# Vectors and Matrices in Matlab

Math 50C — Multivariable Calculus

David Arnold

David-Arnold@Eureka.redwoods.cc.ca.us

**Abstract**

In this exercise you will learn how to enter vectors and matrices in Matlab. Operations involving vectors and matrices will be discussed. *Prerequisites: None*

# Table of Contents

# 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.

# Introducing Matrices

In MATLAB ("MATrix LABoratory"), the basic data structure is the *matrix*, which is simply a rectangular arrangement of numbers, real or complex, in rows and columns. Entering a matrix into MATLAB's workspace

is easy. For example, the matrix

$$A = \begin{bmatrix} \sqrt{2} & \pi & 1/2 \\ e & \sin(\pi/2) & \sqrt{-1} \end{bmatrix}$$

is entered as follows.

```
>> A=[sqrt(2),pi,1/2;exp(1),sin(pi/2),sqrt(-1)]
A =
   1.4142            3.1416            0.5000
   2.7183            1.0000                 0 + 1.0000i
```

Note that commas are used delimit the entries in each row, while semi-colons terminate a row. One can also delimit the entries in a row with spaces, as in

```
>> A=[sqrt(2) pi 1/2;exp(1) sin(pi/2) sqrt(-1)]
A =
   1.4142            3.1416            0.5000
   2.7183            1.0000                 0 + 1.0000i
```

Because matrix $A$ has two rows and three columns, we say that its *size* is 2 by 3. MATLAB can calculate the size of a matrix for you.

```
>> size(A)
ans =
     2     3
```

Even single numbers in MATLAB are matrices.

```
>> a=5;
>> size(a)
ans =
     1     1
```

MATLAB thinks that 5, or any other number, is a matrix with one row and one column.



title page
contents
previous page
next page
back
print doc
close
exit

If $A$ and $B$ are matrices of the same size, then they can be added together. For example,

```
>> A=[1 2;3 4],B=[5,6;7,8],C=A+B
A =
     1     2
     3     4
B =
     5     6
     7     8
C =
     6     8
    10    12
```

You will notice that each element of the matrix $C$ is sum of the corresponding elements in the matrices $A$ and $B$. The same is true for the difference of two matrices. Try `C-A`, and see what you get (you should get $B$).

You can also multiply a matrix by a scalar.

```
>> A=ones(2,2),C=-5*A
A =
     1     1
     1     1
C =
    -5    -5
    -5    -5
```

Note that the matrix $C$ is formed by multiplying each entry of the matrix $A$ by $-5$.



title page
contents
previous page
next page
back
print doc
close
exit

## Vectors

In MATLAB, a *vector* is simply a list of numbers, real or complex, which can be arranged in a row,

```
>> v=[1,2,3]
v =
     1     2     3
```

or in a column.

```
>> w=[1;pi;3-2i]
w =
   1.0000
   3.1416
   3.0000 - 2.0000i
```

Note that commas are used to separate the elements in a row, while semi-colons are used to terminate a row.

MATLAB can easily calculate the length of a vector.

```
>> length(v), length(w)
ans =
     3
ans =
     3
```

There are two important points to be made; (1) both $v$ and $w$ have length 3, and (2) you can enter multiple commands at the MATLAB prompt, separated by commas. Of course, you can also use semi-colons to suppress the output. Try `length(v);length(w)` and see what happens.

MATLAB's *transpose* operator, `.'`, is used to toggle a row vector to a column vector and vice-versa.[1]

[1] MATLAB also has an operator called the *conjugate transpose*, which not only transposes the vector, but takes the complex conjugate of each element of the vector as well. Try `w'` to see this for yourself. Note that the difference is subtle, apostrophe for the conjugate transpose, dot-apostrophe for the regular transpose.



title page
contents
previous page
next page
back
print doc
close
exit

```
>> v.'
ans =
     1
     2
     3
>> w.'
ans =
   1.0000            3.1416            3.0000 - 2.0000i
```

Let's define two important vector operations.

**Definition 1**

Let $\alpha$ be a real or complex number. Let $\mathbf{v}$ and $\mathbf{w}$ be vectors of equal length having real or complex entries.

