Appendix B

A Brief Introduction to MATLAB

B.1 Some Basic Information on MATLAB

What Is MATLAB?

MATLAB stands for MATrix LABoratory. It is an interactive software package for solving problems arising in scientific and engineering computations. MATLAB contains programs for all fundamental matrix computations such as solutions of linear systems, various matrix factorizations, solutions of least-squares problems, eigenvalues and eigenvector computations, and singular values and singular vectors computation. It was developed by Cleve Moler. The most current version contains programs for many other types of computations, including two- and three-dimensional graphics capabilities.

Entering and Exiting MATLAB

On most systems the command **matlab** will let you enter into MATLAB. Give the command **exit** to quit MATLAB.

Two Most Important Commands: HELP and DEMO

The two most important commands are **help** and **demo**. Typing “help” you will get the listing of all the MATLAB functions and other MATLAB capabilities. Typing “help” followed by a MATLAB function name from the list will give you more specific information about that particular function.

Example B.1.

```
>> help norm

 NORM Matrix or vector norm.
  For matrices..
    NORM(X) is the largest singular value of X, max(svd(X)).
    NORM(X,2) is the same as NORM(X).
    NORM(X,1) is the 1-norm of X, the largest column sum,
                     = max(sum(abs((X)))).
    NORM(X,inf) is the infinity norm of X, the largest row sum,
                     = max(sum(abs((X’)))).
    NORM(X,’inf’) is same as NORM(X,inf).
    NORM(X,’fro’) is the F-norm, sqrt(sum(diag(X’*X))).
    NORM(X,P) is available for matrix X only if P is 1, 2, inf or ’fro’.

   For vectors..
    NORM(V,P) = sum(abs(V)^P)^(1/P).
    NORM(V) = norm(V,2).
    NORM(V,inf) = max(abs(V)).
    NORM(V,-inf) = min(abs(V)).

   If X has complex components, z, then abs(z) = sqrt(z*conj(z)).

   See also SIGN, ANGLE, and UNWRAP.   ■
```

Demo teaches you how to use such MATLAB functions as how to enter matrix values into a matrix, how to find its transpose, and how to find the rank of a matrix.

B.2 Most Frequently Used MATLAB Operations and Functions and Examples

Some basic matrix operations:

```
   +      Plus
   -      Minus
   *      Matrix multiplication
   .*     Array multiplication
   ^      power
   \      Backslash or left division
   /      Slash or right division
   &      Logical AND
   |      Logical OR
   ~      Logical NOT
```

Matrix functions—numerical linear algebra:

Matrix analysis.

```
   cond        - Matrix condition number.
   norm        - Matrix or vector norm.
   subspace    - Angle between two subspaces.
   rcond       - LINPACK reciprocal condition estimator.
   rank        - Number of linearly independent rows or columns.
   det         - Determinant.
   trace       - Sum of diagonal elements.
```

```
null         - Null space.
orth         - Orthogonalization.
rref         - Reduced row echelon form.
normest      - Estimate of the matrix 2-norm.
condest      - 1-norm condition number estimate.
```

Linear equations.

```
\ and /      - Linear equation solution; use "help slash".
chol         - Cholesky factorization.
lu           - Factors from Gaussian elimination.
inv          - Matrix inverse.
qr           - Orthogonal-triangular decomposition.
qrdelete     - Delete a column from the QR factorization.
qrinsert     - Insert a column in the QR factorization.
pinv         - Pseudoinverse.
lscov        - Least squares in the presence of known covariance.
cholinc      - Incomplete Cholesky factorization.
```

Eigenvalues and singular values.

```
eig          - Eigenvalues and eigenvectors.
poly         - Characteristic polynomial.
hess         - Hessenberg form.
svd          - Singular value decomposition.
qz           - Generalized eigenvalues.
rsf2csf      - Real block diagonal form to complex diagonal form.
cdf2rdf      - Complex diagonal form to real block diagonal form.
schur        - Schur decomposition.
balance      - Diagonal scaling to improve eigenvalue accuracy.
eigs         - A few eigenvalues.
svds         - A few singular values.
gsvd         - Generalized singular value decomposition.
condeig      - Eigenvalue condition numbers.
ordschur     - Reorder eigenvalues in Schur decomposition.
ordqz        - Reordering eigenvalues in QZ factorization.
ordeig       - Eigenvalues of quasi-triangular matrices.
```

