CONTENTS

HOMEWORK 1

EXERCISE 1

Q: *Derive the Heat Equation for the rod whose heat capacitance c and density ρ vary along the axis and are thus functions of the position x.*

Solution: To derive the one-dimensional Heat Equation where the heat capacity and mass density depend on position, let us first assume that we have a non-uniform rod of length L, which is perfectly insulated and has no externally applied heat source. With this, we can generally define the change in heat, or thermal energy, over some small region Δx as

$$\Delta Q(t) = c(x)m(x)\Delta u(x,t)\Delta x,$$

where $m(x)$ and $c(x)$ are the *mass* and *heat capacity*[1] at position x, respectively, and $u(x,t)$ is the *temperature* of the rod at position x and time t. If the region Δx is sufficiently small, we can assume that the temperature over that region is constant. Further, because the mass density of the rod is $\rho(x) = m(x)/A$, where A is the cross-sectional area of the rod, the above equation can be expressed as

$$\Delta Q(t) = c(x)\rho(x)Au(x,t)\Delta x.$$

The total change in thermal energy on an arbitrary subinterval $[a,b] \subsetneq L$ then becomes

$$\frac{dQ}{dt} = \frac{d}{dt}\int_a^b Ac(x)\rho(x)u(x,t)\ dx = \int_a^b Ac(x)\rho(x)\frac{\partial u}{\partial t}(x,t)\ dx, \tag{1}$$

where differentiation and integration were interchanged due to the fact that x and t are independent variables. For the rod, the only way the temperature changes in the specified region is through the change in *flux* $\phi(x,t)$ through the boundaries of the region specified by the endpoints, where the flux is the rate of how much heat flows through A in a given

[1]The heat capacity $c(x)$ is the amount of energy required to raise the thermal energy of one mass unit by one degree at position x.

amount of time. Positive flux is the flow of thermal energy from left to right and negative flux is the flow of thermal energy from right to left. Thus, flux forms a vector field[2]. Further, because the temperature is decreasing as thermal energy flows from the left to the right, the change per time is negative. Thus,

$$\frac{dQ}{dt} = -(\phi(b,t) - \phi(a,t))A = -\phi(b,t)A + \phi(a,t)A = -A\int_a^b \frac{\partial \phi}{\partial x}(x,t)\ dx, \tag{2}$$

using the Fundamental Theorem of Calculus to rewrite the difference in integral form. In combining equations (1) and (2), we obtain

$$\frac{dQ}{dt} = \int_a^b \left(c(x)\rho(x)\frac{\partial u}{\partial x}(x,t) + \frac{\partial \phi}{\partial x}(x,t) \right)\ dx = 0. \tag{3}$$

Because the limits of integration are arbitrary, this suggests that the integrand must vanish. Thus we can write

$$c(x)\rho(x)\frac{\partial u}{\partial x}(x,t) + \frac{\partial \phi}{\partial x}(x,t) = 0. \tag{4}$$

Lastly, to rewrite equation (4) in a more suggestive manner, having only *one* unknown, we can prescribe the use of Fourier's Heat Law, which states that the flux is equal to negative of the gradient of the temperature, with a constant of proportionality, called the *thermal diffusivity*,

$$\phi(x,t) = -k(x)\nabla u.$$

where in one dimension, $\nabla u = \partial u/\partial x$, and the negative sign describes the flow of thermal energy from areas of high temperature to low temperature. In combining Fourier's Heat Law and equation (4), we can formally write the Heat Equation in the following forms

$$\boxed{\frac{\partial u}{\partial t} = K(x)\frac{\partial^2 u}{\partial x^2}(x,t), \quad K(x) = \frac{k}{c(x)\rho(x)},}$$

$$\boxed{\frac{\partial u}{\partial t} = \frac{1}{c(x)\rho(x)}\frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}\right),}$$

where the first equation is the form where only the heat capacity and mass density depend on position, whereas the second equation is the form where *all* constants depend on the position.

[2]Flux is the integral of the normal component of a vector field, integrated over some boundary:

$$\text{Flux} = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n}\ dS,$$

where $\mathbf{F}$ is the field, $\mathbf{n}$ is the outward unit normal of the boundary $\partial\Omega$, and dS is the arc-length of $\partial\Omega$.

EXERCISE 2

Q: *Attempt to derive the Heat Equation in two dimensions. If you have taken Vector Analysis (Math 152) and/or advanced physics, this should be straightforward. Otherwise you may not be able to complete the derivation but it is still instructive to try. Assume that you have an insulated metal plate. Consider the energy content of an arbitrary region and try to follow the logic of the handout. If you run into a difficulty, write a clear sentence explaining what that difficulty is so that we could discuss it in class at some point.*

Solution: In using an insulated, uniform metal plate, with side lengths L, in the absence of an external heat source, applying the above logic will lead us to the 2-dimensional Heat Equation. In this case, the change in heat capacity over a small, arbitrary region $R = \Delta x \Delta y$, through which we assume the temperature to be constant, is

$$\Delta Q(t) = c\rho u(x,t)\Delta x \Delta y,$$

where the region is finitely small, with Δx and Δy denoting the side lengths of the region. Further, the region is defined by the subintervals $[a,b] \subsetneq L$ and $[c,d] \subsetneq L$, respectively. In looking at the derivative of the thermal energy, the analog of equation (1) becomes

$$\frac{dQ}{dt} = \frac{d}{dt}\int_R c\rho u(x,t)\ dxdy = \int_R c\rho \frac{\partial u}{\partial t}(x,t)\ dxdy. \tag{5}$$

Because the amount of thermal energy only changes through the boundary of the region, due to the change in flux ϕ across the boundary, we can use the formal definition of flux, provided on Page 2, to explicitly describe the rate of change as

$$\frac{dQ}{dt} = -\int_{\partial R} \phi \cdot \mathbf{n}\ ds, \tag{6}$$

where in this case ϕ represents the vector field whose components describe the flow of thermal energy through the plate, and ∂R represents the boundary of the rectangular region. Again, combining equations (5) and (6), we obtain

$$\frac{dQ}{dt} = \int_R c\rho \frac{\partial u}{\partial t}(x,t)\ dxdy\ + \int_{\partial R} \phi \cdot \mathbf{n}\ ds = 0. \tag{7}$$

To rewrite the flux in a form that is more useful, we apply the Divergence Theorem, which states: the integral of the normal component of the flux through the boundary of a region is equal to the divergence, or flux density, over the interior of the region,

$$\int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n}\ dS = \int_R \nabla \cdot \mathbf{F}\ dV$$

where the right-hand side represents the divergence over some volume element. In applying the divergence theorem and Fourier's Heat Law to equation (7), we obtain

$$\frac{dQ}{dt} = \int\limits_R c\rho \frac{\partial u}{\partial t}(x,t)\ dxdy\ + \int\limits_R \nabla \cdot \phi\ dxdy = 0 = \int\limits_R \left(c\rho \frac{\partial u}{\partial t}(x,t) - k\nabla^2 u \right) dxdy = 0.$$

Again, because the boundary is arbitrary, the integrand must vanish and thus, we are left with the Heat Equation in two spatial dimensions,

$$\boxed{\frac{\partial u}{\partial t} = K\nabla^2 u = K\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad K = \frac{k}{c\rho}}$$

EXERCISE 3

Q: *Let $x_1 = 0 < x_2 < \ldots < x_{n1} < x_n = 1$ be an even subdivision of the unit interval $[0, 1]$. Such a subdivision can be generated with the command linspace(0,1,n) in Matlab. For any such subdivision one can construct a nonzero function f with the property:*

$$\int_{x_i}^{x_j} f(x)dx = 0, \quad \textit{for all choices } i \textit{ and } j.$$

This is trivial when n = 2 and easy when n = 3. Do it for n = 4. Provide both the equation $y = f(x)$ and its plot as your solution. Extra points if you produce a scheme for constructing such functions for arbitrary n. In which case, what happens as $n \to \infty$?

Solution: To determine $f(x)$, I guessed that $f(x) = \cos((n-1)\pi x)$, where n is the number of points in $x = \texttt{linspace}(0, 1, n)$. Thus, when $n = 2$, the points are $x_1 = 0$ and $x_2 = 1$. The reason why I chose this function was that f contains an integer multiple of the values of the nodes of the subdivisions. Thus, when f is integrated between any two points, due to the fact that

$$\int_{x_i}^{x_j} \cos((n-1)\pi x)\ dx = \left.\frac{\sin((n-1)\pi x)}{(n-1)\pi}\right|_{x_i}^{x_j} = 0, \quad \text{for all} \quad i, j \in [0, 1] \text{and for all } n,$$

this satisfies and ensures that the integral evaluated between any two subdivisions is zero. However, I definitely had some doubts about this function and I still do, even though it satisfies the conditions. I'm not sure why I still have reservations. For this function, as $n \to \infty$, nothing really changes, which is why I guess I have reservations. But, for a periodic function that depends on an integer frequency that is a $pi-$multiple of every value of the interval, it makes sense. Further, when we are asked to accomplish a specific task in the handout, there is usually an important consequence to be discovered, and from this plot, and perhaps I missed it, there was not much that I could see for higher values of n except that the frequency increases.

Commentary and Corrections: Boy was I wrong. After doing a bit more thinking, the new function that I "created," which works for all values of n, is

$$f(x) = \cos\left(\frac{\pi x}{h}\right),$$

where $h = x_2 - x_1$, the step-size. the reason why this function works is as follows:

$$\int_{x_i}^{x_j} \cos(\pi x/h)\ dx = \left.\frac{\sin(\pi x/h)}{\pi/h}\right|_{x_i}^{x_j} = 0, \quad \text{for all} \quad i, j \in [0,1] \text{ for all } n.$$

Because the function is defined using the step-size, for each x_i, x_j the division by the step-size produces integer values of π, for which sine returns zero. Here is the old code and the new code with its output for $n = 4, 6$.

```
Old Code

%% Exercise  3

L = 1;
n = 4;
x = linspace(0,L,n);
f = @(t) cos((n-1)*pi*t);

for i = 1:n
    for j = 1:n
        area(i,j) = integral(f,x(i),x(j));
    end
end

figure
hold on
plot(x,f(x),'b-');
xlabel('x');
ylabel('f(x) = cos((n-1)\pix)');
title('Exercise 3');

New Code and Output

%% Exercise  3

L = 1;
n = 4;
x = linspace(0,L,n);
h = x(2)-x(1);
f = @(t) cos(pi*t/h);

for i = 1:n
    for j = 1:n
        area(i,j) = quadgk(@(t) f(t),x(i),x(j));
```

```
    end
end

% N = 4

>> area

area =

   1.0e-15 *

        0    0.0069         0   -0.0278
  -0.0069         0    0.0347   -0.0902
        0   -0.0347         0    0.1457
   0.0278    0.0902   -0.1457         0

% N = 6

>> area

area =

   1.0e-15 *

        0    0.0182   -0.0121    0.0069         0    0.2012
  -0.0182         0    0.0234    0.0625   -0.1353   -0.0382
   0.0121   -0.0234         0    0.0963    0.0139    0.2116
  -0.0069   -0.0625   -0.0963         0   -0.2125   -0.0208
        0    0.1353   -0.0139    0.2125         0    0.1232
  -0.2012    0.0382   -0.2116    0.0208   -0.1232         0
```

As we can see, the values returned for all i, j are on the order of magnitude of `eps`, double-precision, which is ultimately zero.
Questions:

1. Are there other functions that satisfy this condition?
2. What does happen as $n \to \infty$ for these functions?
3. Is there a specific pattern to be observed?
4. Why is this result important in the study of PDEs?

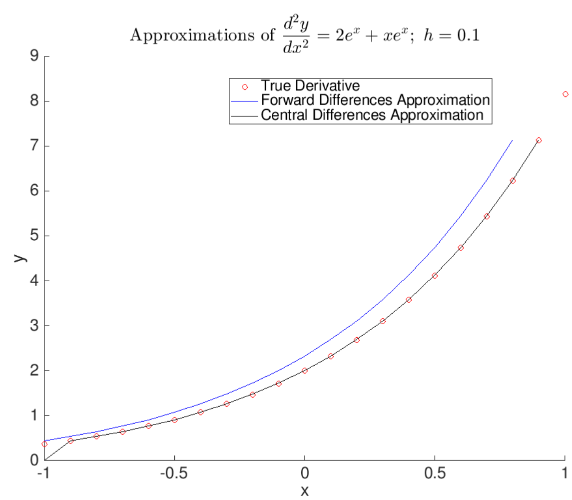
EXERCISE 4

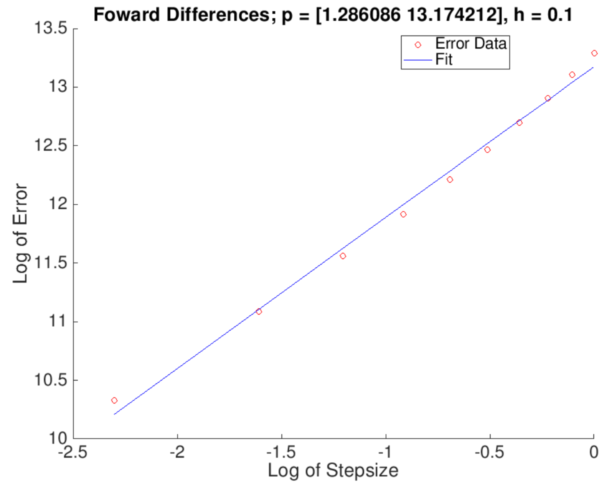
Q: *Let $f(x) = xe^x$. Approximate the second derivative of f on $[1, 1]$ using (i) forward differences and (ii) central differences with step size $h = .1,\ .05,\ .01$. For each method and each step size plot the natural log of the absolute error. Explain the resulting plots: do they confirm the orders of the methods?*

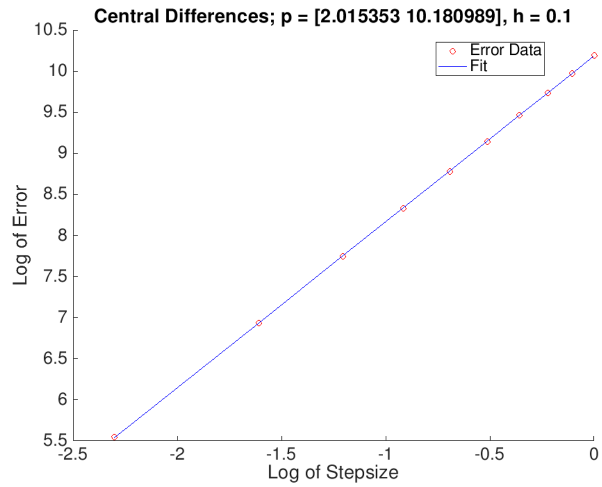
Solution: To obtain the forward and backwards difference approximations, in addition to analyzing the order of both methods, I used the following equations

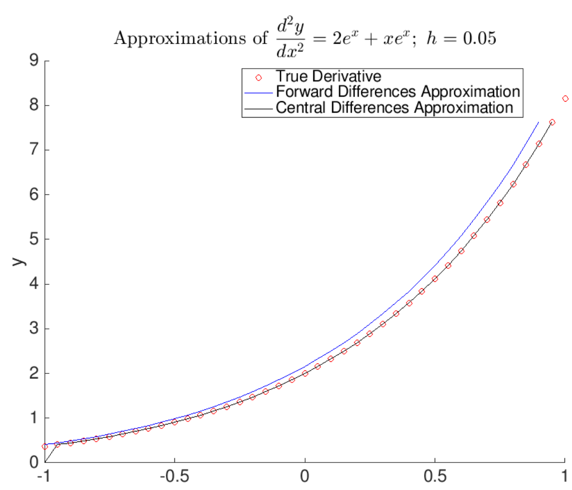
$$f_i''(x) \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}, \qquad f_i''(x) \approx \frac{f_{i+2} - 2f_{i+1} + f_i}{h^2},$$
$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}, \qquad f''(x) \approx \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2},$$

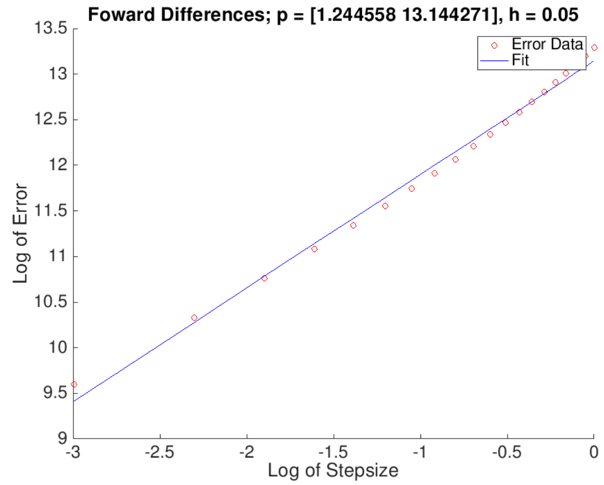
where the two equations on the left denote the central difference approximations to the second derivative and the two on the right are the forward difference approximations. In utilizing these four expressions, I was able to not only determine the approximations, but also the error of the approximations. In progressing through the varying values of h we not only see that the approximations get better, but also the orders for forward and central differences approach 1 and 2, respectively. Thus, although we would need to see what happens in the limit as $h \to 0$ for a formal proof, the resulting plots are sufficient enough as evidence, confirming the orders of both methods. An interesting note: the data for the errors of forward differences appears to have a slight curve. At first, I though this was due to the value at which the errors were computed, but when trying larger and smaller values, the still had a slight curve. I'm not entirely sure why this occurs, but I'm led to propose that this phenomenon occurs due to the fact that the order of forward differences is 1. Here are the plots and the accompanying code.

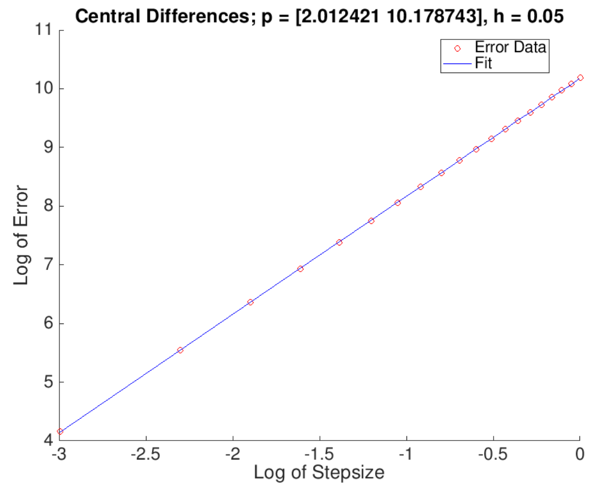


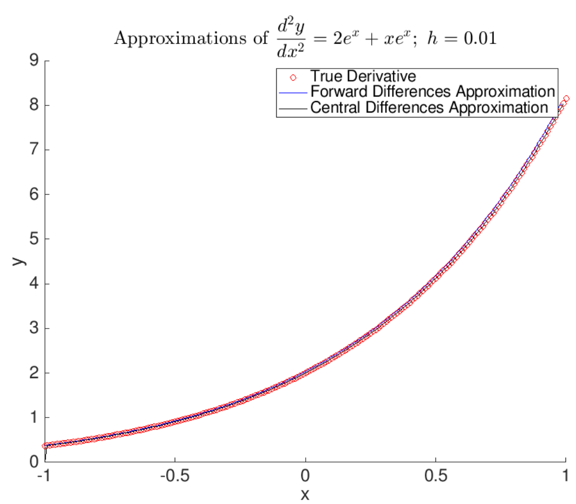

Foward Differences; p = [1.286086 13.174212], h = 0.1
Error Data
Fit
Log of Error
Log of Stepsize
13.5
13
12.5
12
11.5
11
10.5
10
-2.5
-2
-1.5
-1
-0.5
0

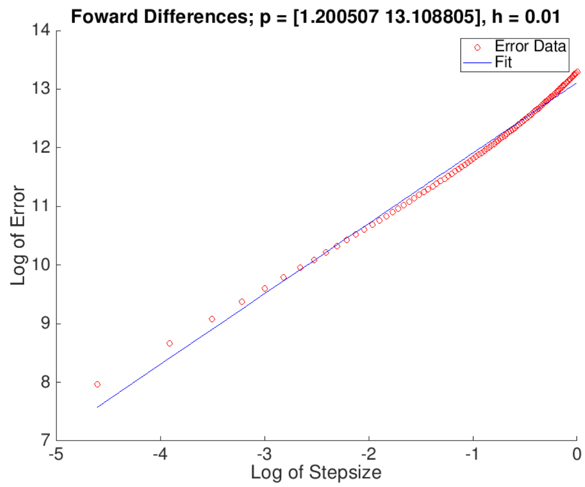

Central Differences; p = [2.015353 10.180989], h = 0.1
Error Data
Fit
Log of Error
Log of Stepsize
10.5
10
9.5
9
8.5
8
7.5
7
6.5
6
5.5
-2.5
-2
-1.5
-1
-0.5
0

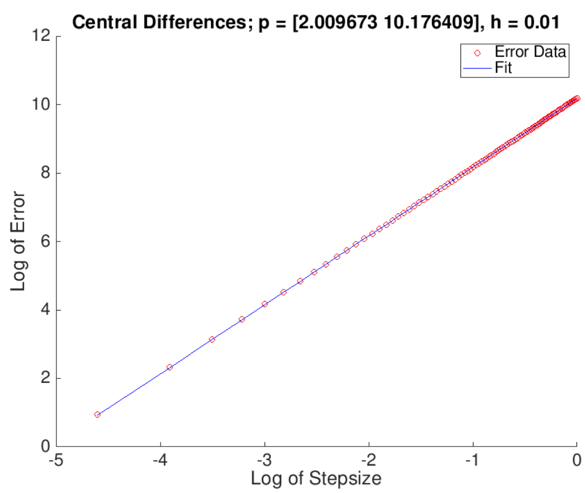

Approximations of $\frac{d^2y}{dx^2} = 2e^x + xe^x;\ h = 0.05$
True Derivative
Forward Differences Approximation
Central Differences Approximation
y
x
9
8
7
6
5
4
3
2
1
0
-1
-0.5
0
0.5
1


Foward Differences; p = [1.244558 13.144271], h = 0.05
Error Data
Fit
Log of Error
Log of Stepsize


Central Differences; p = [2.012421 10.178743], h = 0.05
Error Data
Fit
Log of Error
Log of Stepsize


Approximations of $\frac{d^2y}{dx^2} = 2e^x + xe^x;\ h = 0.01$
True Derivative
Forward Differences Approximation
Central Differences Approximation
y
x






```
%% Exercise 4

h = 0.01; % Generating known data
x = −1:h:1;
x0 = 10;
[m,n] = size(x);
hh = linspace(−1,1,n);


y = x.*exp(x); % The function for the approximations
yy = @(t) t.*exp(t); % The function for the error analysis


d2ydx2 = @(x) 2*exp(x) + x.*exp(x); % The true derivative of y

```

```
% Central Differences (CD)
for l = 2:n-1
    dy(l) = (y(l+1) - 2*y(l) + y(l-1))/(h^2);
end


dyy = (yy(x0 + hh) - 2*yy(x0) + yy(x0 - hh))./(hh.^2); % Error approx ...
    for CD
errc = abs(dyy - d2ydx2(x0));
logh = log(hh(hh>0)); % Use positive values of hh due to log
logerrc = log(errc);
p = polyfit(logh,logerrc(hh>0),1); % Use same values of hh to ensure ...
    same vector length
pfit = polyval(p,logh);


% Forward Differences (FD)
for j = 1:n-2
    dy1(j) = (y(j+2) - 2*y(j+1) + y(j))/(h^2);
end


dyy1 = (yy(x0 + 2*hh) - 2*yy(x0 + hh) + yy(x0))./(hh.^2); % Error ...
    approx for FD
errf = abs(dyy1 - d2ydx2(x0));
logerrf = log(errf);
pp = polyfit(logh,logerrf(hh>0),1);
ppfit = polyval(pp,logh);


figure
hold on
plot(x,d2ydx2(x),'ro'); % Plots the actual derivative
plot(x(1:end-2),dy1,'b-'); % Plots the forward differences approx.
plot(x(1:end-1),dy,'k-'); % Plots the central differences approx.
xlabel('x');
ylabel('y');
legend('True Derivative','Forward Differences Approximation',...
    'Central Differences Approximation');
title('Approximations of $\displaystyle\frac{d^2y}{dx^2} = 2e^x + ...
    xe^x;¬h = 0.01$',...
    'interpreter','latex');
hold off

figure
hold on
plot(logh,logerrc(hh>0),'ro'); % Plot of stepsize vs. CD error; ...
    Maintain same vector length
plot(logh,pfit,'b-'); % Plot of the fit
xlabel('Log of Stepsize');
```

```
ylabel('Log of Error');
title(sprintf('Central Differences; p = [%f %f], h = 0.01',p));
legend('Error Data','Fit');

figure
hold on
plot(logh,logerrf(hh>0),'ro'); % Plot of stepsize vs. FD error; ...
    Maintain same vector length
plot(logh,ppfit,'b-'); % Plot of the fit
xlabel('Log of Stepsize');
ylabel('Log of Error');
title(sprintf('Foward Differences; p = [%f %f], h = 0.01',pp));
legend('Error Data','Fit');
```

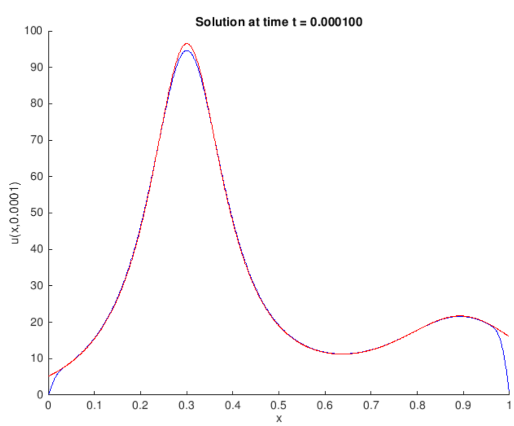
EXERCISE 5

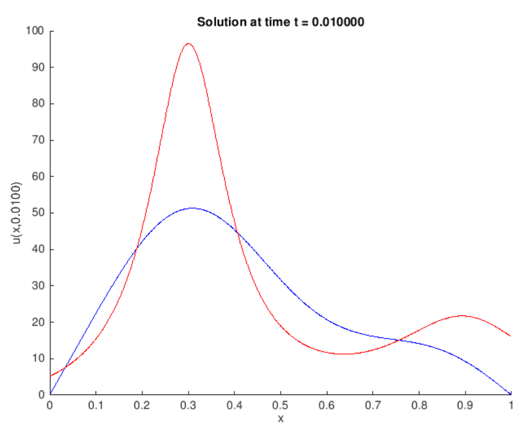
Q: *Modify the above code to solve the Heat Equation with Dirichlet boundary conditions:* $u(0,t) = u(L,t) = 0$. *Plot the solution for a small time, large time, and an intermediate time. What are your observations?*

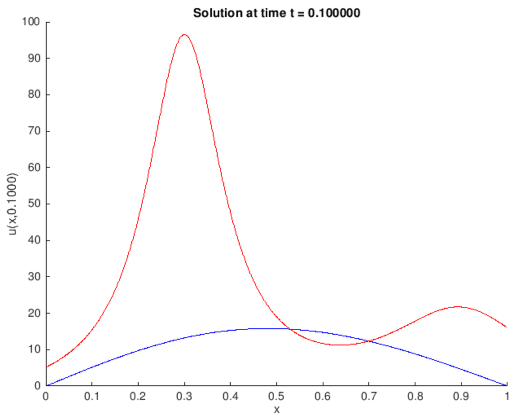
Solution: For Dirichlet conditions, the solution over time decays to zero. This makes physical sense because both boundaries are equal to zero. Thus, because there is no boundary on either end of the rod, the temperature evacuates the metal rod. Here are some plots from at times $t = 0, 0.0001, 0.01, 1$, and the accompanying code.