- The product, $\alpha\mathbf{v}$, is formed by multiplying each entry of $\mathbf{v}$ by $\alpha$.
- The sum, $\mathbf{v} + \mathbf{w}$, is formed by adding the corresponding entries in each vector.

For example,

```
>> alpha=5; v=[1,2,3];
>> alpha*v
ans =
     5    10    15
```

and

```
>> v=[1,2,3]; w=[4,5,6];
>> v+w
ans =
     5     7     9
```

One can calculate the difference of two vectors with $\mathbf{v} - \mathbf{w} = \mathbf{v} + (-\mathbf{w})$, or one can simply subtract the corresponding entries in the prescribed order, as in



title page
contents
previous page
next page
back
print doc
close
exit

```
>> v=[1,2,3]; w=[4,5,6];
>> v-w
ans =
    -3    -3    -3
```

One can also form *linear combinations* of vectors.

**Definition 2**

Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$ be vectors of the same size with real or complex entries. Let $\alpha_1, \alpha_2, \ldots, \alpha_p$ be scalars (numbers), either real or complex. Then

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_p\mathbf{v}_p$$

is called a **linear combination** of the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$.

**Example 1**

Consider the vectors

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} -1 \\ 4 \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{a}_3 = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix},$$

Form and simplify the linear combination $2\mathbf{a}_1 + 3\mathbf{a}_2 - 4\mathbf{a}_3$.

Of course,

$$2\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 3\begin{bmatrix} -1 \\ 4 \\ 0 \end{bmatrix} - 4\begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} + \begin{bmatrix} -3 \\ 12 \\ 0 \end{bmatrix} + \begin{bmatrix} -8 \\ 0 \\ -8 \end{bmatrix} = \begin{bmatrix} -9 \\ 16 \\ -2 \end{bmatrix},$$

but MATLAB simplifies this calculation.



title page
contents
previous page
next page
back
print doc
close
exit

```
>> a1=[1;2;3];a2=[-1;4;0];a3=[2;0;2];
>> 2*v1+3*v2-4*v3
ans =
    -9
    16
    -2
```

## Matrix–Vector Multiplication

In this section we show how to compute the product of a matrix and a vector. Consider

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 2 & 4 & 0 \\ 3 & 0 & 2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 2 \\ 3 \\ -4 \end{bmatrix}.$$

Notice that the columns of matrix $A$ are

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} -1 \\ 4 \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{a}_3 = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix},$$

which are identical to the vectors in **Example 1**. The matrix-vector product $A\mathbf{x}$ is computed by forming a linear combination of the columns of matrix $A$, using the entries of the vector $\mathbf{x}$ as the scalars. Note that the vector $\mathbf{x}$ has three entries, one for each column of matrix $A$. Otherwise, the multiplication would not be possible.

$$\begin{aligned} A\mathbf{x} &= \begin{bmatrix} 1 & -1 & 2 \\ 2 & 4 & 0 \\ 3 & 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ -4 \end{bmatrix} \\ &= 2 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 3 \begin{bmatrix} -1 \\ 4 \\ 0 \end{bmatrix} - 4 \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} -9 \\ 16 \\ -2 \end{bmatrix} \end{aligned}$$

It is extremely important to note that this is the same computation used to find the solution in **Example 1**.

Of course, MATLAB handles this computation with ease. You can enter the matrix $A$ from scratch, but in this case it is easier to build the matrix $A$ from the vectors used in **Example 1**.

```
>> A=[a1,a2,a3]
A =
     1    -1     2
     2     4     0
     3     0     2
```

The ability to build matrices in this manner is one of MATLAB's most powerful features. Create the vector **x** and perform the multiplication.

```
>> x=[2;3;-4];
>> A*x
ans =
    -9
    16
    -2
```

Let's summarize these results in a definition.