Factorization utilities.

```
qrdelete     - Delete a column or row from QR factorization.
qrinsert     - Insert a column or row into QR factorization.
qrupdate     - Rank 1 update to QR factorization.
cholupdate   - Rank 1 update to Cholesky factorization.
planerot     - Givens plane rotation.
```

Use the command help **matfun** to get information on any of the above routines. Here is an example.

Example B.2.

```
>> help lu

 LU Factors from Gaussian elimination.
  [L,U] = LU(X) stores a upper triangular matrix in U and a
  "psychologically lower triangular matrix", i.e. a product
  of lower triangular and permutation matrices, in L , so
  that X = L*U.

   [L,U,P] = LU(X) returns unit lower triangular matrix L, upper
   triangular matrix U, and permutation matrix P so that
   P*X = L*U.

  By itself, LU(X) returns the output from LINPACK'S ZGETRF or
     DGETRF routine.    ■
```

Numerical Examples

To enter a matrix and a vector in MATLAB:

```
>> A=[1 3 5; 2 4 6; 1 3 9]

A =

     1     3     5
     2     4     6
     1     3     9

>> b=[1 1 1]'

b =

     1
     1
     1
```

To solve the linear system $Ax = b$:

```
>> x=A\b

x =

   -0.5000
    0.5000
         0
```

To compute the inverse of A:

```
>> inv(A)

ans =

   -2.2500    1.5000    0.2500
    1.5000   -0.5000   -0.5000
   -0.2500         0    0.2500
```

To compute the eigenvalues of A:

```
>> eig(A)

ans =

   -0.3246
   12.3246
    2.0000
```

To reduce A to an upper Hessenberg matrix:

```
>> hess(A)

ans =

    1.0000   -4.9193    3.1305
   -2.2361    8.6000   -6.2000
         0   -3.2000    4.4000
```

To find the rank of A:

```
>> rank(A)

ans =

     3
```

To compute the 2-norm of A:

```
>> norm(A)

ans =

   13.3538
```

To compute the condition number of A (with respect to 2-norm):

```
>>cond(A)

ans =

     42.1539
```

To compute the QR factorization of A using Householder matrices:

```
>> [q,r] = qr(A) will give the qr factorization of A

q =

   -0.4082    0.5774   -0.7071
   -0.8165   -0.5774   -0.0000
   -0.4082    0.5774    0.7071

r =

   -2.4495   -5.7155  -10.6145
         0    1.1547    4.6188
         0         0    2.8284
```

To compute the LU factorization of A using partial pivoting:

```
>> [l,u] = lu(A)

l =

    0.5000    1.0000         0
    1.0000         0         0
    0.5000    1.0000    1.0000

u =

     2     4     6
     0     1     2
     0     0     4
```

B.3 Writing Your Own Programs Using MATLAB Commands

It is easy to write your own programs using MATLAB commands. We list first some of the most common uses and then give examples.

Some Relational Operators in MATLAB

`<`	less than
`>`	greater than
`<=`	less than or equal to
`>=`	greater than or equal to
`==`	equal

Some Matrix Building Functions

eye	identity matrix
zeros	matrix of zeros
rand	randomly generated matrix with entries between zero and one.
max	max entry
diag	create or extract a diagonal
triu	upper triangular part of a matrix.
tril	lower triangular part of a matrix
hilb	Hilbert matrix
toeplitz	Toeplitz matrix
size	gives the dimension of a matrix
abs	absolute value of the elements of a matrix or a vector.

Examples.

1. rand(5,3) will create a 5 × 3 randomly generated matrix.

