



# Matlab and Matrices

Math 45 — Linear Algebra

David Arnold

**[David-Arnold@Eureka.redwoods.cc.ca.us](mailto:David-Arnold@Eureka.redwoods.cc.ca.us)**

## Abstract

In this exercise you will learn how to enter and edit matrices in Matlab. You will also experiment with some of Matlab's built-in matrix builders. You will also learn how to build matrices from vectors and blocks of matrices. *Prerequisites: None.*

next page

close

exit

# Table of Contents

## Introduction

Working with Matlab

## Entering Matrices

Special Matrices in Matlab

The Transpose Operator

Suppressing Output

Matlab's Workspace

## Indexing Matrices

Advanced Indexing Techniques

## Building Matrices

Building Matrices From Vectors

The Dimensions Need To Match

Building Matrices From Other Matrices

Your Imagination Is Your Only Limit

## Homework

Matlab  
and  
Matrices

title page

contents

previous page

next page

back

print doc

close

exit

## Introduction

This is an interactive document designed for online viewing. We've constructed this onscreen documents because we want to make a conscientious effort to cut down on the amount of paper wasted at the College. Consequently, printing of the onscreen document has been purposefully disabled. However, if you are extremely uncomfortable viewing documents onscreen, we have provided a print version. If you click on the Print Doc button, you will be transferred to the print version of the document, which you can print from your browser or the Acrobat Reader. We respectfully request that you only use this feature when you are at home. Help us to cut down on paper use at the College.

Much effort has been put into the design of the onscreen version so that you can comfortably navigate through the document. Most of the navigation tools are evident, but one particular feature warrants a bit of explanation. The section and subsection headings in the onscreen and print documents are interactive. If you click on any section or subsection header in the onscreen document, you will be transferred to an identical location in the print version of the document. If you are in the print version, you can make a return journey to the onscreen document by clicking on any section or subsection header in the print document.

Finally, the table of contents is also interactive. Clicking on an entry in the table of contents takes you directly to that section or subsection in the document.

## Working with Matlab

This document is a working document. It is expected that you are sitting in front of a computer terminal where the Matlab software is installed. You are not supposed to read this document as if it were a short story. Rather, each time your are presented with a Matlab command, it is expected that you will enter the command, then hit the Enter key to execute the command and view the result. Furthermore, it is expected that you will ponder the result. Make sure that you completely understand why you got the result you did before you continue with the reading.

## Entering Matrices

Entering matrices in Matlab is easy. Enter the following at the Matlab prompt.

**Matlab  
and  
Matrices**

**title page**

**contents**

**previous page**

**next page**

**back**

**print doc**

**close**

**exit**

```
>> A=[1,2,3;4,5,6;7,8,9]
```

Note how semicolons signal the end of a row, while commas are used to separate the entries in a row. Spaces can also be used to separate the entries in a row.

```
>> A=[1 2 3;4 5 6;7 8 9]
```

## Special Matrices in Matlab

Matlab has a number of built-in routines to help create matrices<sup>1</sup>. You can create a matrix of zeros of any size<sup>2</sup>.

```
>> A=zeros(5)  
>> B=zeros(3,5)
```

You easily can create a matrix of zeros that is the same size as a given matrix.

```
>> C=magic(5)  
>> D=zeros(size(C))
```

You can create matrices of ones in a similar manner.

```
>> A=ones(6)  
>> B=ones(2,10)  
>> C=hilb(5)  
>> D=ones(size(C))
```

When performing simulations in Matlab it is useful to build matrices of random numbers. You can create a matrix of uniformly distributed random numbers, each between 0 and 1 (0 inclusive), with the following commands.

<sup>1</sup>For a list of these elementary matrices, type `help elmat` at the Matlab prompt.