**Definition 3**

Let $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ denote the columns of matrix $A$; i.e., $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. Let $\mathbf{x}$ be a column vector having length equal to the number of columns of matrix $A$. Then the matrix–vector product, $A\mathbf{x}$, is defined as

$$A\mathbf{x} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_n\mathbf{a}_n.$$

If the length of the vector $\mathbf{x}$ differs from the number of columns of the matrix $A$, then the product $A\mathbf{x}$ is undefined.



title page
contents
previous page
next page
back
print doc
close
exit

## Matrix–Matrix Multiplication

Now that we know how to form the product of a matrix and vector, it is a simple matter to extend this idea to the product of two matrices.

**Definition 4**

**Matrix Multiplication**. Suppose that matrix $A$ and $B$ have dimensions appropriate for multiplication (the number of columns of $A$ must equal the number of rows of $B$). Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_p$ represent the columns of matrix $B$. Then the matrix product $AB$ is defined as follows:

$$\begin{aligned} AB &= A\left[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_p\right], \\ &= \left[A\mathbf{b}_1, A\mathbf{b}_2, \ldots, A\mathbf{b}_p\right]. \end{aligned}$$

Therefore, the matrix product $AB$ is formed by multiplying each column of the matrix $B$ by the matrix $A$.

**Example 2**

Let $A$ and $B$ represent the following matrices.

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & -2 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$

Find the matrix product $AB$.

Note that the first column of the matrix $B$ is

$$\mathbf{b}_1 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}.$$

The first column of the matrix product $AB$ is $A\mathbf{b}_1$. Note that this multiplication would not be possible if it were not for the fact that the number of rows of $B$ equals the number of columns of $A$.

$$A\mathbf{b}_1 = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & -2 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$
$$= 2\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + 2\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} 2 \\ -2 \\ 4 \end{bmatrix}$$

Since the second, third, and fourth columns of matrix $B$ are identical to the first column of $B$,

$$A\mathbf{b}_1 = A\mathbf{b}_2 = A\mathbf{b}_3 = A\mathbf{b}_4 = \begin{bmatrix} 2 \\ -2 \\ 4 \end{bmatrix}.$$

Consequently,

$$AB = A[\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4],$$
$$= [A\mathbf{b}_1, A\mathbf{b}_2, A\mathbf{b}_3, A\mathbf{b}_4],$$
$$= \begin{bmatrix} 2 & 2 & 2 & 2 \\ -2 & -2 & -2 & -2 \\ 4 & 4 & 4 & 4 \end{bmatrix}.$$

MATLAB will easily duplicate this effort and relieve us of the drudgery of hand calculations.

```
>> A=[1 -1 1;0 1 -2;1 0 1];B=2*ones(3,4);
>> C=A*B
C =
     2     2     2     2
    -2    -2    -2    -2
     4     4     4     4
```

**Remark 1**

Note that it is possible to define $AB$ only if the number of rows in $B$ equals the number of columns in $A$. More precisely, if $A$ is an $m \times n$ matrix and $B$ is a $n \times p$ matrix, then the matrix multiplication is possible, and the product $AB$ is a $m \times p$ matrix.

Mathematicians will say that the matrix multiplication is possible if "the inner dimensions match," and the "outer dimensions give the dimension of the product."

In **Example 2**, $A$ is $3 \times 3$ and $B$ is $3 \times 4$. Since the inner dimensions are both 3, it is possible to multiply these matrices. The outer dimensions of matrix $A$ and $B$ predict that the matrix product will have dimensions $3 \times 4$, which is precisely what we saw in **Example 2**.

**Example 3**

If

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 0 \\ -1 & 2 \\ 4 & -3 \end{bmatrix},$$

use MATLAB to find the matrix products $AB$ and $BA$.

Matrix $A$ is $2 \times 2$ and matrix $B$ is $3 \times 2$. The inner dimensions do not match, so it is not possible to form the matrix product $AB$. We should expect MATLAB to complain.

```
>> A=[1,2;3,4];B=[1,0;-1,2;4,-3];
>> A*B
??? Error using ==> *
Inner matrix dimensions must agree.
```

However, matrix $B$ is $3 \times 2$ and matrix $A$ is $2 \times 2$. This time the inner dimensions match, so it is possible to form the product $BA$. Further, the outer dimensions of $B$ and $A$ predict that the matrix product $BA$ will have dimension $3 \times 2$.

```
>> B*A
ans =
     1     2
     5     6
    -5    -4
```

**Example 4**

If

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

use MATLAB to find $AB$ and $BA$ and comment on the result.

