Math 436
Fall 2008
November 5

Polynomials in MATLAB

1 Polynomial Evaluation

A polynomial is completely known by its coefficients! For example the polynomial

$$p(x) = x^3 - 3x^2 + 1$$

has coefficients, beginning with the largest power, $a_1 = 1$, $a_2 = -3$, $a_3 = 0$ and $a_4 = 1$. Once these numbers are given, it is plain that $p(1) = -1$ or $p(4) = 17$ or $p(-2) = -19$. MATLAB lets you compute polynomials with the `polyval` command. It has the structure

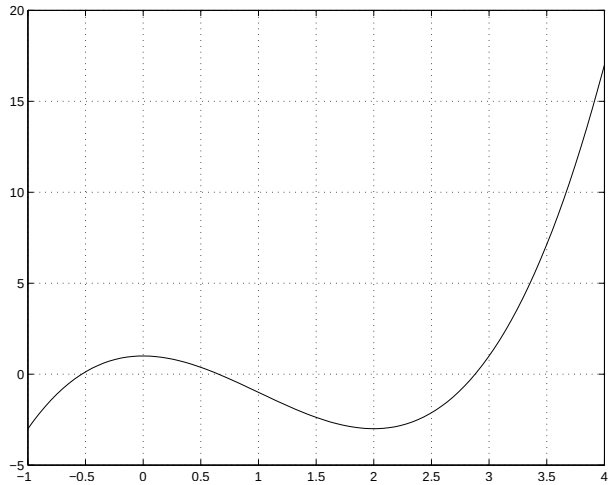
`polyval(coefficient_vector, input_data)`

In our case, we could enter

```
>> a = [1 -3 0 1]
a =
       1             -3              0              1
>> polyval(a,[1 4 -2])
ans =
      -1             17            -19
```

As you can see, we can calculate the three values simultaneously entering a vector as the second argument. For a plot of the curve on the interval $[-1, 4]$ one could proceed as follows:

```
>> a = [1 -3 0 1]
a =
       1             -3              0              1
>> x=linspace(-1,4);    % generate x data
>> p = polyval(a,x);    % calculate polynomial values
>> plot(x,p); grid on   % graph the function
```



2 Roots

It is known that a polynomial of degree n has exactly n, possibly complex roots. Here you have to account for the multiplicity of the roots as well. The `roots` command determines the roots from the coefficients. For our polynomial $p(x) = x^3 - 3x^2 + 1$ we find

```
>> roots([1 -3 0 1])
ans =
    2.8794
    0.6527
   -0.5321
```

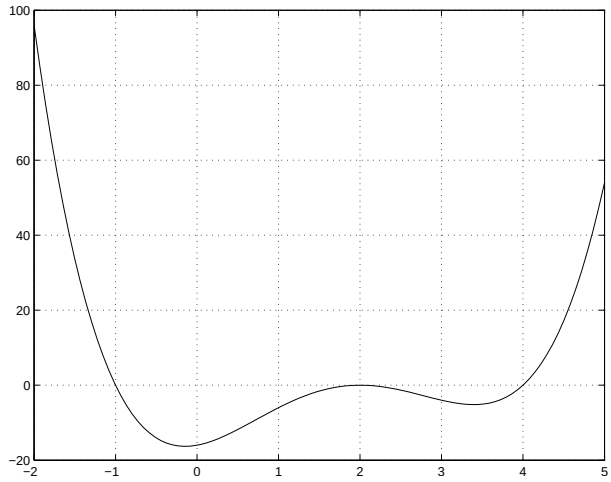
which is consistent with the information from the graph.

The reverse of the `roots` command is `poly`. Here you can easily calculate the coefficients of a polynomial with roots at desired locations. Suppose you want to find a polynomial with simple roots at -1 and 4, and a double root at 2. With `poly` you just need to enter

```
>> poly([-1 4 2 2])
ans =
     1    -7    12     4   -16
```

The desired polynomial is $p(x) = x^4 - 7x^3 + 12x^2 + 4x - 16$. A graph confirms the result:

```
>> a = poly([-1 4 2 2]);
>> x = linspace(-2,5);
>> p = polyval(a,x);
>> plot(x,p), grid on
```



3 Polynomial Interpolation

Polynomial interpolation is built into MATLAB. The command is

$$\texttt{a} = \texttt{polyfit}(\texttt{x\_data}, \texttt{y\_data}, \texttt{degree\_of\_the\_polynomial})$$

It returns the coefficients of the interpolating polynomial. From class there is exactly one polynomial of degree $\leq N-1$ which passes through N given points. MATLAB lets you *approximate* with lower

degree polynomials in the least squares sense. For instance if you take the degree to be one, it will find the least squares regression line. Our usual example with the data

