Math 640-720 Technical Project

Texas A&M University
Instructor: Dr. Boggess
December 4, 2006

by
Jayne Overgard
jayneo@tamu.edu

Math 640-720 Technical Project
Table of Contents

Project Task #1 .... p. 3
Linear Equations

Project Task #2 .... p. 6
Least Squares

Project Task #3 .... p. 9
Least Squares in L^2

Project Task #4 .... p. 12
Signal Filtering

Project Task #5 .... p. 16
Compression

Appendix: Matlab commands .... p. 18

Task #1 .... p. 18

Task #2 .... p. 18

Task #3 .... p. 19

Task #4 .... p. 21

Task #5 .... p. 22

Project Task #1 - Linear Equations

1. Find all solutions to the set of linear equations, Ax = b. Then find a basis for the kernel of A and a basis for the range of A. What is the rank of A?

NOTE: See appendix for significant Matlab computer code used to solve this and every task.

$$A = \begin{pmatrix} 1 & -4 & 3 & 2 & 9 & -1 \\ 2 & 0 & 1 & 0 & -3 & 4 \\ 8 & 0 & -2 & 3 & 7 & 4 \\ -6 & -4 & 5.5 & -1 & 0.5 & -3 \end{pmatrix} \qquad b = \begin{pmatrix} 16.8 \\ -5.9 \\ -11.9 \\ 25.75 \end{pmatrix}$$

To find the rank of A I used the Matlab commands rref(A) and rank(A). Rref(A) produces the reduced row echelon form of A. The results of both commands are shown below:

$$\text{rref(A)} = \begin{pmatrix} 1.0000 & 0 & 0 & .2500 & .0833 & 1.0000 \\ 0 & 1.0000 & 0 & -.8125 & -4.6042 & 2.0000 \\ 0 & 0 & 1.0000 & -.5000 & -3.1667 & 2.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \text{rank(A)} = 3$$

Furthermore, regarding rref(A), since there are three pivots the rank is 3. There are three free variables (corresponding to the non-pivot columns); there are three dependent or basic variables.

Here I used a modified matrix A|b, and used rref on it in Matlab. This is the result:

$$\text{rref} = \begin{pmatrix} 1.0000 & 0 & 0 & .2500 & .0833 & 1.000 & -1.9750 \\ 0 & 1.0000 & 0 & -.8125 & -4.6042 & 2.0000 & -6.1563 \\ 0 & 0 & 1.0000 & -.5000 & -3.1667 & 2.0000 & -1.9500 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

To find all solutions x for Ax=b, the idea here is to solve for the basic variables in terms of the free variables.

Let the solution variable be x = $(u\ v\ w\ x\ y\ z)^T$. Then the reduced row echelon form of the augmented matrix A|b (see previous page) corresponds to the equations:

u+.25x+.0833y +z = -1.975,	so u = -1.9750 - .25x - .0833y - z
v -.8125x - 4.6042y + 2z = -6.1563,	so v = -6.1563 + .8125x + 4.6042y - 2z
w -.5x - 3.1667y + 2z = -1.95,	so w = -1.95 + .5x + 3.1667y - 2z

These lead to the solution:

$$x = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \\ \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} -1.975 - .25\cdot\mathbf{x} - .0833\mathbf{y} - \mathbf{z} \\ -6.1563 + .8125\mathbf{x} + 4.6042\mathbf{y} - 2\mathbf{z} \\ -1.95 + .5\cdot\mathbf{x} + 3.1667\mathbf{y} - 2\mathbf{z} \\ \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix}$$

There are infinitely many solutions because there are four equations but six unknowns. x, y, and z are arbitrary, and u, v, w depend on their values.

To find a basis for the range of A I use the columns of A corresponding to the pivots. So a basis for the range is:

$$\begin{pmatrix} 1 \\ 2 \\ 8 \\ -6 \end{pmatrix}, \begin{pmatrix} -4 \\ 0 \\ 0 \\ -4 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ -2 \\ 5.5 \end{pmatrix}$$

.

To find a basis for the kernel of A I used the rref command. I ended up with this:

$$\text{rref}(A) = \begin{pmatrix} 1.0000 & 0 & 0 & .2500 & .0833 & 1.0000 \\ 0 & 1.0000 & 0 & -.8125 & -4.6042 & 2.0000 \\ 0 & 0 & 1.0000 & -.5000 & -3.1667 & 2.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = U$$

This corresponds to the equations u + .25x + .0833y + z = 0, v - 8.125x - 4.6042y + 2z = 0, and w - .5x - 3.1667y + 2z = 0. The free variables are x, y, and z. Solving, a basis for the kernel of A is:

$$\begin{pmatrix} -.3333 \\ 5.4167 \\ 3.6667 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1.2500 \\ -1.1875 \\ -1.5000 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1.0833 \\ 2.6042 \\ 1.1667 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Another basis for the kernel can be found by using the null(A) command in Matlab.The result is below:

$$\begin{pmatrix} 0.6153 \\ 0.1421 \\ 0.4643 \\ -0.0695 \\ -0.2090 \\ -0.5805 \end{pmatrix}, \begin{pmatrix} 0.1515 \\ -0.1941 \\ -0.1228 \\ -0.9459 \\ 0.1560 \\ 0.0720 \end{pmatrix}, \begin{pmatrix} -0.2019 \\ 0.7859 \\ 0.4124 \\ -0.1808 \\ 0.2991 \\ 0.2222 \end{pmatrix}$$