Enter the matrices and perform the calculations.

```
>> A=[1,2;3,4];B=[1,-1;-1,1];
>> A*B,B*A
ans =
    -1     1
    -1     1
ans =
    -2    -2
     2     2
```

This example clearly indicates that matrix multiplication is *not commutative*. In general, $AB \neq BA$. If you switch the order of multiplication, you cannot expect that the answer will be the same. We will explore further properties of matrices in the exercises.

## Array Operations

There is often a need for operations which operate on an element-by-element basis. In MATLAB these operations are built-in and very easy to use. For example, should you need to multiply two vectors on an element-by-element basis, use MATLAB's `.*` operator.

```
>> v=[1,2,3,4],w=[5,6,7,8],u=v.*w
v =
     1     2     3     4
w =
     5     6     7     8
u =
     5    12    21    32
```

If you look closely, you will see that **u** is a vector of the same size as **v** and **w**, and that each element in **u** is the product of the corresponding elements in **v** and **w**. The MATLAB symbol for this operation is `.*`, and as you have seen, the result is quite different from what `*` does. Try entering `v*w` and see what happens.

The operation we have just defined is called *array multiplication*. There are other array operations. All of them act element-by-element. Try `v./w` and `w./v`. This is *array right division*. Then try `v.\w` and `w.\v`, and compare the results. This is called *array left division*.

For all array operations it is required that the matrices be of exactly the same size. You might try `[1,2;3,4].*[1;1]` to see what happens when they are not.

There is one other array operation — *array exponentiation*. As might be expected, this is also an element-by-element operation. The operation `A.^2` results in every element of the matrix $A$ being raised to the second power. For example, if `A=[1,2;3,4]`, the command

```
>> A=[1,2;3,4],B=A.^2
A =
     1     2
     3     4
B =
     1     4
     9    16
```

raises each entry in $A$ to the second power. Because the command `A^2` is equivalent to `A*A`, you get an entirely different result.



title page
contents
previous page
next page
back
print doc
close
exit

```
>> B=A^2
B =
     7    10
    15    22
```

The built-in MATLAB functions, which we discussed briefly in Chapter 1, are all designed to be *array smart*. This means that if you apply them to a matrix, the result will be the matrix obtained by applying the function to each individual element. For example:

```
>> theta=[0,pi/2,pi,3*pi/2,2*pi],cos(theta)
theta =
         0    1.5708    3.1416    4.7124    6.2832
ans =
    1.0000    0.0000   -1.0000    0.0000    1.0000
```

This is an extremely important feature of MATLAB, as you will discover in the next chapter.

## Correcting Mistakes

What can you do when you make a mistake entering data at the MATLAB prompt? For example, suppose that you mean to enter the vector $v = [1, 2, 3, 4, 5]$, but you inadvertently type

```
>> v=[1,2,3,4,6]
v =
     1     2     3     4     6
```

Of course, you can correct the mistake by retyping your entry, but that would be inefficient for large vectors and/or matrices. MATLAB has a replay feature that is useful in this instance. Simply hit the up-arrow key on your keyboard and MATLAB will replay your last instruction at the MATLAB prompt. Make your correction, hit the **Enter** key, and you're done. Try it!

In fact, every time you press the up-arrow key on your keyboard, MATLAB cycles backward one command in your command history. Pressing the down-arrow cycles in the other direction. Try pressing the up-arrow and down-arrow keys several times in succession to get a feel for this useful feature.



title page
contents
previous page
next page
back
print doc
close
exit

MATLAB also affords you the opportunity to select past commands that start with a particular letter, word, or phase. For example, if you want to replay all past commands that started with the letter `s`, simply type the letter `s` at the MATLAB prompt, then begin striking the up-arrow key on your keyboard. MATLAB will replay all past commands beginning with the letter `s`. Similarly, if you want to replay all commands starting with the word `help`, type `help` at the MATLAB prompt, then continually strike the up-arrow key to recycle through past commands starting with the word `help`.