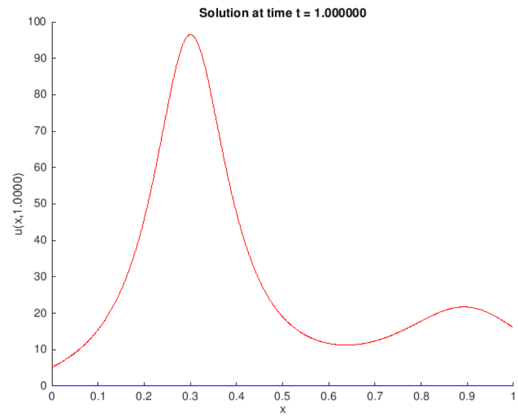
```
%% Exercise 5: Solving the heat equation, discrete case

L = 1;
K = 1;

% Semidiscretization

N = 1000;
h = L/(N+1);
x = (1:N)*h;

% Forming the matrix with Dirichlet Conditions

A = toeplitz([-2 1 zeros(1,N-2)]);

% Eigenval decomp for A

[V,D] = eig(A);
lambda = K*diag(D)/(h^2);

% Solution construction

f = @humps;
```

```
u0 = f(x)';
t = 0.0001;
C = V\u0;
u = V*(exp(lambda*t).*C);
tt = linspace(0,0.1,N);

% Plotting the solution

figure
hold on
p = plot(x,u,'b-');
plot(x,u0,'r-');
xlabel('x');
ylabel(sprintf('u(x,%1.4f)',t));
title(sprintf('Solution at time t = %f',t));

% Animating the solution

for t = tt
    u = V*(exp(lambda*t).*C);
    if ishandle(p)
        set(p,'ydata',u);
        title(sprintf('t=%1.4f',t),'FontSize',20);
        pause(0.001);
    end
end
```





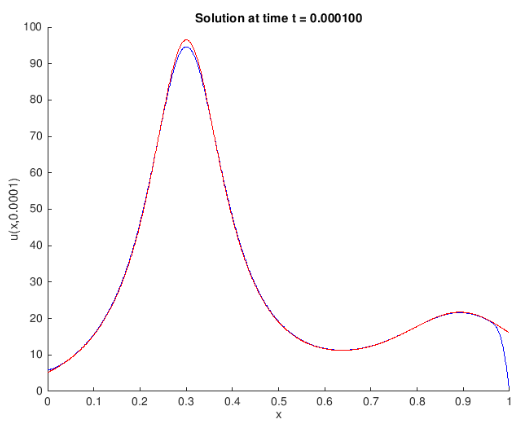


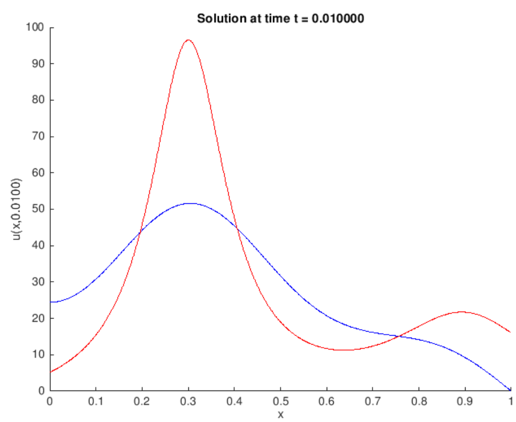


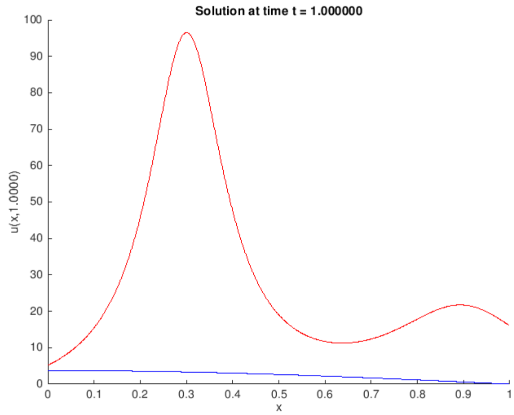
EXERCISE 6

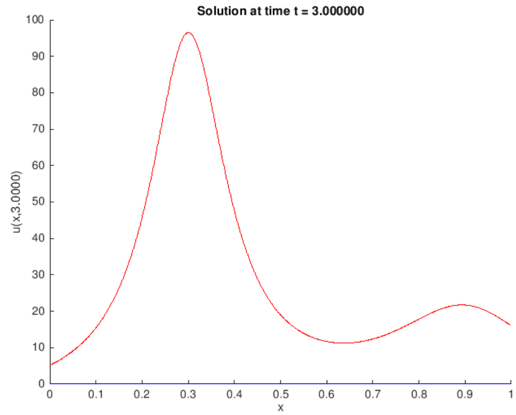
Q: *Repeat the previous exercise for mixed boundary conditions:* $\frac{\partial u}{\partial x}(0,t) = 0,\ u(L,t) = 0.$

Solution: The difference between this exercise and the last is that the first element of the coefficient matrix is -1 instead of being a TST with -2's on the main diagonal. These mixed conditions convey that the left end of the rod is perfectly insulated whereas the right end of the rod has no boundary. This suggests that over time, the temperature will decay to zero, but would take more time than the previous case. The times for which the solution is plotted for the mixed conditions is $t = 0.0001, 0.01, 1, 3$.









EXERCISE 7

Q: *Run the code of this section. Then plot several of the last eigenvectors of V (corresponding to small eigenvalues). What do the plots look like? Try to prove your conjecture.*

Solution: The plots look like cosine functions that alternate after every iteration. Here are the following plots and the code.

```
% Configuring the eigenvectors for plotting

[lambda,l] = sort(abs(diag(D)));
V = V(:,l);
lambda = lambda./(h^2);

for k = 1:9
    subplot(3,3,k);
    plot(x,V(:,k)./V(1:k),'r.');
    hold on
    ylim([-1 1]);
end
```

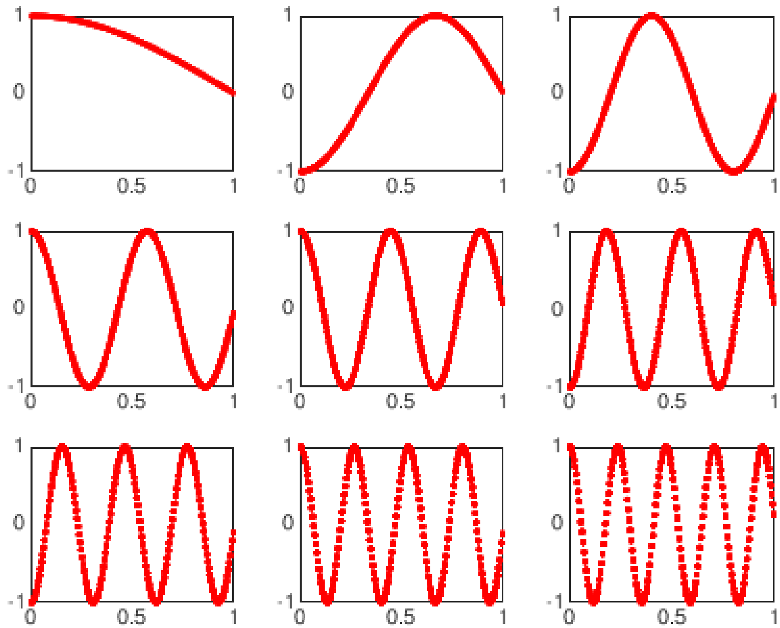


Figure 1: A plot of the first nine eigenvectors.

HOMEWORK 2

EXERCISE 1

(1) Consider the Dirichlet problem for the heat equation:

$$\begin{aligned}
&\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < \pi, \quad t > 0, \\
&u(x,0) = f(x), \\
&u(0,t) = u(\pi,t) = 0.
\end{aligned}$$

Semi-discretize the problem and investigate the eigendecomposition of the resulting matrix in Matlab. Plot a few eigenvectors (corresponding to small eigenvalues) and conjecture what they are. You may not be able to derive exact formulas, as in Theorem 2. Still, you should be able to see enough structure to derive a formula for the solution similar to Equation (4.1).

Solution: To derive the formula for the Heat Equation, let us present a general formula for the eigenvectors and eigenvalues of the Dirichlet Laplacian, which is given by the following matrix

$$A = \begin{bmatrix}
-2 & 1 & & & & & \\
1 & -2 & 1 & & & & \\
& 1 & -2 & 1 & & & \\
& & \ddots & \ddots & \ddots & & \\
& & & 1 & -2 & 1 & \\
& & & & 1 & -2 & 1 \\
& & & & & 1 & -2
\end{bmatrix},$$

which is a Tridiagonal Symmetric Toeplitz matrix (TST, or *"tasty"* if you want to be fancy).

Theorem 1. *The general expression for the eigenvalues of the $N \times N$ matrix A is given by*

$$\lambda_n = 2\sin\left(\frac{\pi(N-2n+1)}{2(N+1)}\right) - 2 \tag{i}$$

$$= 2\cos\left(\frac{\pi n}{N+1}\right) - 2, \quad n = 1, \ldots, N. \tag{ii}$$

The corresponding eigenvectors $\mathbf{v}_n$ have components given by

$$\mathbf{v}_n(k) = \sqrt{\frac{2}{N+1}} \sin\left(\frac{\pi n k}{N+1}\right), \quad k = 1, \ldots, N. \tag{iii}$$

Equation (i) was derived numerically and primarily through looking for patterns, in addition to lots of crying. To confirm that equation (i) was indeed a correct formulation, I created a function in MATLAB and compared the outputs of both equations (i) and (ii). Here is the code for that function and some outputs with different N, n values.

```
function [lambda,lambda1] = dir_eig(N,n)

lambda = 2*sin((pi*(N - 2*n + 1))/(2*N + 2)) - 2; % The derived formula
lambda1 = 2*cos((pi*n)/(N + 1)) - 2; % The actual formula

end

Outputs:
% N = 2, n = 2
l =

    -3


l1 =

   -3.0000

>> [l,l1] = dir_eig(9,9)
% N = n = 9
l =

   -3.9021


l1 =

   -3.9021


>> [l,l1] = dir_eig(250,32)
% N = 250, n = 32
l =

   -0.1583


l1 =

   -0.1583
```

I was able to successfully derive equation (iii) numerically, but analytically, it would have been impossible, especially with my formulation for the eigenvalues, so I referenced a handout from the University of Chicago to see that the analytic solution was indeed exactly the same as the solution I derived from numerical analysis. In plotting the eigenvectors, shown in Figure 1, I was able to discern that they behaved like the sine function, and this is where the analysis began.

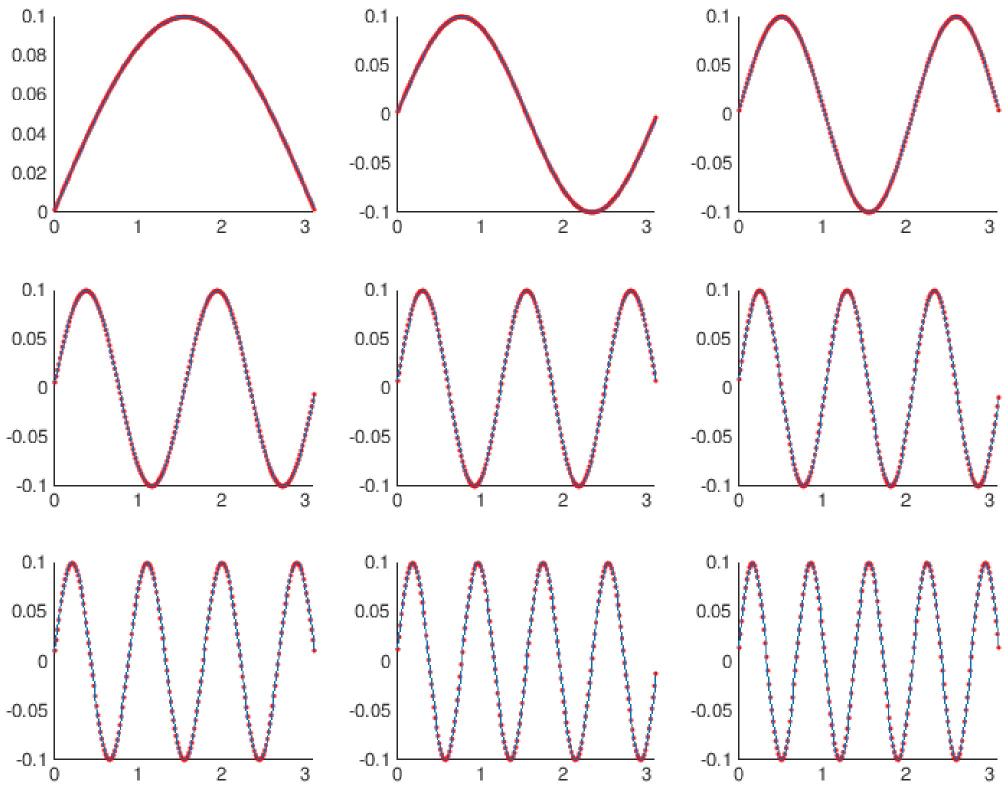


Figure 2: The eigenvectors of the discrete Dirichlet Laplacian. Although it is not that easy to see, the plots contain the red dots, which are the plots of the eigenvectors as determined by `eig`, and blue lines, which represent the numerically derived solution.

So, I began with $\sqrt{2/N}\sin(kx)$ to see if I could match the plots, based on the logic in the handout. After some trial and error, I found that the denominator in the square root was $N+1$, which is what produced the above plots. Then, from there, I realized that I could rewrite the aforementioned expression in terms of the step-size h, which is in turn defined as $h = x_n = nL/(N+1)$, where the index n ranges from $n = 0, \ldots, N+1$. Thus, in making the substitution, I was able to find the formula that is equation (iii), where $L = \pi$. To confirm that this was indeed the formula, again, I compared the MATLAB code against my formulation and found agreement. Of course when deriving formulas as presented in theorem 1, we should always appeal to more rigorous and more formal proofs. Anyhow, here is the code that produced the above plots.

```
%% Exercise 1: Solving the Dirichlet Problem of the Heat Equation

L = pi;
```

```
K = 1;

% Semidiscretization

N = 200;
h = L/(N+1);
x = (1:N)*h;

% Forming the matrix with Dirichlet Conditions

A = toeplitz([-2 1 zeros(1,N-2)]);

% Eigenvalue decomposition for A

[V,D] = eig(A);
lambda = K*diag(D)/(h^2);

% Solution construction

f = @humps;
u0 = f(x)';
t = 0.0001;
C = V\u0;
u = V*(exp(lambda*t).*C);
tt = linspace(0,0.1,N);

% Plotting the solution

figure
hold on
p = plot(x,u,'b-');
plot(x,u0,'r-');
xlabel('x');
ylabel(sprintf('u(x,%1.4f)',t));
title(sprintf('Solution at time t = %f',t));

% Configuring the eigenvectors for plotting

[lambda,l] = sort(abs(diag(D)));
V = V(:,l);
lambda = lambda./(h^2);

% Animating the solution

% for t = tt
%     u = V*(exp(lambda*t).*C);
%     if ishandle(p)
%         set(p,'ydata',u);
%         title(sprintf('t=%1.4f',t),'FontSize',20);
%         pause(0.001);
%     end
```

```
% end

for k = 1:9
    subplot(3,3,k);
    hold on
    plot(x,sign(V(1,k)).*V(:,k),'r.');
    plot(x,sqrt(2/(N+1))*sin(k*x));
    legend('Eigenvectors from Eig','Eigenvectors from Derivation');
end
```

The General Formula for the Dirichlet Problem

To derive the general formula for the solution of the Dirichlet Problem of the heat equation, we proceed using the logic in the handout. We begin our derivation at the general solution to the semi-discretized Heat Equation, given by

$$\mathbf{u} = Ve^{Kh^{-2}\lambda_n t}V^T\mathbf{u}_0 = \sum_{n=1}^{N} e^{Kh^{-2}\lambda_n t}\langle \mathbf{v}_n, \mathbf{u}_0\rangle \mathbf{v}_n = \sum_{n=1}^{N} e^{Kh^{-2}\lambda_n t} P_n(\mathbf{u}_0), \tag{1}$$

where λ_n are the eigenvalues of the matrix A and $P_n(\mathbf{u}_0)$ represents the n-th projection of the initial condition $\mathbf{u}_0$ onto the n-th eigenspace of A. We need to determine the behavior of equation (1) in the limit as $h \to 0$ and as $N \to \infty$. In doing so, equation (1) becomes

$$\mathbf{u} = \sum_{n=1}^{\infty} e^{t\ \lim_{h\to 0}\lambda_n^h} P_n(u), \tag{2}$$

where $\lambda_n^h = Kh^{-2}\lambda_n$. To evaluate the limit in the exponential, we take

$$\begin{aligned}
\lim_{h\to 0}\lambda_n^h &= \lim_{h\to 0} Kh^{-2}\left(2\cos\left(\frac{n\pi}{N+1}\right) - 2\right)\\
&= \lim_{h\to 0} Kh^{-2}\left(2 - \left(\frac{n\pi}{N+1}\right)^2 + \mathcal{O}(N^-4) - 2\right)\\
&= \lim_{h\to 0} K\left(-\left(\frac{n\pi}{h(N+1)}\right)^2 + h^{-2}\mathcal{O}(N^-4)\right)\\
&= -K\frac{n^2\pi^2}{L^2}.
\end{aligned}$$

We come to this conclusion because $h = L/(N+1)$ and the fact that $N \to \infty$ faster than $h \to 0$ for the $\mathcal{O}(N^{-4})$ terms. For the orthogonal projections, the inner product $V^T\mathbf{u}_0$ is

$$V^T\mathbf{u}_0 = \langle \mathbf{v}_n, \mathbf{u}_0\rangle = \sqrt{\frac{2}{N+1}}\sum_{k=1}^{N} f(hk)\sin\left(\frac{\pi nk}{N+1}\right)$$

where the $\mathbf{v}_n$'s are the components of V^T and $\mathbf{u}_0 = f(\mathbf{x})$ has components $f(hk)$ for $k = 1, \ldots, N$. In taking the m-th component of $P_n(\mathbf{u}_0) = \langle \mathbf{v}_n, \mathbf{u}_0\rangle \mathbf{v}_n$, we have

$$\begin{aligned}&\sin\left(\frac{n\pi m}{N+1}\right)\frac{2}{N+1}\sum_{k=1}^{N} f(hk)\sin\left(\frac{\pi nk}{N+1}\right)\\&=\sin\left(\frac{n\pi m}{N+1}\right)\frac{2}{h(N+1)}\sum_{k=1}^{N} f(hk)\sin\left(\frac{\pi nkh}{h(N+1)}\right)h.\end{aligned}$$

In making our substitution for h and then taking the limit as $h \to 0$, we obtain

$$\left[\frac{2}{L}\int_0^L f(y)\sin\left(\frac{\pi ny}{L}\right)dy\right]\sin\left(\frac{n\pi m}{N+1}\right).$$

It is easy to see from plotting that as $N \to \infty$ the term $\sin\left(\frac{n\pi m}{N+1}\right) \to \sin\left(\frac{n\pi x}{L}\right)$, so in the limit, the projections become

$$P_n(u_0) = \left[\frac{2}{L}\int_0^L f(y)\sin\left(\frac{\pi ny}{L}\right)dy\right]\sin\left(\frac{n\pi x}{L}\right).$$

Just to make sure all bases are covered, let's examine the initial condition projected onto the first eigenspace of A. The m-th component would like

$$\langle \mathbf{v}_1, \mathbf{u}_0\rangle \mathbf{v}_1 = \sin\left(\frac{\pi m}{N+1}\right)\frac{2}{N+1}\sum_{k=1}^{N} f(hk)\sin\left(\frac{\pi k}{N+1}\right)$$

If we take the limit of the above expression as $N \to \infty$, this first projection $P_1(\mathbf{u}_0) \to 0$. Thus, the solution to the Dirichlet Problem to the Heat Equation becomes

$$\boxed{u(x,t) = \sum_{n=1}^{\infty} e^{-K\frac{n^2\pi^2}{L^2}t}\left[\frac{2}{L}\int_0^L f(y)\sin\left(\frac{\pi ny}{L}\right)dy\right]\sin\left(\frac{n\pi x}{L}\right)}.$$

To validate this solution, I use the following MATLAB code with the associated plots.

```
L = pi;
K = 1;

% Semidiscretization

N = 100;
h = L/(N+1);
x = (1:N)*h;

% Forming the matrix with Dirichlet Conditions

A = toeplitz([-2 1 zeros(1,N-2)]);

```

```
% Eigenvalue decomposition for A

[V,D] = eig(A);
lambda = K*diag(D)/(h^2);

% Solution construction

f = @humps;
u0 = f(x)';
t = 0.01;
C = V\u0;
u = V*(exp(lambda*t).*C);
tt = linspace(0,1,N);

% Configuring the eigenvectors for plotting

% [lambda,l] = sort(abs(diag(D)));
% V = V(:,l);
% lambda = lambda./(h^2);

% Analytic Solution

for n = 1:N
    g = @(x) sin(pi.*x.*n/L);
    gg = @(x) f(x).*g(x);
    proj(:,n) = (2/L)*quadgk(gg,0,L)*g(x);
    en(n) = exp(-(n^2)*t);
end

uu = proj*en';

% Plotting the Analytic and Numerical Solutions

figure
hold on
p = plot(x,u,'b-'); % Numerical Solution
plot(x,u0,'r-'); % Initial Condition
plot(x,uu,'ko');
xlabel('x');
```

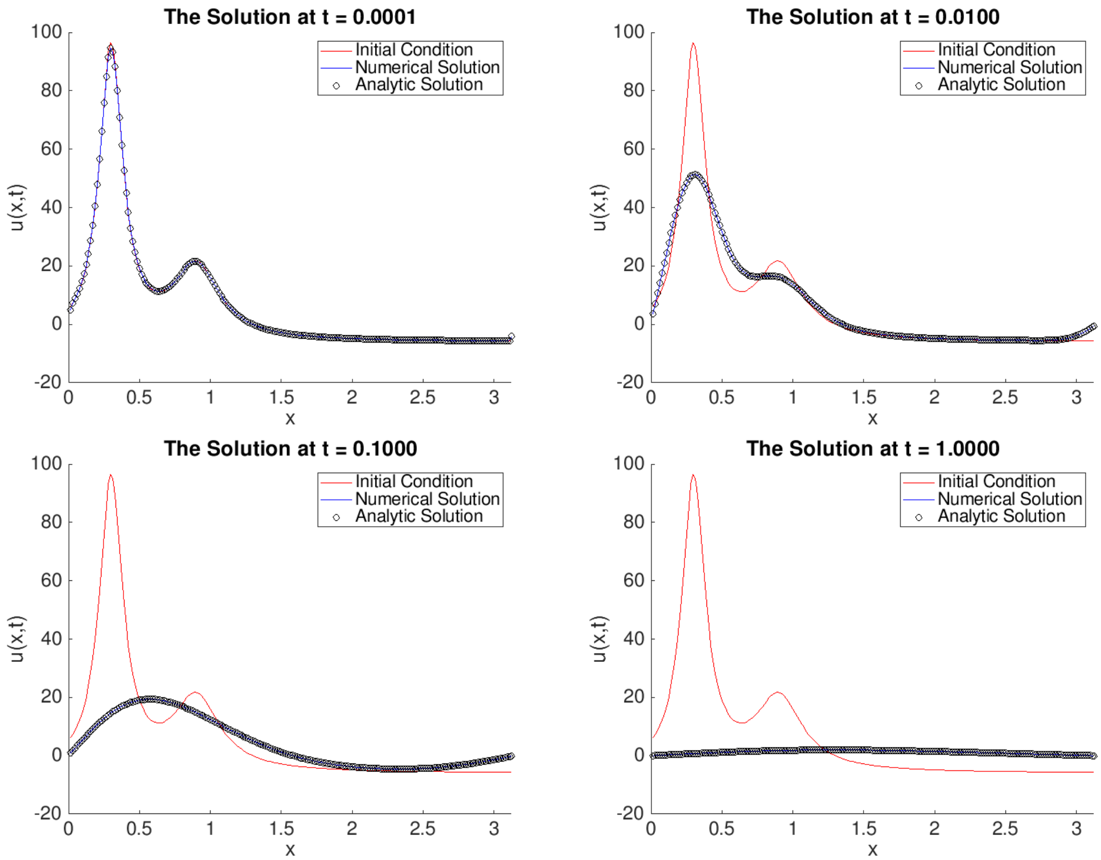


Figure 3: The analytic solution versus the numerical solution for confirmation.

EXERCISE 2

(2) Consider the Neumann problem for the wave equation:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, \quad t > 0,$$
$$u(x,0) = f(x), \quad \frac{\partial u}{\partial t}(x,0) = 0$$
$$\frac{\partial u}{\partial x}(0,t) = \frac{\partial u}{\partial x}(L,t) = 0.$$

Use the logic of this handout to find a formula for the solution similar to Equation (4.1)

Solution: To derive the solution to the wave equation, it would first be useful to understand the solution to a second order ODE of the form

$$\frac{d^2 u}{dt^2} = -\omega^2 u, \quad u(0) = u_0, \quad \dot{u}(0) = \dot{u}_0. \tag{3}$$

The solution to the above equation is given by

$$u(t) = C_1 e^{-i\omega t} + C_2 e^{i\omega t} = C_1 \cos(\omega t) + C_2 \sin(\omega t),$$

where Euler's formula[3] was applied to the exponential expression to obtain the trigonometric one. After applying the initial conditions, the general form of the solution to equation (1) is

$$u(t) = u_0 \cos(\omega t) + \frac{\dot{u}_0 \sin(\omega t)}{\omega}.$$

Using semi-discretization, we have the wave equation in the form

$$\frac{d^2\mathbf{u}}{dt^2} = A\mathbf{u}, \quad \mathbf{u}_0 = f(\mathbf{x}_0), \quad \dot{\mathbf{u}}_0 = f'(\mathbf{x}_0) = \mathbf{0}. \tag{4}$$

where the A is the diagonalizable matrix

$$A = \begin{bmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -1 \end{bmatrix},$$

representing the discrete Neumann Laplacian. This implies that the solution can be written in the form

$$\mathbf{u} = \cos\left(t\sqrt{-A}\right)\mathbf{u}_0 + \frac{\sin\left(t\sqrt{-A}\right)}{\sqrt{-A}}\dot{\mathbf{u}}_0, \tag{5}$$

following from the solution to equation (3). It at first seems counter-intuitive to have a matrix as part of the argument of the trigonometric functions, but it is essential to understanding the formation of the solution. Further, it is also worth mentioning that because A is diagonalizable, we can write $A = VDV^{-1} = VDV^T$. Using Taylor's Theorem, we can then rewrite the cosine and sine terms as

$$\begin{aligned}
\cos\left(t\sqrt{-A}\right) &= I + t^2\frac{A}{2!} + t^4\frac{A^2}{4!} + t^6\frac{A^3}{6!} + \cdots \\
&= VIV^T + t^2\frac{VDV^T}{2!} + t^4\frac{VDV^TVDV^T}{4!} + t^6\frac{VDV^TVDV^TVDV^T}{6!} + \cdots \\
&= V\left(I + t^2\frac{D}{2!} + t^4\frac{D^2}{4!} + t^6\frac{D^3}{6!} + \cdots\right)V^T \\
&= V\cos\left(t\sqrt{-D}\right)V^T,
\end{aligned}$$

[3] "Wheeler's" formula is given by $e^{i\theta} = \cos(\theta) + i\sin(\theta)$.

and

$$\frac{\sin\left(t\sqrt{-A}\right)}{\sqrt{-A}} = V\frac{\sin\left(t\sqrt{-D}\right)}{\sqrt{-D}}V^T,$$

respectively. However, because the sine term is scaled by the zero vector, due to the initial conditions, it vanishes. Thus, the general solution to the discrete Neumann problem to the heat equation is

$$\mathbf{u} = V\cos\left(t\sqrt{-D}\right)V^t\mathbf{u}_0 = \sum_{n=1}^{N}\cos(\sqrt{-\lambda_n}t)\langle\mathbf{v}_n,\mathbf{u}_0\rangle\mathbf{v}_n = \sum_{n=1}^{N}\cos(\sqrt{-\lambda_n}t)P_n(\mathbf{u}_0). \tag{6}$$

In taking the continuous limit, as per the handout, we know that

$$\begin{aligned}
\lambda_n &= -\frac{n^2\pi^2}{L^2},\\
P_1(u) &= \frac{1}{L}\int_0^L f(y)\ dy,\\
P_n(u) &= \left[\frac{2}{L}\int_0^L f(y)\cos\left(\frac{\pi ny}{L}\right)dy\right]\cos\left(\frac{n\pi x}{L}\right).
\end{aligned}$$

Thus we can formally write the complete solution to the Neumann problem to the heat equation:

$$\boxed{u(x,t) = \frac{1}{L}\int_0^L f(y)\ dy + \sum_{n=1}^{\infty}\cos\left(\frac{n\pi t}{L}\right)\left[\frac{2}{L}\int_0^L f(y)\cos\left(\frac{\pi ny}{L}\right)dy\right]\cos\left(\frac{n\pi x}{L}\right).}$$

Here are a few plots of the numerical and analytic solutions, with the corresponding code.