Jayne Overgard
Math 640 Technical Project

Project Task #2 - Least Squares

2. Find the "closest least squares solution" to the equation Ax=b where

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 4 \\ 2 & 3 & -1 & 4 \\ 4 & 2 & 3 & 8 \end{pmatrix} \qquad b = \begin{pmatrix} -1 \\ 2 \\ -3 \\ 4 \\ 9 \\ 1 \end{pmatrix}$$

To find the least squares solution, first find $K = A^T*A$. In Matlab the command used is A'*A.

$$K = A^T*A = \begin{pmatrix} 1 & 1 & 0 & 0 & 2 & 4 \\ 1 & 2 & 1 & 0 & 3 & 2 \\ 0 & 1 & 3 & 1 & -1 & 3 \\ 0 & 0 & 1 & 4 & 4 & 8 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 4 \\ 2 & 3 & -1 & 4 \\ 4 & 2 & 3 & 8 \end{pmatrix} = \begin{pmatrix} 22 & 17 & 11 & 40 \\ 17 & 19 & 8 & 29 \\ 11 & 8 & 21 & 27 \\ 40 & 29 & 27 & 97 \end{pmatrix}$$

The closest least squares solution is found by solving $Kx^* = f$

$$f = A^Tb = \begin{pmatrix} 1 & 1 & 0 & 0 & 2 & 4 \\ 1 & 2 & 1 & 0 & 3 & 2 \\ 0 & 1 & 3 & 1 & -1 & 3 \\ 0 & 0 & 1 & 4 & 4 & 8 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ -3 \\ 4 \\ 9 \\ 1 \end{pmatrix} = \begin{pmatrix} 23 \\ 29 \\ -9 \\ 57 \end{pmatrix}$$

Here the Matlab command was A'*b

$$x^* = K^{-1}f = \begin{pmatrix} 0.3285 & -0.1603 & 0.0024 & -0.0882 \\ -0.1603 & 0.1750 & -0.0007 & 0.0140 \\ 0.0024 & -0.0007 & 0.0742 & -0.0214 \\ -0.0882 & 0.0140 & -0.0214 & 0.0485 \end{pmatrix} \begin{pmatrix} 23 \\ 29 \\ -9 \\ 57 \end{pmatrix} = \begin{pmatrix} -2.1427 \\ 2.1904 \\ -1.8536 \\ 1.3323 \end{pmatrix}$$

$$v^* = Ax^* = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 4 \\ 2 & 3 & -1 & 4 \\ 4 & 2 & 3 & 8 \end{pmatrix} \begin{pmatrix} -2.1427 \\ 2.1904 \\ -1.8536 \\ 1.3323 \end{pmatrix} = \begin{pmatrix} 0.0477 \\ 0.3845 \\ -2.0381 \\ 3.4756 \\ 9.4687 \\ 0.9076 \end{pmatrix}$$

A*(inv(A'*A)*(A'*b)) is the Matlab command that I used to find v*.

Is your solution, X, unique? Why? Compute the norm of the residual, ||AX - b||, where X is the solution. In Matlab, let x = X + 0.1*rand(4,1) and compute ||Ax - b||. If your answer X is correct, then ||AX - b|| should be less than or equal to ||Ax - b||.

(Note: Allow x^* = X for the work below and in Matlab.)

Yes, the solution is unique. The kernel of $K=A^TA = \{0\}$, and so by Theorem 4.8 the solution is unique .

$$K = \begin{pmatrix} 22 & 17 & 11 & 40 \\ 17 & 19 & 8 & 29 \\ 11 & 8 & 21 & 27 \\ 40 & 29 & 27 & 97 \end{pmatrix}$$

null(K) = Empty matrix: 4-by-0

To find the residual the Matlab command I used was the norm(A*(inv(A'*A)*(A'*b))-b) command.
The result was ||AX - b|| = 2.2662.

The Matlab command I used to calculate x was x = (inv(A'*A)*(A'*b))+0.1*rand(4,1)

$$\text{This returned } x = \begin{pmatrix} -2.1017 \\ 2.2798 \\ -1.8478 \\ 1.3676 \end{pmatrix}$$

||Ax - b|| = 2.4278

||AX - b|| < ||Ax - b||.

The second time I generated x using the rand command the result was x = $\begin{pmatrix} -2.0614 \\ 2.1914 \\ -1.8397 \\ 1.3526 \end{pmatrix}$

||Ax - b|| = 2.3455

||AX - b|| < ||Ax - b||.