One final note. Striking the escape key (Esc) on your keyboard clears the command line. Useful when things get so garbled you want to start over.

## Indexing

You can access the individual entries of a vector by using MATLAB's indexing capabilities. For example, if you want the third element of the vector $v$, simply enter

```
>> v(3)
ans =
     8
```

If you want the second through fourth entries of the vector, enter

```
>> v([2,3,4])
ans =
     5     8    11
```

Simple. MATLAB even provides an easy way of accessing the last element of a vector.

```
>> v(end)
ans =
    14
```

MATLAB also has powerful indexing procedures for matrices. To demonstrate some of these indexing routines, first enter



title page
contents
previous page
next page
back
print doc
close
exit

```
>> A=magic(6)
A =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11
```

to create a so-called "magic" matrix, one whose rows, columns, and diagonals all have identical sums.[2]. To examine the value of the entry in the third row, second column, simply enter

```
>> A(3,2)
ans =
     9
```

Indexing makes it easy to edit an individual entry in your matrix.

```
>> A(3,2)=100
A =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31   100     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11
```

You can access much more than an individual entry. For example, suppose you wanted to store the data lying in the first two rows and last three columns in the variable *D*.

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

[2] Type `sum(A,1)`, `sum(A,2)`, `sum(diag(A))`, and `sum(diag(fliplr(A)))` to find the sum of each column, row, main diagonal, and off diagonal, respectively



title page
contents
previous page
next page
back
print doc
close
exit

```
    26    19    24
    21    23    25
```

MATLAB provides a special indexing technique to access an entire row or column of a matrix. For example, if you want to store the third row of the matrix $A$ in the list $v$, enter

```
>> v=A(3,:)
v =
    31   100     2    22    27    20
```

Similarly, you can store the contents of the fourth column of the matrix $A$ in the list $w$ with

```
>> w=A(:,4)
w =
    26
    21
    22
    17
    12
    13
```

One final example shows how to store the fourth and fifth columns of the matrix $A$ in the variable $C$.

```
>> C=A(:,[4,5])
C =
    26    19
    21    23
    22    27
    17    10
    12    14
    13    18
```

Veteran MATLAB users pronounce this last construct “every row, fourth and fifth column.”



title page
contents
previous page
next page
back
print doc
close
exit

## The Workspace

Typing whos at the MATLAB prompt reveals the variables in MATLAB's workspace.[3]

```
>> whos
  Name        Size            Bytes  Class

  A           6x6               288  double array
  B           2x2                32  double array
  C           6x2                96  double array
  D           2x3                48  double array
  a           1x1                 8  double array
  a1          3x1                24  double array
  a2          3x1                24  double array
  a3          3x1                24  double array
  alpha       1x1                 8  double array
  u           1x4                32  double array
  v           1x6                48  double array
  w           6x1                48  double array
  x           3x1                24  double array

Grand total is 88 elements using 704 bytes
```

The whos command reveals the name, size, class, and number of bytes required to store each variable currently registered in your command window workspace. The information provided by the whos command is particularly useful when deciphering error messages you encounter when you ask MATLAB to do something you shouldn't.

You can examine the contents of any variable in your workspace by typing its name, as in

```
>> D
D =
```

[3] Your workspace may look slightly different, depending on the variables you've initialize during your current MATLAB session.

```
    26     19     24
    21     23     25
```

but you can also browse the contents of your workspace by clicking on the Workspace Browser icon on the MATLAB toolbar. Once the Workspace Browser window opens, recent versions of MATLAB allow you to edit the contents of a variable in your workspace interactively, by opening the variable in the MATLAB editor.

You can clear the variable `A` from your workspace with the command `clear A` and you can clear all variables from your workspace with the command `clear`.[4]

If you want to save your workspace for another day, then select `Save Workspace` from the `File` menu. When you return to your computer, you can load any saved workspace with the `Load Workspace` command from the `File` menu.

