Derek Frank
dmfrank@ucsc.edu
AMS 147
3/10/10

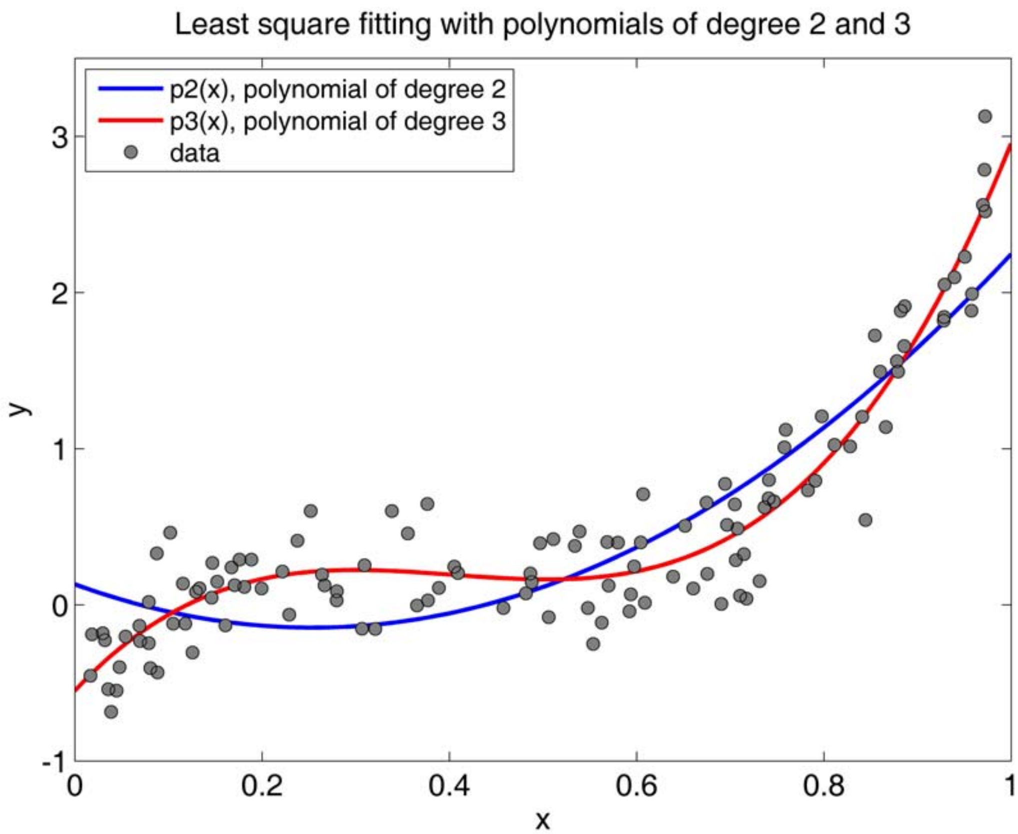
Homework #8

1.

a. In this first problem, I am to take *x* and *y* data points found in the file "data6.txt" use the least square method to fit a poylnomial of degree 2 and another of degree 3 to the given data. I am to plot these functions along with the given data.

b. Using Matlab, I will implement the least square method by trying to solve for the minimum values of *b* to satisfy $p2(x)=b_1g_1+b_2g_2+b_3g_3$ and $p3(x)=b_1g_1+b_2g_2+b_3g_3+b_4g_4$, the polynomials of degree 2 and 3 respectively, where $g_1=1$, $g_2=x$, $g_3=x^2$, and $g_4=x^3$. *x* and *y* are the values given in "data6.txt."