The third time I generated x using the rand command the result was x = $\begin{pmatrix} -2.1229 \\ 2.2508 \\ -1.8264 \\ 1.3522 \end{pmatrix}$

||Ax - b|| = 2.3403

||AX - b|| < ||Ax - b||.

The fourth time I generated x using the rand command the result was x = $\begin{pmatrix} -2.1412 \\ 2.2651 \\ -1.8091 \\ 1.4255 \end{pmatrix}$

||Ax - b|| = 2.6121

||AX - b|| < ||Ax - b||.

The fifth time I generated x using the rand command the result was x = $\begin{pmatrix} -2.0961 \\ 2.2323 \\ -1.7690 \\ 1.3848 \end{pmatrix}$

||Ax - b|| = 2.5314

||AX - b|| < ||Ax - b||.

I am satisfied that my X is the correct answer, the closest least squares solution to Ax=b.

Project Task #3 - Least Squares in L^2

3. Find the closest tenth degree polynomial to sin(2pi*x) on the interval 0<=x<=1 using the $L^2[0,1]$ norm. If P(x) is your answer, then compute the residual ||P(x) - sin(2pi*x)||. Plot y = P(x) and y = sin(2pi*x) over the interval to see if they closely match with each other.

To find the closest tenth degree polynomial to sin(2pi*x) one must find $P(x)=a + bx + cx^2 + dx^3 + \ldots + jx^{10}$. We use the monomial basis, using the Hilbert matrix, 11x11. The Matlab code was most easily found using K = hilb(11).

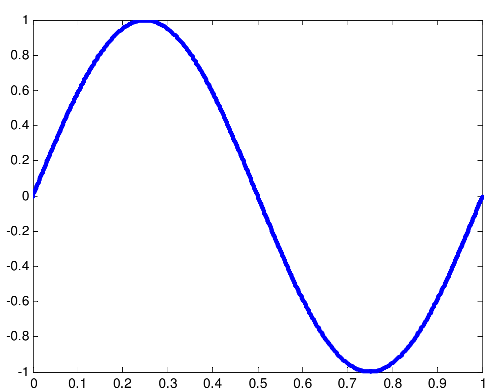
$$K = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} \\ \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} \\ \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} \\ \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} \\ \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} & \frac{1}{20} \\ \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} & \frac{1}{20} & \frac{1}{21} \end{pmatrix}$$

The idea is to find Kc =f using the $L^2[0,1]$ norm. We need to solve Kc=f. The right hand side of the equation, f, is the inner product of the x^i with $\sin(2\pi x) = \langle x^i, \sin(2\pi x)\rangle = \int_0^1 x \cdot \sin(2\pi x)\, dx$. The c's are the coefficients of the tenth degree polynomial that is the closest to $\sin(2\pi x)$ on the interval [0,1].

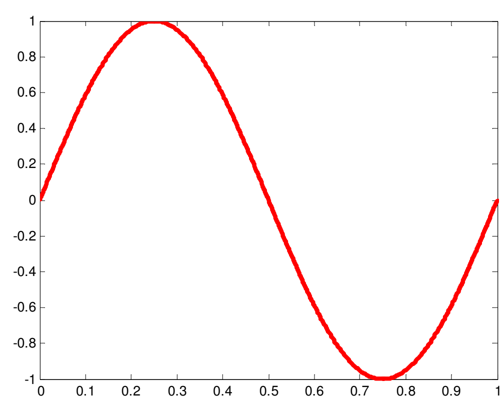
$$f = \begin{pmatrix} 0 \\ -0.1591549 \\ -0.1591549 \\ -0.1349663 \\ -0.1107776 \\ -0.0907802 \\ -0.074974 \\ -0.0625764 \\ -0.0528045 \\ -0.0450293 \\ -0.038775 \end{pmatrix} \qquad K^{-1}f = c = \begin{pmatrix} 1.7233353 \times 10^{-5} \\ 6.2809256 \\ 0.072763 \\ -42.346967 \\ 7.3942337 \\ 49.4452123 \\ 86.9064317 \\ -223.8437484 \\ 149.2600045 \\ -33.1688899 \\ -0 \end{pmatrix}$$

The coefficients are the result of solving the equation Kc = f. These are the coefficients of the closest tenth degree polynomial to sin(2πx). The result, P(x), is below:

$$P(x) = .000017233353 + 6.2809256x + 0.072763x^2 - 42.346967x^3 + 7.3942337x^4 + 49.4452123x^5 + 86.9064317x^6 - 223.8437484x^7 + 149.2600045x^8 - 33.1688899x^9 - 0x^{10}$$



y = sin(2πx)



Closest tenth degree polynomial P(x)

In order to check how close this is I calculated the residual, ||P(x) - sin(2πx)||. To do this I used several steps in Matlab. I set

```
y=P - sin(2*pi*x)
a=abs(y.^2)
b=trapz(X,a)
n=sqrt(b)
```

and got ||P(x) - sin(2πx)|| = 3.6594e-006.