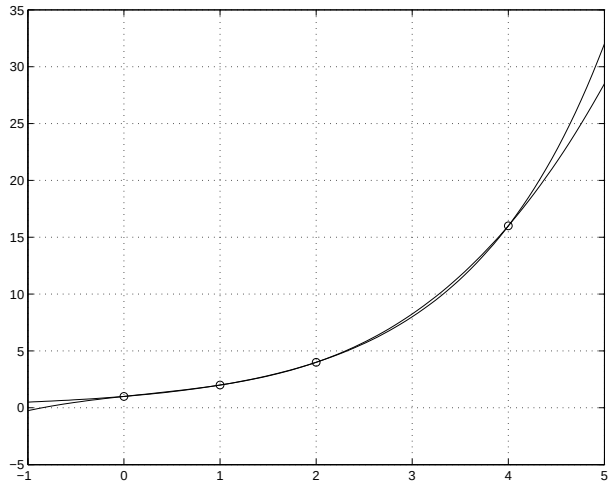
x	0	1	2	4
y	1	2	4	16

can be treated as follows (I switched to `format rat`)

```
>> X = [0 1 2 4], Y = [1 2 4 16]
X =
       0              1              2              4
Y =
       1              2              4             16
>> a = polyfit(X,Y,3)
a =
       5/24          -1/8           11/12           1
>> polyval(a,3)  %  p(3)
ans =
      33/4
```

The vector `a` contains the coefficients of the polynomial, i.e. the interpolating polynomial is $P_4(x) = \frac{5}{24}x^2 - \frac{1}{8}x^2 + \frac{11}{12}x + 1$ and its value at 3 is $P_4(3) = 33/4 = 8.25$. For a graph of the interpolating polynomial along with the underlying function $f(x) = 2^x$ and the data points on the interval $[-1, 5]$ continue as follows

```
>> x=linspace(-1,5);  % generate x values
>> f = 2.^x;          % underlying function
>> p = polyval(a,x);  % interpolating polynomial
>> plot(X,Y,'o',x,f,x,p), grid on   %  graph
```

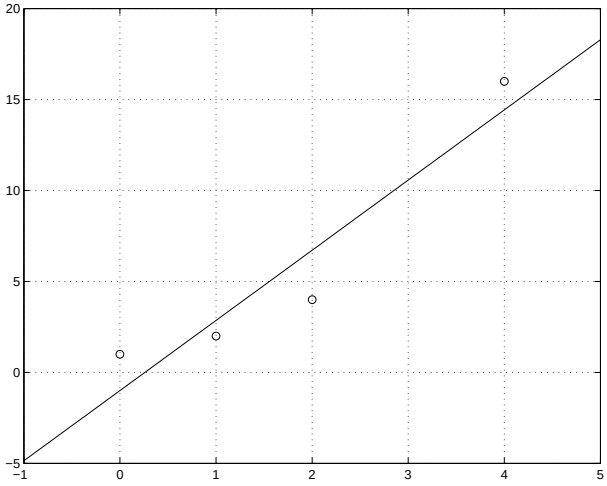


3.1 Least Squares Line

Suppose you want to find the least squares regression line for the data. Here you are looking to a polynomial of degree one to approximate the data `X` and `Y`, and we enter 1 for the degree in `polyfit` (here it is assumed the definitions for the vectors `X` and `Y` are still in effect).

```
>> b = polyfit(X,Y,1)
b =
    3.8571   -1.0000
>> y = polyval(b,x);           % for a plot
>> plot(x,y,X,Y,'o'), grid on
```

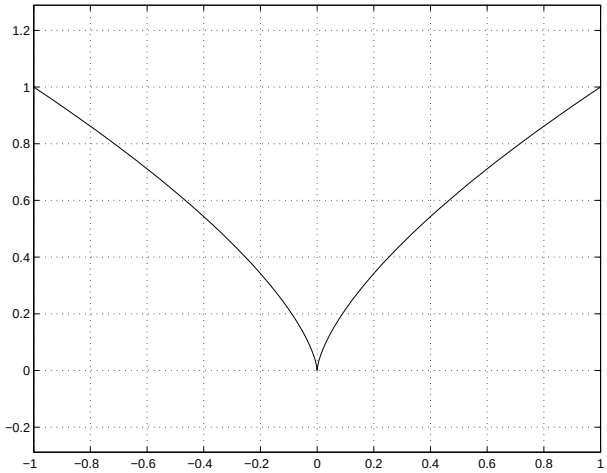
Thus the least square regression line is $y = 3.8571x - 1$.



4 Parametric Curves

For parametric curves $(x(t), y(t)$ you need to display the x-values versus the y with t as a parameter. For example, the curve $x(t) = t^3$ and $y(t) = t^2$ with $-1 \leq t \leq 1$ can be generated as follows:

```
>> t = linspace(-1,1);
>> x = polyval([1 0 0 0],t);
>> y = polyval([1 0 0],t);
>> plot(x,y), grid on, axis('equal')
```



This, of course, is the graph of the function $y = \sqrt[3]{x^2}$.