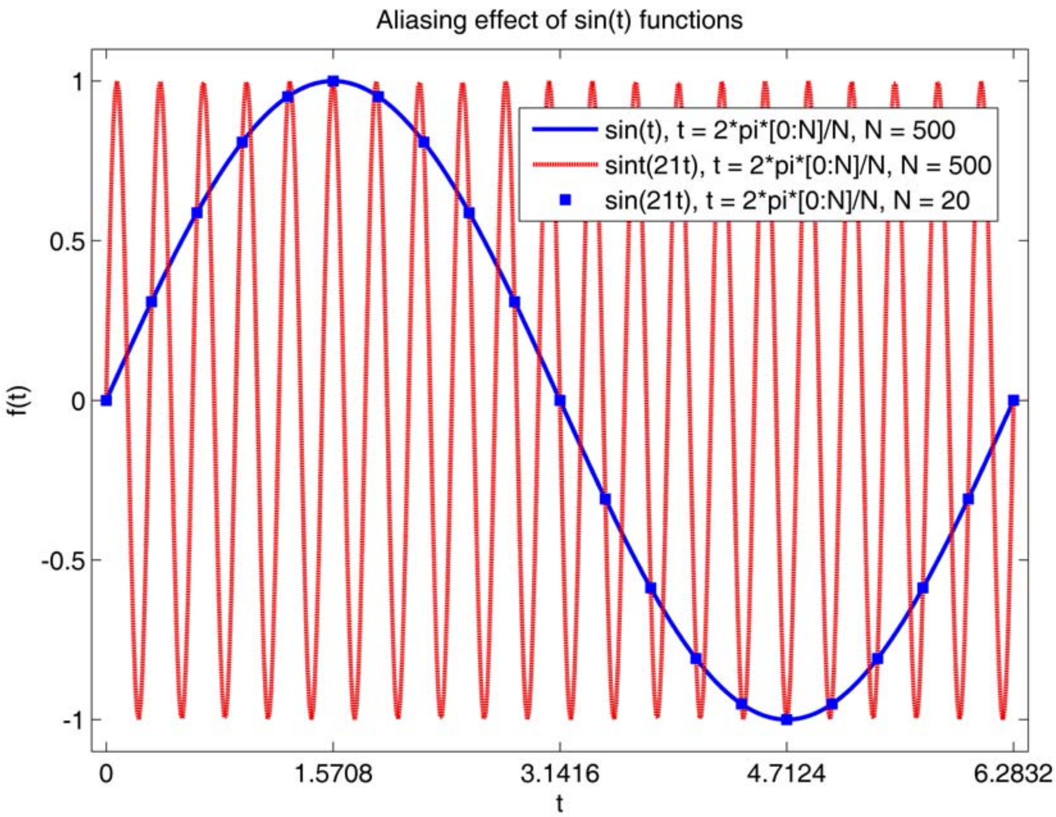
c.



d. The results show that using a linear combination of a higher degree polynomial create a more accurate fitting function. Here the least square fitting of a polynomial of degree 3 fits the data points significantly better than the least square fitting of a polynomial of degree 2. Although neither fitting function is near perfect, due to all the noise in the given data, they do a good job of following the given data on its interval.

2.

a. In this second problem. I am to plot functions *sin(t)* and *sin(21t)* on an interval for $t\epsilon[0,2\pi]$. I am to plot *sin(t)* once with 500 points and *sin(21t)* twice, once with 500 points and the other with 20 points.
b. Using Matlab, I am to implement these functions using two different time, *t*, sets. The first time contains 500 points and the second contains 20 points. Both time sets range for $t\epsilon[0,2\pi]$.
c.



d. The results show an aliasing effect. *sin(21t)* with 20 points appears the same as *sin(t)* on the same interval $t\epsilon[0,2\pi]$.

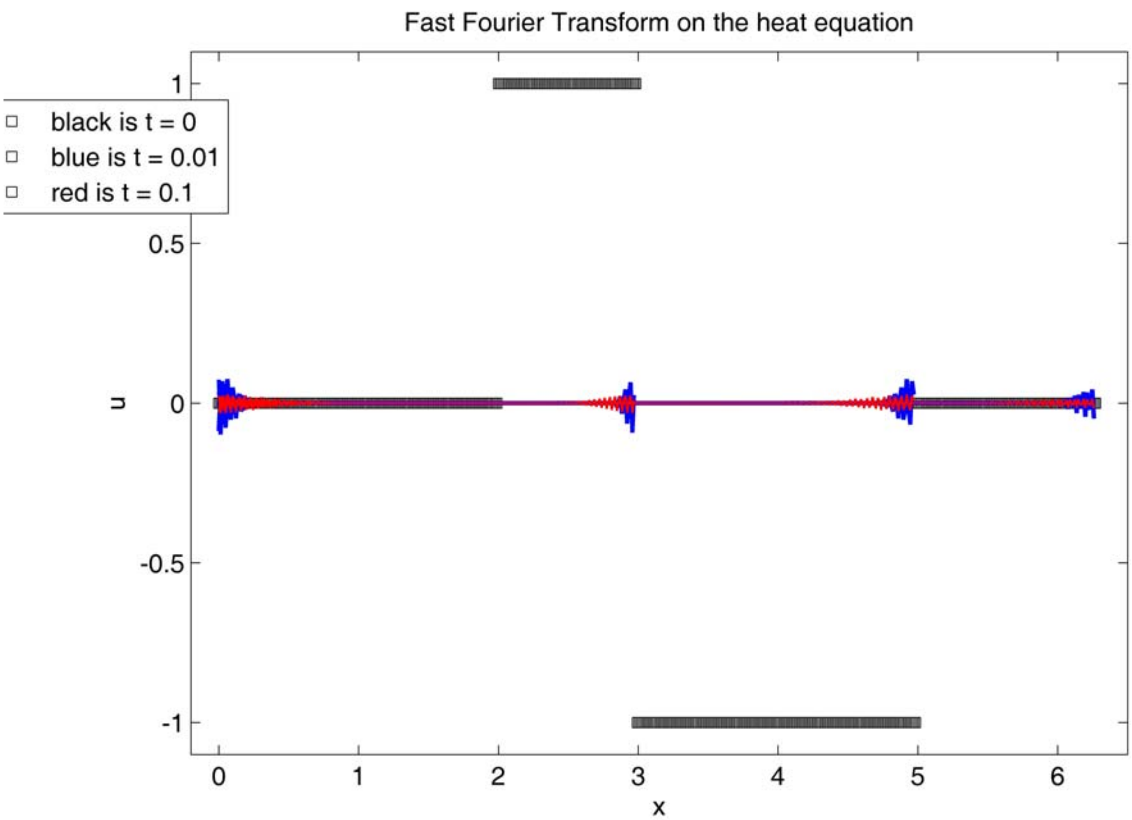
3.

