgCalc;C:\
MATLABR11\work\EGN3420;C:\MATLABR11\work\EEL5891
```

Existing directories (not on the current search path) can be added to the current path using the 'addpath' command or the 'path' command with several arguments. Directories are removed from the MATLAB search path with the 'rmpath' command.

Example 5.6.4

```
addpath C:\MATLABR11\help -end % Append new directory
      % C:\MATLABR11\help to end of MATLAB search path
path % Show modified MATLAB search path

MATLABPATH

C:\MATLABR11\toolbox\matlab\general
C:\MATLABR11\toolbox\matlab\ops
.
.
.
C:\MATLABR11\work\EGN3420
C:\MATLABR11\work\EEL5891
C:\MATLABR11\help

P=Path; % Save the current MATLAB search path
new_dir='c:\MATLABR11\work\EELxxxx'; % Define another directory to be
      % added to MATLAB search path
path(P,new_dir) % Append new_dir to MATLAB search path
% path(path, new_dir) does the same thing

path % Show modified MATLAB search path
% path(new_dir,path) prepends new_dir to search path
```

```
MATLABPATH
```

```
C:\MATLABR11\toolbox\matlab\general  
C:\MATLABR11\toolbox\matlab\ops  
.  
.  
.  
C:\MATLABR11\work\EGN3420  
C:\MATLABR11\work\EEL5891  
C:\MATLABR11\help  
c:\MATLABR11\work\EELxxxx
```

```
rmpath C:\MATLABR11\help C:\MATLABR11\work\EELxxxx % Remove the new  
% directories C:\MATLABR11\help and C:\MATLABR11\work\EELxxxx
```

```
path % Show current MATLAB search path
```

```
MATLABPATH
```

```
C:\MATLABR11\toolbox\matlab\general  
C:\MATLABR11\toolbox\matlab\ops  
.  
.  
.  
C:\MATLABR11\work\EGN3420  
C:\MATLABR11\work\EEL5891
```

5.7 MATLAB at Startup

At startup, an M-file `matlabrc.m` is executed. Typing `'help matlabrc'` in the Command window explains its basic function.

Example 5.7.1

`help matlabrc`

```
MATLABRC Master startup M-file.
```

```
MATLABRC is automatically executed by MATLAB during startup.  
It establishes the MATLAB path, sets the default figure size,  
and sets a few uicontrol defaults.
```

```
On multi-user or networked systems, the system manager can put  
any messages, definitions, etc. that apply to all users here.
```

```
MATLABRC also invokes a STARTUP command if the file 'startup.m'  
exists on the MATLAB path.
```

Like any other MATLAB M-file, it can be viewed in the Command window by using the `'type'` command.

At the end of `'matlabrc.m'` is a check for the existence of an optional M-file called `'startup.m'`. A user can specify a personalized list of MATLAB commands to be executed at startup. For example, specific directories can be added to the MATLAB search path, plots can be customized, display formats can be set, other M-files can be run, etc. The following startup M-file initiates all MATLAB sessions. It is stored in a directory on the default MATLAB search path.

Example 5.7.3

```
% Create a MATLAB startup file with a greeting and current date
```

```
% MATLAB startup file
```

```
clc  
disp ('Good Day Dr. Klee')  
disp('Today is')  
disp(date)
```

```
Good Day Dr. Klee  
Today is  
26-Sep-2000
```