<sup>2</sup>Remember, help is readily available for any Matlab function. For example, type `help zeros` to receive help on the zeros function.

```
>> A=rand(6)
>> B=rand(5,3)
```

Scalar multiplication works with matrices in exactly the same manner as with vectors.

```
>> C=10*rand(5)
```

Matlab has a variety of routines that round numbers.

```
>> D=floor(C)
>> D=ceil(C)
>> D=round(C)
>> D=fix(C)
```

The identity matrix has ones on its main diagonal and zeros everywhere else.<sup>3</sup>

```
>> I=eye(5)
```

Other types of diagonal matrices are possible with Matlab's `diag` command.

```
>> E=diag([1,2,3,4,5])
>> F=diag([1,2,3,4,5],-1)
>> G=diag(1:5,1)
```

## The Transpose Operator

The transpose operator (single apostrophe) plays the same role with matrices as it does with vectors. Rows are changed to columns and columns are changed to rows.

```
>> J=[1 2 3;4 5 6;7 8 9]
>> J'
```

<sup>3</sup>The import of the identity matrix will be investigated in later activities.

## Suppressing Output

Remember, appending a semicolon to any Matlab command will suppress the output. This is useful if the result is large and you have no desire to view it.

```
>> K=rand(100);
```

## Matlab's Workspace

Take a moment to examine Matlab's workspace with the whos command.

```
>> whos
```

Note the size of each of the variables in your workspace is given. Of course, you could also find the size of the matrix I by entering

```
>> size(I)
```

## Indexing Matrices

The following notation is used to denote a matrix with 3 rows and 3 columns.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Note that  $a_{12}$  refers to the entry in row 1, column 2. In general,  $a_{ij}$  refers to the entry in row  $i$ , column  $j$ .

Matlab uses similar notation to identify the elements of a matrix.

```
>> A=pascal(5)
```

```
>> A(1,2)
```

```
>> A(3,4)
```

In general,  $A(i, j)$  refers to the element in row  $i$ , column  $j$  of matrix  $A$ .

You can easily change an entry in a matrix.

```
>> A(3,3)=11111
```

## Advanced Indexing Techniques

When indexing a matrix, your subscripts can also be vectors. This is a powerful tool which allows you to easily select a submatrix of a given matrix.

```
>> A=magic(6)
>> A([1,2],[3,4,5])
```

The notation `A([1,2],[3,4,5])` references the submatrix formed using the elements that appear in rows 1 and 2 *and* in columns 3,4, and 5 of the matrix A.

The command

```
>> A([1,3,5],[1,2,3,4,5,6])
```

produces a submatrix composed of rows 1, 3, and 5 of matrix A.

If you recall that the notation `1:6` represents the vector `[1,2,3,4,5,6]`, so that `A(1,1:6)` is equivalent to `A(1,[1,2,3,4,5,6])`.

```
>> A(1,1:6)
```

Using a colon by itself in place of a subscript denotes *all* of the corresponding row or column. Thus,

```
>> A(:,1)
```

produces the first column of the matrix A, and

```
>> A(3,:)
```

produces the third row of the matrix A. In a sense, you can read the notation `A(3,:)` as follows: “Third row, every column.” The command

```
>> A(1:3,:)
```

produces a submatrix composed of the first three rows of the matrix A, while the command

```
>> A(:,1:2:6)
```

produces a submatrix composed of columns 1, 3, and 5 of matrix A.

## Building Matrices

Matlab allows you to build complex matrices out of vectors and matrices.

### Building Matrices From Vectors

Create three row vectors with the following commands.

```
>> v1=1:3
```

```
>> v2=4:6
```

```
>> v3=7:9
```

The command

```
>> M=[v1;v2;v3]
```

builds a matrix with the vectors v1, v2, and v3 each forming a row of the matrix M.

The command

```
>> N=[v1,v2,v3]
```

produces a significantly different but logical result.

Change the vectors v1, v2, and v3 into column vectors with the transpose operator.

```
>> v1=v1'  
>> v2=v2'  
>> v3=v3'
```

The command

```
>> P=[v1,v2,v3]
```

builds a matrix with the vectors  $v_1$ ,  $v_2$ , and  $v_3$  each forming a column of the matrix  $P$ . Of course, you could have achieved the same result by taking the transpose of the matrix  $M$ .

```
>> M'
```

## The Dimensions Need To Match

When building matrices, be sure that each row and column has the same number of elements. For example, the following sequence of commands will produce an error.

```
>> w1=1:3,w2=4:6,w3=7:10  
>> Q=[w1;w2;w3]
```

## Building Matrices From Other Matrices

It is a simple matter to augment a row or column vector to a matrix. Remember, each row and column of a matrix must have the same number of elements. Thus,

```
>> A=[1,2,3,4;5,6,7,8;9,10,11,12]  
>> b=[1,1,1]'  
>> M=[A,b]
```

is valid, but

```
>> M=[A;b]
```

is not.

