Philip Rego Project 2

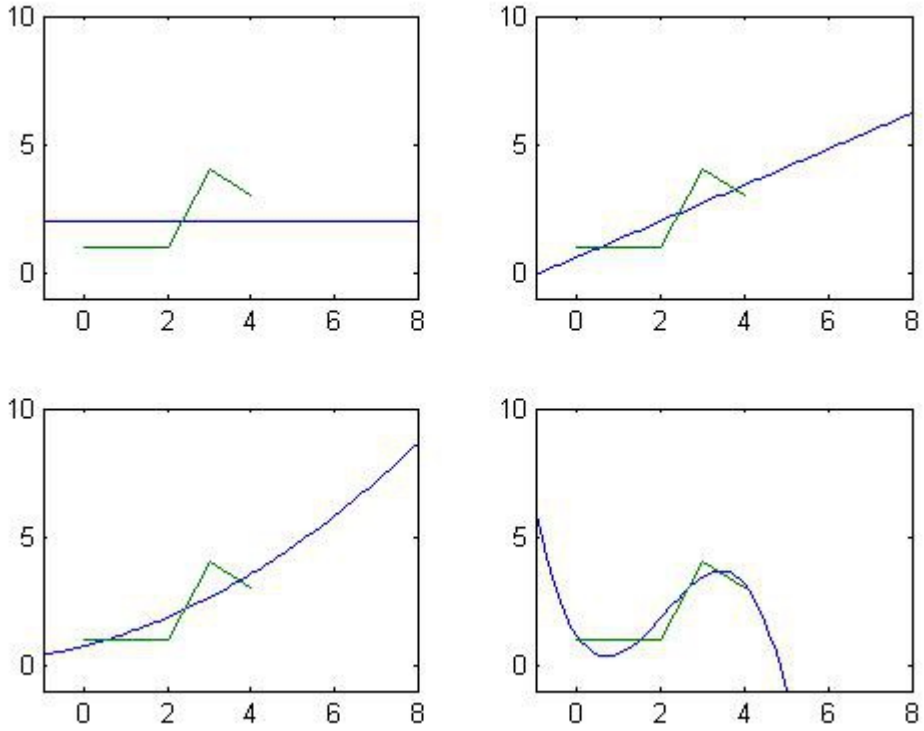
Code Explanation
The leastSquares function starts by building a matrix with n columns of x. If the basis is monomial then each column will have the function .^(i-1) applied to itself. If the basis is Legendre then 'a' and 'b' are selected, and Legendre is applied to each element with j and t as input. I wrote my own Legendre function, the implementation is recursive and straightforward. Finally R'*R*c = R'*b is calculated to Ax=b, and it's solved with Gaussian elimination with partial pivoting. C And the condition number are returned.

Display_Least_Squares is the main function. First it calculates the least squares with monomials for the given points from the first part and graphs them on the same page. The green line goes through the given points. Next, the c's and condition numbers are calculated for the second part. Finally the relative error is calculated.

Analysis
The condition number is a very good indicator of the accuracy of the solutions. As the condition number rises so does the error. As 'n' increases the condition number grows and the results become less accurate. The Legendre function is clearly the better basis function. The Legendre function gives the correct 'c' for the first two elements [1.5 .5] and gives very little error, less than 1^-15 for the rest of the c's. Conversely, the monomial function doesn't give the correct value for the first two c's and rest of the values are much higher then Legendre's. The condition numbers for Legendre are low <5 while the monomial's get very large.

For the first degree of the Legendre function matlab calculates .5, and the exact value is .5, but when calculating the error the answer is nonzero. Perhaps matlab calculates the difference before storing the number. I couldn't get the relative error past the first degree because the exact solution has 0s. I looked at the absolute errors to see how they correspond to the condition number.

Monomial Least Squares (0, 1),(1, 1),(2, 1),(3, 4),(4, 3)



```
>> Display_Least_Squares
```
The condition numbers and c's for n 1 to 4 monomials
```
conditionNo =
     1
one =
     2
conditionNo =
   4.738720018687268
two =
   0.600000000000000
   0.700000000000000
conditionNo =
  27.112831810234844
three =
   0.742857142857143
   0.414285714285715
   0.071428571428571
conditionNo =
     2.196193738089254e+02
four =
   1.142857142857158
  -2.452380952381060
   2.071428571428643
  -0.333333333333345
```
Condition numbers and c's for the second part
```
conditionNo =
     6.435041115113421e+02
cM5 =
   1.000000000000018
   0.999999999999562
   0.000000000002053
  -0.000000000003235
   0.000000000001621
conditionNo =
     6.270238221988386e+05
cM9 =
   1.000000000064264
   0.999999993561445
   0.000000119618848
  -0.000000888038407
   0.000003320893622
  -0.000006847956991
   0.000007900868713
  -0.000004778804547
   0.000001179834462
conditionNo =
     6.784275332692783e+08
```

```
cM13 =
   1.000000008638006
   0.999996723818991
   0.000143287034453
  -0.002462419558893
   0.022192755212847
  -0.119541611067897
   0.412323541785298
  -0.943528076010938
   1.449135775879531
  -1.477214218758130
   0.958676353757357
  -0.358541622050801
   0.058819504676923
conditionNo =
         2.82049993706709
cL5 =
                    1.5
                    0.5
      6.24211603859681e-17
     -1.49647859791453e-16
     -7.88482465030553e-18
conditionNo =
         3.94071292268996
cL9 =
                    1.5
                    0.5
      5.02165560456391e-17
     -1.51491946102498e-16
     -3.05742464879645e-17
      1.78650142667887e-16
      2.56563659721851e-17
     -1.53208051570309e-16
      1.05032096714451e-16
conditionNo =
          4.9250447922908
cL13 =
                    1.5
                    0.5
       4.5142500426875e-17
     -1.46824587397638e-16
     -3.98769539162697e-17
      1.85286415421375e-16
      1.18746309295994e-17
     -1.4565824744563e-16
      8.64967402017315e-17
      3.65787124906277e-16
      7.70263963537493e-17
     -3.72304483682114e-16
```

1.12109436373103e-19

The relative errors

L5Error =

1.0e-15 *

0

0.222044604925031

Inf

Inf

Inf

L9Error =

1.0e-15 *

0

0.222044604925031

Inf

Inf

Inf

Inf

Inf

Inf

Inf

L13Error =

1.0e-15 *

0

0.222044604925031

Inf

Inf

Inf

Inf

Inf

Inf

Inf

Inf

Inf

Inf

Inf

M5Error =

1.0e-12 *

0.018429702208778

0.438316050122012

Inf

Inf

Inf

M9Error =

1.0e-08 *

0.006426370546819

0.643855468940302

Inf

```
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
M13Error =
   1.0e-05 *
   0.000863800586615
   0.327618100870453
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
           Inf
```