a. In this final problem, I am to solve the heat equation with initial conditions:

$u_0=0, \quad x\epsilon[0,2)$
$u_0=1, \quad x\epsilon[2,3)$
$u_0=-1, \quad x\epsilon[3,5)$
$u_0=0, \quad x\epsilon[5,2\pi]$

Implementing and using the Fast Fourier Transform, I will find 512 points on the interval for $x\epsilon[0,2\pi]$ use two constant times, $t=0.01$ and $t=0.1$.
b. Using Matlab, I will implement the Fast Fourier Transform and the heat equation. I will then solve for the initial given conditions, where $t=0$, to solve for the two times.

c.



d. The results show the effect of solving the heat equation using the Fast Fourier Transform. Although the initial conditions are discontinuous, the function tries to smooth out with times greater than zero. As well, the larger the time, the smoother the plot.

Appendix:

1. "LS_p2_p3.m"

```
% This code fits g(x) = b1*g1(x)+b2*g2(x)+b3*g3(x) to the data
% where
% g1(x) = 1
% g2(x) = x
% g3(x) = x^2
% g4(x) = x^3
%
clear
figure(4)
clf reset
axes('position',[0.15,0.13,0.75,0.75])
%
load data6.txt
x=data6(:,1);
y=data6(:,2);
%
g1=ones(size(x));
```

```
g2=x;
g3=x.^2;
g4=x.^3;
G2=[g1, g2, g3];
G3=[g1, g2, g3, g4];
%
A2=G2'*G2;
A3=G3'*G3;
G2y=G2'*y;
G3y=G3'*y;
b2=A2\G2y;
b3=A3\G3y;
%
xp=[0:0.01:1];
g_2=b2(1)+b2(2)*xp+b2(3)*xp.^2;
g_3=b3(1)+b3(2)*xp+b3(3)*xp.^2+b3(4)*xp.^3;
plot(xp,g_2,'b-','linewidth',2.0)
hold on
plot(xp,g_3,'r-','linewidth',2.0)
hold on
plot(x,y,'ko','markerfacecolor',[0.5,0.5,0.5])
axis([0,1,-1.0,3.5])
set(gca,'xtick',[0:0.2:1])
set(gca,'ytick',[-1:1:3])
set(gca,'fontsize',14)
xlabel('x')
ylabel('y')
h1=legend('p2(x), polynomial of degree 2','p3(x), polynomial of
degree 3','data',2);
set(h1,'fontsize',12)
title('Least square fitting with polynomials of degree 2 and 3')
```

"data6.txt"

```
1.6721343e-02  -4.5469628e-01
1.8476811e-02  -1.8959377e-01
2.9829671e-02  -1.8130540e-01
3.2282005e-02  -2.2683302e-01
3.5818257e-02  -5.4111601e-01
3.8781999e-02  -6.8574500e-01
4.4600301e-02  -5.4881850e-01
4.7810847e-02  -3.9875197e-01
5.4283700e-02  -2.0326853e-01
6.9136405e-02  -1.3443281e-01
6.9729072e-02  -2.3241730e-01
7.9062059e-02   1.8678560e-02
7.9108419e-02  -2.4620879e-01
8.0708472e-02  -4.0604119e-01
8.7588041e-02   3.2876267e-01
8.8374466e-02  -4.3319406e-01
1.0188167e-01   4.6202129e-01
1.0515025e-01  -1.2114410e-01
1.1556552e-01   1.3587735e-01
1.1806926e-01  -1.2104627e-01
1.2568773e-01  -3.0506966e-01
1.2939403e-01   8.3160902e-02
1.3327284e-01   1.0566028e-01
1.4623570e-01   4.5854203e-02
1.4692485e-01   2.6801257e-01
1.5211340e-01   1.4851365e-01
```

