



# Matlab and Vectors

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 vectors in Matlab. Operations involving vectors and scalars will be discussed. *Prerequisites: None*

next page

close

exit

# Table of Contents

## Introduction

Working with Matlab

## Entering Row Vectors

Ranges

Suppressing Output

Matlab's Workspace

## Entering Column Vectors

The Transpose Operator

Indexing Vectors

## Vector Operations

Vector-Scalar Operations

Vector-Vector Operations

Elementwise Operations

More Complicated Expressions

## Homework Assignment

Matlab  
and  
Vectors

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 Row Vectors

Entering row vectors is easy in Matlab. Enter the following command at the Matlab prompt<sup>1</sup>.

Matlab  
and  
Vectors

title page

contents

previous page

next page

back

print doc

close

exit

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

There are a number of Matlab constructs introduced in this command. To enter a vector, first enter a left bracket, the elements of the vector separated by spaces, and a closing right bracket. You can also use commas to delimit the entries of your vector.

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

The = sign is Matlab's assignment operator. You use this operator to assign Matlab items to variables. To see that the row vector [1,2,3] has been assigned to the variable v, enter the following command at the Matlab prompt.

```
>> v
```

## Ranges

Sometime you will need to enter a vector of equally spaced entries. This is easily accomplished in Matlab with the construct `start:increment:finish`. If you don't supply an increment, Matlab assumes that the increment is 1.

```
>> x1=0:10
```

You can select your own increment.

```
>> x2=0:2:10
```

You can even go backwards.

```
>> x3=10:-2:0
```

And you can get very fancy indeed.

```
>> x4=0:pi/2:2*pi
```

<sup>1</sup>The >> is the Matlab prompt. You must enter what comes after the prompt. Then you must hit the Enter key to execute the command.

There will be times, particularly when plotting functions, that you will need a large number of entries in a vector.

```
>> x=0:.1:10
```

## Suppressing Output

You can suppress the output of a Matlab command by appending a semicolon.

```
>> x=0:.1:10;
```

This is particularly useful when the output is large and you have no desire to see it.

## Matlab's Workspace

You can get a listing of your variables in your workspace at any time by issuing the following command at the Matlab prompt.

```
>> who
```

You can get even more information about the variables in your workspace by entering

```
>> whos
```

Note that the size of each of the variables in your workspace is given. You can also find the size of the vector  $v$  by entering

```
>> size(v)
```

The information supplied by the result of this last command indicates that the vector  $v$  has 1 row and 3 columns. Although you can think of the vector  $v$  as a matrix with 1 row and 3 columns, you can also think of it as a row vector with length 3. Try the following command at the Matlab prompt<sup>2</sup>.

---

<sup>2</sup>You can easily get help on any of Matlab's commands. For example, enter `help length` at the Matlab prompt to get help on the `length` command.

```
>> length(v)
```

## Entering Column Vectors

Entering column vectors is also easy in Matlab. Enter the following command at the Matlab prompt.

```
>> w=[4;5;6]
```

Note that semicolons delimit the rows of a column vector. Try the following commands.

```
>> w  
>> who  
>> whos  
>> size(w)
```

The information supplied by the result of this last command indicates that the vector  $w$  has 3 rows and 1 column. Although you can think of the vector  $w$  as a matrix with 3 rows and 1 column, you can also think of it as a column vector with length 3. Try the following command at the Matlab prompt.

```
>> length(w)
```

## The Transpose Operator

You can use Matlab's transpose operator (a single apostrophe) to change a row vector to a column vector.

```
>> y=(1:10)'
```

Or a column vector to a row vector.

```
>> y=y'
```

## Indexing Vectors

Once you have defined a vector, you can easily access any of its elements with Matlab's powerful indexing commands. For example, enter the following vector.

```
>> x=[10,13,19,23,27,31,39,43,51]
```

Now try each of the following commands.

```
>> x(2)
```

```
>> x(7)
```

You can easily change an element of a vector.

```
>> x(6)=100
```

You can also access a range of elements.

```
>> x([1,3,5])
```

```
>> x(1:3)
```

```
>> x(1:2:length(x))
```

## Vector Operations

There are numerous operations involving vectors and scalars that are handled easily in Matlab.