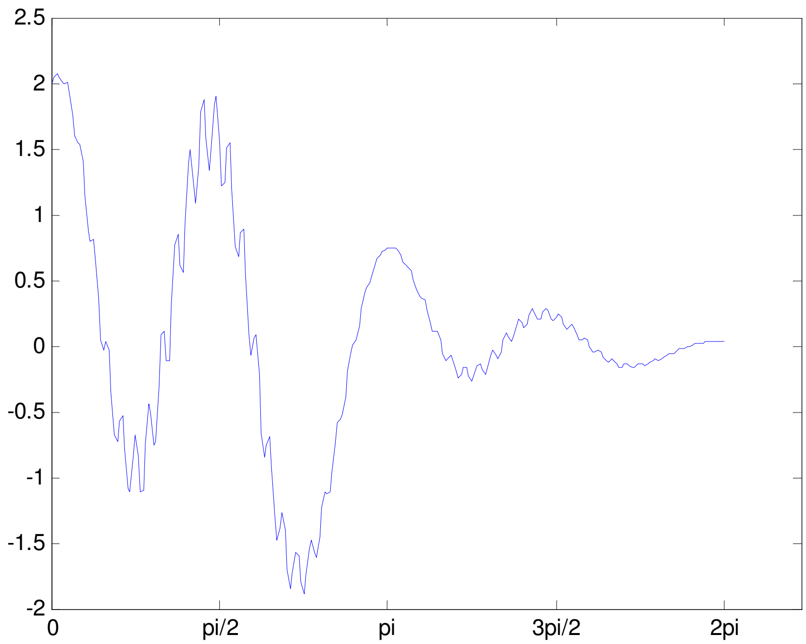
Computing the residual after adding a random number to the coefficients (c+0.1*rand(1,1) to be exact), I got
$||PR(x) - \sin(2\pi x)|| = 0.2739$
$||PR(x) - \sin(2\pi x)|| = 0.2027$
$||PR(x) - \sin(2\pi x)|| = 0.2552$
$||PR(x) - \sin(2\pi x)|| = 0.1391$
$||PR(x) - \sin(2\pi x)|| = 0.1731$

For each value my original residual was always less, and the plots are nearly identical. Therefore I am confident that the coefficients are correct.

Project Task #4 - Signal Filtering

4. Plot the following function over the interval [0,2pi].

$f(t) = e^{-t2/10}(\sin(2t) + 2\cos(4t) + 0.4\sin(t)\sin(50t))$



$f(t) = e^{-t2/10}(\sin(2t) + 2\cos(4t) + 0.4\sin(t)\sin(50t))$

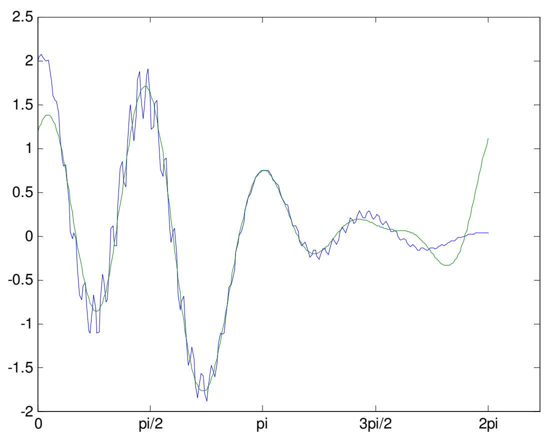
Our goal is to filter out the high frequency noise. We do this by discretizing the interval $[0,2\pi]$ into $N=2^8= 256$ intervals. **In Matlab, yhat=fft(y)**. The vector yhat contains N+1=257 entries which represent the frequency information from the vector

$y=f(t) = e^{-t2/10}(\sin(2t) + 2\cos(4t) + 0.4\sin(t)\sin(50t)$.

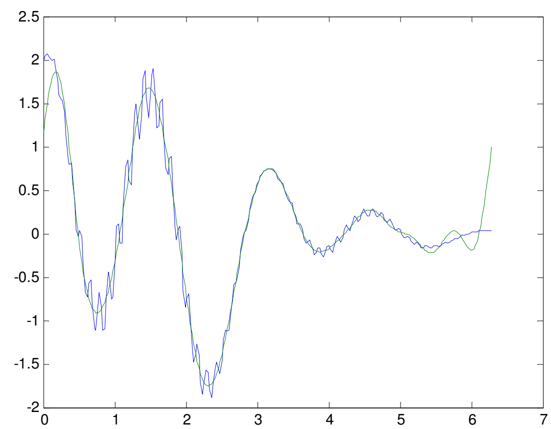
$\zeta(N-k) = \zeta(k)$; the values of $\zeta(1),...,\zeta(m),0,...,0,\zeta(N-m+1),...,\zeta(N)$ for small values of m contain the low frequency components, while the other part, $\zeta(m+1),...,\zeta(N-m)$ contains the high frequency components. We intend to remove these.