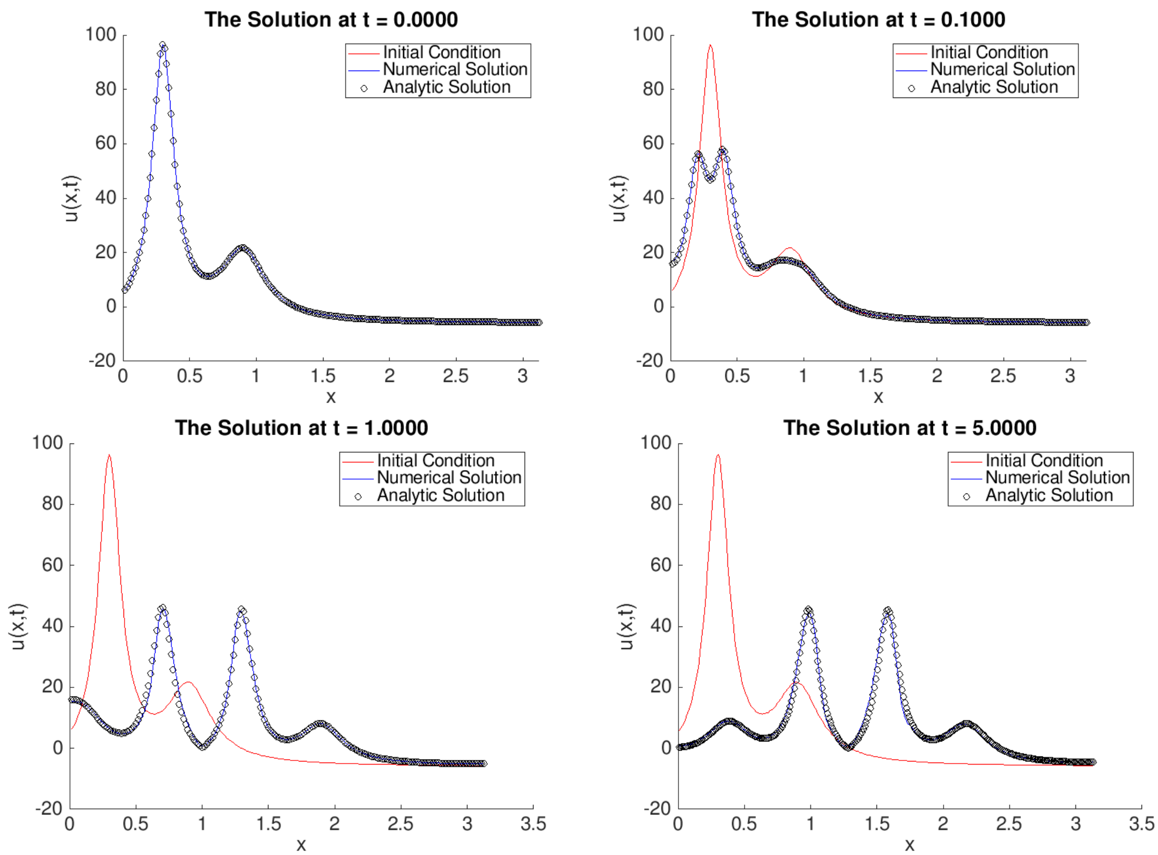


Figure 4: The numerical and analytic solutions to the wave equation.

```
%% Exercise 2 The Neumann Problem for the Wave Equation

L = pi;
K = 1;

% Semidiscretization

N = 300;
h = L/(N+1);
x = (1:N)*h;

% Forming the matrix with Neumann Conditions

A = toeplitz([-2 1 zeros(1,N-2)]);
A(1,1) = -1;
A(N,N) = -1;

% Eigenvalue decomposition for A

[V,D] = eig(A);
```

```
lambda = K*diag(D)/(h^2);

% Solution construction

f = @humps;
u0 = f(x)';
t = 5;
C = V\u0;
u = V*(cos(sqrt(-lambda)*t).*C);
tt = linspace(0,5,N);

% Analytic Solution

for n = 1:N
    g = @(x) cos(pi.*x.*n/L);
    gg = @(x) f(x).*g(x);
    proj(:,n) = (2/L)*quadgk(gg,0,L)*g(x);
    en(n) = cos(pi.*t.*n/L);
end

uu = (1/L)*quadgk(f,0,L) + proj*en';

% Plotting the solution

figure
hold on
plot(x,u0,'r-'); % Initial Condition
p = plot(x,u,'b-'); % Numerical Solution
plot(x,uu,'ko'); % Analytic Solution
title(sprintf('The Solution at t = %1.4f',t));
legend('Initial Condition','Numerical Solution','Analytic Solution');
ylabel('u(x,t)');
xlabel('x');
```

EXERCISE 3

Let f be a function on $[0, L]$. Define

$$S_N(f) = \frac{1}{L}\int_0^L f(y)\ dy + \sum_{n=1}^{N}\left[\frac{2}{L}\int_0^L f(y)\cos\left(\frac{\pi n y}{L}\right)dy\right]\cos\left(\frac{n\pi x}{L}\right).$$

and set $R_N(f) = fS_N(f)$. Think of $S_N(f)$ as a Fourier approximation to f with $R_N(f)$ being the error, or residual. Let us agree to measure the magnitude of the residual as follows: This is called the infinity- or sup-norm. For each of the functions below produce a `loglog` *plot of the sup-norm of the residual against N:*

$$f = x$$
$$f = x(L - x)$$
$$f = \frac{1}{2 + \cos(\omega x)}, \quad \omega = \frac{\pi}{L},$$

You can set $L = \pi$ or any positive number of your choice. If you can compute integrals analytically, do that. If you have to approximate integrals numerically, use ***quadgk****. Write your observations.*

Solution: Here are the plots for the various functions, the observations, and the corresponding code.

```
%% Fourier Error Analysis

% Conditions

L = 7*pi/3;
N = 1:100;
x = linspace(0,L,100);
w = pi/L;

% The functions

%f = @(x) x;
%f = @(x) x.*(L - x);
f = @(x) 1./(2 + cos(w*x));

err = zeros(size(N));
ss = zeros(length(x),length(x));

for n=N
    s = zeros(size(x));
    for ii=1:n
        g = @(x) cos(pi*ii/L*x);
        gf = @(x) g(x).*f(x);
        s = s + 2/L*quadgk(gf,0,L)*g(x);
    end
    s = s + 1/L*quadgk(f,0,L)*ones(size(s));
    SS(n,:) = s;
    R = f(x) - s;
    E(n) = max(abs(R));
end

logN = log(N);
logE = log(E);
p = polyfit(logN(end-20:end),logE(end-20:end),1);
order = p(1);
```

```
figure
hold on
plot(logN,logE,'ro')
plot(0:5,polyval(p,0:5),'b—')
xlabel('Log of N')
ylabel('Log of Error')
title(sprintf('Order of convergence: %f for f(x) = 1/(2 + ...
    cos(wx))',order));
```

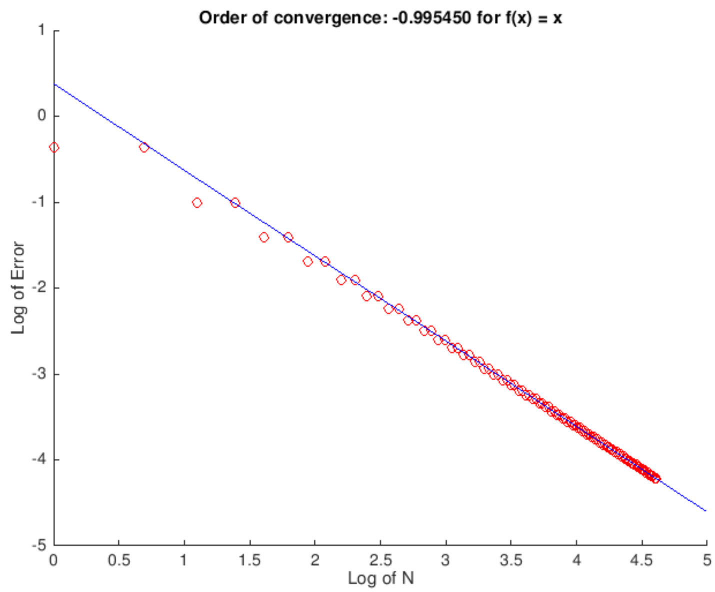


Figure 5: The plot of the error with $f(x) = x$. The Fourier approximation converges to the solution with an order of 1.

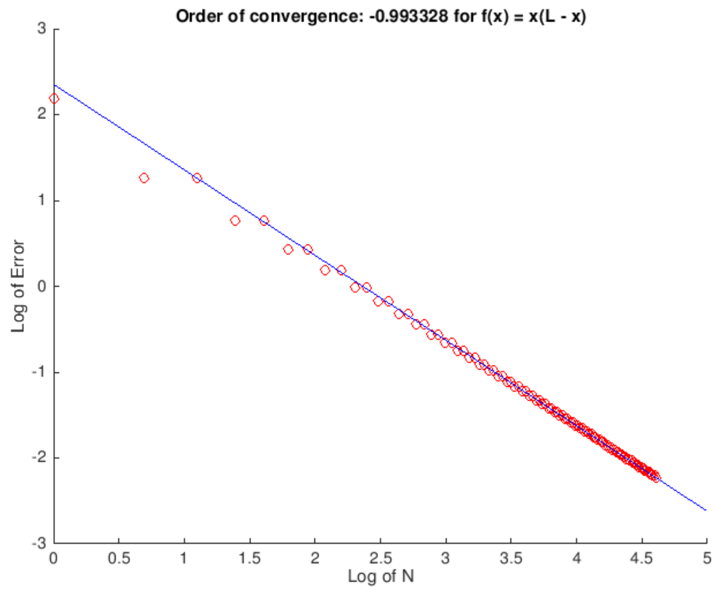


Figure 6: Here is the plot of $f(x) = x(L - x)$, which also converges with an order of 1.

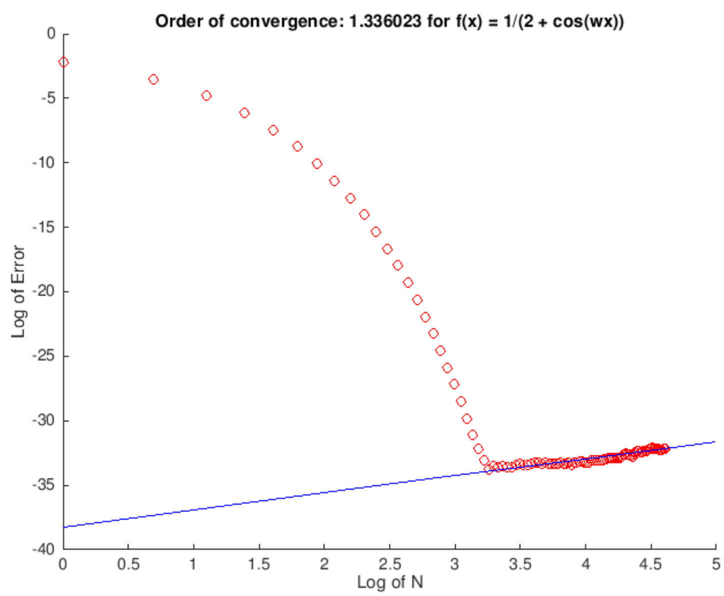


Figure 7: This is the last plot of $f(x) = 1/(2 + \cos(\omega x))$ with $\omega = 3/7$. This function also converges with an order of 1, but this function converges much more slowly compared to the other two functions. This is makes sense, given that the first two functions are simple compared to this function. Further, the convergence of the series oscillates slightly about the fit, suggesting that $N = 100$ is not a sufficient amount of terms.

HOMEWORK 3

EXERCISE 1

Use separation of variables to derive a general formula for the solution of the following problem:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad -\pi < x < \pi, \quad t > 0, \tag{1}$$
$$u(x, 0) = f(x),$$
$$u(\pi, t) = u(-\pi, t) = 0.$$

Test your solution using the following initial conditions:

(a) $f = \sin(3x)$

(b) $f = |x|$

(c) $f = \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases}$

By testing, I mean that you should compare the solution obtained numerically—through semi-discretization—with the analytic formula. If you do things correctly, the agreement will be noticeable. Think about the best way of comparing the numerical solution and the analytic solution.

Solution: To solve the Heat Equation using separation-of-variables, let $u(x, t) = v(t)w(x)$. By substituting u into equation (1), we obtain a system of linear ODE

$$\frac{dv}{dt} = -\omega^2 v \tag{2}$$
$$\frac{d^2 w}{dx^2} = -\omega^2 w, \quad w(\pi) = w(-\pi) = 0 \tag{3}$$

where $\omega^2 = \lambda$. The respective solutions to equations (2) and (3) are

$$v(t) = Ae^{-\omega^2 t} = Ae^{\lambda t} \tag{4}$$
$$w(x) = a\cos(\omega x) + b\sin(\omega x) = a\cos\left(\sqrt{-\lambda}x\right) + b\sin\left(\sqrt{-\lambda}x\right). \tag{5}$$

To satisfy the initial boundary conditions, we obtain the following matrix vector system

$$A\mathbf{c} = \begin{bmatrix} \cos\left(\sqrt{-\lambda}\pi\right) & \sin\left(\sqrt{-\lambda}\pi\right) \\ \cos\left(\sqrt{-\lambda}\pi\right) & -\sin\left(\sqrt{-\lambda}\pi\right) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0}. \tag{6}$$

In order for equation (6) to be true, we can either take the trivial solution, where $a = b = 0$, or we can determine λ in order that the determinant is equal to zero. To obtain non-trivial solutions, we have

$$\begin{aligned} \det(A) &= -2\sin\left(\sqrt{-\lambda}\pi\right)\cos\left(\sqrt{-\lambda}\pi\right) = 0 \\ \implies& \sin\left(\sqrt{-\lambda}\pi\right)\cos\left(\sqrt{-\lambda}\pi\right) = 0. \end{aligned} \tag{7}$$

Using the zero-product property, we can determine the arguments of both the sine and cosine terms so that equation (7) is true. As a result, we obtain two eigensolutions to equation (3), given by

$$\begin{aligned} \lambda_{\text{sin}} &= -n^2 \\ \lambda_{\text{cos}} &= -\left(n + \frac{1}{2}\right)^2 \end{aligned}$$

where λ_{sin} is the result of letting sine equal zero and λ_{cos} is the result of letting cosine equal zero. Thus, the eigensolutions of equation (1) with eigenvalues λ_{sin} are of the form

$$u_n(x,t) = e^{-n^2 t}\big(a_n\cos(nx) + b_n\sin(nx)\big),$$

and by linearity, we have

$$u(x,t) = \sum_{n=0}^{\infty} e^{-n^2 t}\big(a_n\cos(nx) + b_n\sin(nx)\big).$$

In order to determine the coefficients a_n, b_n, we examine the initial condition, where

$$u(x,0) = a_0 + \sum_{n=1}^{\infty}\big(a_n\cos(nx) + b_n\sin(nx)\big) = f(x). \tag{8}$$

Further, let us define the inner product of two functions f and g to be

$$\langle f, g\rangle = \int_{-\pi}^{\pi} f(x)g(x)\ dx,$$

which is called the L^2-inner product. Now, in taking inner-products of both sides of equation (8) with $\cos(mx)$ and $\sin(mx)$, we have

$$\begin{aligned} \Big\langle f, \cos(mx)\Big\rangle &= a_0\Big\langle 1, \cos(mx)\Big\rangle + \sum_{n=1}^{\infty} a_n\Big\langle \cos(nx), \cos(mx)\Big\rangle + b_n\Big\langle \sin(nx), \cos(mx)\Big\rangle \\ \Big\langle f, \sin(mx)\Big\rangle &= a_0\Big\langle 1, \sin(mx)\Big\rangle + \sum_{n=1}^{\infty} a_n\Big\langle \cos(nx), \sin(mx)\Big\rangle + b_n\Big\langle \sin(nx), \sin(mx)\Big\rangle. \end{aligned}$$

Using the property of the orthonormality of the eigenfunctions of the Laplacian, we have

$$\begin{aligned} a_0 &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\ dx, \\ a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)\ dx, \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)\ dx. \end{aligned}$$

In rewriting equation (8), we have the solution

$$\begin{aligned} u(x,t) = &\frac{1}{2\pi}\int_{-\pi}^{\pi} f(y)\ dy \\ &+ \sum_{n=1}^{\infty} e^{-n^2 t}\left(\left[\frac{1}{\pi}\int_{-\pi}^{\pi} f(y)\cos(ny)\ dy\right]\cos(nx) + \left[\frac{1}{\pi}\int_{-\pi}^{\pi} f(y)\sin(ny)\ dy\right]\sin(nx)\right) \end{aligned}$$

Here are plots of the numerical and analytic solutions for all cases including the code that generated the plots.

(a) $\sin(3x)$

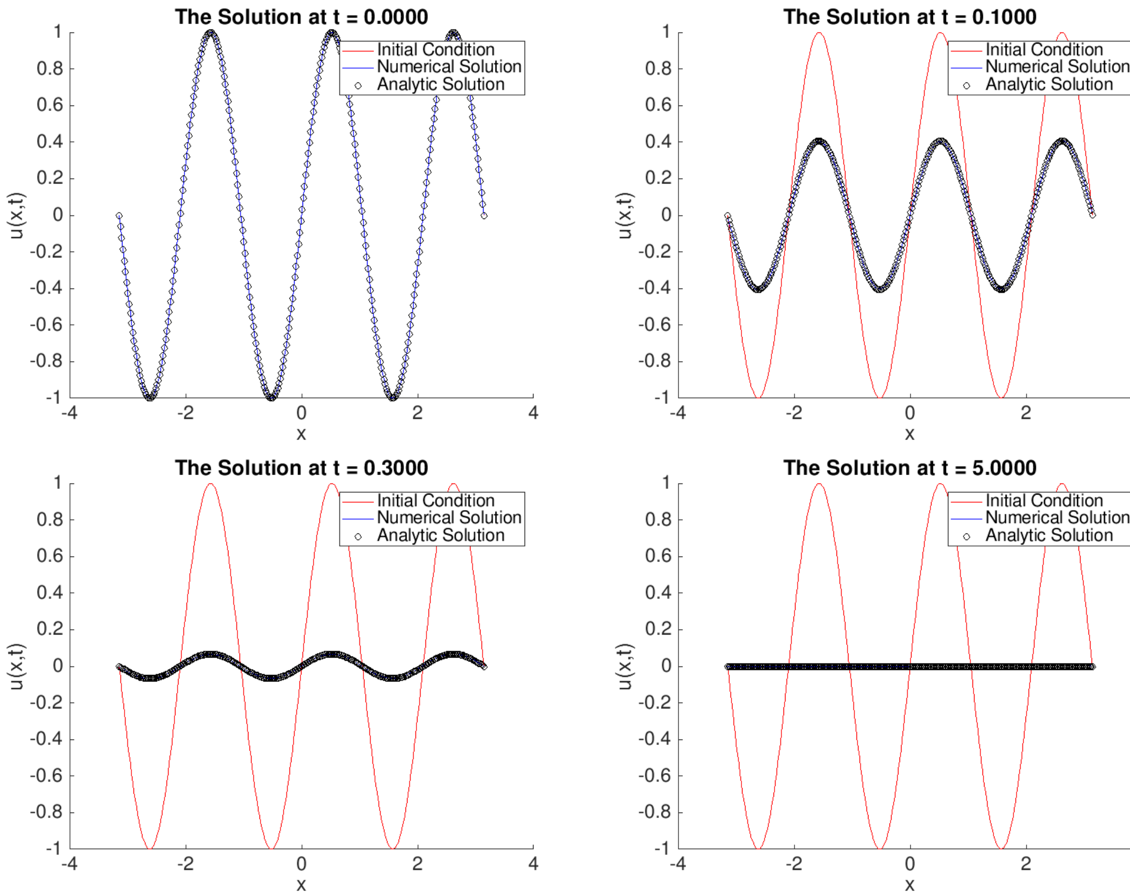
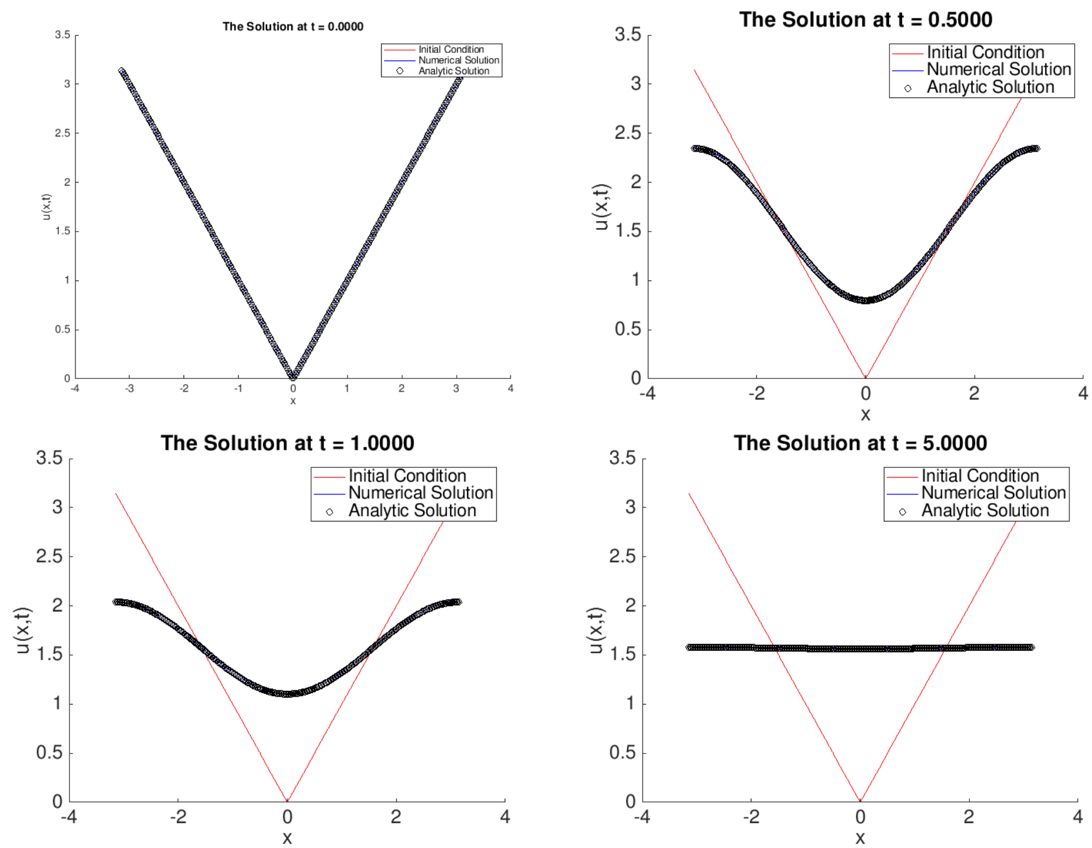
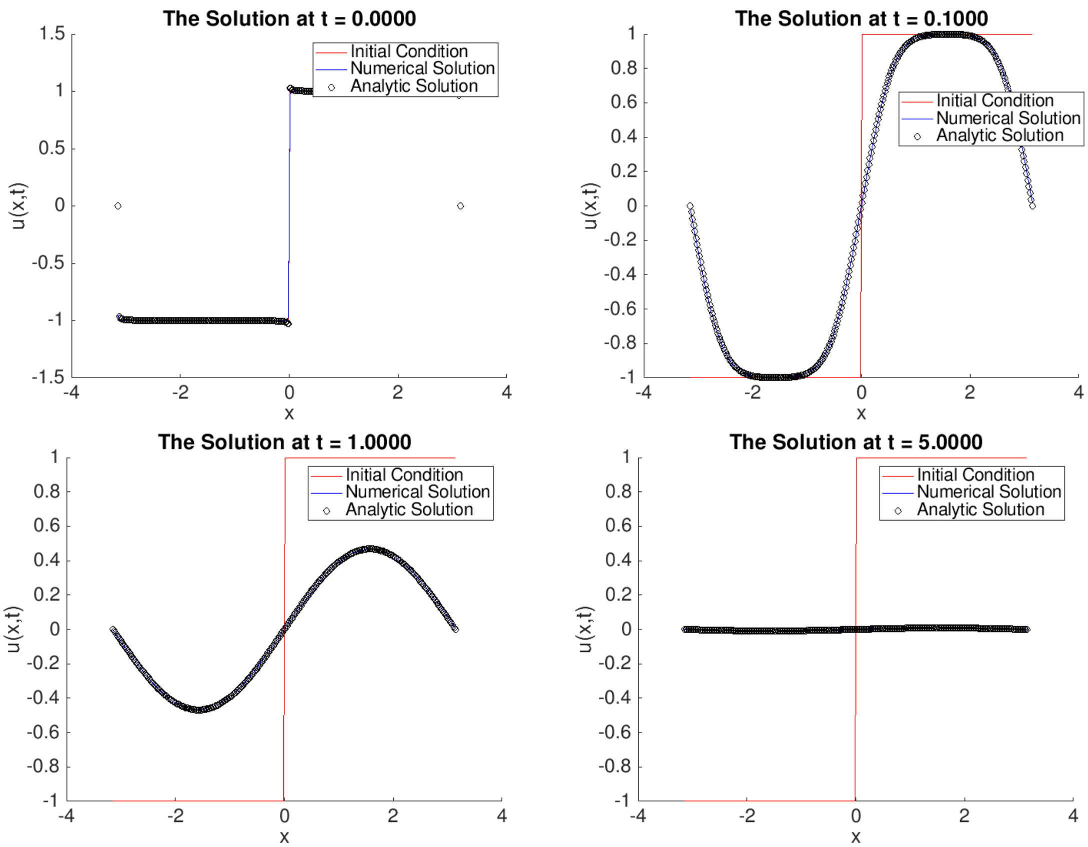


Figure 8: The numerical and analytic solutions to the Heat Equation with periodic boundary conditions.