You can also augment two or more matrices. Remember, each row and column of a matrix must have the same number of elements. Thus,

```
>> A=magic(3),B=ones(3,4)
>> M=[A,B]
```

is valid, but

```
>> N=[A;B]
```

is not.

## Your Imagination Is Your Only Limit

Matlab's matrix building capability is extremely flexible. Here is an interesting example.

```
>> A=zeros(3),B=ones(3),C=2*ones(3),D=3*ones(3)
>> M=[A,B;C,D]
```

Here is something called a *Vandermonde* matrix.

```
>> x=[1,2,3,4,5]'  
>> N=[ones(size(x)),x,x.^2,x.^3,x.^4]
```

And here is one final piece of chicanery.

```
>> B=zeros(8)
>> B(1:3,1:3)=[1 2 3;4 5 6;7 8 9]
>> B(4:8,4:8)=magic(5)
```

## Homework

Answer questions 1-11 on regular notebook paper. Be sure to test your responses at the computer.

1. Write the Matlab commands that generate each of the following matrices.

a. 
$$\begin{pmatrix} 1 & 1 & 3 & 0 \\ -3 & -1 & 6 & 8 \\ 4 & 4 & 10 & -7 \end{pmatrix}$$

b. 
$$\begin{pmatrix} 10 & 5 & -6 \\ -5 & -8 & -4 \\ -5 & -10 & 3 \\ 8 & 8 & -5 \end{pmatrix}$$

2. Write a simple command that will create a  $3 \times 5$  matrix where each entry is the number  $-3$ .

3. Craft a Hilbert matrix with the following commands.

```
>> format rat
>> H=hilb(5)
>> format
```

Develop a formula for the entry in row  $i$ , column  $j$  of the matrix  $H$ . That is, find an expression for  $H(i,j)$  in terms of  $i$  and  $j$ .

4. Explain the difference between the commands `floor`, `ceil`, `round`, and `fix`. Demonstrate with examples in Matlab.

5. Write a Matlab command that will generate a  $4 \times 4$  matrix filled with random integers between  $-5$  and  $5$ .

6. Technically, `'` is Matlab's conjugate transpose operator. To see this, enter the matrix `A=[1, 2+i, sqrt(2); 3, 3-i, sqrt(-1)]`, then type `A'`. Describe the result. What happens when you enter `A.'`? Explain the difference between the transpose operators `'` and `.'`.

7. What is the entry in row 5 column 7 of matrix `pascal(10)`? What Matlab commands do you use to find this entry?

[title page](#)

[contents](#)

[previous page](#)

[next page](#)

[back](#)

[print doc](#)

[close](#)

[exit](#)

8. Let  $T = \text{toeplitz}(1:7)$ . Write a Matlab command that will

- place the even rows of matrix  $T$  into the matrix  $B$ .
- place the odd columns of matrix  $T$  in matrix  $C$ .
- place the last column of the matrix  $T$  in the vector  $b$ .

9. The system of equations

$$x_1 + 2x_2 - 3x_3 = 4$$

$$2x_1 - 3x_3 = -2$$

$$x_2 + x_3 = 0$$

has *coefficient matrix* and *answer vector*

$$A = \begin{pmatrix} 1 & 2 & -3 \\ 2 & 0 & -3 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 4 \\ -2 \\ 0 \end{pmatrix},$$

respectively. Form the *augmented matrix*  $M = [A, b]$  and place the augmented matrix in *reduced row echelon form* with the command `rref(M)`.

10. Let  $x = (0:\pi/2:2*\pi)$ . What Matlab commands will create a matrix whose first row is  $x$ , whose second row is the sine of each entry in  $x$ , and whose third row is the cosine of each entry in  $x$ ?

11. Let  $x = \text{linspace}(0, 10)$ . What Matlab commands will create a matrix whose first column is  $x$ , whose second column is the square of each entry in  $x$ , and whose third column is the reciprocal of each entry in  $x$ ?

[title page](#)

[contents](#)

[previous page](#)

[next page](#)

[back](#)

[print doc](#)

[close](#)

[exit](#)