We define a new vector yfhat = [yhat(1),...,yhat(m),0,...,0,yhat(N-m+1),...yhat(N)] to remove the middle part. I used m=6 first to zero out all the the high frequency components from 7 on, leaving only 1 through 6 and 252 through 257. We define yf=ifft(yfhat) in Matlab to reverse the FFT of the vector yfhat.

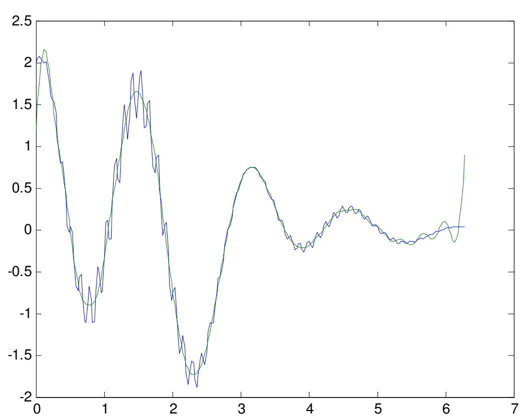
The result is shown on the graph below labeled as "m=6". The original signal is in blue the the new signal with the noise removed is in green. There are several other choices for m shown, each labeled by their m number.



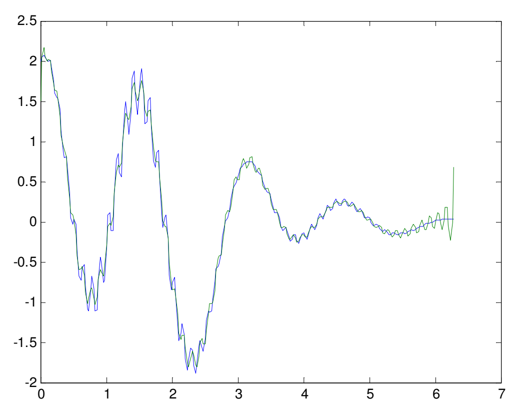
m=6



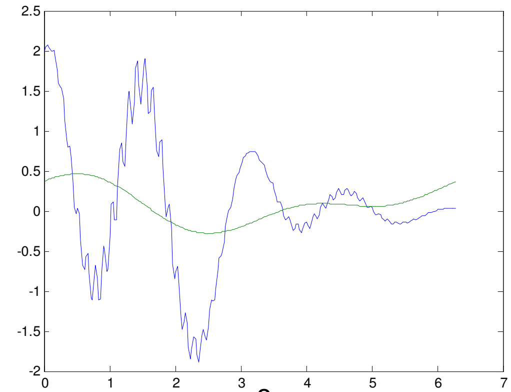
m=11



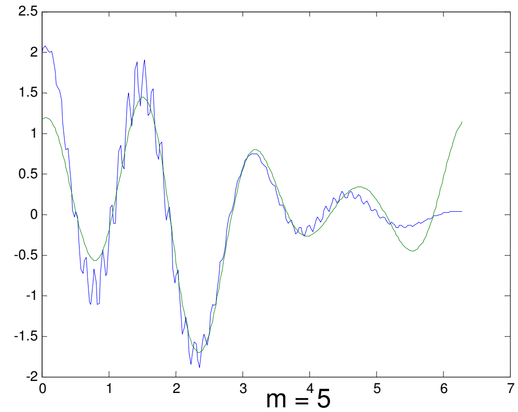
m = 20



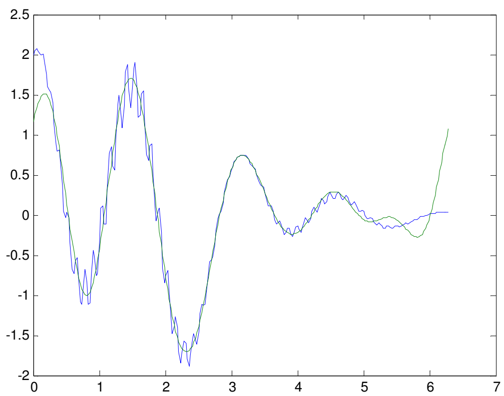
m = 50



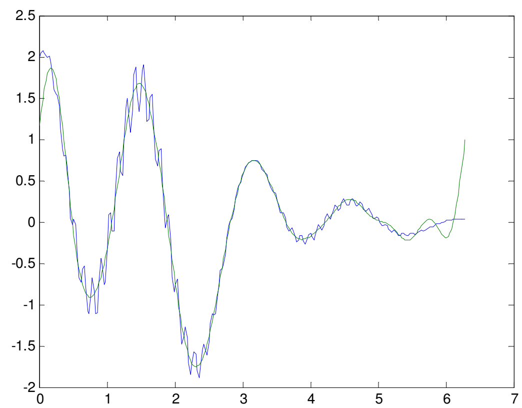
m = 2



m = 5



m=7



m=11

Analysis: The comparison graphs when $m<6$, ie: when a significant number of high frequency coefficients are zeroed out, shows too much of the original signal is lost, both high and low frequencies. When $6<m<12$ or so, the graph seems to be the best, with most of the noise gone but the integrity of the original lower frequencies of the signal intact. When $m>19$, the green line begins to get noise in it, and so while the signal is improved it still contains noise.