(b) $|x|$


The Solution at t = 0.0000
The Solution at t = 0.5000
The Solution at t = 1.0000
The Solution at t = 5.0000
Initial Condition
Numerical Solution
Analytic Solution
u(x,t)
x

(c) $f = \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases}$




```
%% Exercise 1: Solving the Heat Equation w/ Periodic Boundary Conditions

L = 2*pi;
K = 1;

% Semidiscretization

N = 300;
h = L/(N+1);
x = linspace(-pi,pi,N);

% Forming the matrix with Periodic Boundary Conditions

A = toeplitz([-2 1 zeros(1,N-2)]);
A(1,N) = 1;
A(N,1) = 1;

% Eigenvalue decomposition for A
```

```

[V,D] = eig(A);
lambda = K*diag(D)/(h^2);

% Solution construction

%f = @humps;
%f = @(x) sin(3*x);
f = @(x) abs(x);
%f = @sign;
u0 = f(x)';
t = 5;
C = V\u0;
u = V*(exp(lambda*t).*C);
tt = linspace(0,1,N);

% Analytic Solution

for n = 1:N
    g = @(x) sin(x.*n);
    gg = @(x) f(x).*g(x);
    g1 = @(x) cos(x.*n);
    gg1 = @(x) f(x).*g1(x);
    proj(:,n) = (1/pi)*quadgk(gg,-pi,pi)*g(x);
    proj1(:,n) = (1/pi)*quadgk(gg1,-pi,pi)*g1(x);
    en(n) = exp(-(n^2)*t);
end

uu = (1/(2*pi))*quadgk(f,-pi,pi) + (proj + proj1)*en';

% Plotting the Analytic and Numerical Solutions

figure
hold on
plot(x,u0,'r-'); % Initial Condition
p = plot(x,u,'b-'); % Numerical Solution
plot(x,uu,'ko'); % Analytic Solution
title(sprintf('The Solution at t = %1.4f',t));
legend('Initial Condition','Numerical Solution','Analytic Solution');
ylabel('u(x,t)');
xlabel('x');
```

EXERCISE 2

Repeat the previous exercise with:

$$\begin{aligned}
&\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}, \quad -\pi < x < \pi, \quad t > 0, \qquad (9)\\
&u(x,0) = f(x),\\
&u(\pi,t) = u(-\pi,t) = 0,\\
&\frac{\partial u}{\partial t}(x,0) = 0.
\end{aligned}$$

Solution: The solution construction of equation is not dissimilar from that of the Heat Equation with periodic boundary conditions, the only difference being the presence of a second partial derivative in time. Thus, in substituting $u(x,t) = v(t)w(x)$ into equation (9), we obtain

$$\frac{d^2 v}{dt^2} = -\omega^2 v, \quad \dot{v}(0) = 0. \tag{10}$$

$$\frac{d^2 w}{dx^2} = -\omega^2 w, \quad w(\pi) = w(-\pi) = 0 \tag{11}$$

with the respective solutions of

$$v(t) = c_1\cos(\omega t) + c_2\sin(\omega t) = c_1\cos\left(\sqrt{-\lambda}t\right) + c_2\sin\left(\sqrt{-\lambda}t\right). \tag{12}$$

$$w(x) = a\cos(\omega x) + b\sin(\omega x) = a\cos\left(\sqrt{-\lambda}x\right) + b\sin\left(\sqrt{-\lambda}x\right). \tag{13}$$

By applying the initial condition, we find that $c_2 = 0$. Because the process for determining λ has already performed, we can thus construct the general solution to the Wave Equation with Periodic Boundary conditions:

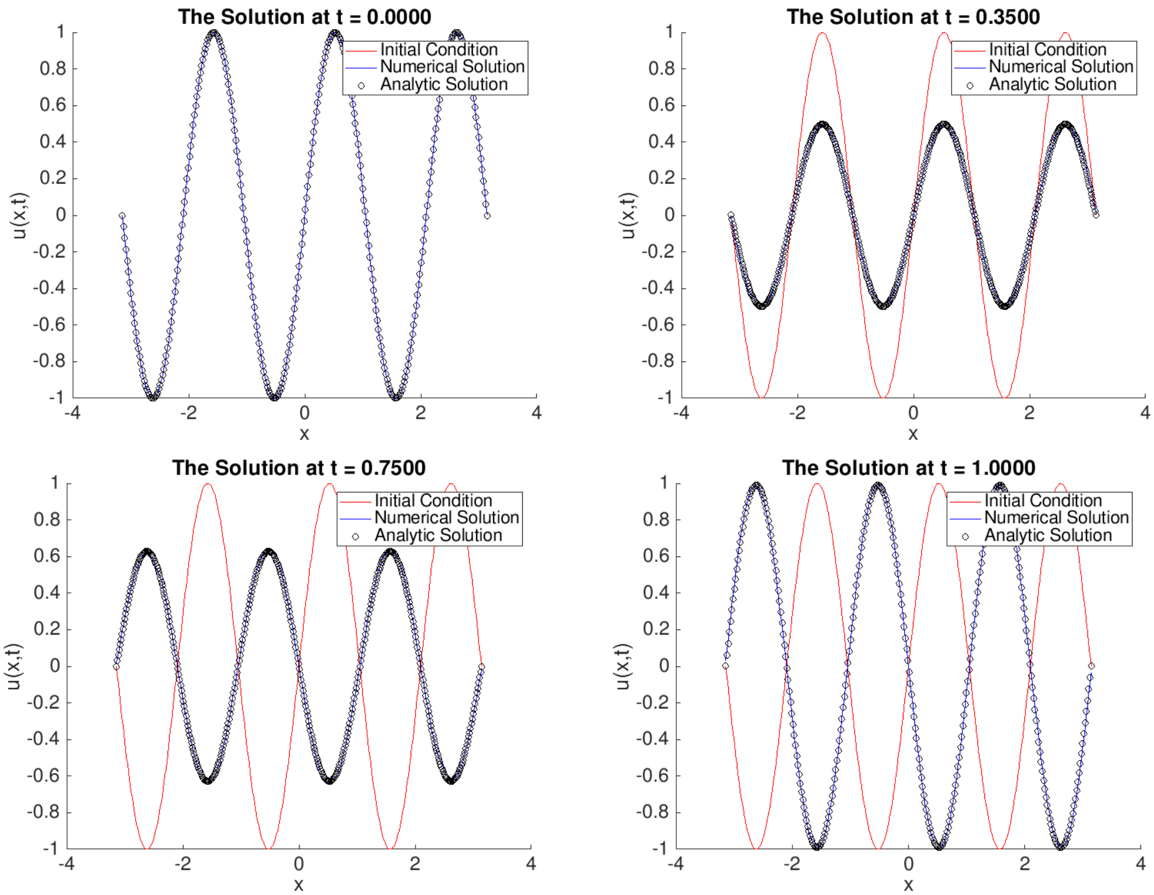
$$u(x,t) = \sum_{n=0}^{\infty}\cos(nt)\big(a_n\cos(nx) + b_n\sin(nx)\big).$$

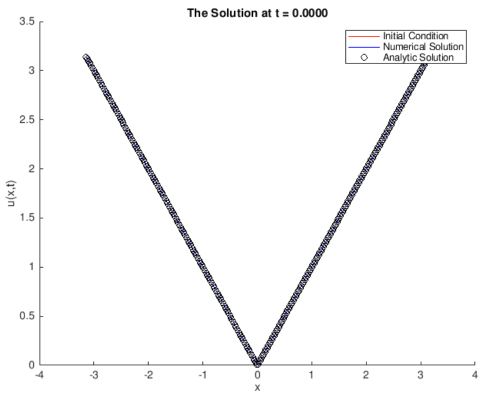
Using the same logic to obtain the Fourier Coefficients, we obtain the solution

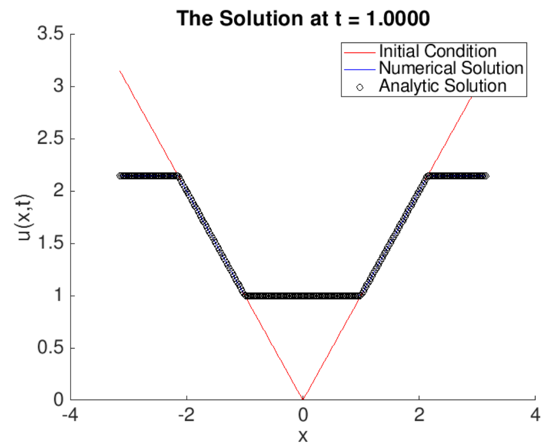
$$\begin{aligned}
u(x,t) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(y)\ dy\\
&+ \sum_{n=1}^{\infty}\cos(nt)\left(\left[\frac{1}{\pi}\int_{-\pi}^{\pi} f(y)\cos(ny)\ dy\right]\cos(nx) + \left[\frac{1}{\pi}\int_{-\pi}^{\pi} f(y)\sin(ny)\ dy\right]\sin(nx)\right).
\end{aligned}$$

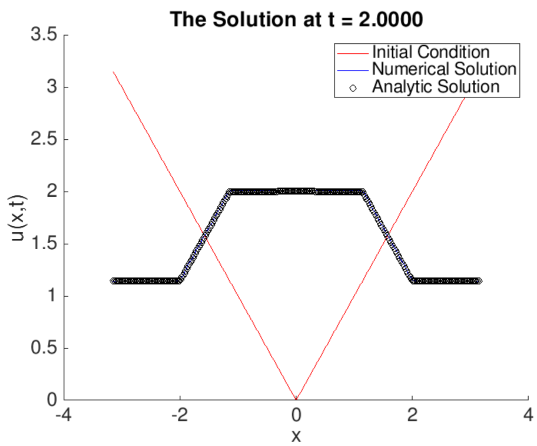
Here are the plots and the code for the solution to the Wave Equation with periodic boundary conditions.

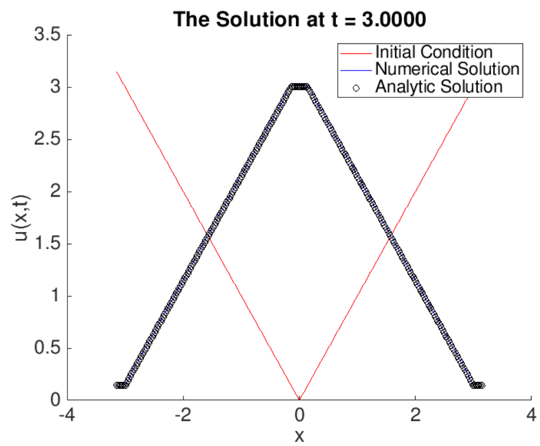
(a)$sin(3x)$

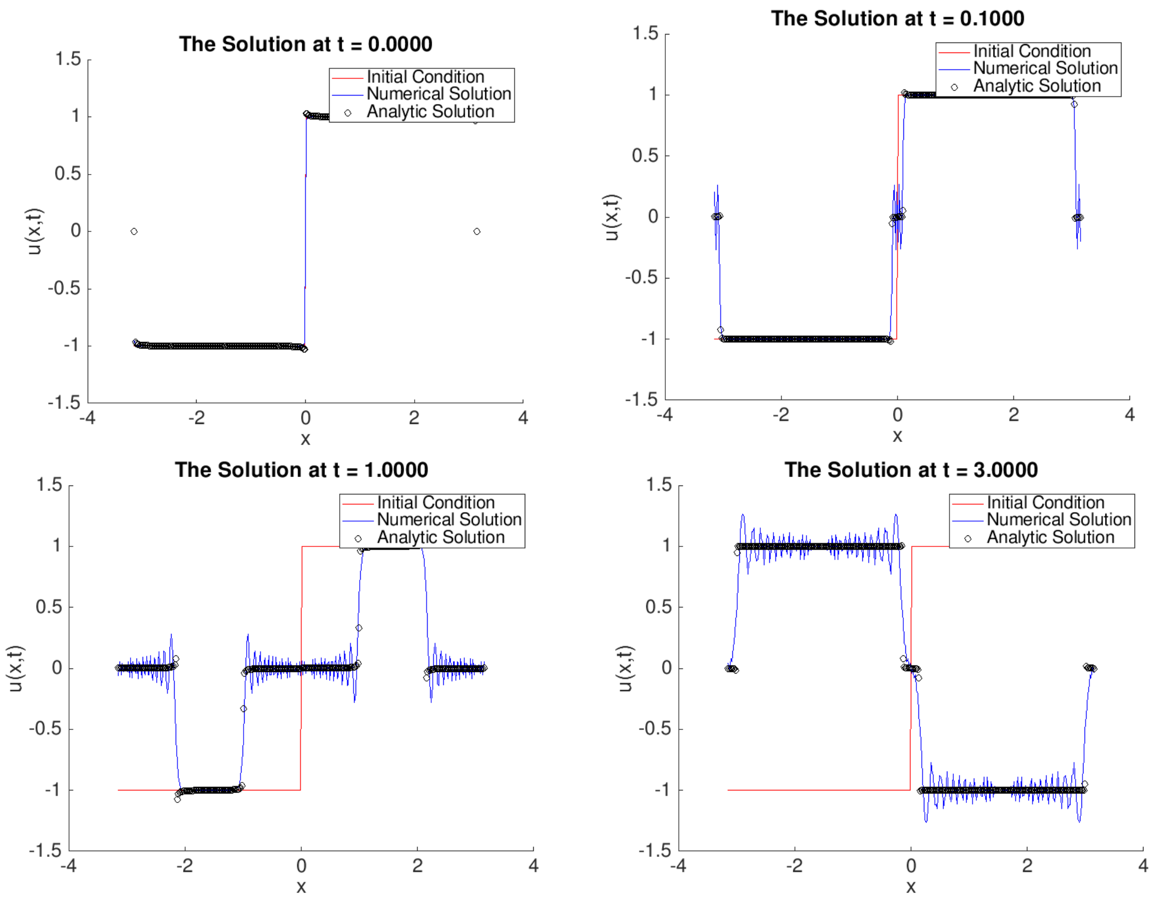

The Solution at t = 0.0000
The Solution at t = 0.3500
The Solution at t = 0.7500
The Solution at t = 1.0000
Initial Condition
Numerical Solution
Analytic Solution
u(x,t)
x


The Solution at t = 0.0000
Initial Condition
Numerical Solution
Analytic Solution
u(x,t)
x


The Solution at t = 1.0000
Initial Condition
Numerical Solution
Analytic Solution
u(x,t)
x


The Solution at t = 2.0000
Initial Condition
Numerical Solution
Analytic Solution
u(x,t)
x


The Solution at t = 3.0000
Initial Condition
Numerical Solution
Analytic Solution
u(x,t)
x




```
%% Exercise 2: Solving the Wave Equation w/ Periodic Boundary Conditions

L = 2*pi;
K = 1;

% Semidiscretization

N = 300;
h = L/(N+1);
x = linspace(-pi,pi,N);

% Forming the matrix with Periodic Boundary Conditions

A = toeplitz([-2 1 zeros(1,N-2)]);
A(1,N) = 1;
A(N,1) = 1;

% Eigenvalue decomposition for A

[V,D] = eig(A);
```

```
lambda = K*diag(D)/(h^2);

% Solution construction

%f = @humps;
%f = @(x) sin(3*x);
%f =@(x) abs(x);
f =@sign;
u0 = f(x)';
t = 3;
C = V\u0;
u = V*(cos(sqrt(-lambda)*t).*C);
tt = linspace(0,1,N);

% Analytic Solution

for n = 1:N
    g = @(x) sin(x.*n);
    gg = @(x) f(x).*g(x);
    g1 = @(x) cos(x.*n);
    gg1 = @(x) f(x).*g1(x);
    proj(:,n) = (1/pi)*quadgk(gg,-pi,pi)*g(x);
    proj1(:,n) = (1/pi)*quadgk(gg1,-pi,pi)*g1(x);
    harm(n) = cos(n.*t);
end

uu = (1/(2*pi))*quadgk(f,-pi,pi) + (proj + proj1)*harm';

% Plotting the Analytic and Numerical Solutions

figure
hold on
plot(x,u0,'r-'); % Initial Condition
p = plot(x,u,'b-'); % Numerical Solution
plot(x,uu,'ko'); % Analytic Solution
title(sprintf('The Solution at t = %1.4f',t));
legend('Initial Condition','Numerical Solution','Analytic Solution');
ylabel('u(x,t)');
xlabel('x');
set(gca,'FontSize',18);
```

EXERCISE 3

Consider the most general first order PDE in two variables:

$$F(x, t, u, u_x, u_t) = 0$$

Under what conditions can one separate variables? Illustrate your hypothesis with an example.

Solution: Separation of variables can always be performed only when we have a linear homogeneous differential equation whose boundary conditions are also linear and homogeneous.

HOMEWORK 4

EXERCISE 1

In the handout we made the following remark about the L^1-norm of the Dirichlet kernel: $||D_N||_1 = C\log(N)$. Write a Matlab code to estimate the constant C.

Solution: In order to determine the constant C, the following MATLAB code was produced with the following plot in Figure 1.

```
%% Exercise 1

N = 5000;

one_norm = zeros(1,N);

for n = 1:N
    dir_ker = @(s) abs(sin((n+0.5)*s)./sin(s/2));
    one_norm(n) = integral(dir_ker,0,2*pi)/(2*pi);
end

x = log(1:N);
y = one_norm;

p = polyfit(x,y,1);
pp = polyval(p,x);

figure
hold on,
plot(x,y,'b-');
plot(x,pp,'ro');
xlabel('$$log(N)$$','interpreter','latex');
ylabel('$$L^{1}$$-Norm of $$D_{N}$$','interpreter','latex');
legend('Data','Fit');
title(sprintf('Constant = %f',p(1)));
set(gca,'FontSize',18);
```

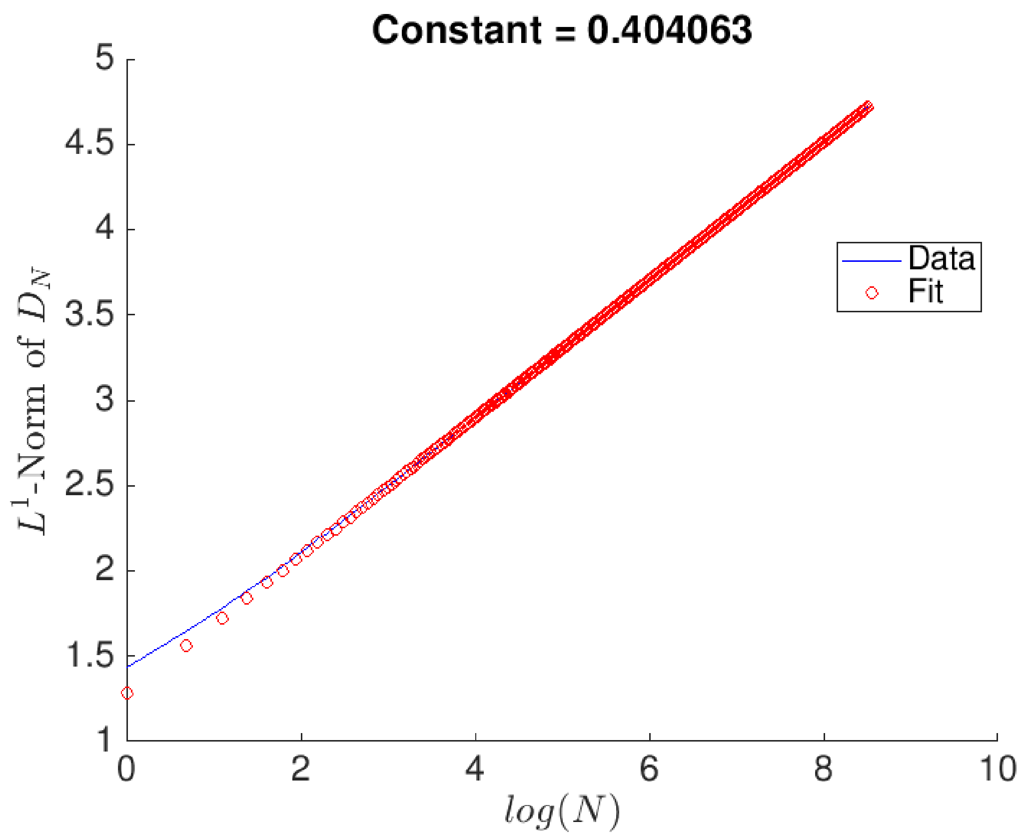


Figure 9: Plot of the L^1-norm of the Dirichlet Kernel against $\log(N)$.

Interestingly enough, as $N \to \infty$, the constant approaches $4/\pi^2 \approx 0.405$. Thus, the relationship is $||D_N||_1 = \frac{4}{\pi^2}\log(N)$.

Exercise 2

Study the following code snippet:

```
f = @(x) pi-abs(x-pi);
N = 50;
c = zeros(1,N);

for n=1:N
c(n) = quadgk(@(x)exp(-1i*n*x).*f(x),0,2*pi)/(2*pi);
end

xx = 1:N;
yy = abs(c);
```

```
ind = yy>eps('single');
xx = log(xx(ind));
yy = log(yy(ind));
p = polyfit(xx,yy,1);

figure
plot(xx,yy,'ro')
hold on
plot(xx,polyval(p,xx))
title(sprintf('p = [%f %f]',p))
```

The script computes fifty Fourier coefficients c_n, $n = 1, \ldots, 50$ of the function $f(x) = \pi - |x-\pi|$, selects the nonzero ones, plots the magnitudes in logarithmic coordinates and fits a straight line to the resulting data. Evidently, the magnitudes of the Fourier coefficients decay quadratically, $c_n = \mathcal{O}(n^{-2})$. and the same conclusion can be reached for any other continuous non-differentiable function. As the handout hints, differentiability accelerates convergence. Replace the function in the above script with a C^1 function of your choice and find the rate of decay of Fourier coefficients. Repeat the exercise with functions from C^2 and C^3: you may need to think about a general construction of a function that has prescribed number of derivatives. Try to formulate a conjecture about the rate of decay of Fourier coefficients of functions in C^k. What might happen if $f \in C^\infty$?

Solution: The functions that were used for analysis were of the form

$$f(x) = |x|^{k+1},$$

which are k-times differentiable. Further, the Fourier coefficients were computed over the interval $[-\pi, \pi]$, which doesn't change the computation because the interval still has length 2π. Here is the code that determined the rate of convergence of the C^1, C^2, and C^3 functions, and the associated plots.

```
%% Exercise 2

%f = @(x) abs(x);
%f = @(x) abs(x).^2; % C1 function
%f = @(x) abs(x).^3; % C2 function
%f = @(x) abs(x).^4; % C3 function
%f = @(x) abs(x).^500; % C499 function
N = 50;
nn= 50;

% Determining the first 50 Fourier Coefficients

c = zeros(1,N);
```

```
for n = 1:N
    c(n) = integral(@(x)exp(-1i*n*x).*f(x),-pi,pi)/(2*pi);
end

% Determining how the Order changes with Differentiability

for k = 1:620
    g = @(x) abs(x).^((k-1)+1));
    for l = 1:nn
        c1(l) = integral(@(x)exp(-1i*l*x).*g(x),-pi,pi)/(2*pi);
    end
    uu = 1:620;
    vv = abs(c1);
    ind = vv>eps('single');
    uu = log(uu(ind));
    vv = log(vv(ind));
    p1(k,:) = polyfit(uu,vv,1);
end

% Fitting the Order of Convergence as a function of Differentiability

logkk = log(1:620)';
logp1 = log(p1(:,1));
pk = polyfit(logkk(end-400:end),logp1(end-400:end),1);
ppk = polyval(pk,logkk(end-400:end));

% Plotting the Log of Convergence vs. Log of Differentiability

figure
hold on
plot(logkk,logp1,'b-');
plot(logkk(end-400:end),ppk(end-400:end),'ro');
ylabel('Log of Rate of Convergence');
xlabel('Log of Differentiability');
legend('Convergence vs. Differentiability','Data Fit');
title(sprintf('Order of Convergence: p = %1.4f, N = %1.0f',[pk(1),nn]));
set(gca,'FontSize',18);

% Plotting Order of Convergence vs. Differentiability

figure
hold on
plot((1:620),abs(p1(:,1)),'b-');
xlabel('k');
ylabel('Order of Convergence');
title('The Rate of Convergence r(k,N)');
set(gca,'FontSize',18);

% Fitting the data for the order of convergence

xx = 1:N;
```

```
yy = abs(c);
ind = yy>eps('single');
xx = log(xx(ind));
yy = log(yy(ind));
p = polyfit(xx(end-20:end),yy(end-20:end),1);

figure
hold on
plot(xx(end-20:end),yy(end-20:end),'b-');
plot(xx(end-20:end),polyval(p,xx(end-20:end)),'ro');
xlabel('log(N)');
ylabel('Log of Fourier Coefficients');
title(sprintf('Rate of Convergence of |x|^{4} = [%1.4f %1.4f]',p));
set(gca,'FontSize',18);
```

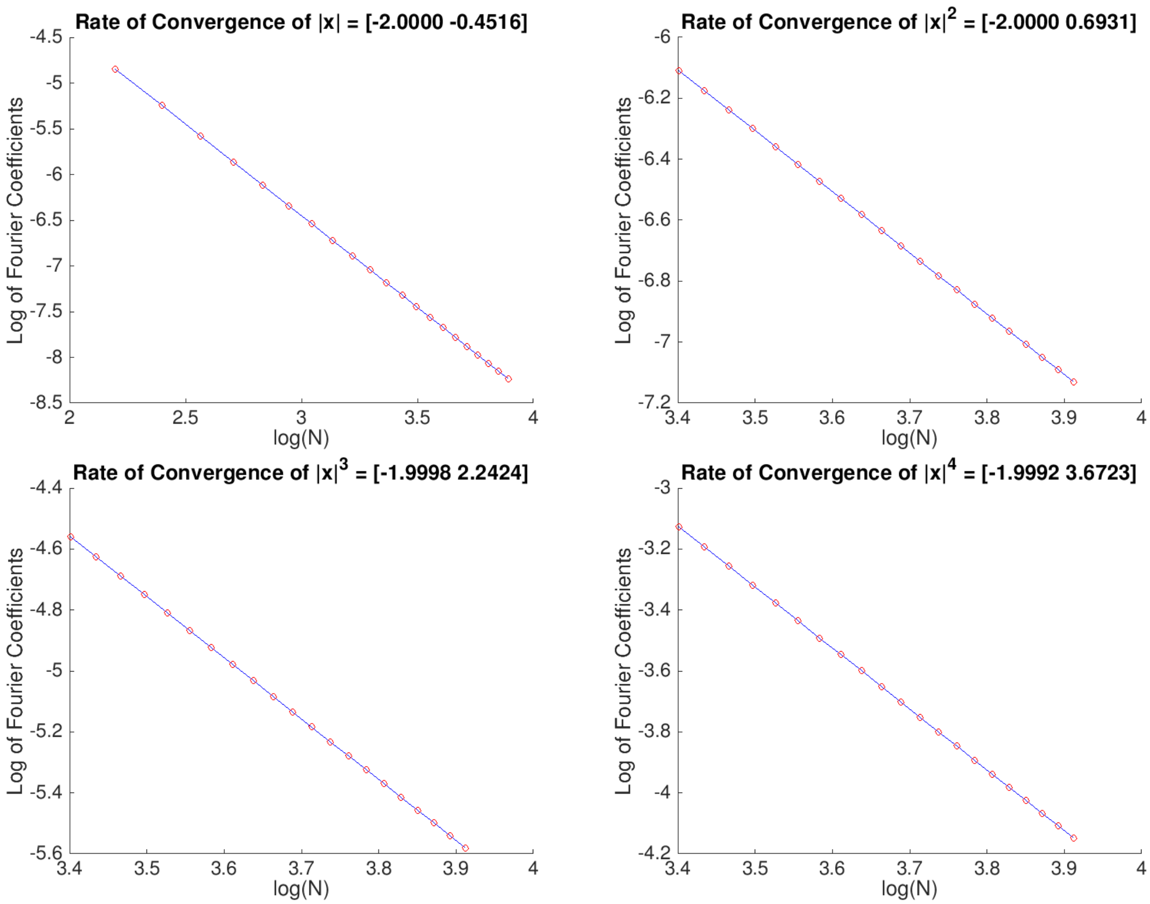


Figure 10: Plots of $f(x) = |x|^{k+1}$ for the values of $k = 0, 1, 2, 3$.

In looking at the above plots in Figure 2, it is not at all obvious that as differentiability increases, the rate of decay of Fourier coefficients decreases, meaning that the order of convergence increases. Because the magnitude of the Fourier coefficients decreases less and less as k increases, this implies that the convergence of the Fourier series converges more quickly. We can see this behavior in Figure 3, where the function in question is $f(x) = |x|^{500}$. Further, this becomes very numerically ill-conditioned for increasing powers, hence the terrible "fit."

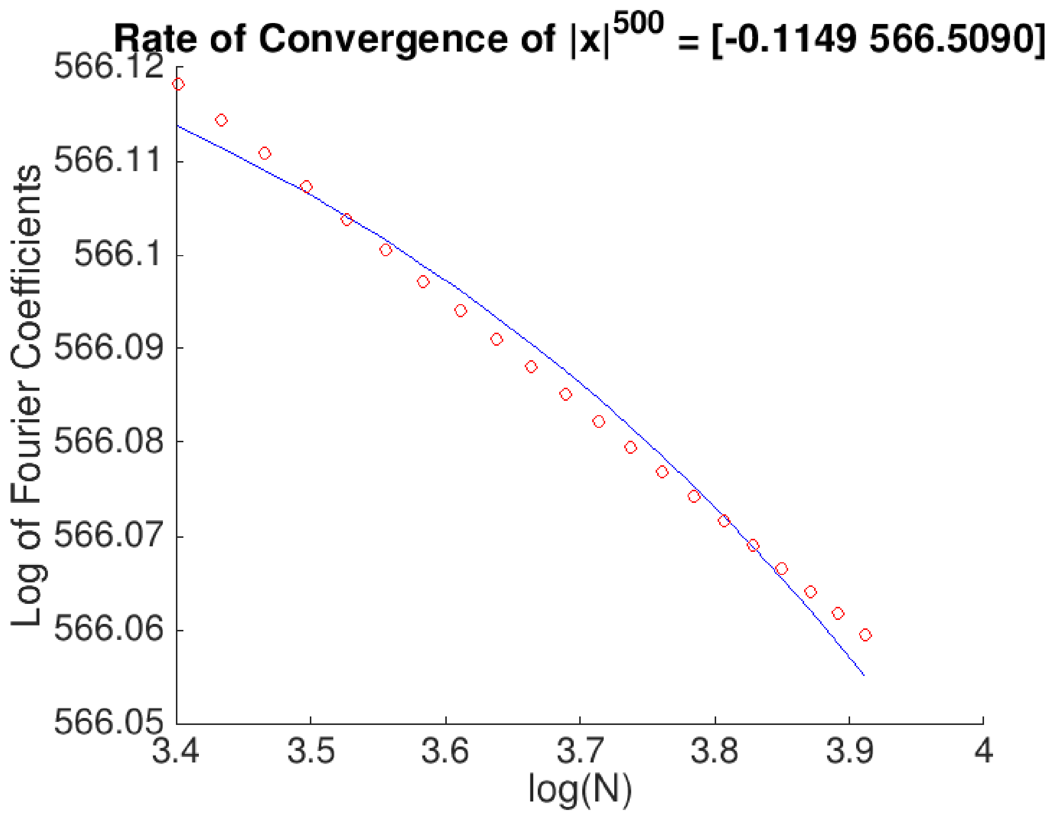


Figure 11: This plot conveys that the differentiability k drastically affects the rate of convergence.

However, I decided to measure the rate of decay of the Fourier coefficients as a function of differentiability. The resulting plots are on the following page. Further, not only does the differentiability affect the rate of convergence of Fourier Series, but also the number of Fourier coefficients. As obvious as this seems, it is wickedly cool to see this effect graphically. In Figure 4, in computing the Fourier coefficients for the functions $f(x) = |x|^{k+1}$ for $k = 1, \ldots, 620$[4], we see, in the upper left plot, that the rate of decay decreases quickly for $N = 50$, and begins to level out as N increases. This result is a direct consequence of Riemann's Lemma, and for functions $f \in C^{\infty}$, I think that the rate of decay will become constant. We see that as N increases, the graph continues to become less and less steep, suggesting that, eventually, the graph will shallow out having a constant rate of decay. This perhaps further

[4]The reason why the value 620 was chosen was because any higher values caused the algorithm to be ill-conditioned. In practice, though, as long as $k < \infty$, the functions would be k-times differentiable.

implies that the number of Fourier coefficients N has a greater effect than the differentiability k, but I'm not sure how to prove that conjecture, at least numerically.