1.6084705e-01	-1.3216332e-01
1.6734918e-01	2.3919576e-01
1.7070362e-01	1.2609116e-01
1.7590539e-01	2.8990790e-01
1.8110402e-01	1.1436405e-01
1.8854758e-01	2.8981638e-01
1.9941288e-01	1.0459050e-01
2.2174387e-01	2.1347103e-01
2.2908284e-01	-6.2857190e-02
2.3776679e-01	4.1061289e-01
2.5197669e-01	6.0076800e-01
2.6403344e-01	1.9256969e-01
2.6682907e-01	1.2413478e-01
2.7973318e-01	2.8158626e-02
2.7991057e-01	8.6781429e-02
3.0678418e-01	-1.5261882e-01
3.0955554e-01	2.5330464e-01
3.2100351e-01	-1.5465111e-01
3.3871821e-01	6.0087554e-01
3.5577278e-01	4.5739763e-01
3.6560044e-01	-4.1159937e-03
3.7652374e-01	6.4606188e-01
3.7724585e-01	2.8455690e-02
3.8876813e-01	1.0867237e-01
4.0538826e-01	2.4375816e-01
4.0921129e-01	2.0292506e-01
4.5785979e-01	-2.0530710e-02
4.8183968e-01	7.2610840e-02
4.8661098e-01	2.0087707e-01
4.8799801e-01	1.4399834e-01
4.9706724e-01	3.9472108e-01
5.0607910e-01	-7.8844483e-02
5.1111783e-01	4.2120651e-01
5.3401440e-01	3.7697005e-01
5.3901169e-01	4.6978344e-01
5.4807116e-01	-2.0781463e-02
5.5368413e-01	-2.5124201e-01
5.6288815e-01	-1.1438732e-01
5.6857540e-01	4.0225173e-01
5.7002458e-01	1.2371362e-01
5.8000837e-01	3.9845058e-01
5.9235541e-01	-4.2415819e-02
5.9389354e-01	6.7059576e-02
5.9740739e-01	2.4591763e-01
6.0447429e-01	3.9985614e-01
6.0693252e-01	7.0802915e-01
6.0888250e-01	1.3403008e-02
6.3900312e-01	1.8074124e-01
6.5169431e-01	5.0567648e-01
6.6044429e-01	1.0451391e-01
6.7471668e-01	6.5492388e-01
6.7562236e-01	1.9844492e-01
6.9045804e-01	5.2761253e-03
6.9452884e-01	7.7443819e-01
6.9658076e-01	5.1134889e-01
7.0496501e-01	6.4287154e-01
7.0584207e-01	2.8593087e-01
7.0786490e-01	4.8844460e-01
7.1054691e-01	5.7950080e-02

```
7.1458461e-01   3.2494658e-01
7.1732409e-01   3.9623237e-02
7.3134518e-01   1.5132860e-01
7.3646681e-01   6.2414741e-01
7.4092188e-01   6.8221555e-01
7.4138065e-01   7.9992007e-01
7.4687020e-01   6.6174480e-01
7.5787388e-01   1.0078196e+00
7.5931116e-01   1.1202550e+00
7.8305517e-01   7.3293228e-01
7.9100259e-01   7.9386963e-01
7.9773815e-01   1.2073058e+00
8.1132522e-01   1.0241088e+00
8.2808907e-01   1.0138354e+00
8.4103943e-01   1.2047647e+00
8.4458266e-01   5.4299430e-01
8.5452280e-01   1.7249862e+00
8.6001259e-01   1.4935936e+00
8.6604460e-01   1.1384665e+00
8.7811440e-01   1.5608384e+00
8.7936530e-01   1.4934267e+00
8.8212455e-01   1.8820694e+00
8.8571388e-01   1.6569570e+00
8.8633356e-01   1.9113807e+00
9.2806552e-01   1.8188949e+00
9.2853637e-01   1.8435150e+00
9.2877206e-01   2.0505207e+00
9.3950275e-01   2.0971502e+00
9.5055131e-01   2.2279939e+00
9.5764149e-01   1.8832683e+00
9.5823881e-01   1.9906070e+00
9.6998226e-01   2.5602668e+00
9.7166076e-01   2.7851760e+00
9.7220160e-01   3.1276034e+00
9.7230211e-01   2.5194295e+00
```