2. If x is a vector, diag(x) is the diagonal matrix with entries of x on the diagonal. diag(A) is the vector consisting ot the diagonal of A.

3. hilb(5) will create the 5 × 5 Hilbert matrix.

4. max(A(:,2)) will give the maximum value of the second column of A. max(max(A)) will give the maximum entry of the whole matrix A.

Colon Notation

The colon notation (:) is used to conveniently indicate a whole row or column.

Examples.

1. A(:,2) gives the entire second column of A.

2. A(1:3,:) gives the first three rows of A.

3. A(:,[2,5,7]) gives the second, fifth, and seventh columns of A.

The statement A(:,[2,5,7]) = B(:,1:3) will replace the columns 2,5,7 of A with the first three columns of B.

for, while, if commands

These commands are most useful in writing MATLAB programs for matrix algorithms. The uses of these commands will be illustrated in the following examples.

Computing Elapsed Time of an Algorithm

The function **cputime** returns the CPU time in seconds that has been used by the MATLAB process since the MATLAB process started. For example,

```
t=cputime; your_operation;  cputime - t
```

returns the CPU time to run your operation. Since the PC version of MATLAB does not have a **cputime** function, MATCOM contains a program cputime.m. If the version of MATLAB that you use contains the **cputime** function, then you can delete the cputime.m program from MATCOM. See also MATLAB commands **TIC**, **TOC**, **CLOCK**, **DATENUM**.

Saving a MATLAB Program

The command save `<filename>` will store all variables in the filename `filename.mat`.

The command load `<filename>` will restore all the variables from the file named `filename.mat`.

Getting a Hard Copy

`diary<filename>` will cause a copy of all subsequent terminal input and most of the resulting output to be written to the named file.

`diary off` suspends it.

B.4 Examples of Some Simple MATLAB Programs

Example B.3. The following code will make the elements below the diagonal of the 4×4 matrix $A = a(i, j)$ zero:

```
a = rand(4,4)
for i = 1:4
  for j = 1:4
      if i > j
         a(i,j) = 0
      end;
  end;
end;
```

■

Example B.4. The following code will create a matrix A such that the (i, j)th entry of the matrix $A = a(i, j)$ is $(i + j)$:

```
         a= zeros(4,4)

         for i = 1:4
           for j = 1:4
               a(i,j) = i+j
           end;
         end;
```

■

Example B.5. The following MATLAB program computes the matrix-matrix product of two upper triangular matrices. This program will be called **matmat.m**:

```
% Matrix-Matrix product with upper triangular matrices
% input U and V two upper triangular matrices of order n
% output C = U * V
% function  C = matmat(U,V)
  function  C = matmat(U,V)
        [n,m] = size(U)
         C = zeros(n,n)
         for i = 1:n
          for j = i:n
            for k = i:j
                C(i,j) = C(i,j) + U(i,k) * V(k,j)
            end;
          end;
         end;

        end;
```

■

Example B.6 (MATLAB implementation of Algorithm 4.1).

```
% Computing  the two norm of a vector x
% input x
% output  nrm, the two norm of the vector x
% function  nrm = twonorm(x)
  function  nrm = twonorm(x)
        [n,m] = size(x)
        r = max(abs(x))
        y=x/r
        s = 0
        for i=1:n
        s = s + y(i)^2
        end;
        nrm = r * s^0.5
        end;
```

■

Use of “diary” Command and Printing The Output

Sometimes you may want a listing of your MATLAB program step by step as it was executed. The **diary** command can be used to create a listing. For example,

```
>>diary B:diary8     <----- sets the diary on
>> C = matmat(U,V)   <----- execute your program
>> diary off         <----- set the diary off
```

The above command will store all the commands that are executed by the program matmat.m in the file diary8. Only data which are printed on the screen are stored in diary8. This file can then be printed. If you do not want the output to be printed on the screen, place a semicolon at the end of the line.

To write a comment line put the % sign in the first column. Examples of some M files are given in this appendix.