Near the endpoints $t = 0$ and $t = 2\pi$ the function $e^{int} = 1$ because $e^{2\pi i} = \cos(2\pi) + i\sin(2\pi) = 1$ and $e^0 = 1$. So the filtered signal misbehaviors near these endpoints.

A little more explanation of the signal filtering mathematics is warranted. The idea here, and what the Matlab fft and ifft commands did for us, is to find a trigonometric polynomial of the form $p(x) = c_0 + c_1e^{it} + \ldots + c_{n-1}e^{i(n-1)t}$ which agrees at t_j with $f(t_j) = f_j$.

Let $\omega = e^{2\pi i/n}$. We need $f_j = \sum_{k=0}^{n-1} c_{(k)}\omega^{jk}$ in the following matrix:

$$\begin{pmatrix} f_0 \\ . \\ . \\ f_{n-1} \end{pmatrix} = \begin{bmatrix} 1 & \omega & . & . & \omega^{n-1} \\ 1 & \omega^2 & . & . & \omega^{2(n-1)} \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & \omega^{n-1} & . & . & \omega^{(n-1)^2} \end{bmatrix} \begin{pmatrix} c_0 \\ . \\ . \\ c_{n-1} \end{pmatrix}$$

Solving for c ends up:

$$c = \begin{pmatrix} c_0 \\ . \\ . \\ c_{n-1} \end{pmatrix} = 1/n \begin{bmatrix} 1 & 1 & . & . & 1 \\ \frac{1}{\omega} & \frac{1}{\omega^2} & . & . & \frac{1}{\omega^{n-1}} \\ . & . & . & . & . \\ . & . & . & . & . \\ \frac{1}{\omega^{n-1}} & \frac{1}{\omega^{2(n-1)}} & . & . & \frac{1}{\omega^{(n-1)^2}} \end{bmatrix} \begin{pmatrix} f_0 \\ . \\ . \\ f_{n-1} \end{pmatrix}$$ or c = F(f), the Discrete Fourier Transform

of the data f. The inverse f = (IF)c = $(F)^{-1}c$ is called the Inverse Discrete Fourier Transform of c. The fft and ifft is an efficient way of calculating these.

Project Task #5 - Compression

5. Here we store theessential frequency information of the following function over the interval [0,2pi]. We throw away the rest.

$$f(t) = e^{-t2/10}(\sin(2t) + 2\cos(4t) + 0.4\sin(t)\sin(50t))$$

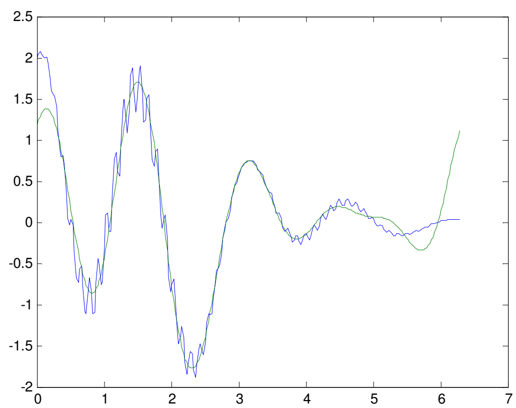
The function f(t) is defined above. Here are the other key equations also used in task #4:

y = f(t)
yhat = fft(y)

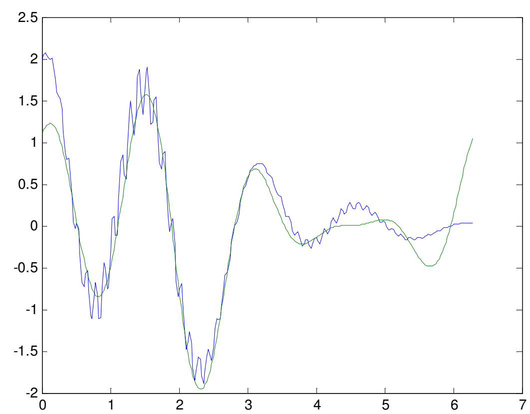
and we define a new vector, cyhat.

cyhat(j) = yhat(j) if its absolute value is larger than a given tolerance and zero otherwise.

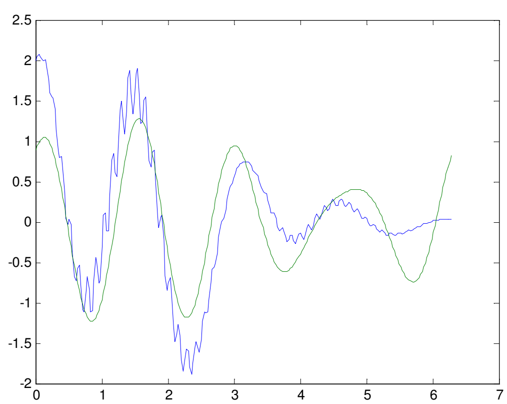
Below is a comparison to f and yc=ifft(cyhat) for different tolerances, indicated by tol=value.