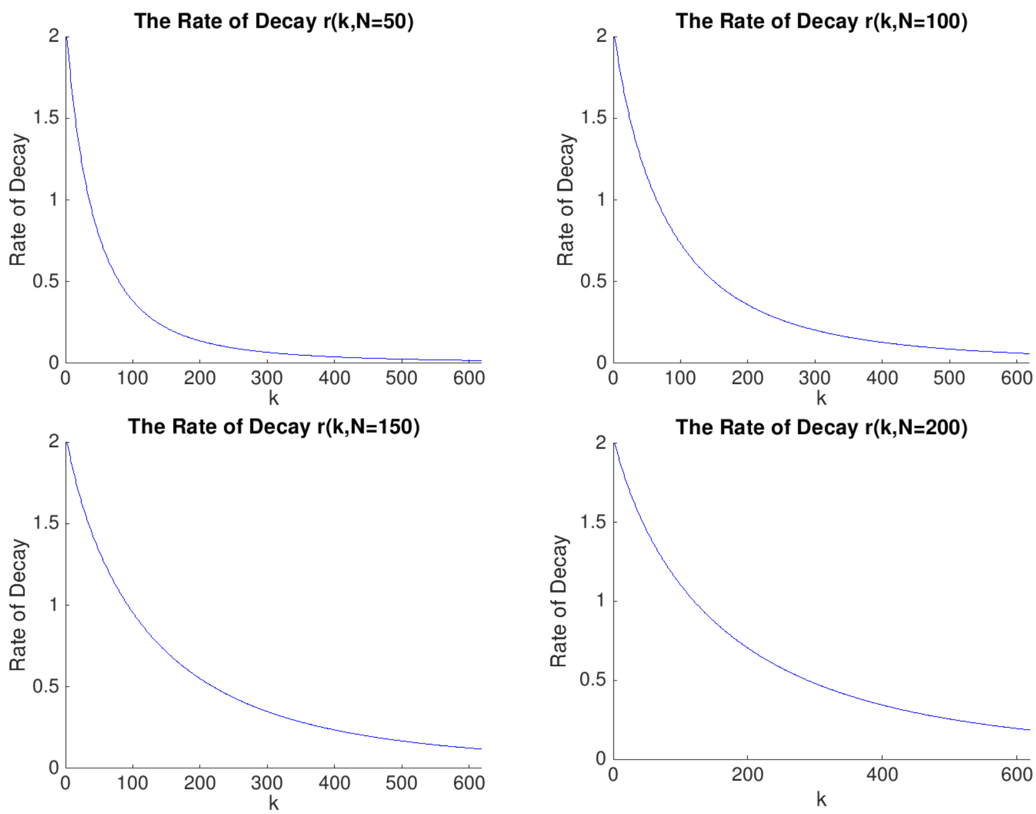


Figure 12: Plots of the order of convergence as a function of differentiability k and the number of Fourier coefficients N.

EXERCISES 3 AND 4

Write a script that tests the rate of convergence of Bernstein polynomials to a continuous function in the spirit of the previous exercise. You can measure the rate of convergence at a point or use infinity norm convergence—your choice.

Solution: Bernstein polynomials have a consistent rate of convergence; the rate of convergence is $\mathcal{O}(n^{-1})$. This was determined using the sup-norm, which provides a much stronger sense of convergence, i.e., uniform convergence. Further, as mentioned in the handout, uniform convergence implies more often than not, pointwise convergence. We can see this from the following plots in Figure 5.

```
%% Exercises 3 and 4

%f = @(x) abs(x);
```

```
%f = @(x) abs(x).^2;
%f = @(x) abs(x).^3;
%f = @(x) abs(x).^4;
%f = @(x) sign(x);
%f = @(x) sin(x);
%f = @(x) 1.*(x<0.5) + 0.*(x>0.5);
N = 200;

x = linspace(0,1,100);
sum_b = zeros(size(x));
%xx = zeros(N,N);

for l = 1:N
    for k = 1:l
        for j = 0:k
            b = @(x) f(j/N).*nchoosek(N,j).*(x.^j).*((1-x).^(N-j));
        end
    end
    sum_b = sum_b + b(x);
    R = (sum_b - f(x))/l;
    E(k) = max(abs(R));
end

xx = log(1:N-1);
yy = log(E(1:end-1));
p = polyfit(xx,yy,1);

% Plot of the Polynomial againss the Actual Function

figure
hold on
plot(x,f(x),'b-');
plot(x,sum_b,'ro');
xlabel('x');
ylabel('f(x)');
legend('Graph','Bernstein Fit');
title('Bernstein Fit of f(x) = sin(x)');
set(gca,'FontSize',18);

% The Order of Convergence

figure
hold on
plot(xx,yy,'b-');
plot(xx,polyval(p,xx),'ro');
xlabel('log(N)');
ylabel('log(E)');
legend('Error','Fit');
title(sprintf('Order of Convergence = [%1.3f %1.3f]',p));
set(gca,'FontSize',18);
```

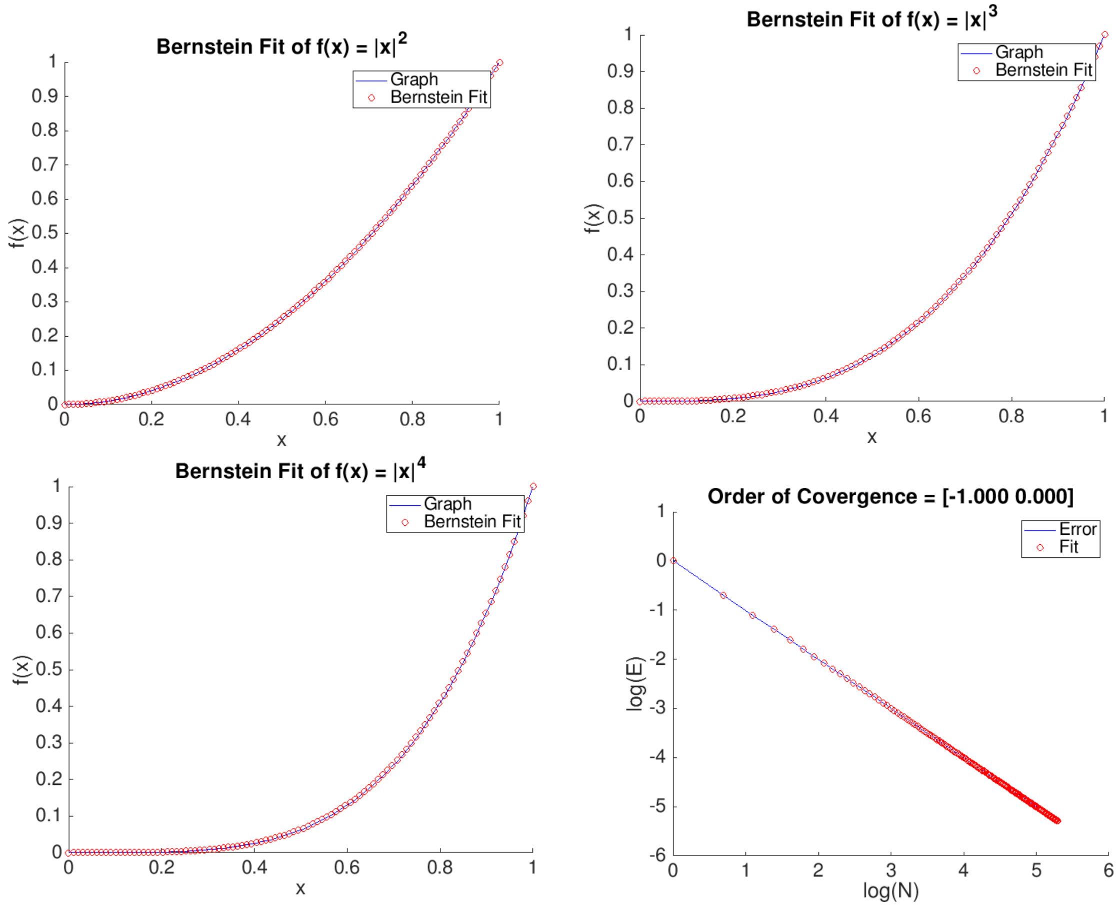


Figure 13: Figures of the Bernstein polynomials for $f(x) = |x|^{k+1}$ for the values of $k = 0, 1, 2, 3$. The last figure is the order of convergence for all three cases.

To further see the relation, Figure 6 is a plot of the direct error.

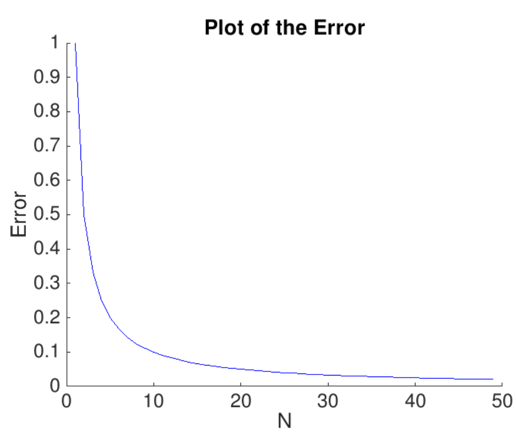


Figure 14: A plot of the direct error.

This trend holds for all k-times differentiable functions, for the piecewise function

$$f(x) = \begin{cases} 1, & x < \frac{1}{2} \\ 0, & x > \frac{1}{2} \end{cases}, \tag{1}$$

as seen in Figure 7, and for functions $f \in C^{\infty}$. For $f(x) = \sin(x)$, the following plots were generated, as shown in Figure 8.

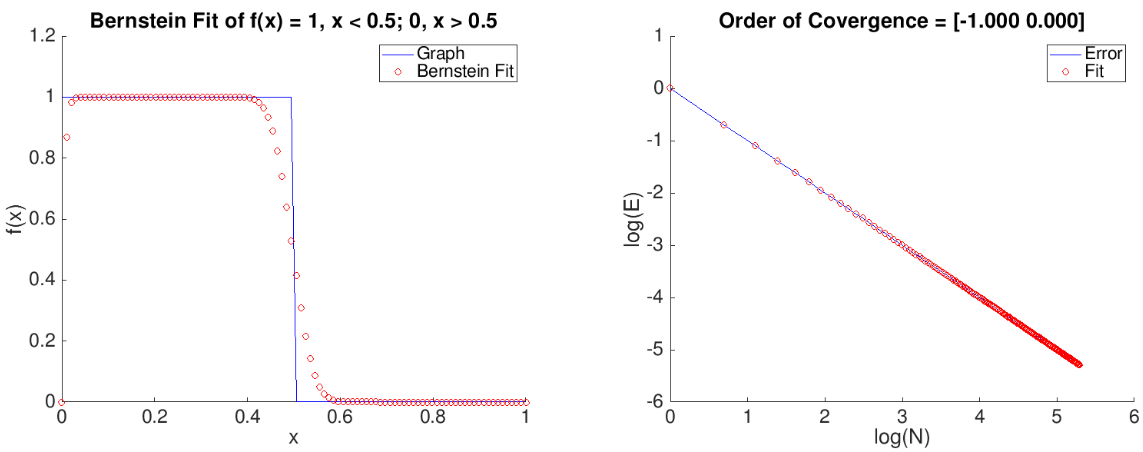


Figure 15: Plots of the fit for equation (1) and the associated rate of convergence.

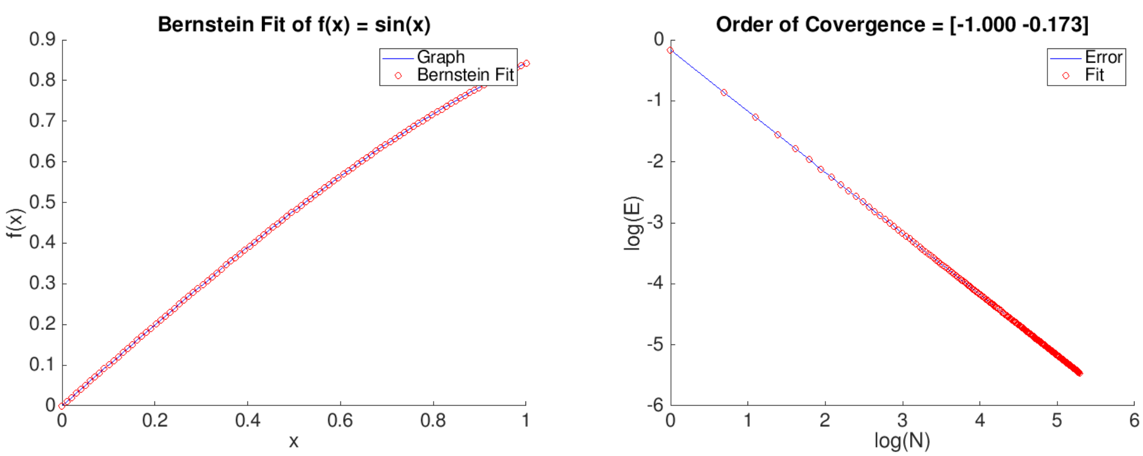


Figure 16: A plot of the Bernstein fit for $f(x) = \sin(x)$ and the order of convergence.

As we can see from the first plot in Figure 7 that there is no Gibbs phenomenon that occurs. This perhaps follows from the fact that polynomials are being used to approximate the function as opposed to other methods. This suggests that as $N \to \infty$, $B_N(x) \to f$. Because this is true for C^k and C^{∞} functions, this would further imply that $B_N(x)$ always converges to f on the specified interval $[0, 1]$.

EXERCISE 5

Let $f = sign(\pi - x)$ and let $S_N(f)$ be the N-th Fourier sum. Show experimentally that $S_N(f)$ converges to f in L^2-norm. In fact, find the rate of convergence. Also, demonstrate with a Matlab script that convergence is not uniform, that is, there is no convergence in infinity norm.

Solution: As mentioned in Exercises 3 and 4, uniform convergence imposes a strong condition on the convergence of sequences of functions. The definition of uniform convergence is as follows:

> **Definition 1.1.** *A sequence $\{f_n\}$ defined on a set S converges uniformly to a function f if for every $\epsilon > 0$ there exists an N so that $n \geq N$ implies that $|f_n(x) - f(x)| < \epsilon$ holds for all $x \in S$.*

This means that no matter how small we pick ϵ, there will be an N associated with that ϵ such that we can always find an $n \geq N$ where the sequence will approximate the function so closely that the magnitude of the difference will be less than ϵ. The reason why uniform convergence is stronger than pointwise convergence is because pointwise convergence depends on the set for which the function f is defined, whereas uniform convergence does not. In looking at the N-th Fourier sum of $f(x) = \text{sign}(\pi - x)$, we have the following plots in Figure 9.

```
%% Exercise 5

L = pi;
N = 100;
x = linspace(-L,L,N);

f = @(x) sign(pi - x);

err_l2 = zeros(size(1:N));
err_inf = zeros(size(1:N));
ss = zeros(length(x),length(x));
SS = zeros(length(x),length(x));

% Infinity-norm and L2-norm convergence tests

for n = 1:N
    ss = zeros(size(x));
    for m = 1:n
        g = @(x) cos(m*x);
        g1 = @(x) sin(m*x);
        gg = @(x) g(x).*f(x);
        gg1 = @(x) g1(x).*f(x);
        ss = ss + 1/L*quadgk(gg,-L,L)*g(x) + 1/L*quadgk(gg1,-L,L)*g1(x);
    end
```

```
    ss = ss + quadgk(f,-L,L)*ones(size(ss))./(2*L);
    R = abs(f(x) - ss);
    err_l2(n) = norm((R.^2).*(2*L)/n,2);
    err_inf(n) = max(abs(R));
end

% L2-norm fit

logN = log(1:N);
logE_l2 = log(err_l2);
p_l2 = polyfit(logN(end-20:end),logE_l2(end-20:end),1);
order_l2 = p_l2(1);

% Infinity-norm "fit"

logN = log(1:N);
logE_inf = log(err_inf);
p_inf = polyfit(logN(end-20:end),logE_inf(end-20:end),1);
order_inf = p_inf(1);

% Plot of L2-norm convergence

figure
hold on
plot(logN,logE_l2,'ro');
plot(logN,polyval(p_l2,logN),'b-');
xlabel('Log of N');
ylabel('Log of Error');
title(sprintf('$$L^{2}$$-Norm Convergence: ...
    %f',order_l2),'interpreter','latex');
set(gca,'FontSize',18);

% Plot of Infinity-norm convergence

figure
hold on
plot(logN,logE_inf,'ro');
plot(logN,polyval(p_inf,logN),'b-');
xlabel('Log of N');
ylabel('Log of Error');
title(sprintf('Infinity Norm Convergence: %f',order_inf));
set(gca,'FontSize',18);

% Plot of function and Fourier Approximation

figure
hold on
plot(x,f(x),'b-');
plot(x,ss,'ro');
xlabel('x');
ylabel('f(x)');
```

```
legend('Function','Fourier Approximation');
title('The Function and Fourier Approximation');
set(gca,'FontSize',18);
```

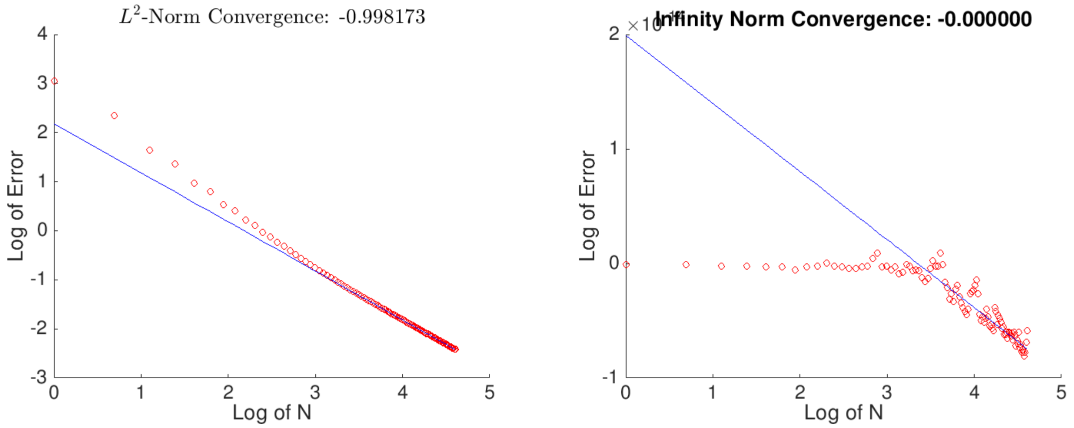


Figure 17: The L^2- and L^∞-norms for $|S_N(x) - f(x)|$, where $f(x) = \text{sign}(\pi - x)$.

Upon typing this, I was thinking about why uniform convergence fails and why the L-2 norm doesn't, and hence plotted the errors for both against N.

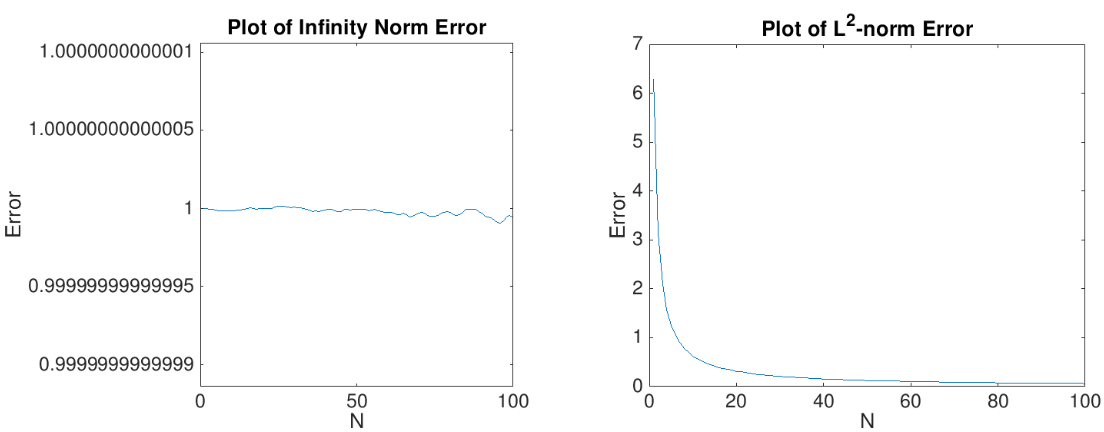


Figure 18: Plots of the

Although I did not include the plots for increasing values of N, it is true that for infinity norm convergence, the values vary about the value 1, in a very inconsistent manner. However, for L2-norm convergence, we see that the error decreases as $\mathcal{O}(n^{-1})$.

EXERCISE 6

If you had advanced physics, you may be familiar with Legendre polynomials $\{P_n\}$ *the first five of which are listed below:*

$$\begin{aligned}
P_0 &= 1\\
P_1 &= x\\
P_2 &= \frac{3}{2}x^2 - \frac{1}{2}\\
P_3 &= \frac{5}{2}x^3 - \frac{3}{2}x^2\\
P_4 &= \frac{35}{8}x^4 - \frac{15}{4}x^3 + \frac{3}{8}
\end{aligned}$$

Legendre polynomials are orthogonal on the interval $[1,1]$:

$$\langle P_n, P_m\rangle = \int_{-1}^{1} P_n(x)P_m(x)\ dx = \begin{cases} 0, & \text{if } n \neq m;\\ \frac{2}{2n+1}, & \text{if } n = m \end{cases}$$

As a matter of fact, the above orthogonality relation can be used to define $P_n's$. *The following code snippet plots the first five Legendre polynomials. It is not very efficient but you can adapt it for simple computations; alternatively you can write your own code for generating Legendre polynomials (this can be done with a three-term recurrence relation).*

```
x = linspace(-1,1);
figure
hold on
for n=1:5
    P = legendre(n,x);
    plot(x,P(1,:))
end
```

If f *is a function on* $[1,1]$ *then its Legendre series is:*

$$\sum_{n=0}^{\infty} \frac{\langle f, P_n\rangle}{\langle P_n, P_n\rangle} = \sum_{n=0}^{\infty} \frac{2n+1}{2}\langle f, P_n\rangle P_n(x).$$

Obviously, Legendre series are special kinds of Fourier series and all of the questions about convergence of Fourier series can be asked about Legendre series. Plot the partial sums of the Legendre series for the function $f = sign(x)$ *for* $N = 2, 4, 8, 16, 32, 64$. *Then try to answer the following questions. Does the series converge? If so, at what rate? Do Legendre series exhibit Gibbs phenomenon?*

Solution: Here is the code and the plots of the desired Legendre series.

```
%% Exercise 6 Legendre Polynomials

N = 100;
x = linspace(-1,1,1000);

f = @(x) sign(x);

s = zeros(size(x));
ss = zeros(size(x));

for n = 1:N
    g = @(x) f(x).*legendre_n(x,(n-1));
    proj = ((2*(n-1)+1)/2)*quadgk(g,-1,1).*legendre_n(x,(n-1));
    s = s + proj;
    R = f(x) - s;
    E(n) = max(abs(R));
end

for j = 1:6
    subplot(2,3,j);
    ss = zeros(size(x));
    for nn = 1:(2^j)
        gg = @(x) f(x).*legendre_n(x,(nn-1));
        proj1 = ((2*(nn-1)+1)/2)*quadgk(gg,-1,1).*legendre_n(x,(nn-1));
        ss = ss + proj1;
    end
    hold on
    plot(x,f(x),'b-');
    plot(x,ss,'r-');
    title(sprintf('N = %1.0f',nn));
    set(gca,'FontSize',18);
end

xx = 1:N;
yy = E;
p = polyfit(xx,yy,1);
pp = polyval(p,xx);

% Fit

figure
hold on
plot(xx,yy,'b-');
plot(xx,pp,'ro');
xlabel('N');
ylabel('Error');
title(sprintf('Legendre Polynomial Convergence: %1.3f',p(1)));
set(gca,'FontSize',18);
```

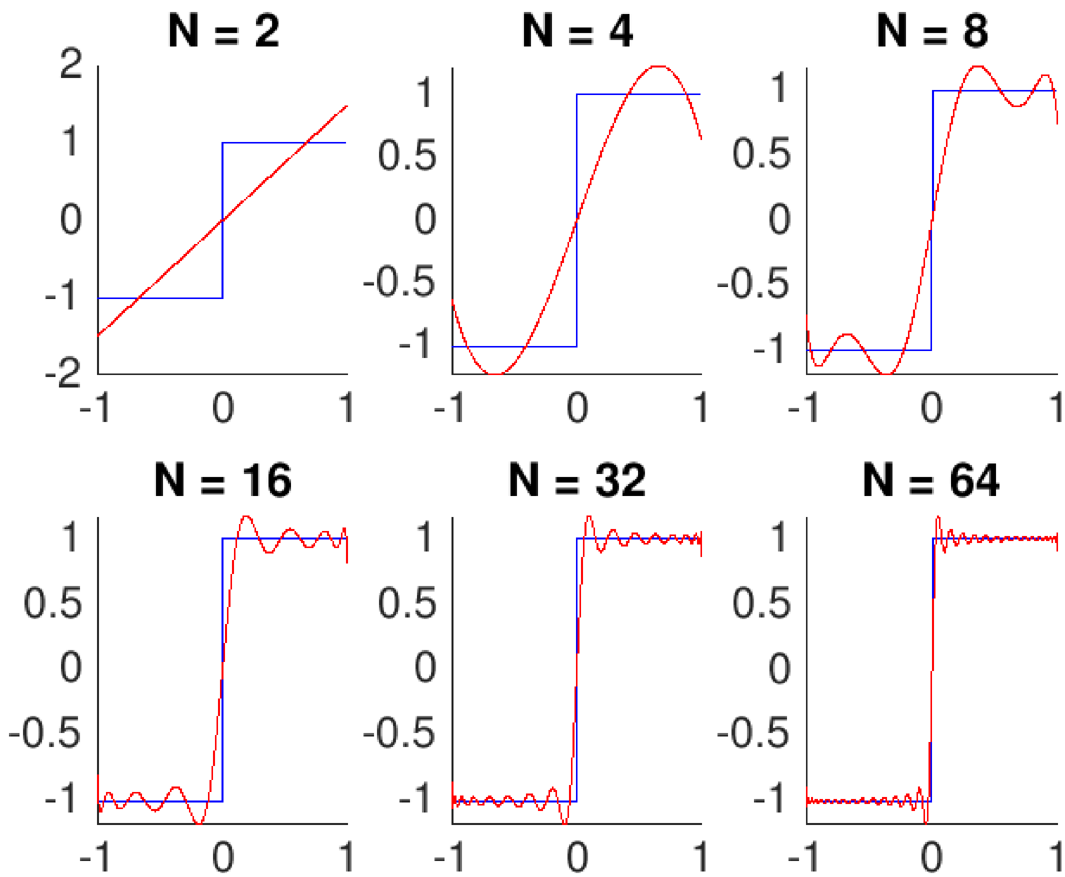


Figure 19: Plots of the Legendre series for $f(x) = \text{sign}(x)$ for $N = 2, 4, 8, 16, 32, 64$.

We see that the Legendre polynomials approximate the function well, and indeed do exhibit Gibbs phenomenon, as conveyed by the following code block.

```
>> abs(max(s) - max(f(x)))/max(f(x)) .*100 % N = 64

ans =

   17.9127
```

Further, the Legendre polynomials do indeed converge to the function, with an order of 0.001, as demonstrated in Figure 12.

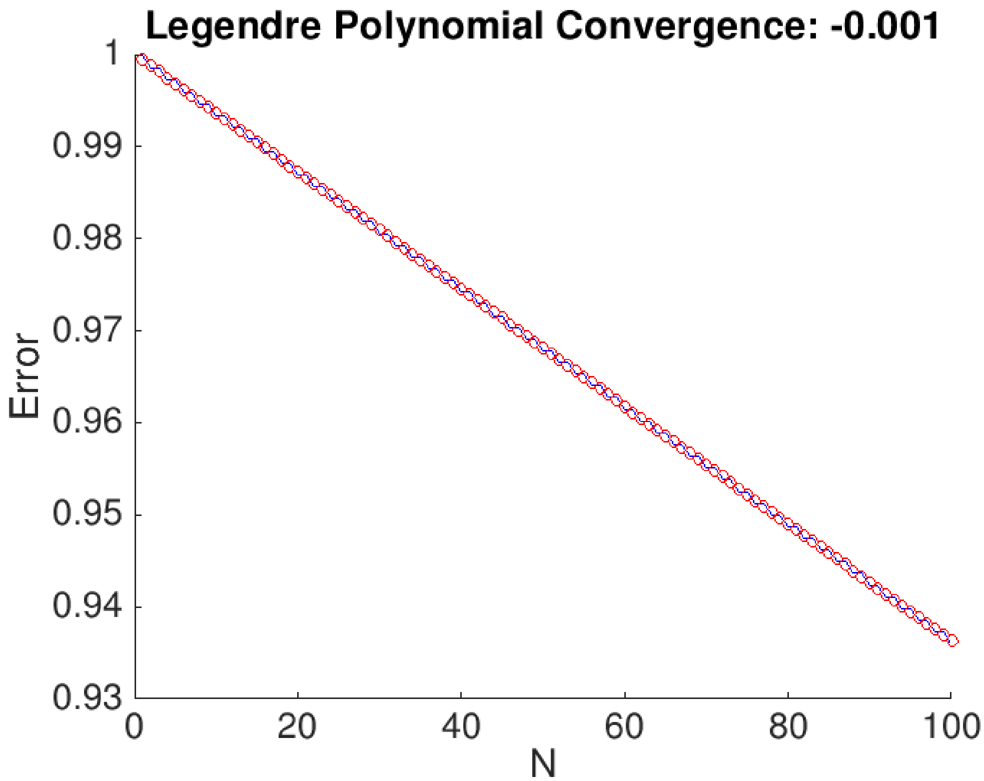


Figure 20: The convergence of the Legendre Series of $f(x) = \text{sign}(x)$.