## Exercises

1. Enter `ones(2), ones(3), ones(2,5)` and examine the output. Write a sentence or two describing the output of each of the the commands.

   a. `2*ones(1000)`

   b. `-3*ones(1000,1)`

2. Enter the matrices

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 1 \\ 5 & 5 \end{bmatrix}, \quad \text{ans} \quad D = \begin{bmatrix} -1 & 1 & 3 \\ 2 & 2 & 2 \end{bmatrix}$$

   into MATLAB's workspace and evaluate each of the following expressions. Explain any error messages that occur.

   a. `2*A+3*C`
   b. `A-4*D`
   c. `B^2`
   d. `B.^2`
   e. `A*B`
   f. `B*A`
   g. `C*D`
   h. `C.*D`
   i. `A*B+D`

3. Consider the scalars $x_1 = -2$, $x_2 = 1$, $x_3 = -4$, and the vectors

[4] For a complete description of the `clear` command, enter `help clear` at the MATLAB prompt.



title page
contents
previous page
next page
back
print doc
close
exit

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{a}_3 = \begin{bmatrix} -1 \\ 3 \\ -3 \end{bmatrix}.$$

a. Use MATLAB to calculate the linear combination $x_1\mathbf{a}_1 + x_2\mathbf{v}_2 + x_3\mathbf{a}_3$.

b. Use MATLAB to evaluate the matrix-vector product $A\mathbf{x}$ that will yield the same answer found in part (a).

4. Consider the scalars $x_1 = -2$, $x_2 = 1$, $x_3 = -4$, $x_4 = 5$ and the vectors

$$\mathbf{a}_1 = \begin{bmatrix} -1 \\ 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} -2 \\ 0 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{a}_3 = \begin{bmatrix} -1 \\ 2 \\ 3 \\ -3 \end{bmatrix}, \quad \mathbf{a}_4 = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}.$$

a. Use MATLAB to calculate the linear combination $x_1\mathbf{a}_1 + x_2\mathbf{v}_2 + x_3\mathbf{a}_3 + x_4\mathbf{a}_4$.

b. Use MATLAB to evaluate the matrix-vector product $A\mathbf{x}$ that will yield the same answer found in part (a).

5. Consider the matrix and vectors

$$A = \begin{bmatrix} 1 & -2 \\ 2 & 2 \end{bmatrix}, \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{v}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

a. Compute the matrix-vector products $A\mathbf{v}_1$ and $A\mathbf{v}_2$.

b. Form the matrix $V = [\mathbf{v}_1, \mathbf{v}_2]$ and compute the matrix product $AV$. Write a short paragraph expressing your findings when you compare this result with those found in part (a).

6. Consider the matrix and vectors

$$A = \begin{bmatrix} 1 & -2 & 2 \\ 2 & 1 & 2 \\ 3 & -4 & 0 \end{bmatrix}, \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ -1 \\ 3 \end{bmatrix}.$$

a. Compute the matrix-vector products $A\mathbf{v}_1$ and $A\mathbf{v}_2$.

b. Form the matrix $V = [\mathbf{v}_1, \mathbf{v}_2]$ and compute the matrix product $AV$. Write a short paragraph expressing your findings when you compare this result with those found in part (a).

7. Consider the scalars $x_1 = -2$, $x_2 = 3$, $x_3 = 2$, and the row vectors

$$\mathbf{v}_1 = [1, 2, 3], \quad \mathbf{v}_2 = [0, -4, 5], \quad \text{and} \quad \mathbf{v}_3 = [-1, -1, 2].$$

a. Use MATLAB to calculate the linear combination $x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + x_3\mathbf{v}_3$.

b. Form the row vector `x=[x1,x2,x3]` and the matrix `V=[v1;v2;v3]` and calculate the vector-matrix product `x*V`. Write a short paragraph explaining what happens when you multiply a matrix on the left by a row vector.

8. Create a $5\times5$ diagonal matrix with `D=diag([2,4,6,8,10])` and a $5\times5$ matrix of ones with `A=ones(5)`.

a. Use MATLAB to calculate both $AD$ and $DA$.

b. What happens when you multiply a matrix on the right by a diagonal matrix? On the left?