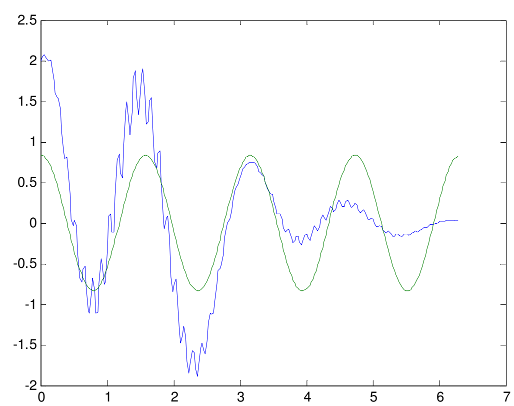
Plot of f in blue and cy=ifft(cyhat) in green, tol = 5



tol = 30



tol = 50



tol = 100

Upon evaluation of the entries of cyhat when the tolerance was 5, there were 12 nonzero entries. This corresponds to a (245/257)*100 = 95.3307% compression rate. When the tolerance was 30, there were 10 nonzero entries corresponding to a 96.1089% compression rate. At tol = 50 there were 4 nonzero entries, a 98.4436% compression rate. At tol = 100 there were 2 nonzero entries, a 99.2218% compression rate.

In computing the relative l^2 norm of the error, $(||y - yc||_t^2)/||y||_t^2$:

For tol = 5, the relative l^2 norm of the error was .3077. Compression rate = 95.3307%

For tol = 30, the relative l^2 norm of the error was 0.3471. Compression rate = 96.1089%

For tol = 50, the relative l^2 norm of the error was 0.5789. Compression rate = 98.4436%

For tol = 100, the relative l^2 norm of the error was 0.6967. Compression rate = 99.2218%

We observe that the higher the compression rate the larger the error for tol > 22. For tol < 22, the error remains at .3077.

APPENDIX

Matlab Commands

Project Task #1 - Linear Equations

	Task	Matlab Commands
1.	To enter the matrix A:	A = [1 04 3 2 9 -1;2 0 1 0 -3 4;8 0 -2 3 7 4;-6 -4 5.5 -1 .5 -3]
2.	To find the rank of A: Also rref(A) shows 3 pivots, indicating a rank of 3	rank(A)
3.	To help solve, I used rref on a modified matrix A|b.	
4.	To find a basis for the kernel of A:	null(A)

Project Task #2 - Least Squares

	Task	Matlab Commands
1.	First find $K = A^T*A$	K = A'*A
2.	Find $f = A^Tb$.	f = A'*b
3.	Solve $Kx^* = f$.	X = inv(K)*f
4.	To find $v^* = Ax^*$ $=A*K^{-1}*f$	V = A*(inv(A'*A)*(A'*b)

	Task	Matlab Commands
5.	To find $\|\|AX - b\|\| = \|\|Ax^* - b\|\|$	N = norm(A*(inv(A'*A)*(A'*b)) - b)
6.	To generate x with a small random 4 dimensional column vector added to X	x = (inv(A'*A)*(A'*b)) + 0.1*rand(4,1)
7.	To find $\|\|Ax - b\|\|$	R = norm(A*x - b)

Project Task #3 - Least Squares in L^2

	Task	Matlab Commands
1.	First find K an 11x11 Hilbert Matrix	K = hilb(11) **-or-** x=linspace(0,1,11); for i=1:11 for j=1:11 K(i,j) = 1/(i+j-1); end end
2.	$\int_0^1 \mathbf{x} \cdot \sin(2\pi \mathbf{x})\, d\mathbf{x}$ to find f = <x^i, sin(2pi*x)>	x = 0:.001:1 for n = 0:10 y = (x.^n).*sin(2*pi*x); f(n+1)=trapz(x,y) end

Verification of a correct f was accomplished by finding each entry;

```
syms t real
f1=vpa(int((t^0)*(sin(2*pi*t)),t,0,1))
f2=vpa(int((t^1)*(sin(2*pi*t)),t,0,1))
f3=vpa(int((t^2)*(sin(2*pi*t)),t,0,1))
f4=vpa(int((t^3)*(sin(2*pi*t)),t,0,1))
f5=vpa(int((t^4)*(sin(2*pi*t)),t,0,1))
f6=vpa(int((t^5)*(sin(2*pi*t)),t,0,1))
f7=vpa(int((t^6)*(sin(2*pi*t)),t,0,1))
f8=vpa(int((t^7)*(sin(2*pi*t)),t,0,1))
f9=vpa(int((t^8)*(sin(2*pi*t)),t,0,1))
f10=vpa(int((t^9)*(sin(2*pi*t)),t,0,1))
f11=vpa(int((t^10)*(sin(2*pi*t)),t,0,1))
f1=double(f1)
f2=double(f2)
f3=double(f3)
f4=double(f4)
f5=double(f5)
f6=double(f6)
f7=double(f7)
f8=double(f8)
f9=double(f9)
f10=double(f10)
f11=double(f11)
f=[f1;f2;f3;f4;f5;f6;f7;f8;f9;f10;f11]
```