2. “alias_sin.m”

```
% This code plots different sin functions
% g1(t) = sin(t)        t = 2*pi*[0:N]/N, N = 500
% g2(t) = sin(21t)      t = 2*pi*[0:N]/N, N = 500
% g3(t) = sin(21t)      t = 2*pi*[0:N]/N, N = 20
%
clear
figure(5)
clf reset
axes('position',[0.15,0.13,0.75,0.75])
%
N1=500; N2=20;
t1=2*pi*[0:N1]/N1; t2=2*pi*[0:N2]/N2;
g1=sin(t1); g2=sin(21*t1); g3=sin(21*t2);
%
plot(t1,g1,'b-','linewidth',2.0)
hold on
plot(t1,g2,'r--','linewidth',2.0)
hold on
plot(t2,g3,'bs','markerfacecolor','b')
axis([-.1,2*pi+.1,-1.1,1.1])
set(gca,'xtick',[0:pi/2:7])
set(gca,'ytick',[-1:.5:1])
```

```
set(gca,'fontsize',12)
xlabel('t')
ylabel('f(t)')
h1=legend('sin(t), t = 2*pi*[0:N]/N, N = 500','sint(21t), t =
2*pi*[0:N]/N, N = 500','sin(21t), t = 2*pi*[0:N]/N, N = 20',2);
set(h1,'fontsize',12)
title('Aliasing effect of sin(t) functions')
```

3. “golden_search.m”

```
% This code uses the Fourier transform to solve
% the heat equation
%
clear
figure(6);
clf reset
axes('position',[0.15,0.13,0.75,0.75])
%
L=2*pi;
c0=4*pi^2/L^2;
%
N=512;
dx=L/N;
% N = [0,162],[163,243],[244,406],[407,511]
x=[0:N-1]*dx;
u0=[0,1,-1,0];
y=fft(u0);
ind1=[0:162/2,-162/2+1:-1];
ind2=[163/2:243/2,-243/2+1:163/2-1];
ind3=[244/2:406/2,-406/2+1:244/2-1];
ind4=[407/2:511/2,-511/2+1:407/2-1];
%
t=0.01;
y2=y(1)*exp(-c0*ind1.^2*t);
u_001_1=real(ifft(y2));
y2=y(2)*exp(-c0*ind2.^2*t);
u_001_2=real(ifft(y2));
y2=y(3)*exp(-c0*ind3.^2*t);
u_001_3=real(ifft(y2));
y2=y(4)*exp(-c0*ind4.^2*t);
u_001_4=real(ifft(y2));
%
t=0.1;
y2=y(1)*exp(-c0*ind1.^2*t);
u_01_1=real(ifft(y2));
y2=y(2)*exp(-c0*ind2.^2*t);
u_01_2=real(ifft(y2));
y2=y(3)*exp(-c0*ind3.^2*t);
u_01_3=real(ifft(y2));
y2=y(4)*exp(-c0*ind4.^2*t);
u_01_4=real(ifft(y2));
%
plot(x(1:163),u0(1),'k-','linewidth',1.0)
hold on
plot(x(164:244),u0(2),'k-','linewidth',2.0)
hold on
plot(x(245:407),u0(3),'k-','linewidth',1.0)
hold on
plot(x(408:N),u0(4),'k-','linewidth',2.0)
hold on
```

```
%
plot(x(1:162),u_001_1,'b-','linewidth',2.0)
hold on
plot(x(1:243),u_001_2,'b-','linewidth',2.0)
hold on
plot(x(1:406),u_001_3,'b-','linewidth',2.0)
hold on
plot(x(1:N-1),u_001_4,'b-','linewidth',2.0)
hold on
%
plot(x(1:162),u_01_1,'r-','linewidth',1.0)
hold on
plot(x(1:243),u_01_2,'r-','linewidth',1.0)
hold on
plot(x(1:406),u_01_3,'r-','linewidth',1.0)
hold on
plot(x(1:N-1),u_01_4,'r-','linewidth',1.0)
%
axis([-.2,6.5,-1.1,1.1])
set(gca,'fontsize',12)
set(gca,'xtick',[0:1:6])
set(gca,'ytick',[-1:.5:1])
xlabel('x')
ylabel('u')
legend('black is t = 0','blue is t = 0.01','red is t = 0.1')
title('Fast Fourier Transform on the heat equation')
```