Basic Commands

Table of Contents

Here are some basic commands to get you started.

Help!

One of the most useful and useless commands in Matlab is the `help` command. It is sometimes useful because it will give you what you want. For example

```
help sin
```

```
         SIN    Sine of argument in radians.
            SIN(X) is the sine of the elements of X.
 
            See also ASIN, SIND.
 
            Overloaded methods:
               sym/sin
               codistributed/sin
               gpuArray/sin
 
            Reference page in Help browser
               doc sin
```

lets you see that the `sin` command assumes the argument is in radians and also, under `See also` tells you that you might be interested in `asin` and `sind`. You can probably guess what they do. That's useful! On the other hand it's useless because for example

```
help diff
```

```
         DIFF Difference and approximate derivative.
            DIFF(X), for a vector X, is [X(2)-X(1)  X(3)-X(2) ... X(n)-X(n-1)].
            DIFF(X), for a matrix X, is the matrix of row differences,
               [X(2:n,:) - X(1:n-1,:)].
            DIFF(X), for an N-D array X, is the difference along the first
               non-singleton dimension of X.
            DIFF(X,N) is the N-th order difference along the first non-singleton
               dimension (denote it by DIM). If N >= size(X,DIM), DIFF takes
               successive differences along the next non-singleton dimension.
            DIFF(X,N,DIM) is the Nth difference function along dimension DIM.
               If N >= size(X,DIM), DIFF returns an empty array.
 
            Examples:
               h = .001; x = 0:h:pi;
```

```
   diff(sin(x.^2))/h is an approximation to 2*cos(x.^2).*x
   diff((1:10).^2) is 3:2:19

   If X = [3 7 5
           0 9 2]
   then diff(X,1,1) is [-3 2 -3], diff(X,1,2) is [4 -2
                                                   9 -7],
   diff(X,2,2) is the 2nd order difference along the dimension 2, and
   diff(X,3,2) is the empty matrix.

See also GRADIENT, SUM, PROD.

Overloaded methods:
   sym/diff
   fints/diff
   gpuArray/diff
   umat/diff
   iddata/diff

Reference page in Help browser
   doc diff
```

doesn't tell you anything particularly useful about the `diff` command, which we'll use a lot.

But the `help` command is always worth a shot, try it before you try anything else

Arithmetic and More

Previously you did `2+2` and no doubt you were suitably impressed. Here are some other calculations to get you warmed up.

```
10*20
```

```
ans =

   200
```

```
sqrt(2)
```

```
ans =

    1.4142
```

```
2^7
```

```
ans =

   128
```

```
sin(2*pi/7)


        ans =

            0.7818


log(3)


        ans =

            1.0986
```

Notice that results are returned as decimals which means you often get just an approximation as is the case with `sqrt(2)` and the `sin` and `log` functions.

Also notice that the `log` function represents the natural logarithm (the `help` command would tell you that). Computer scientists and mathematicians differ on notation sometimes and this is one of those places. If you want Matlab to do base 10 logarithms you must use `log10`.

Can you figure out what the following commands do? Try them all!

```
factorial(5)
round(pi*6)
primes(40)
factor(600)
gcd(120,90)
date
```

Precision and Other Display Stuff

Whenever you are performing mathematical computations on a computer, you must realize that some rounding of values is necessarily involved. A computer stores each value in a relatively small *finite* region of memory. How could you possibly store a value like *pi*, which has an infinite non-repeating decimal expansion, in a small space? You can't. So computers will necessarily round messy numbers. Some programming languages round more than others, but calculations in Matlab are handled fairly precisely. What we mean is that each numerical value is stored using a lot of digits. For example, calculations involving the constant `pi` use the value `3.141592653589793` instead of `3.14` or `3.1416`.

However, the internal precision of values in Matlab will not always get displayed when Matlab shows you an answer. Try the following:

```
pi


        ans =

            3.1416
```

This answer might seem like a disappointing level of precision. The good news is that the internal value really is stored very precisely, but Matlab assumes that most people just don't want to see all of those ugly digits, so the value is rounded before it is displayed. If you really want to see all of the digits in your answers to computations, try the following:

```
format long
pi
```

```
ans =

   3.141592653589793
```

Aha! There are those digits. If you want to go back to showing fewer digits in Matlab's answers, just use:

```
format short
```

Don't forget that values used in computations are always represented internally using full precision, regardless of whether you specify the long or short format. The format command only affects the way the answer is displayed on the screen.

```
format compact
```

This command suppresses extra line feeds and keeps your Matlab output, from then on, more compact and with no extra spaces between lines. If you want to return it to normal use

```
format loose
```

Published with MATLAB® 8.0