T. Henson

Math 7 Elementary Linear Algebra
INTRODUCTION TO MATLAB
3. POLYNOMIALS IN MATLAB

Commands learned:

- Defining a polynomial in MatLab
- polyval
- polyfit

DEFINING A POLYNOMIAL

Defining a Polynomial as an Matrix of *y*-Values. There are three different ways in which you can define a polynomial in MatLab. You will learn two of them in this introduction to MatLab.

One method is to use the same process we used for entering the equation of a line (which after all, is a polynomial): define a domain and enter an expression for the polynomial, which generates a matrix of values. However, the syntax for defining the expression is a little different from what you use when typing expressions into your graphing calculator.

In MatLab, the expression $x \wedge 3$ is interpreted as multiplication of a matrix named *x* by itself three times, *i.e.*, $x \wedge 3 = x \bullet x \bullet x$. However, the matrix must be square to carry out this operation. Recall from the MatLab document on graphing linear equations that the variable *x* is the name of the set of *x* coordinates (the *x*-interval), and is not a square matrix. If you think about the way you evaluate a polynomial, what you actually want to do is raise each entry in the matrix *x* to the third power. To accomplish this, use the notation $x. \wedge 3$ (use the period key on the keyboard).

Enter the lines below to create a "polynomial" matrix whose entries are the values of the polynomial $y = x^4 - 4x^3 + 10$ (in other words, a matrix of *y* values) over the interval (domain) $[-2, 5]$, using an *x*-increment of 0.1. **Note** the use of a semicolon at the end of each line to suppress the output.

```
x = [-2 : 0.1 : 5];
y = x. ^ 4 - 4 * x. ^ 3 + 10;
```

If all you want to do is graph a polynomial, then this method of defining the polynomial is fine. You can now graph the polynomial by simply entering the plot command

```
plot(x, y)
```

NOTE: It is important to understand that what you have actually created is simply a set of (x, y) coordinates of points on the graph of the polynomial. The polynomial is not a symbolic expression.

Defining a Polynomial as a Coefficient Matrix. There are some disadvantages to defining a polynomial as a matrix of *y* values. To take advantage of the way MatLab represents mathematical objects (as matrices) we can use an alternative approach that allows us to make use of a built-in polynomial evaluation command called **polyval**.

First, to define the polynomial $y = x^4 - 4x^3 + 10$, we enter a matrix. From left to right, the entries in the matrix are the coefficients of the polynomial, with the polynomial arranged in descending order. Any missing powers of the variable must be represented by 0. So the polynomial $p(x) = x^4 - 4x^3 + 10$ is defined by

$$p = [1 \quad -4 \quad 0 \quad 0 \quad 10]$$

The zero entries represent the missing powers of *x*: x^2 and x.

Evaluating a Polynomial Defined By a Coefficient Matrix. Having defined the polynomial as the coefficient matrix *p*, you can use the MatLab command **polyval** to evaluate the polynomial for a specific value of *x* or for a range of *x* values.

The general syntax for this command is: polyval$(p, xvalue)$, where *p* is the name of the polynomial and *xvalue* is the value of *x* or a matrix of *x* values.

Example 1: To evaluate the polynomial $p(x) = x^4 - 4x^3 + 10$ at $x = 3$, enter

polyval$(p, 3)$

Example 2: To evaluate the polynomial $p(x) = x^4 - 4x^3 + 10$ over the set of *x* values $\{-2, -1, 0, 1, 2, 3, 4, 5\}$, enter the following lines below. The semicolons have been omitted so you can see the output.

xval $= [-2:5]$
polyval$(p, 3)$

(NOTE: the colon between the -2 and 5 tells MatLab to create a matrix of values starting at -2 and ending at 5. The default increment is 1.)

Graphing a Polynomial Defined By a Coefficient Matrix. Having defined the polynomial by the coefficient matrix *p*, to graph the polynomial, you must create matrices of *x* and *y* values that MatLab can plot. To do this,

- First define a matrix for the *x*-interval: $x = [-2:0.1:5]$;
- Then define a matrix for the corresponding *y* values by using polyval:

$$y = \text{polyval}(p, x);$$

Be sure you end the lines with semicolons to suppress the output.

Once you defined the matrices of x and y values, use plot:

$$\text{plot}(x, y)$$

NOTE: The **polyval** command can be nested inside the plot command, allowing you to skip the step of defining the matrix of *y* values:

$$\text{plot}\left(x, \text{polyval}\left(p, x\right)\right)$$

The third method of defining a polynomial in MatLab uses the Symbolic Toolbox to create a symbolic expression which acts like an algebraic object. However, we will not be making use of this method at this time.

Polynomial Curve Fitting. The MatLab function **polyfit** is used to find the coefficients of a polynomial *p* of degree *n*. If *X* and *Y* are row (or column) matrices containing the *x* and *y* coordinates of the known points, and *n* is the degree of the polynomial to be found, then

$$p = \text{polyfit}\left(X, Y, n\right)$$

will return a matrix containing the coefficients of the polynomial *p*. This polynomial can be evaluated using **polyval** and graphed as discussed above.

Example: find a third degree polynomial that fits the points

$$\left(1, 5.5\right), \left(2, 43.\right), \left(3, 128\right), \left(4, 290.7\right), \left(5, 498.4\right)$$

SOLUTION:

Define matrices *X* and *Y* containing the *x* and *y* coordinates, respectively, of the points given:

$X = \left[1:5\right]$

$Y = \left[5.5 \quad 43.1 \quad 128 \quad 290.7 \quad 498.4\right]$

Use **polyfit**, with $n = 3$, to find the coefficients of the third degree polynomial that best fits this set of points:

$$p = \text{polyfit}\left(X, Y, 3\right)$$

COMMENT: You can use $p = \text{rats}\left(\text{polyfit}\left(X, Y, 3\right)\right)$ if you would like to express the coefficients in rational form. However, this form is not recommended when graphing.

REMINDER: the entries in the matrix *p* are the coefficients of the third degree polynomial in descending order. Thus the polynomial is

$$p\left(x\right) = -0.1917x^3 + 31.5821x^2 - 60.3262x + 35.34$$

To see how well this polynomial fits the data, you can plot the original set of points along with the polynomial. To plot the polynomial, define matrices of *x* and *y* values to be plotted by entering:

$x1 = [1:0.1:5];$

$y1 = \text{polyval}(p, x1);$

Now plot the original points and polynomial together. The ‘o’ entry in the plot command below will plot the original set of points as open circles. Without this entry, MatLab would plot the points and connect them with straight lines.

$\text{plot}(X, Y, 'o', x1, y1)$