EXAM 1

EXERCISE 1

1. Let $a > 1$. consider the following Dirichlet problem for the Heat Equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad -a < x < a, \quad t > 0, \tag{1}$$
$$u(-a,t) = u(a,t) = 0, \quad u(x,0) = \begin{cases} 1, & \text{if } |x| \leq 1 \\ 0, & \text{otherwise} \end{cases} .$$

(a) (20 Points) Derive the series solution.

(b) (20 Points) Set $a = 2$ and plot your symbolic solution for $t = 0.1$. Superimpose on the same plot the numerical approximations obtained through semidiscretization with $N = 20, 40, 80$. (Provide code and don't forget to add a legend to the plot.) Comment on the plot: does it validate the symbolic solution?

(c) (10 Points) Explain what happens to the (symbolic) solution as $a \to \infty$.

Solution

(a): Let us assume that the solution u is of the form $u(x,t) = h(t)g(x)$. By doing so, we can invoke separation of variable in order to solve equation (1). By substituting u into equation (1), we obtain a system of linear ODE

$$\frac{dh}{dt} = -\omega^2 h \tag{2}$$
$$\frac{d^2 g}{dx^2} = -\omega^2 g, \quad g(-a) = g(a) = 0 \tag{3}$$

where $-\omega^2 = \lambda$. The respective solutions to equations (2) and (3) are

$$h(t) = Ae^{-\omega^2 t} = Ae^{\lambda t} \tag{4}$$
$$g(x) = \alpha \cos(\omega x) + \beta \sin(\omega x) = \alpha \cos\left(\sqrt{-\lambda} x\right) + \beta \sin\left(\sqrt{-\lambda} x\right). \tag{5}$$

To satisfy the initial boundary conditions, we obtain the following matrix vector system

$$\begin{bmatrix} \cos\left(\sqrt{-\lambda} a\right) & \sin\left(\sqrt{-\lambda} a\right) \\ \cos\left(\sqrt{-\lambda} a\right) & -\sin\left(\sqrt{-\lambda} a\right) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{6}$$

In order for equation (6) to be true, we can either take the trivial solution, where $a = b = 0$, or we can determine λ in order that the determinant is equal to zero. To obtain non-trivial solutions, we have

$$\begin{aligned}\det(A) &= -2\sin\left(\sqrt{-\lambda}a\right)\cos\left(\sqrt{-\lambda}a\right) = 0\\ &\implies \sin\left(\sqrt{-\lambda}a\right)\cos\left(\sqrt{-\lambda}a\right) = 0.\end{aligned} \tag{7}$$

Using the zero-product property, we can determine the arguments of both the sine and cosine terms so that equation (7) is true. As a result, we obtain two eigensolutions to equation (3), given by

$$\lambda_{\sin} = -\frac{n^2\pi^2}{a^2}$$
$$\lambda_{\cos} = -\frac{\left(n+\frac{1}{2}\right)^2\pi^2}{a^2}$$

where $\lambda_{\sin}$ is the result of letting sine equal zero and $\lambda_{\cos}$ is the result of letting cosine equal zero. Thus, the eigensolutions[5] of equation (1) with eigenvalues $\lambda_{\sin}$ are of the form

$$u_n(x,t) = e^{-\frac{n^2\pi^2}{a^2}t}\left(a_n\cos\left(\frac{n\pi x}{a}\right) + b_n\sin\left(\frac{n\pi x}{a}\right)\right),$$

and by linearity, we have

$$u(x,t) = \sum_{n=0}^{\infty} e^{-\frac{n^2\pi^2}{a^2}t}\left(a_n\cos\left(\frac{n\pi x}{a}\right) + b_n\sin\left(\frac{n\pi x}{a}\right)\right).$$

In order to determine the coefficients a_n, b_n, we examine the initial condition for the most general case, where

$$u(x,0) = a_0 + \sum_{n=1}^{\infty}\left(a_n\cos\left(\frac{n\pi x}{a}\right) + b_n\sin\left(\frac{n\pi x}{a}\right)\right) = f(x). \tag{8}$$

Further, let us define the inner product of two functions f and g to be

$$\langle p, q\rangle = \int_{-a}^{a} p(x)q(x)\ dx.$$

Now, in taking inner-products of both sides of equation (8) with $\cos\left(\frac{m\pi x}{a}\right)$ and $\sin\left(\frac{m\pi x}{a}\right)$, we have

[5] Of course, the eigensolutions with $\lambda_{\cos}$ also form a solution of equation (1).

$$
\begin{aligned}
&\left\langle f, \cos\left(\frac{m\pi x}{a}\right)\right\rangle = a_0\left\langle 1, \cos\left(\frac{m\pi x}{a}\right)\right\rangle \\
&+ \sum_{n=1}^{\infty} a_n\left\langle \cos\left(\frac{n\pi x}{a}\right), \cos\left(\frac{m\pi x}{a}\right)\right\rangle + b_n\left\langle \sin\left(\frac{n\pi x}{a}\right), \cos\left(\frac{m\pi x}{a}\right)\right\rangle \\
&\left\langle f, \sin\left(\frac{m\pi x}{a}\right)\right\rangle = a_0\left\langle 1, \sin\left(\frac{m\pi x}{a}\right)\right\rangle \\
&+ \sum_{n=1}^{\infty} a_n\left\langle \cos\left(\frac{n\pi x}{a}\right), \sin\left(\frac{m\pi x}{a}\right)\right\rangle + b_n\left\langle \sin\left(\frac{n\pi x}{a}\right), \sin\left(\frac{m\pi x}{a}\right)\right\rangle.
\end{aligned}
$$

When $m \neq n$, both inner products return zero because the sine and cosine terms are orthogonal. When $m = n = 0$, the inner product $\left\langle f, \sin\left(\frac{m\pi x}{a}\right)\right\rangle = 0$, whereas

$$
\left\langle f, \cos\left(\frac{m\pi x}{a}\right)\right\rangle = \langle f, 1\rangle = a_0\langle 1, 1\rangle = a_0\int_{-a}^{a} 1\ dx = 2aa_0,
$$

which implies that

$$
a_0 = \frac{1}{2a}\langle f, 1\rangle = \frac{1}{2a}\int_{-a}^{a} f(x)\ dx,
$$

which is the average temperature. In applying the same analysis for the condition $n = m \neq 0$, for all, we have for the inner products

$$
\begin{aligned}
a_n &= \frac{1}{a}\left\langle f, \cos\left(\frac{n\pi x}{a}\right)\right\rangle = \frac{1}{a}\int_{-a}^{a} f(x)\cos\left(\frac{n\pi x}{a}\right)\ dx, \\
b_n &= \frac{1}{a}\left\langle f, \sin\left(\frac{n\pi x}{a}\right)\right\rangle = \frac{1}{a}\int_{-a}^{a} f(x)\sin\left(\frac{n\pi x}{a}\right)\ dx.
\end{aligned}
$$

With the desired coefficients, we can now construct the general solution:

$$
\begin{aligned}
&u(x,t) = \frac{1}{2a}\int_{-a}^{a} f(y)\ dy \\
&+\sum_{n=1}^{\infty} e^{-\frac{n^2\pi^2}{a^2}t}\left(\left[\frac{1}{a}\int_{-a}^{a} f(y)\cos\left(\frac{n\pi y}{a}\right)\ dy\right]\cos\left(\frac{n\pi x}{a}\right)+\left[\frac{1}{a}\int_{-a}^{a} f(y)\sin\left(\frac{n\pi y}{a}\right)\ dy\right]\sin\left(\frac{n\pi x}{a}\right)\right).
\end{aligned} \tag{9}
$$

Now we can comfortably apply the initial condition. It would be useful to rewrite the initial condition as such

$$u(x,0) = \begin{cases} 1, & \text{if } -1 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases},$$

where we replace the absolute value inequality with a compound inequality, for clarity. In applying the initial condition for the coefficients, we have

$$\begin{aligned} a_0 &= \frac{1}{2a}\int_{-a}^{a} f(y)\ dy = \frac{1}{2a}\int_{-1}^{1} 1\ dy = \frac{1}{a}, \\ a_n &= \frac{1}{a}\int_{-a}^{a} f(y)\cos\left(\frac{n\pi y}{a}\right)\ dy = \frac{1}{a}\int_{-1}^{1}\cos\left(\frac{n\pi y}{a}\right)\ dy = \frac{2\sin\left(\frac{\pi n}{a}\right)}{\pi n}, \\ b_n &= \frac{1}{a}\int_{-a}^{a} f(y)\sin\left(\frac{n\pi y}{a}\right)\ dy = \frac{1}{a}\int_{-1}^{1}\sin\left(\frac{n\pi y}{a}\right)\ dy = 0. \end{aligned}$$

In rewriting equation (9), the symbolic solution becomes

$$\boxed{u(x,t) = \frac{1}{a} + \sum_{n=1}^{\infty} e^{-\frac{n^2\pi^2}{a^2}t}\left[\frac{2\sin\left(\frac{\pi n}{a}\right)}{\pi n}\right]\cos\left(\frac{n\pi x}{a}\right)}. \tag{10}$$

(b): To examine the analytic solution numerically, the following MATLAB code was produced, as well as the plots. *Note:* The `init_temp` function was produced by this code:

```
function y = init_temp(x)

y = 1.*(abs(x) ≤ 1) + 0.*(x < −1) + 0.*(x > 1);

end
```

```
%% Exercise 1: Solving the Heat Equation w/ General Periodic Boundary ...
    Conditions

a = 2;
L = 2*a;
K = 1;

% Semidiscretization

N = 20,40,80;
h = L/(N+1);
x = linspace(−a,a,N);

% Forming the matrix with Periodic Boundary Conditions

A = toeplitz([−2 1 zeros(1,N−2)]);
```

```
A(1,N) = 1;
A(N,1) = 1;

% Eigenvalue decomposition for A

[V,D] = eig(A);
lambda = K*diag(D)/(h^2);

% Solution construction

f = @init_temp;
u0 = f(x)';
t = 0.1;
C = V\u0;
u = V*(exp(lambda*t).*C);
tt = linspace(0,100,N);

% Analytic Solution

en = zeros(N,1);
proj = zeros(length(x),N);

for n = 1:N
    g = @(x) cos(pi.*x.*n/a);
    gg = @(x) f(x).*g(x);
    proj(:,n) = (1/a)*quadgk(gg,-1,1)*g(x);
    en(n) = exp(-((n^2 * pi^2)/(a^2))*t);
end

uu = (1/(2*a))*quadgk(f,-1,1) + proj*en;

% Plotting the Analytic and Numerical Solutions

figure
hold on
plot(x,u0,'r-'); % Initial Condition
p = plot(x,u,'b-'); % Numerical Solution
plot(x,uu,'ko'); % Analytic Solution
title(sprintf('The Solution at t = %1.4f and N = %1.4f',[t N]));
legend('Initial Condition','Numerical Solution','Analytic Solution');
ylabel('u(x,t)');
xlabel('x');
set(gca,'FontSize',18);
```

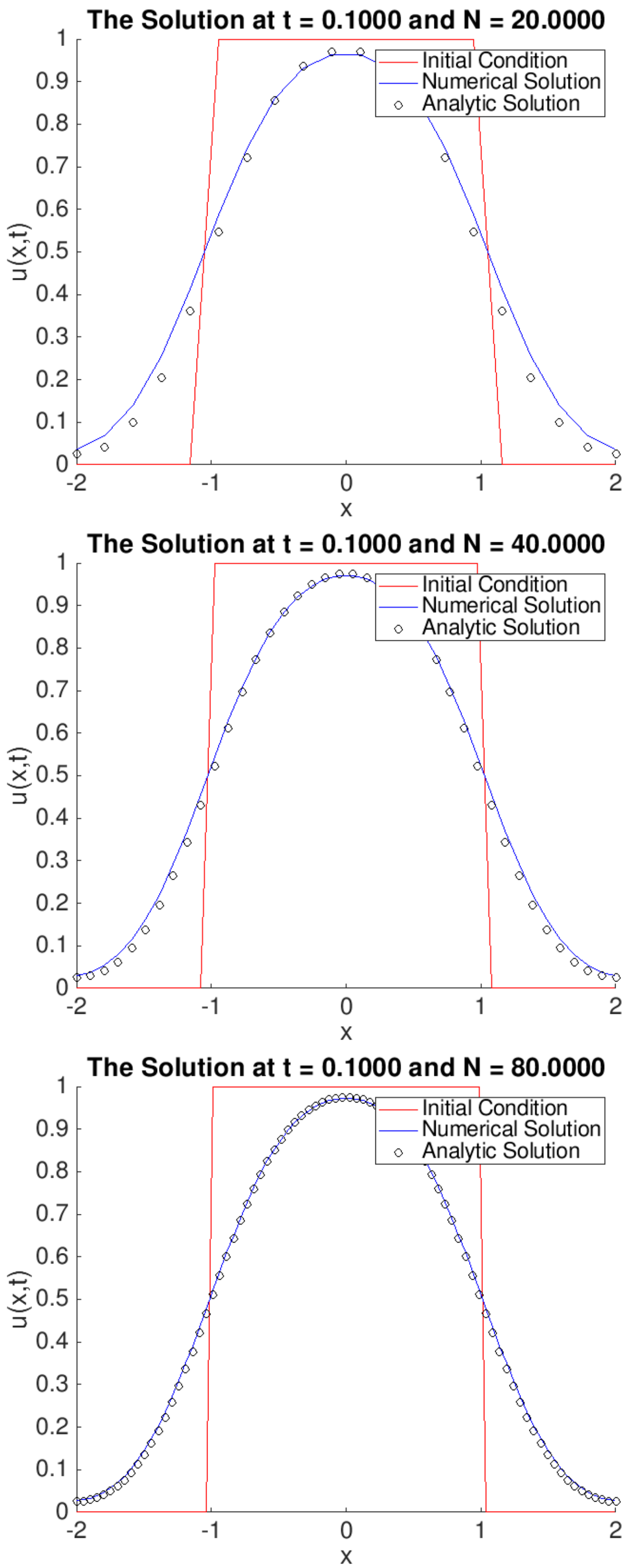


Figure 21: The solution to the Heat Equation with Dirichlet conditions and general periodic boundary conditions.

As we can see, the analytic and numerical solutions agree well, especially as N increases, which is expected.

(c): To observe what happens to the solution as $a \to \infty$, it would be best to look at equation (9). When we take the limit, firstly, the initial condition goes to zero. Then, we can interchange integration and summation. In doing so, and treating the sum as a Riemann sum, we can write

$$\boxed{u(x,t) = \frac{1}{\pi}\int_0^\infty \left(\cos(\xi x)e^{-\xi^2 t}\left[\int_{-\infty}^{\infty} f(y)\cos(\xi y)\ dy\right] + \ \sin(\xi x)e^{-\xi^2 t}\left[\int_{-\infty}^{\infty} f(y)\sin(\xi y)\ dy\right]\right)\ d\xi},$$

where we set $\xi = \pi s$, and s is represented by n/a, the subdivisions in the Riemann sum.

EXERCISE 2

Let $V = \{f \in C^2([0,1]) \mid f(0) + f'(0) = 0, \quad f'(1) = 0\}$. That is, V is a vector space of twice-differentiable function on [0,1] with continuous second derivatives and boundary conditions: $f(0)+f'(0) = 0, \quad f'(1) = 0$. Denote by L the one-dimensional Laplacian: $L(f) = f''$. Compute the first nine eigenfunctions of L (the ones with smallest eigenvalues.) Plot the resulting eigenfunctions on the same figure (using **`subplot`***) and put the corresponding eigenvalues in the titles. Then try to answer the following questions: What is the rate of growth of eigenvalues? What happens to the eigenfunctions as the magnitude of the eigenvalues increases?* ***Note:*** *You may need to approximate the eigenvalues numerically.*

Solution: The eigenfunctions of the Laplacian are given by

$$\frac{d^2 f}{dx^2} = \lambda f, \quad f = a\cos\left(\sqrt{-\lambda}x\right) + b\sin\left(\sqrt{-\lambda}x\right).$$

In satisfying the conditions, we have this matrix-vector system:

$$\begin{bmatrix} -\sqrt{-\lambda}\sin\left(\sqrt{-\lambda}\right) & \sqrt{-\lambda}\cos\left(\sqrt{-\lambda}\right) \\ 1 & \sqrt{-\lambda} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

whose solutions depend on the following condition

$$\lambda\sin\left(\sqrt{-\lambda}\right) = \sqrt{-\lambda}\cos\left(\sqrt{-\lambda}\right).$$

In plotting both sides against λ, and determining where the graphs intersect, we obtain the eigenvalues of the Laplacian. This is shown in Figure 2 below.

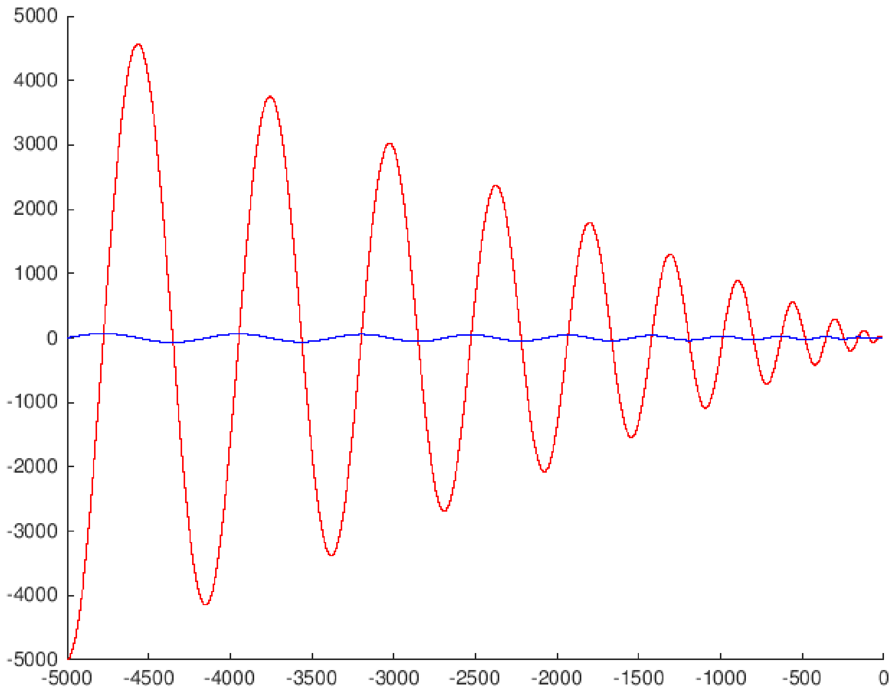


Figure 22: Solving for the eigenvalues of the Laplacian.

As we can see, the intersection points give the eigenvalues, where the eigenvalues are on the x-axis. In order to determine their rate of growth, I used the following code to extract the unique eigenvalues. Then, I used a `loglog` plot in order to determine the growth rate.

```
%% Exercise 2

l = linspace(-5000,0,100000);
y1 = @(l) l.*sin(sqrt(-l));
y2 = @(l) sqrt(-l).*cos(sqrt(-l));
dif = @(l) y2(l)-y1(l);  % Difference between both graphs

% Determining the eigenvalues

x = -1000000:20:0;
n = numel(x); % Produces the number of elements in x
z = zeros(1,n); % Creating an array

for j = 1:numel(x)
    z(j) = fzero(dif,x(j)); % Determines the eigenvalue from the dif
end

z = uniquetol(z,1e-6); % Provides the unique eigenvalues
```

```
z = sort(abs(z)); % Sorts the eigenvlues

% Plotting the first 9 eigenfunctions

x1 = linspace(0,10,1000);

% Plotting the first Nine Eigenfunctions

for k = 1:9
    subplot(3,3,k);
    hold on
    plot(x1,(cos(x1*sqrt(z(k))) + sin(x1*sqrt(z(k))))); % The ...
        eigenfunctions
    title(sprintf('lambda = %1.4f',z(k)));
end

% Determining the rate of growth of eigenvectors

x2 = linspace(0,320);
p = polyfit([1:length(z)],z,2);
pp = polyval(p,x2);

% Confirming that the rate of growth is quadratic

dz = diff(z);
p1 = polyfit([1:length(z)-1],dz,1);
pp1 = polyval(p1,x2(1:end-1));

yy = log(z);
pfit = polyfit(log([2:length(z)]),yy(2:end),1);
pval = polyval(pfit,log([2:length(z)]));

% Loglog plot

figure
hold on
plot(log([2:length(z)]),yy(2:end),'b-');
plot(log([2:length(z)]),pval,'ro');
title(sprintf('p = [%1.4f %1.4f]',[pfit]));

figure
hold on
plot(z,'b-');
plot(x2,pp,'ro');

figure
hold on
plot(dz,'b-');
plot(x2(1:end-1),pp1,'ro');
```

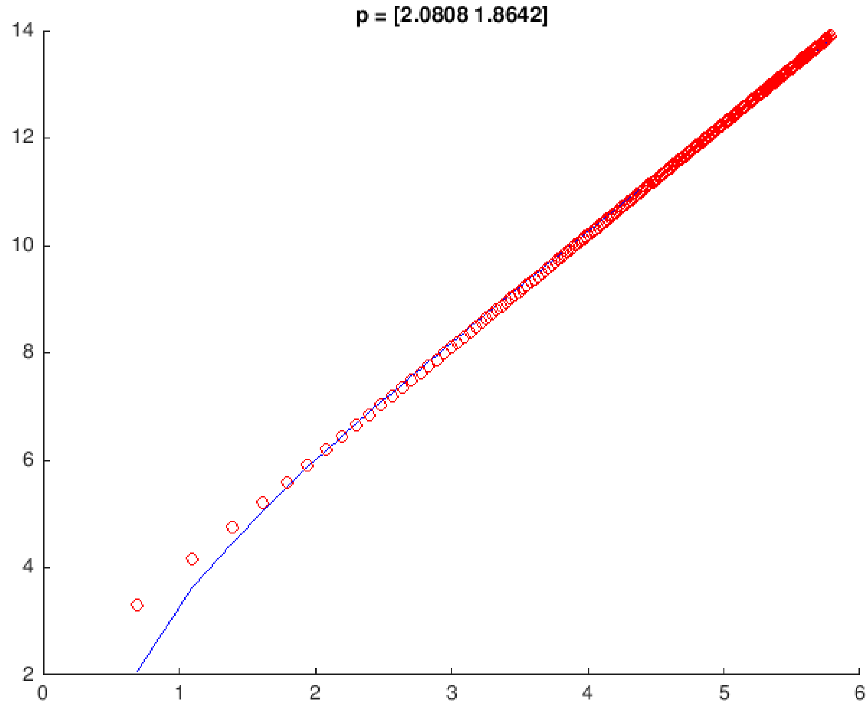


Figure 23: The order of growth for the eigenvalues.

It is clear that the magnitude of the eigenvalues increases quadratically. To confirm this, I fit a second order polynomial to the data of the eigenvalues, and proceeded to plot its first derivative to determine the linearity of the rate of change.

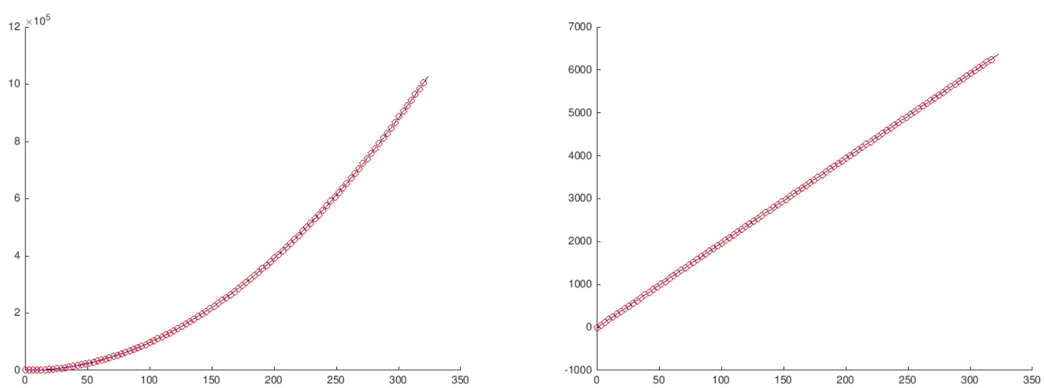


Figure 24: The plot of the eigenvalues on the left and the rate of change of the eigenvalues on the right, which is indeed linear as expected.

Now that we have the eigenvalues, we can comfortably plot the first nine eigenfunctions of the Laplacian.

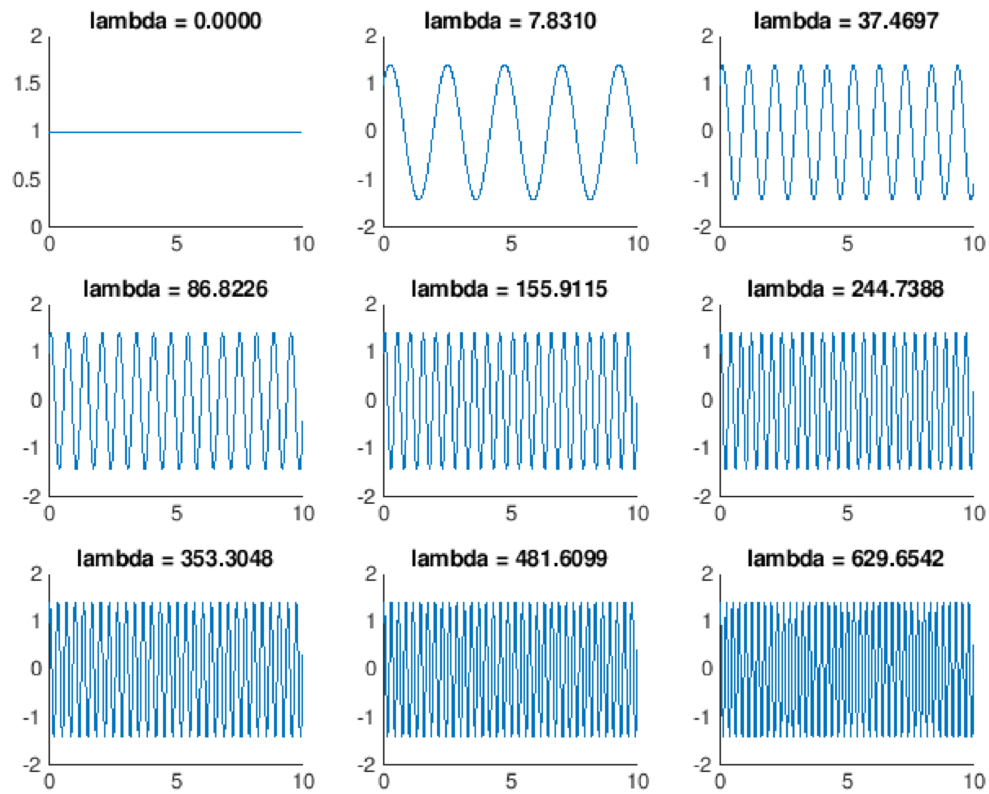


Figure 25: The first nine eigenfunctions of the Laplacian.

As we can see, as the magnitude of the eigenvalue increases, the frequency of the eigenfunctions increase as well, as expected.

EXERCISE 3

Solution: Well, I computed the quadratic numerically and found that it is

$$v = \frac{1}{2}x^2 + x + \frac{1}{2},$$

I hope. Here is the code that determined the quadratic.

```
%% Exercise 3

f = @(c) ((2.*(21.*c(1).^2 + 70.*c(1).*c(3) + 35.*c(2).^2 + ...
    105.*c(3).^2 + 15 - 42.*c(2)))/(105)).^2;
```

```

guess = [10000 20000 30000];
min = fminsearch(f,guess);

for g = 1:10000
    guess(g,:) = [g g*2 g*3];
    min(g,:) = fminsearch(f,guess(g,:));
end

min = mean(min>eps('double'));

x = linspace(-1,1);

p = @(x) x.^3;
q = @(c,x) c(1).*x.^2 + c(2).*x + c(3);

q = @(x) q(min,x);

h = @(x) p(x)-q(x);

norm = sqrt(quadgk(h,-1,1));
```

I'm not sure if there is an analytic way to accomplish this, but if there is, I don't know it. I would have asked you more questions about this exercise, but you were busy with a meeting. I'm sure I have to project $f = x^3$ onto the space V and then determine the norm of the difference between f and its projection onto the space, but I will do that later.