9. Let

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & -2 \\ 0 & -1 \end{bmatrix}.$$

Which of the following is a valid identity for the given matrices?

a. $AB = BA$
b. $(AB)^2 = A^2B^2$
c. $(A+B)^2 = A^2 + 2AB + B^2$
d. $(A+B)(A-B) = A^2 - B^2$

10. Let

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \quad \text{and} \quad O = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that $A + O = O + A = A$. This is why $O$ is called the *zero matrix* or the *additive identity*. If $a$ and $b$ are real numbers and $ab = 0$, then either $a = 0$ or $b = 0$. This is **not** the case with matrices. Find a non-zero matrix $B$ so that $AB = O$. Verify your result with MATLAB.

11. Enter `eye(2), eye(3), eye(4)` at the MATLAB prompt. Write a short sentence describing the matrix output by the command `eye(100)`.

12. Enter `format rat, H=hilb(5), I=eye(5)`, then calculate `H*I` and `I*H` and examine the results. Write a short sentence explaining why $I$ is called the *identity matrix*.

13. You saw that MATLAB's transpose operator will convert a row vector to a column vector, and vice-versa. Create the matrix $B$ with `A=1:7, B=repmat(A,5,1)`. Take the transpose of matrix $B$ with `B.'` and write a sentence or two explaining the result.

14. Enter the following matrices in MATLAB's workspace.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & -4 \\ -3 & 2 \end{bmatrix}$$

Enter the real number, $\alpha = -3$, then use MATLAB to check the validity of each of the following identities for the given matrices.

a. $(A+B)^T = A^T + B^T$
b. $(A-B)^T = A^T - B^T$
c. $(AB)^T = A^T B^T$
d. $(AB)^T = B^T A^T$
e. $(\alpha A)^T = \alpha A^T$
f. $(A^T)^T = A$

Which of properties (a)–(f) appears to be a valid property of the transpose?

15. A matrix $A$, having the property $A^T = A$, is called a *symmetric matrix*. Generate a matrix $A$ with `B=pascal(5), A=B+B.'` and show that $A$ is symmetric.

    a. Write a short sentence or two explaining why the word symmetric is used to describe this class of matrices.

    b. Enter matrix $A$ with `A=floor(10*rand(6))`. Which of the following is a symmetric matrix?

    i. $A - A^T$
    ii. $A + A^T$
    iii. $AA^T$
    iv. $A^T A$

16. A matrix $A$ having the property $A = -A^T$, is called a *skew-symmetric* matrix. Enter the matrix $A$ with `B=ceil(10*rand(6)), A=A-A.'`.

a. Show that $A$ is a skew-symmetric matrix.

b. Repeat the command `B=ceil(10*rand(6)), A=A-A.'` to generate a number of different skew-symmetric matrices. What property does the diagonal of a skew-symmetric matrix possess?

17. The conjugate of the matrix $A$, $\overline{A}$, is created by taking the conjugate of each entry of the matrix $A$. The operator * is traditionally used to denote the *conjugate transpose* of a matrix; i.e., $A^* = \overline{A}^T$. MATLAB uses ' for the conjugate transpose operator *. Enter `A=rand(2)+i*rand(2)` and note that the resulting matrix has complex entries.

a. Take the conjugate transpose of $A$ with `conj(A).'`.

b. Take the conjugate transpose of $A$ with `A'`. Does this give the same result as that found in part (a)?

18. Create a vector **v** with `v=1:5:51`. Test your response to each of the following questions by entering your command at the MATLAB prompt.

a. What command will reveal the length of the vector **v**?

b. What command will reveal the third entry of the vector **v**?

c. What command will change the second entry to 115?

d. Explain what happens when you enter `v(1:2:end)`.

e. What command will pick off all of the entries in the even positions of the vector **v**?

f. Explain what happens when you enter `v(20)=999`.