3. Solve Kc = f; this is to find the polynomial coefficients. $c = K^{-1}f$

```
c=inv(K)*f
```

4. To plot y = sin(2pi*x)

```
x = linspace(0,1,1001);
plot(x, sin(2*pi*x))
```

5. To plot P(x), the closest tenth degree polynomial to y:

```
X=linspace(0,1,1001)
P=(.17233352868514957990845559e-4)
+(6.2809255569714349124163264538*X.^1) +
(.727630306782133804565043732988831e-1*X.^2) +
(-42.346967006495459247695153508487*X.^3) +
(7.3942336848582538928134118804415*X.^4) +
(49.445212325951506806281901066890*X.^5) +
(86.906431706648672910860552300746*X.^6) +
(-223.84374835410459407414159709180*X.^7) +
(149.26000447129658706558252070802*X.^8) +
(-33.168889882510352673578262630539*X.^9) +
(-.29122177869232253451546788866031e-17*X.^10)
plot(X,P)
```

6. To check the residual, ||P(X) - sin(2pi*x)||:

```
y = P - sin(2*pi*X)
a = abs(y.^2)
b = trapz(X,a)
```

```
n = sqrt(b)
```

7. To compute the residual after adding a random number to the coefficients (c + 0.1*rand(1,1)):

```
PR=((.1723335e-4+.1*rand(1,1)))
+((6.2809255+.1*rand(1,1))*X.^1)
+((.7276303e-1+.1*rand(1,1))*X.^2)
+((-42.3469670+.1*rand(1,1))*X.^3)
+((7.3942336+.1*rand(1,1))*X.^4)
+((49.4452123+.1*rand(1,1))*X.^5)
+((86.9064317+.1*rand(1,1))*X.^6)
+((-223.8437483+.1*rand(1,1))*X.^7)
+((149.2600044+.1*rand(1,1))*X.^8)
+((-33.1688898+.1*rand(1,1))*X.^9)
+((-.2912217e-17+.1*rand(1,1))*X.^10)
y = PR - sin(2*pi*X)
a = abs(y.^2)
b = trapz(X,a)
n = sqrt(b)
```

Project Task #4 - Signal Filtering

Task	Matlab Commands

1. Plot f(t)	t=linspace(0,2*pi,257);
	y= exp(-1*t.^2/10).*(sin(2*t) + 2*cos(4*t)+ 0.4*sin(t).*sin(50*t));
	plot(t,y);
	set(gca,'XTick',0:pi/2:2*pi)
	set(gca,'XTickLabel',{'0','pi/2','pi','3pi/2','2pi'})

2. Enter yhat	yhat=fft(y)

3. Enter yfhat, yf and plot y and yf on the same set of axes.	yfhat = [yhat(1:6) zeros(1,245) yhat(252:257)]
	yf=ifft(yfhat);
	plot(t,y,t,yf);
	set(gca,'XTick',0:pi/2:2*pi)
	set(gca,'XTickLabel',{'0','pi/2','pi','3pi/2','2pi'})

4. To plot the other graphs I adjusted m and N-m+1 accordingly in the yfhat matrix definition.	

Project Task #5 - Compression

Task / Matlab Commands

1. Using tol=5.0 define cyhat and plot y with yc
.4*sin(t).*sin(50*t));

```
t=linspace(0,2*pi,257);
y= exp(-1*t.^2/10).*(sin(2*t) + 2*cos(4*t) +

yhat=fft(y)
yfhat = [yhat(1:6) zeros(1,245) yhat(252:257)]
yf=ifft(yfhat);
tol=5.0
cyhat=yfhat
        for j=1:257
                if abs(cyhat(j)) < tol
                cyhat(j) = 0
                else cyhat(j) = cyhat(j)
                end
        end
yc=ifft(cyhat)
plot(t,y,t,yc)
```

2. I changed tol value several times.

3. To compute the relative l^2 norm of the error:

```
ndif=norm(y-yc)
n=norm(y)
ndif/n
```

4. To aid in the counting of the zeros and nonzeros:

```
t=linspace(0,2*pi,257);
y= exp(-1*t.^2/10).*(sin(2*t)+
2*cos(4*t)+0.4*sin(t).*sin(50*t));
yhat=fft(y)
yfhat = [yhat(1:6) zeros(1,245) yhat(252:257)]
yf=ifft(yfhat);
tol=23.0
cyhat=yfhat
counter=0
        for j=1:257
                if abs(cyhat(j)) < tol
                cyhat(j) = 0
                counter=counter + 1
                else cyhat(j) = cyhat(j)
                end
        end
yc=ifft(cyhat)
plot(t,y,t,yc)
nonzero=257-counter
counter
```

5. Percentage of compression:

```
(counter/257)*100
```