Correction and Commentary: I should have gone with my initial calculation. The actual answer that I obtained numerically was

$$v = \frac{3}{5}x.$$

This was obtained by solving the Grammian system, where the matrix entries were all possible inner products of the basis vectors ($V = \text{span}\{1, x, x^2\}$), the uncknown was the coefficients of the general expression for $v = c_1 + c_2 x + c_3 x^2$, and the vector of unknowns is the inner product of f with each element of the basis:

$$\begin{bmatrix} \langle \alpha_1, \alpha_1 \rangle & \langle \alpha_1, \alpha_2 \rangle & \langle \alpha_1, \alpha_3 \rangle \\ \langle \alpha_2, \alpha_1 \rangle & \langle \alpha_2, \alpha_2 \rangle & \langle \alpha_2, \alpha_3 \rangle \\ \langle \alpha_3, \alpha_1 \rangle & \langle \alpha_3, \alpha_2 \rangle & \langle \alpha_3, \alpha_3 \rangle \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \langle a_1, f \rangle \\ \langle a_2, f \rangle \\ \langle a_3, f \rangle \end{bmatrix},$$

where the α_i are the elements of V. Here is the new code.

```
%% Exercise 3

% Better Code

% basis of V
```

```

v1 = @(x) 1;
v2 = @(x) x;
v3 = @(x) x.^2;

f = @(x) x.^3;

% Gram Matrix

g = zeros(3,3);

g(1,1) = integral(@(x) v1(x).*v1(x),-1,1,'arrayvalued',true);
g(1,2) = integral(@(x) v1(x).*v2(x),-1,1);
g(1,3) = integral(@(x) v1(x).*v3(x),-1,1);
g(2,1) = quadgk(@(x) v2(x).*v1(x),-1,1);
g(2,2) = quadgk(@(x) v2(x).*v2(x),-1,1);
g(2,3) = quadgk(@(x) v2(x).*v3(x),-1,1);
g(3,1) = quadgk(@(x) v3(x).*v1(x),-1,1);
g(3,2) = quadgk(@(x) v3(x).*v2(x),-1,1);
g(3,3) = quadgk(@(x) v3(x).*v3(x),-1,1);

% Solution Vector

b = zeros(3,1);

b(1,1) = quadgk(@(x) v1(x).*f(x),-1,1);
b(2,1) = quadgk(@(x) v2(x).*f(x),-1,1);
b(3,1) = quadgk(@(x) v3(x).*f(x),-1,1);

% Solution

c = g\b;

>> c

c =

    0.0000
    0.6000
   -0.0000
```

HOMEWORK 5

EXERCISE 1

Prove the translation property of the Fourier transform: $\widehat{f(x+a)} = e^{i\xi a}\widehat{f}$.

Solution:

$$\begin{aligned}
\widehat{f(x+a)} &= \int_{-\infty}^{\infty} e^{-i\xi x}\ f(x+a)\ dx, \quad u = x+a,\ du = dx \\
&= \int_{-\infty}^{\infty} e^{-i\xi(u-a)}\ f(u)\ du \\
&= \int_{-\infty}^{\infty} e^{-i\xi u + i\xi a}\ f(u)\ du \\
&= \int_{-\infty}^{\infty} e^{-i\xi u} e^{i\xi a}\ f(u)\ du \\
&= e^{i\xi a}\int_{-\infty}^{\infty} e^{-i\xi u}\ f(u)\ du \\
&= \boxed{e^{i\xi a}\widehat{f}(\xi)}.
\end{aligned}$$

Just for fun I also proved the following two properties of the Fourier Transform:

$$\begin{aligned}
\widehat{f(cx)} &= \int_{-\infty}^{\infty} e^{-i\xi x}\ f(cx)\ dx, \quad u = cx,\ du = c\ dx \\
&= \frac{1}{c}\int_{-\infty}^{\infty} e^{-i\frac{\xi}{c}u}\ f(u)\ du \\
&= \boxed{\frac{1}{c}\widehat{f}\left(\frac{\xi}{c}\right)}.
\end{aligned}$$

$$
\begin{aligned}
\widehat{ixf(x)} &= \int_{-\infty}^{\infty} e^{-i\xi x}\ ixf(x)\ dx \\
&= \int_{-\infty}^{\infty} \frac{d}{d\xi}\Big[-e^{-i\xi x}\ f(x)\Big]\ dx
\end{aligned}
$$

Here, we can interchange integration and differentiation because both variables are independent of one another.

$$
\begin{aligned}
\int_{-\infty}^{\infty} \frac{d}{d\xi}\Big[-e^{-i\xi x}\ f(x)\Big]\ dx &= \frac{d}{d\xi}\left[-\int_{-\infty}^{\infty} e^{-i\xi x}\ f(x)\ dx\right] \\
&= \frac{d}{d\xi}\big[-\widehat{f}(\xi)\big] \\
&= \boxed{-\widehat{f}\,'(\xi)}.
\end{aligned}
$$

Exercise 2

Find the Fourier transform of the function:

$$
f(x) = \begin{cases} -1, & -1 \le x \le 0, \\ +1, & 0 < x \le 1. \end{cases}
$$

Solution:

$$
\begin{aligned}
\widehat{f(x)} &= \int_{-\infty}^{\infty} e^{-i\xi x} f(x)\ dx \\
&= \int_{-1}^{1} e^{-i\xi x} \begin{cases} -1, & -1 \le x \le 0, \\ +1, & 0 < x \le 1. \end{cases} \quad dx \\
&= -\int_{-1}^{0} e^{-i\xi x}\ dx + \int_{0}^{1} e^{-i\xi x}\ dx \\
&= \left[\frac{e^{-i\xi x}}{i\xi}\right]_{-1}^{0} + \left[\frac{e^{-i\xi x}}{-i\xi}\right]_{0}^{1} \\
&= \frac{2-\left(e^{i\xi}+e^{-i\xi}\right)}{i\xi} = \boxed{\frac{2-2\cos(\xi)}{i\xi}}.
\end{aligned}
$$

EXERCISE 3

Solve the Wave Equation using Fourier transform:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}, \quad -\infty < x < \infty, \quad t > 0,$$
$$u(x,0) = f(x),$$
$$\frac{\partial u}{\partial t}(x,0) = g(x).$$

Solution:

$$\begin{aligned}
\widehat{\frac{\partial^2 u}{\partial t^2}} = \widehat{\frac{\partial^2 u}{\partial x^2}} &\implies \frac{d^2 \widehat{u}}{dt^2} = -\xi^2 \widehat{u}, \quad \widehat{u(x,0)} = \widehat{f}, \quad \widehat{\frac{\partial u}{\partial t}(x,0)} = \widehat{g} \\
&\implies \widehat{u} = c_1 \cos(\xi t) + c_2 \sin(\xi t), \quad \widehat{u}' = -c_1 \xi \sin(\xi t) + c_2 \xi \cos(\xi t) \\
&\implies c_1 = \widehat{f}, \quad c_2 = \frac{\widehat{g}}{\xi} \\
&\implies \widehat{u} = \cos(\xi t)\widehat{f} + \sin(\xi t)\frac{\widehat{g}}{\xi} \\
&\implies \boxed{u = \left(\cos(\xi t)\widehat{f} + \frac{\sin(\xi t)\widehat{g}}{\xi}\right)^{\vee}}.
\end{aligned}$$

HOMEWORK 6

EXERCISE 1

(1) Use the power series method to re-derive the solutions of the following familiar ODE:

(a) $\frac{dy}{dt} = ky$

(b) $\frac{d^2y}{dt^2} = -\omega^2 y$

Solution: To begin, let us rewrite the first ODE as

$$\frac{dy}{dt} - ky = 0,$$

and define

$$y(t) = \sum_{n=0}^{\infty} a_n t^n,$$

as our solution. In substituting y into the equation, we obtain

$$\sum_{n=1}^{\infty} n a_n t^{n-1} - k \sum_{n=0}^{\infty} a_n t^n = 0.$$

We can rewriting the sums so as to shift the index by making the substitution $n = k + 1$ for the first sum and $k = 0$ for the second. In doing so, we can combine the sums and obtain the recursion relation

$$a_{m+1} = \frac{k a_m}{(m+1)},$$

with the first few terms being

$$\begin{gathered} a_1 = k a_0 \\ a_2 = k\frac{a_1}{2} = k^2 \frac{a_0}{1 \cdot 2} \\ a_3 = k\frac{a_2}{3} = k^3 \frac{a_0}{1 \cdot 2 \cdot 3}. \end{gathered}$$

As such, the general expression is

$$a_n = \frac{a_0 k^n}{n!}$$

In substituting this expression into y, we obtain

$$\boxed{y(t) = a_0 \sum_{n=0}^{\infty} \frac{(kt)^n}{n!} = a_0 e^{kt}},$$

which is an exponential as expected. For the second ODE, we apply the same logic as before. In doing so, we obtain two recursion relations:

Odd integers:	Even integers:
$a_{2n+1} = \dfrac{a_1(-1)^n \omega^{2n+1}}{(2n+1)!}$	$a_n = \dfrac{a_0(-1)^n \omega^{2n}}{(2n)!}$

In adding up both terms, we have

$$\begin{aligned} y(t) &= a_0 \sum_{n=0}^{\infty} \frac{(-1)^n(\omega t)^{2n}}{(2n)!} + \frac{a_1}{\omega} \sum_{n=0}^{\infty} \frac{(-1)^n(\omega t)^{2n+1}}{(2n+1)!} \\ &= \boxed{a_0 \cos(\omega t) + \frac{a_1}{\omega}\sin(\omega t)}, \end{aligned}$$

where the extra factor of ω is a consequence of the formula. The solution satisfies the equation because it is the exact answer that is expected.

EXERCISE 2

(2) Find all values of z for which J_0 is convergent. Explain your reasoning by citing appropriate tests of convergence from Calculus.

Solution: Let us apply the ratio test, which is as follows:

> **Ratio Test.** Suppose that we have the series $\sum_n a_n$. Define the following:
>
> $$\lim_{n\to\infty} \left| \frac{a_{n+1}}{a_n} \right| = L.$$
>
> Then,
>
> 1. If $L < 0$, the series is absolutely convergent, and hence convergent.
> 2. If $L > 0$, the series is divergent.
> 3. If $L = 1$, the series may be divergent, conditionally convergent, or absolutely convergent.

The Bessel functions of the first kind $J_0(z)$ are absolutely convergent for all values of z.

Proof. In applying the ratio test, we have

$$\begin{aligned}
\lim_{n\to\infty}\left|\frac{(-1)^n(-1)(z)^{2n}z}{2^{2n}2^2(n+1)n!(n+1)n!}\cdot\frac{2^{2n}n!n!}{(-1)^n(z)^{2n}}\right| &= \lim_{n\to\infty}\left|\frac{(-1)z^2}{4(n+1)}\right| \\
&= \frac{1}{4}\lim_{\to\infty}\left|\frac{z^2}{n+1}\right| \\
&= 0,
\end{aligned}$$

which means that Bessel functions of the first kind are absolutely convergent for all values of z. □

EXERCISE 3

(3) The Derivative of J_0 is another Bessel function: $\frac{d}{dz}J_0(z) = -J_1(z)$. Find the series representing $J_1(z)$ and produce its plot.

Solution: The function $J_n(z)$ is given by the following formula, which has been derived in exercise (5):

$$J_n(z) = \sum_{n=0}^{\infty}\frac{(-1)^n z^{2n+m}}{2^{2n+m}\ n!(n+m)!}. \tag{1}$$

In substituting $m = 0$, we obtain $J_0(z)$. Taking the derivative of $J_0(z)$ produces the following:

$$\begin{aligned}
\frac{d}{dz}[J_0(z)] &= \frac{d}{dz}\left[\sum_{n=0}^{\infty}\frac{(-1)^n z^{2n}}{2^{2n}(n!)^2}\right] \\
&= \sum_{n=1}^{\infty}\frac{(-1)^n(2n)z^{2n-1}}{2^{2n}(n!)^2} \\
&= \sum_{n=1}^{\infty}\frac{(-1)^n z^{2n-1}}{2^{2n-1}(n-1)!n!}, \qquad \text{let } n = k+1 \\
&= \sum_{k=0}^{\infty}\frac{(-1)^{k+1}z^{2k+1}}{2^{2k+1}k!(k+1)!} \\
&= -\sum_{k=0}^{\infty}\frac{(-1)^k z^{2k+1}}{2^{2k+1}k!(k+1)!} \\
&= -J_1(z).
\end{aligned}$$

This is easily verified by substituting $m = 1$ into equation (1). Here is the code and the plot of $J_1(z)$.

```
%% Exercise 3

% My Bessel 1

n = 100;
J = zeros(n,1);
z = linspace(0,10,n);

for k = 1:n
    j = 0;
    for l = 0:n
        j = j + ...
            (((-1)^l)*z(k)^(2*l+1))/(2^(2*l+1)*factorial(l)*factorial(l+1));
    end
    J(k,:) = j;
end

% Actual Besssel

y = besselj(1,z);

% Comparison

figure
hold on
plot(z,y,'ro');
plot(z,J,'b-');
xlabel('x','Interpreter',"latex");
ylabel('y','Interpreter',"latex");
title('Bessel Function of the First Kind, Order One');
legend('Actual Bessel Function','My Bessel Function');
set(gca,'fontsize',20);
```

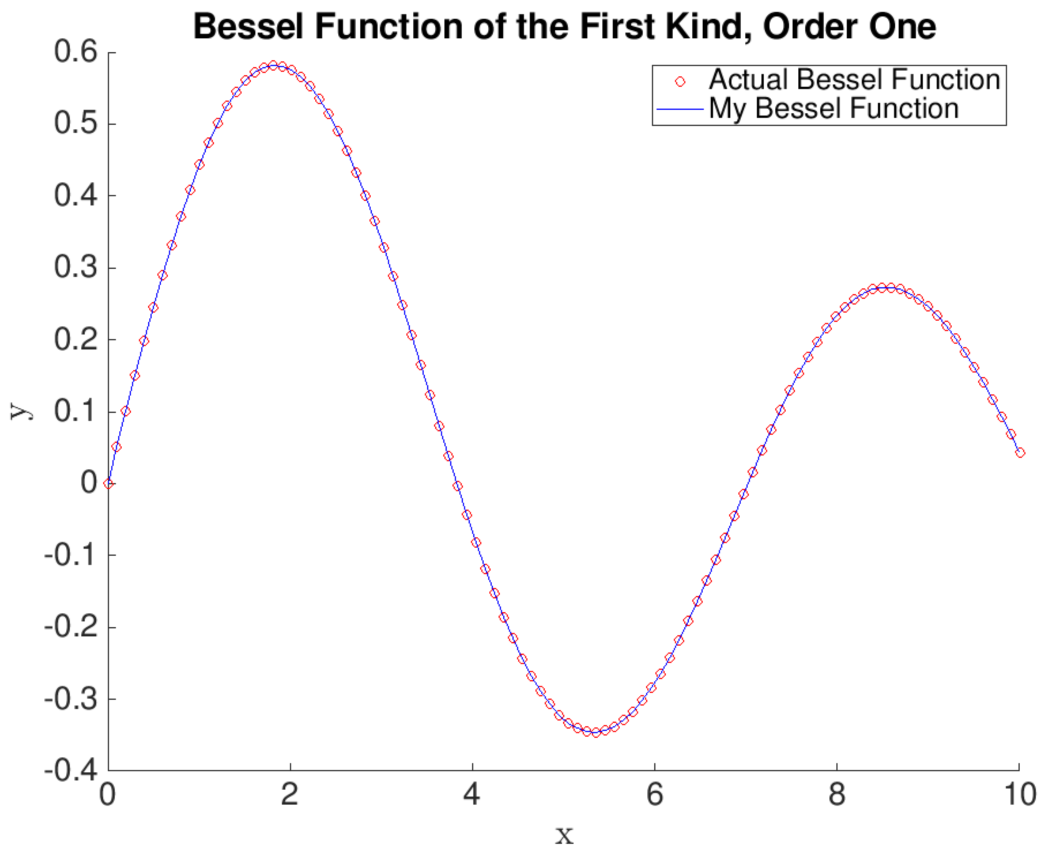


Figure 26: The plot of $J_1(z)$

EXERCISE 4

(4) Use software to confirm the orthogonality of $J_0(z_n r)$ *for* $n = 0, 1, 2, 3$. *Also compute*

$$\int_0^1 J_0^2(z_n r)r\ dr$$

for $n = 0, 1, 2, 3$.

Solution: I don't have very much to say about this one. Here is my code and the resulting outputs.

```
%% Exercise 4

% Orthogonality of Bessel Functions

n = 100;
z = linspace(0,10,100);
```

```
J0 = @(z) besselj(0,z);

% Generating a list of zeros

for k = 1:20
    zn(k) = fzero(J0,k);
end

% Obtaining unique zeros

zn = uniquetol(zn,eps(1.0));
zn = zn(1:4);

>> zn

zn =

    2.4048    5.5201    8.6537   11.7915

prod1 = zeros(4,4);

% Checking the orthogonality of the bessel function

for ii = 1:length(zn)
    for jj = 1:length(zn)
        prod1(ii,jj) = ...
            quadgk(@(r)besselj(0,zn(ii)*r).*besselj(0,zn(jj)*r).*r,0,1);
    end
end

>> prod1

prod1 =

    0.1348    0.0000    0.0000    0.0000
    0.0000    0.0579   -0.0000    0.0000
    0.0000   -0.0000    0.0368    0.0000
    0.0000    0.0000    0.0000    0.0270

prod2 = zeros(length(zn),1);

% Integrating the first four terms: n = 0,1,2,3

for kk = 1:length(zn)
    prod2(kk,:) = quadgk(@(r)besselj(0,zn(kk)*r).^(2).*r,0,1);
end

>> prod2

prod2 =

```

```
      0.1348
      0.0579
      0.0368
      0.0270
```

The `prod1` matrix confirm the orthogonality because when $i \neq j$, the inner product returns zero and when $i = j$, the inner product returns a value. Of course, if the Bessel functions where orthonormal, we'd expect an identity matrix, but the absence of such suggests that they are indeed just orthogonal.

Exercise 5

If the initial temperature in the disk is not radial, one has to solve the full problem for the Heat Equation in polar coordinates:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial r^2} + \dfrac{1}{r}\dfrac{\partial u}{\partial r} + \dfrac{1}{r^2}\dfrac{\partial^2 u}{\partial \theta^2}, \quad 0 < r < 1, \quad -\pi < 0 < \pi, \quad t > 0, \\ u(1, \theta, t) = 0, \\ u(r, \theta, 0) = u_0(r, \theta). \end{cases} \tag{2}$$

Follow the methodology of this handout to derive the series solution for the case when the initial temperature is given by

$$u(r, \theta, 0) = f_0(r) + f_1(r)\cos(\theta).$$

If you are adventurous, you can try the general case.

Solution: To begin, let us utilize the method of separation of variables to obtain the solution. Further, let us define our domain as the following: $D = \{r, \theta \in \mathbb{R} \mid 0 < r < 1,\ -\pi < \theta < \pi\}$. Thus we assume that our solution has the form $u(r, \theta, t) = y(t)g(r, \theta)$. Substituting this guess into equation (2), we have the following system

$$\frac{dy}{dt} = -\lambda y, \tag{3}$$

$$\frac{\partial^2 g}{\partial r^2} + \frac{1}{r}\frac{\partial g}{\partial r} + \frac{1}{r^2}\frac{\partial^2 g}{\partial \theta^2} + \lambda g = 0, \quad u(r, \theta, 0) = u_0(r, \theta), \tag{4}$$

where equation (4) can be further separated as follows, using $g(r, \theta) = h(r)\phi(\theta)$, producing

$$\frac{d^2\phi}{d\theta^2} = -\mu\phi, \tag{5}$$

$$r^2\frac{d^2 h}{dr^2} + r\frac{dh}{dr} + (\lambda r^2 - \mu)h = 0, \quad h(1) = 0. \tag{6}$$

The solutions to equations (3) and (5) are trivial, given by

$$y(t) = e^{-l^2 t},$$
$$\phi(\theta) = a\cos(m\theta) + b\sin(m\theta),$$

where we denote $\lambda = l^2$ and $\mu = m^2$. The solution to equation (6) requires care, and as such, we use the Frobenius Method to determine the solution. Equation (6) is known as Bessel's Differential Equation. In making the substitution $z = lr$, we obtain a more familiar form of Bessel's Equation,

$$z^2 \frac{d^2h}{dz^2} + z\frac{dh}{dz} + (z^2 - m^2)h = 0. \tag{7}$$

We begin the derivation of the solution to Bessel's Differential Equation by assuming $h = \sum_{n=0}^{\infty} a_n z^{n+c}$ as opposed to the standard power series guess, which produces a trivial solution. Thus, in substituting our guess into equation (7), we obtain

$$\sum_{n=0}^{\infty}(n+c)(n+c-1)a_n z^{n+c} + \sum_{n=0}^{\infty}(n+c)a_n z^{n+c} + \sum_{n=0}^{\infty} a_n z^{n+c+2} - m^2 \sum_{n=0}^{\infty} a_n z^{n+c} = 0.$$

Next, we shift the indices such that each sum has the same power, and after simplification we have to solve the following

$$[c^2 - m^2]a_0 = 0$$
$$[(c+1)^2 - m^2]a_1 = 0$$
$$(n+c+2)(n+c+1)a_{n+2} + (n+c+2)a_{n+2} + a_k - m^2 a_{n+2} = 0,$$

to determine the appropriate coefficients. We first solve the first equation, which suggests that $c = \pm m$, implying that $a_1 = 0$. Further, because we have seen Bessel functions before, this means that the odd coefficients all equal zero because they depend on a_1. Let us then take the solutions where $c = m$. In doing so gives the following recursion relation

$$a_{n+2} = \frac{-a_n}{(n+m+2)^2 - m^2}.$$

The first few even terms are

$$a_2 = \frac{-a_0}{4(m+1)}$$
$$a_4 = \frac{-a_2}{8(m+2)} = \frac{a_0}{4 \cdot 8(m+2)(m+1)}$$
$$a_6 = \frac{-a_4}{12(m+3)} = \frac{-a_0}{4 \cdot 8 \cdot 12(m+3)(m+2)(m+1)}$$

Thus, overall we find that a_n is

$$a_{2n} = \frac{(-1)^n a_0}{2^{2n}\ n!(m+n)\cdots(m+2)(m+1)}.$$

Further, we can strategically define

$$a_0 = \frac{1}{2^m m!},$$

which would give

$$a_{2n} = \frac{(-1)^n}{2^{2n+m}\ n!(n+m)!}$$

In substituting this into our initial guess, be have

$$h(r) = \sum_{n=0}^{\infty} \frac{(-1)^n (z)^{2n+m}}{2^{2n+m}\ n!(n+m)!} = J_m(lr),$$

which is the formula for Bessel's function of the first kind of order m. Before forming the full solution, we apply boundary conditions. Because $h(1) = 0$, this suggests that $J_m(l) = 0$, where $l = z_n$, for $n = 0, 1, 2, 3, \ldots$. This means that for each m, the z_n's are the zeros to the Bessel function of order m. Now we can form the full solution to equation (2), given by

$$u(r,\theta,t) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} J_m(z_n r)\big[a_{mn}\cos(m\theta) + b_{mn}\sin(m\theta)\big]e^{-z_n^2 t},$$

In applying the initial condition, we have

$$u(r,\theta,0) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} J_m(z_n r)e^{-z_n^2 t}\big[a_{mn}\cos(m\theta) + b_{mn}\sin(m\theta)\big] = u_0(r,\theta).$$

The coefficients a_{mn} and b_{mn} are obtained by projecting twice, once for the Bessel function and once for each $\sin(m\theta)$ and $\cos(m\theta)$,

$$a_{mn} = \frac{\int\limits_D u_0(r,\theta)J_m(z_n r)\cos(m\theta)\ rdrd\theta}{\int\limits_D J_m^2(z_n r)\cos^2(m\theta)\ rdrd\theta},$$

$$b_{mn} = \frac{\int\limits_D u_0(r,\theta)J_m(z_n r)\cos(m\theta)\ rdrd\theta}{\int\limits_D J_m^2(z_n r)\cos^2(m\theta)\ rdrd\theta}.$$

Thus, the solution, in all its glory, is given by

$$u(r,\theta,t) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{\int\limits_D u_0(r,\theta) J_m(z_n r)\cos(m\theta)\ r dr d\theta}{\int\limits_D J_m^2(z_n r)\cos^2(m\theta)\ r dr d\theta} J_m(z_n r)\cos(m\theta) e^{-z_n^2 t}$$
$$+ \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{\int\limits_D u_0(r,\theta) J_m(z_n r)\cos(m\theta)\ r dr d\theta}{\int\limits_D J_m^2(z_n r)\cos^2(m\theta)\ r dr d\theta} J_m(z_n r)\sin(m\theta) e^{-z_n^2 t},$$

where $u_0(r,\theta)$ is replaced by the initial condition $f_0(r) + f_1\cos(\theta)$.

EXAM 2

EXERCISE 1

1. In this problem we consider the wave equation with Dirichlet boundary conditions:

$$\begin{aligned} &\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad (x,y) \in D, \quad t > 0, \\ &u\big|_{\partial\Omega} = 0, \\ &u\big|_{t=0} = 0, \quad \frac{\partial u}{\partial t}\bigg|_{t=0} = v(x,y). \end{aligned} \tag{1}$$

The domain D is a square centered at the origin with sides of length $2a$ parallel to the coordinate axes: symbolically, $D = \{(x,y) \in \mathbb{R}^2 \mid -a < x, y < a\}$. The initial velocity is constant on S—a smaller square within D, similarly oriented, with sides of length $2b$—and is zero outside:

$$v(x,y) = \begin{cases} V, & \text{if } (x,y) \in S \\ 0, & \text{otherwise.} \end{cases}$$

Your task is to do the following:

(a) Derive a symbolic formula for the solution of (1). Be sure to accompany your derivation with clear verbal explanations.

(b) Use the formula from Part 1 to produce surface plots of the solution at time $t = 1$ using the following parameters: (i) $a = 2$,$b = 1$, $V = 1$; (ii) $a = 10$, $b = 1$, $V = 1$;(iii) $a = 2$, $b = .1$, $V = 10$. Comment on the plots: do they conform with your intuition?

(c) For each set of parameters in the previous part, plot the velocity of the center of the membrane as a function of time for $0 < t < 10$.

(d) Explain what happens to the solution of (1) when $a \to \infty$ with all other parameters held fixed.