19. Create a matrix $A$ with `A=magic(6)`. Test your response to each of the following questions at the MATLAB prompt.

a. What command will reveal the dimensions of the matrix $A$?

b. What command will reveal the entry in the second row, fifth column of the matrix $A$?

c. What command will store the contents entries lying in the first two rows **and** the first two columns in a matrix $B$?

d. What command will store the fourth row in a vector **v**?

e. What command will store the second column in a vector **w**?

f. What command will store the fifth and sixth columns in a matrix $C$?

g. Explain what happens when you enter `A(7,9)=0`?

20. Create the matrices $A$ and $B$ with `A=magic(3)` and `B=pascal(3)`

a. Use the appropriate indexing command to store the columns of the matrix `B` in the vectors `v1`, `v2`, and `v3`.

b. Execute each of the commands `A*v1`, `A*v2`, and `A*v3` and compare these individual results with the matrix product `A*B`. Compare the output of this matrix product with the previous matrix-column products and write a sentence or two explaining the connection between the two.

c. Use the appropriate indexing command to store the rows of the matrix `a` in the vectors `r1`, `r2`, and `r3`.

d. Execute each of the commands `r1*B`, `r2*B`, and `r3*B` and compare these individual results with the matrix product `A*B`. Compare the output of this matrix product with the previous row-matrix products and write a sentence or two explaining the connection between the two.

21. Enter the vectors `x=[1,2,3,4,5]` and `y=[6,7,8,9,10]`. Use array operations to answer each of the following questions. Verify your response at the MATLAB prompt. Note: In the questions that follow, the notation $x_j$ represents the $j$ th entry of the vector **x**, etc.

a. Create a vector **v** whose $j$ th entry is $v_j = 2x_j - 3y_j$.

b. Create a vector **v** whose $j$ th entry is $v_j = x_j^2$.

c. Create a vector **v** whose $j$ th entry is $x_j / y_j$.

d. Create a vector $\mathbf{v}$ whose $j$ th entry is $1/y_j$.

e. Create a vector $\mathbf{v}$ whose $j$ th entry is $x_j + 2$.

f. Create a vector $\mathbf{v}$ whose $j$ th entry is $x_j^2 - 2x_j + 7$.

g. Create a vector $\mathbf{v}$ whose $j$ th entry is $y_j e^{y_j}$.

h. Create a vector $\mathbf{v}$ whose $j$ th entry is $\cos 2x_j + 3 \sin y_j$.

22. Create two matrices $X$ and $Y$ with the command `[X,Y]=meshgrid([1,2,3,4,5])` . Use array operations to answer each of the following questions. Verify your response at the MATLAB prompt. Note: In the questions that follow, the notation $x_{ij}$ represents the entry in the $i$ th row and $j$ th column of the matrix $X$, etc.

a. Create a matrix $A$ whose $ij$ th entry is $a_{ij} = 2x_{ij} - 3y_{ij}$.

b. Create a matrix $A$ whose $ij$ th entry is $a_{ij} = x_{ij}^2$.

c. Create a matrix $A$ whose $ij$ th entry is $a_{ij} = y_{ij} - 3$.

d. Create a matrix $A$ whose $ij$ th entry is $a_{ij} = x_{ij}^2 + y_{ij}^2$.

e. Create a matrix $A$ whose $ij$ th entry is $a_{ij} = e^{x_{ij}^2 + y_{ij}^2}$.

23. Enter `whos` at the MATLAB prompt and examine the variables in your workspace. You can use the `clear` command to clear any individual variable from your workspace. For example, if `A` is a variable in your workspace you wish to remove, enter `clear A` at the MATLAB prompt. Enter `whos` and note that the variable `A` is gone from the workspace. Next, select `save Workspace as` from the `File` menu, choose a directory or folder in the manner required by your operating system, name the file `test.mat` and click `OK`. After that, clear all variables from your workspace with the command `clear all` and investigate the result of this command with the command `whos`. Your workspace should now be empty. Finally, select `Load Workspace` from the `File` menu, browse to the appropriate folder and select `test.mat`, then click `OK`. Enter `whos` at the MATLAB prompt and note that your workspace variables have been returned.