### Vector-Scalar Operations

Scalar-vector operations are straightforward. Technically, you cannot add or subtract a scalar from a vector. However, this is such a common operation that Matlab allows it. Thus, if  $y$  is a vector, the command  $y+2$  will add 2 to each entry in the vector. Try each of the following Matlab commands and study the resulting output.

```
>> y=1:5
```

```
>> y+2
```

```
>> y-2
```

```
>> 2*y
```

```
>> y/2
```

Of course, these operations are equally valid when using column vectors.

```
>> w=(1:3:20)'  
>> w+3  
>> w-11  
>> .1*w  
>> w/10
```

## Vector-Vector Operations

First enter the following vectors.

```
>> a=1:3  
>> b=4:6
```

Adding and subtracting vectors is natural and easy. Try the following Matlab commands<sup>3</sup>.

```
>> a,b,a+b  
>> a,b,a-b
```

Of course, these operations are equally valid when using column vectors.

```
>> a=(1:3)',b=(4:6)'  
>> a+b,a-b
```

However, you can arrive at unexpected results if you do not remember that Matlab is a *matrix* environment.

```
>> a,b,a*b
```

This last command leads to an error because `*` is Matlab's symbol for *matrix* multiplication and you are not using it properly<sup>4</sup>.

<sup>3</sup> Because no semicolons are present to prevent suppression of output, the command `a,b,a+b` will display the vector `a`, display the vector `b`, then display the vector `a+b`.

<sup>4</sup> We'll look at Matlab's matrix multiplication operator in later activities.

You can also get into trouble if you attempt to add (or subtract) vectors of unequal lengths.

```
>> a=1:3,b=4:7,a+b
```

## Elementwise Operations

If you wish to multiply the vectors  $a$  and  $b$  on an *element-by-element* basis, you enter the following Matlab command.

```
>> a=(1:3)',b=(4:6)'  
>> a,b,a.*b
```

The symbol `.*` is Matlab's elementwise multiplication operator. The output is computed by multiplying the first elements of the vectors  $a$  and  $b$ , then the second elements of each vector, etc.

Matlab's elementwise division operator, `./`, works in much the same manner.

```
>> a,b,a./b
```

If you wish to exponentiate on an element-by-element basis, use `.^`.

```
>> a,a.^2
```

## More Complicated Expressions

With a little practice you will soon learn to evaluate quite complicated expressions. Suppose, for example, that you wish to evaluate the expression  $x^2 - 2x - 3$  for values of  $x$  ranging from 1 through 10, in increments of 1.

```
>> x=1:10  
>> y=x.^2-2*x-3
```

Take a moment to check these results with a calculator.

Let's look at another example. Suppose that you would like to evaluate the expression  $\sin x/x$  for values of  $x$  ranging from  $-1$  to  $1$  in increments of  $0.1$ <sup>5</sup>.

```
>> x=-1:.1:1  
>> y=sin(x)./x
```

Check these results with a calculator.

Matlab's elementwise operators will also work on column vectors.

```
>> xdata=(1:10)'  
>> xdata.^2
```

## Homework Assignment

*Answer questions 1-9 on regular notebook paper. Be sure to test your responses at the computer.*

- Write the Matlab commands that generate each of the following vectors.
  - $\begin{pmatrix} 1 \\ 2 \\ -3 \end{pmatrix}$
  - $(1, 2, -1, 3)$
  - A column vector containing all the odd numbers from 1 to 1000.
  - A row vector containing all of the even numbers from 2 to 1000.
- If  $x=0:2:20$ , write the Matlab command that will square each entry of  $x$ .
- If  $x=[0, 1, 4, 9, 16, 25]$ , write the Matlab command that will take the square root of each entry of  $x$ .

<sup>5</sup>To examine a list of Matlab's elementary functions, type `help elfun` at the Matlab prompt.

4. If  $x=0: .1:1$ , write the Matlab command that will raise each entry of  $x$  to the power of  $2/3$ .
5. If  $x=0:\pi/2:2*\pi$ , write the Matlab command that will take the cosine of each entry of  $x$ .
6. If  $x=-1: .1:1$ , write the Matlab command that will take the inverse sine of each entry of  $x$ .
7. If  $x=-1: .1:1$ , write the Matlab command that will take the exponential of each entry of  $x$ .
8. If  $x=linspace(0,2*\pi,1000)$ , what is the 50th entry of  $x$ ? What is the length of  $x$ ?
9. If  $k=0:100$ , what is the 12th entry of  $y=0.5.^k$ ?
10. Obtain a printout of the graph resulting from the following sequence of commands.

```
t=linspace(0,2*pi,1000);  
x=2*cos(3*t);  
y=3*sin(5*t);  
plot(x,y)  
xlabel('x-axis')  
ylabel('y-axis')  
title('The graph of  $x = 2*\cos(3*t)$ ,  $y = 3*\sin(5*t)$ .')
```

Finally, just for fun, try the following command sequence.

```
close all, comet(x,y)
```