Solution: To begin, let us define $u(x,t) = f(t)g(x,y)$ so as to use the method of separation of variables, which can be used because equation (1) is linear homogeneous. As such, we can substitute u into equation (1) to obtain the following system of DE,

$$\frac{d^2 f}{dt^2} = -\lambda f, \quad f(0) = 0, \tag{2}$$

$$\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} = -\lambda g, \tag{3}$$

where equation (2) is ordinary and equation (3) is partial. Further, separation of variables can be used a second time for equation (3), using $g(x,y) = h(x)p(y)$ to produce the following ODE,

$$\frac{d^2h}{dx^2} = -\mu h, \quad h(-a) = h(a) = 0; \tag{4}$$
$$\frac{d^2p}{dy^2} = -(\lambda - \mu)p, \quad g(-a) = g(a) = 0. \tag{5}$$

Because equations (2), (4), and (5) are variations of the simple harmonic oscillator, their solutions are trivial to form and are given by the following, respectively:

$$f(t) = A\cos(\sqrt{\lambda}t) + B\sin(\sqrt{\lambda}t), \tag{6}$$
$$h(x) = a\cos(\sqrt{\mu}x) + b\sin(\sqrt{\mu}x), \tag{7}$$
$$p(y) = c\cos(\sqrt{(\lambda-\mu)}y) + d\sin(\sqrt{(\lambda-\mu)}y). \tag{8}$$

To begin forming the solution, we must first solve the boundary-value problem for equations (7) and (8) to determine the eigenvalues λ, μ; so, for $h(x)$, we must solve the following matrix-vector system:

$$\begin{bmatrix} \cos\left(\sqrt{\mu}a\right) & \sin\left(\sqrt{\mu}a\right) \\ \cos\left(\sqrt{\mu}a\right) & -\sin\left(\sqrt{\mu}a\right) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{9}$$

In doing so, in order that the solution is non-trivial for the system, we find that the possible eigenvalues μ are

$$\mu_m = \frac{m^2\pi^2}{a^2}, \quad \text{or}$$
$$\mu_m = \frac{\left(m+\frac{1}{2}\right)^2\pi^2}{a^2}.$$

In taking the first of the two as our choice—for, each set of eigenvalues produces a set of solutions to equation (1)—this implies that $a = 0$,

$$\begin{aligned} h(-a) = h(a) &= a\cos(m\pi) + b\sin(m\pi) \\ &= a\cos(m\pi) = 0 \\ &\implies a = 0. \end{aligned}$$

Thus, the eigenfunctions are

$$h_m(x) = b_m \sin\left(\frac{m\pi x}{a}\right) \quad m = 1, 2, 3, \ldots \tag{10}$$

Because we have the eigenvalues μ_m, we can determine the eigenvalues from the other boundary condition, and thus the other eigenfunctions. In solving a system similar to that in equation (9) for $g(y)$, we find that the eigenvalues are

$$\lambda_{mn} = \frac{n^2\pi^2}{a^2} + \frac{m^2\pi^2}{a^2},$$

where in a likewise manner as before, we find that the eigenfunctions, due to the boundary conditions, are

$$p_n(y) = d_n \sin\left(\frac{n\pi y}{a}\right). \quad n = 1, 2, 3, \ldots$$

Lastly, because we have λ_{mn}, we have

$$f_{mn}(t) = A_{mn} \cos\left(\frac{\sqrt{m^2+n^2}\pi t}{a}\right) + B_{mn} \sin\left(\frac{\sqrt{m^2+n^2}\pi t}{a}\right).$$

Now, we can form the general series solution to equation (1):

$$\begin{aligned} u(x,y,t) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} f_{mn}(t) g_{mn}(x,y) \\ &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \left[A_{mn} \cos\left(\frac{\sqrt{m^2+n^2}\pi t}{a}\right) + B_{mn} \sin\left(\frac{\sqrt{m^2+n^2}\pi t}{a}\right)\right] \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{a}\right), \end{aligned} \tag{11}$$

where the constant terms were absorbed into A_{mn} and B_{mn}. It is easy to check that this is indeed a solution to equation (1). Now, we must satisfy the initial conditions. In doing so, we find that A_{mn} must be equal to zero.

$$\begin{aligned} u(x,y,0) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{a}\right) = 0 \\ \implies A_{mn} = 0. \end{aligned}$$

Further, the initial velocity is

$$u_t(x,y,0) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} B_{mn} \frac{\sqrt{m^2+n^2}\pi}{a} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{a}\right) = v(x,y),$$

where the coefficients $B_{mn}\frac{\sqrt{m^2+n^2}\pi}{a}$ are obtained by projecting the initial velocity twice, once onto each eigenspace, utilizing the inner-product and orthogonality of the eigenbasis:

$$B_{mn}\frac{\sqrt{m^2+n^2}\pi}{a} = \frac{\int\limits_D v(x,y) \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{a}\right) dxdy}{\int\limits_D \sin^2\left(\frac{m\pi x}{a}\right) \sin^2\left(\frac{n\pi y}{a}\right) dxdy}.$$

In substituting this expression into equation (11), taking into account initial conditions, the general solution becomes

$$\boxed{u(x,y,t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{\int\limits_D v(x',y')\sin\left(\frac{m\pi x'}{a}\right)\sin\left(\frac{n\pi y'}{a}\right)dx'dy'}{\int\limits_D \sin^2\left(\frac{m\pi x'}{a}\right)\sin^2\left(\frac{n\pi y'}{a}\right)dx'dy'} \sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{a}\right)\sin\left(\frac{\sqrt{m^2+n^2}\pi t}{a}\right)}.$$

However, because we need to determine the velocity of the center of the membrane with respect to time, it would be more convenient to use the cosine solution of the above equation, which is obtained in the same way as was computed for the sine solution, using instead :

$$u(x,y,t) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{\int\limits_D v(x',y')\cos\left(\frac{(m+\frac{1}{2})\pi x'}{a}\right)\cos\left(\frac{(n+\frac{1}{2})\pi y'}{a}\right)dx'dy'}{\int\limits_D \cos^2\left(\frac{(m+\frac{1}{2})\pi x'}{a}\right)\cos^2\left(\frac{(n+\frac{1}{2})\pi y'}{a}\right)dx'dy'}$$
$$\cos\left(\frac{(m+\frac{1}{2})\pi x}{a}\right)\cos\left(\frac{(n+\frac{1}{2})\pi y}{a}\right)\sin\left(\frac{\sqrt{(m+\frac{1}{2})^2+(n+\frac{1}{2})^2}\pi t}{a}\right).$$

In using the cosine solution, we can adequately visualize the solution for the specified parameters, provided down below, in addition to the code that produced the plots.

```
% Exercise 1

N = 100;
a = 2;
b = 0.1;
t = 1;
V = @(x,y) 10*(x<b & x>-b & y<b & y>-b) + 0*(x>b & x<-b & y>b & y<-b);
[x,y] = meshgrid(linspace(-a,a,N));

u = zeros(size(x));

% Sine Solution

for m = 1:5
    for n = 1:5
        u = u + integral2(@(x,y)V(x,y).*sin(m*pi*x/a).*sin(n*pi*y/a)...
        ,-a,a,-a,a).*sin(m*pi*x/a).*sin(n*pi*y/a).*sin(sqrt(m^2 + n^2)...
        *pi*t/a)/integral2(@(x,y) (sin(m*pi*x/a).^2).*(sin(n*pi*y/a).^2)...
        ,-a,a,-a,a);
    end
end

```

```
% Cosine Solution

for m = 1:5
    for n = 1:5
        u = u + integral2(@(xx,yy) V(xx,yy).*cos((m+1/2)*pi*xx/a)...
        .*cos((n+1/2)*pi*yy/a),-a,a,-a,a).*cos((m+1/2)*pi*x/a).*...
        cos((n+1/2)*pi*y/a).*sin(sqrt((m+1/2)^2 + (n+1/2)^2)*pi*t/a)/...
        integral2(@(xx,yy) ...
            (cos((m+1/2)*pi*xx/a).^2).*(cos((n+1/2)*pi*yy/a).^2)...
        ,-a,a,-a,a);
    end
end

tt = linspace(0,10,1000);
du = zeros(size(tt));
x1 = 0.01;
y1 = 0.01;
ddu = zeros(size(tt));
xx1 = 0;
yy1 = 0;

% Sine Velocity: Always Zero

for n = 1:10
    for m = 1:10
        du = du + (sqrt(m^2 + n^2)*pi/a)*integral2(@(xx,yy) ...
            V(xx,yy).*sin(m*pi*xx/a).*sin(n*pi*yy/a),-a,a,-a,a)...
        .*sin(m*pi*x1/a).*sin(n*pi*y1/a).*cos(sqrt(m^2 + ...
            n^2).*pi.*tt/a)/...
        integral2(@(xx,yy) ...
            ((sin(m*pi*xx/a)).^2).*((sin(n*pi*yy/a)).^2),-a,a,-a,a);
    end
end

% Cosine Velocity

for m = 1:15
    for n = 1:15
        ddu = ddu + sqrt((m+1/2)^2 + ...
            (n+1/2)^2)*integral2(@(xx,yy)V(xx,yy).*cos((m+1/2)*pi*xx/a).*...
        cos((n+1/2)*pi*yy/a),-a,a,-a,a)...
        .*cos((m+1/2)*pi*xx1/a).*cos((n+1/2)*pi*yy1/a)...
        .*sin(sqrt((m+1/2)^2 ...
            +(n+1/2)^2)*pi*tt/a)/integral2(@(xx,yy)(cos((m+1/2)*pi*xx/a).^2)
        .*(cos((n+1/2)*pi*yy/a).^2),-a,a,-a,a);
    end
end


figure
mesh(x,y,u);
```

```
xlabel('$x$','interpreter','latex');
ylabel('$y$','interpreter','latex');
zlabel('$u(x,y,t)$','interpreter','latex');
title(sprintf('$u(x,y,1)$: $a$ = %1.0f, $b$ = %1.1f, $V$ = 10',[a ...
    b]),'interpreter','latex');
set(gca,'fontsize',18);

figure
plot(tt,ddu);
xlabel('Time (sec)','interpreter','latex');
ylabel('$u_{t}(0,0,t)$','interpreter','latex');
title(sprintf('Velocity: $a$ = %1.0f, $b$ = %1.1f, V = 10',[a ...
    b]),'interpreter','latex');
set(gca,'fontsize',18);
```

For the plots, the cosine solution is first, the sine solution second, and lastly lies the velocity of the membrane at the center as a function of time. Further, the first five terms of the sum for each solution were computed, whereas the first ten terms of the sum were used for the velocity.

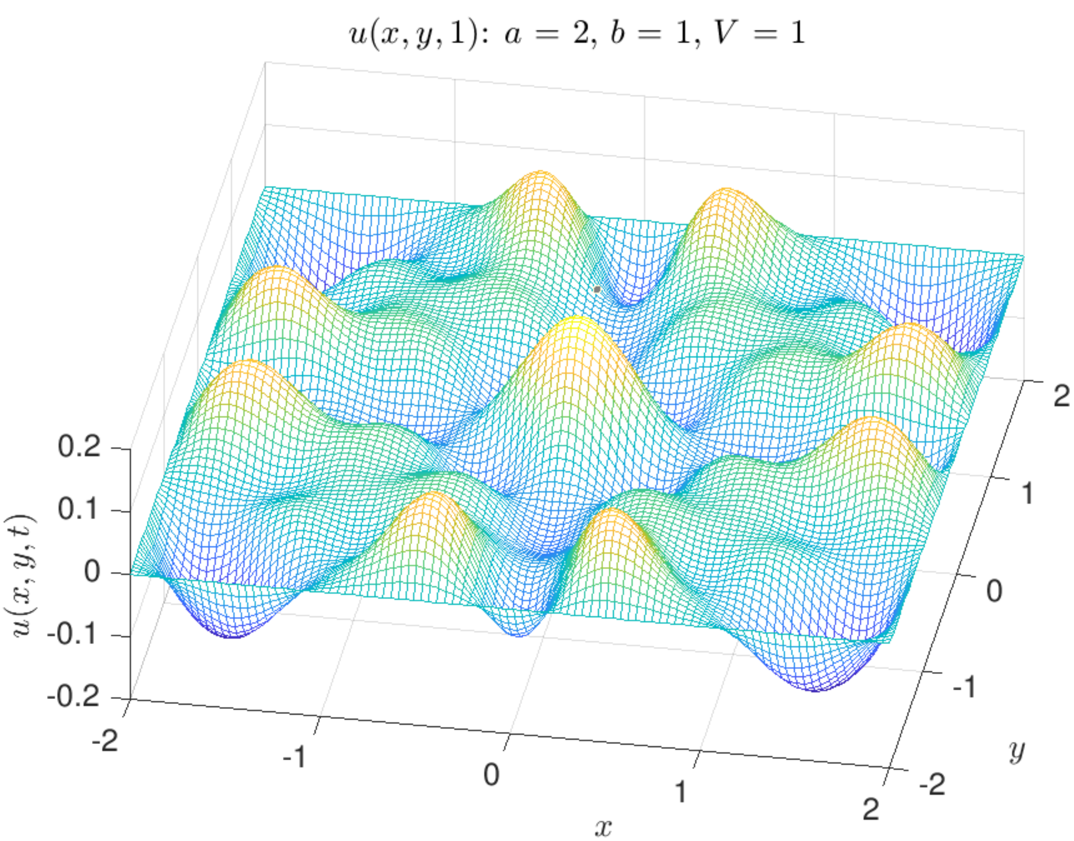


Figure 27: The cosine solution for condition (a).

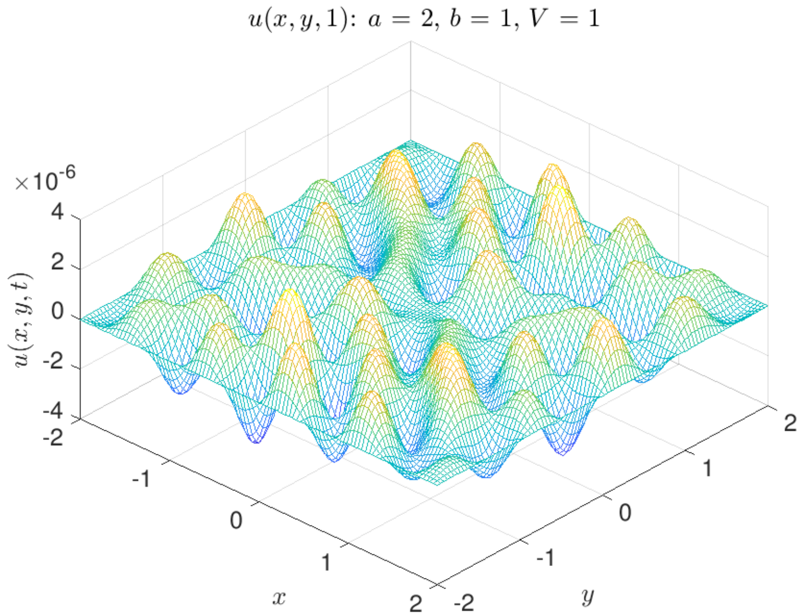


Figure 28: The sine solution for condition (a).

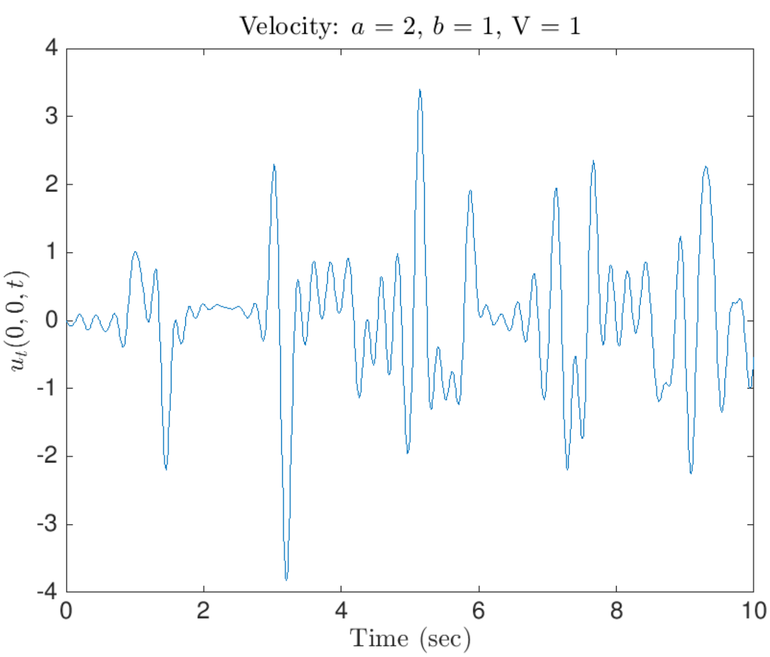


Figure 29: The velocity of the center for condition (a).

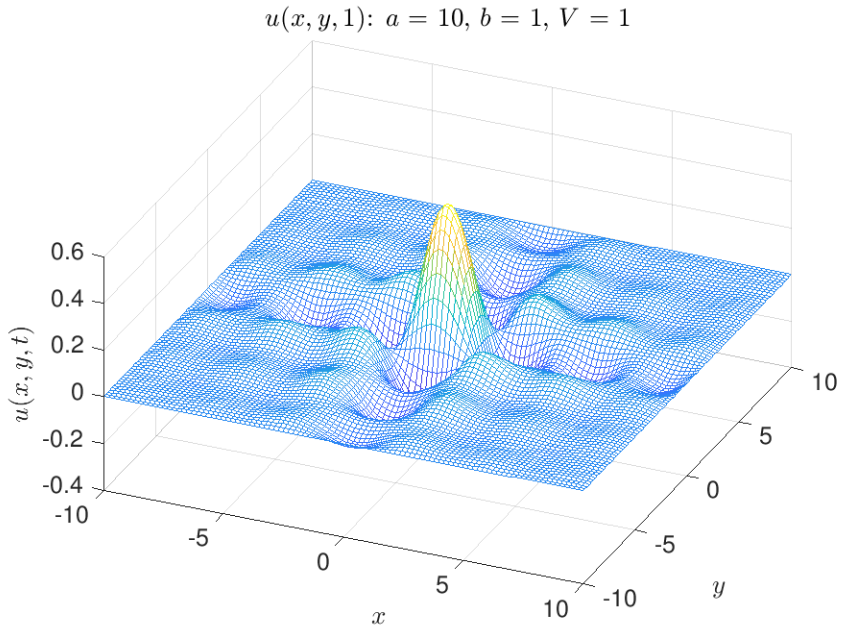


Figure 30: The cosine solution for condition (b).

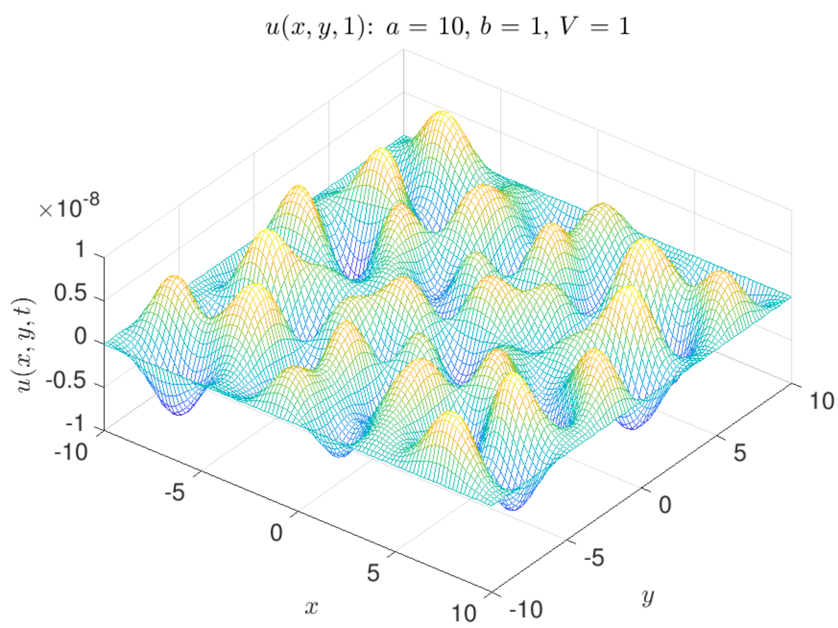


Figure 31: The sine solution for condition (b).

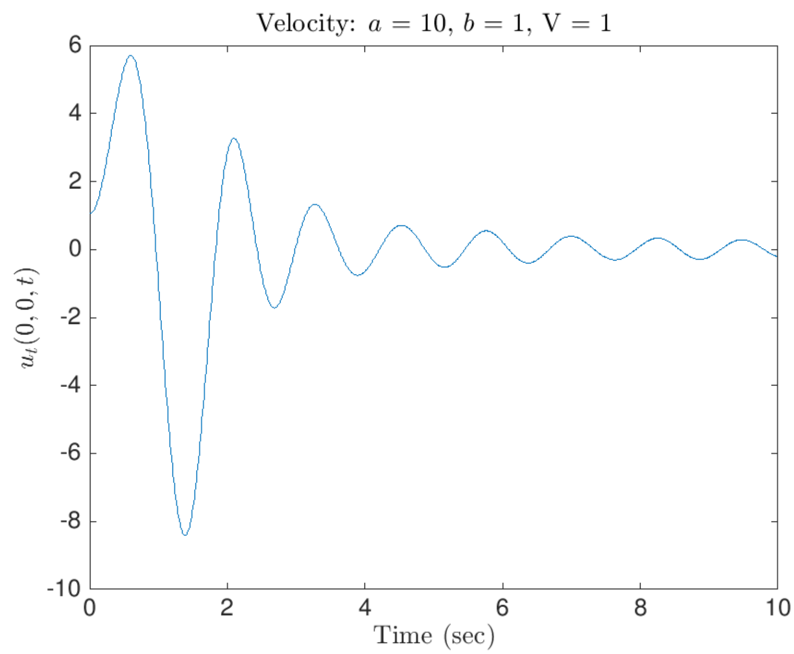


Figure 32: The velocity of the center for condition (b).

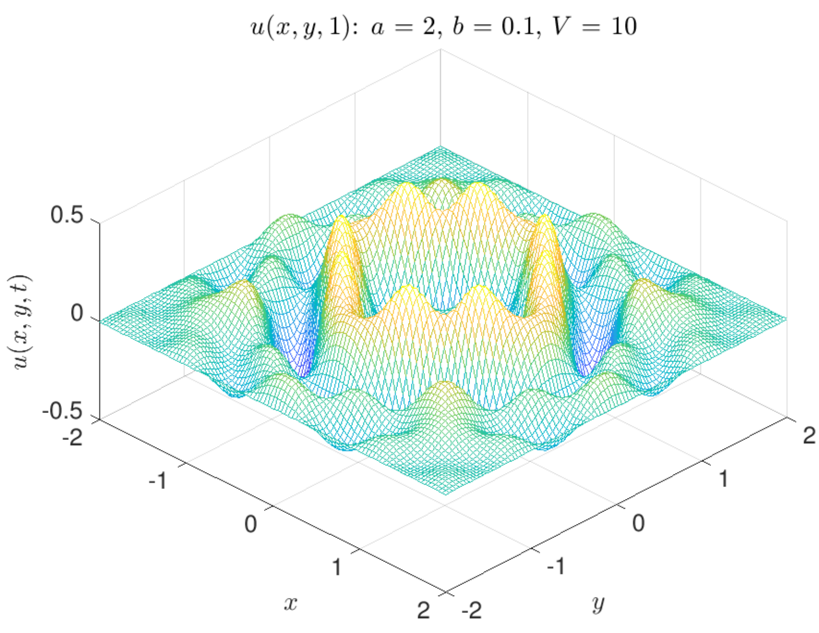


Figure 33: The cosine solution for condition (c).

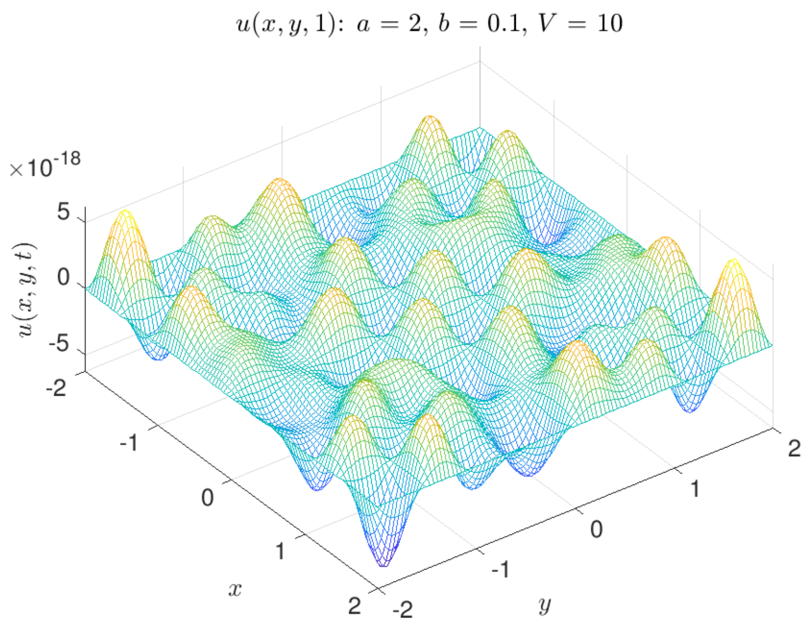


Figure 34: The sine solution for condition (c).

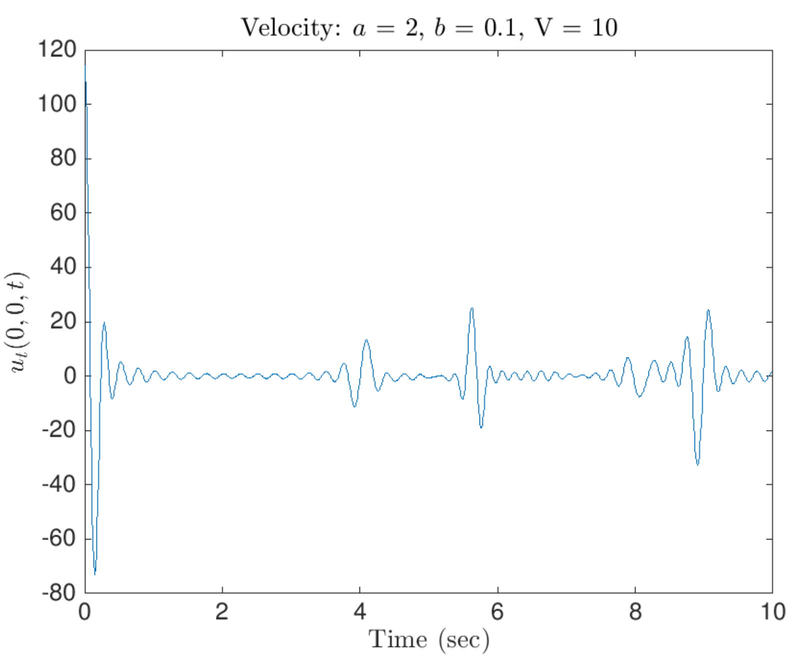


Figure 35: The velocity of the center for condition (c).

There are a few things to notice about the plots. First, in observing the contrast between the sine and cosine solutions, the sine solution is obviously odd, where the cosine solution has more of a radial symmetry, if anything. As a result, we observe that the sine solution has a constant velocity of zero at its center, whereas the cosine solution has non-zero velocities at its center on the specified time interval. Further, the sine graphs are much less intuitive compared to their cosine counter parts. The cosine graphs look as though someone has struck the membrane at its center, thus creating the ripples that propagate throughout the membrane. Further, in varying the parameters, we see that when the parameters are quite different in value from each other, the behavior of the membrane changes drastically. In Figure 1, there are a decent number of peaks, which contrasts with the plot in Figure 4, where only the center is considerably prominent. Even still, the graph in Figure 7 has a circular deformation at its center, due to the effect of the velocity and changing the domain on which it's defined. Having a high initial velocity on a small patch of the membrane is going to have an appreciable affect because the velocity is in a more concentrated region, as in Figure 7, whereas in Figures 1 and 4, the initial velocity is spread out over more of the membrane. Ultimately, because there is a change in velocity in the membrane as time progresses, there is an acceleration present, suggesting that there is internal force. And so, concentrated forces produce drastic results, as in Figure 7. As a result the plots of the cosine solution are what I would expect with intuition, especially Figures 4 and 7. Figure 1 is not as obvious, but it still makes sense. The plots of the sine solutions are not at all intuitive, but rather slightly confusing due to the odd nature of the plots. After trying some different parameters, the graphs became much clearer.

To see what happens as $a \to \infty$, first interchange summation and integration. Also, it would be useful to know that the denominator in the boxed equation above produces two factors of a, suggesting that in place of the denominator, scale the solution by $\frac{1}{a^2}$. After interchanging summation and integration, in treating the summations as Riemann sums, which is why we need both factors of $1/a$, we essentially apply a Fourier Sine Transform of the initial data and then we apply the Inverse Fourier Sine Transform to obtain the continuous solution:

$$\boxed{u(x,y,t) = \frac{1}{\pi^2}\int_0^\infty \int_0^\infty \sin\left(\sqrt{\xi^2+\eta^2}t\right)\sin(\xi x)\sin(\eta y)\left[\int_{-\infty}^\infty \int_{-\infty}^\infty v(x',y')\sin(\xi x')\sin(\eta y')dx'dy'\right]d\xi d\eta},$$

where $r = \frac{n}{a}$, $s = \frac{m}{a}$, $\eta = \pi r$, and $\xi = \pi s$. The same is true for the cosine solution; the only difference is that the sine terms that depend on space are replaced with cosine. The sine with time dependence stays the same to satisfy the initial conditions.

Correction and Commentary: This is definitely my favorite class by far, and my favorite exam, despite having to have that painfully honest conversation with you. Anyhow, the correction I need to make is that the solution should include a factor of

$$\frac{1}{\pi a\sqrt{m^2+n^2}},$$

further suggesting that I should divide by its analog in the expression that was derived as $a \to \infty$. This doesn't, however affect the overall appearance of the solutions in the plots.

EXERCISE 2

2. Let D be the same square as in the previous problem. Solve the Laplace equation on D with Dirichlet boundary conditions

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (x, y) \in D \quad u\big|_{\partial\Omega} = f, \tag{12}$$

where the function f equals 1 on vertical sides of the square and (-1) on the horizontal sides. Plot the solution for a = 2 and comment on the plot: does it make sense?

Solution: Because the boundary conditions are not homogeneous, it would be useful to separate each condition so as to solve the Laplace Equation for each boundary of the square. Then, in taking the superposition of the four boundary conditions individually, this will produce the full solution that satisfies all boundary conditions. Thus, the boundary-value problem we have is shown in Figure 10 below. To begin solving the Laplace equation, let us guess that $u(x, y) = f(x)g(y)$. Let us solve the BVP for u_4. In substituting u into equation (12), we obtain

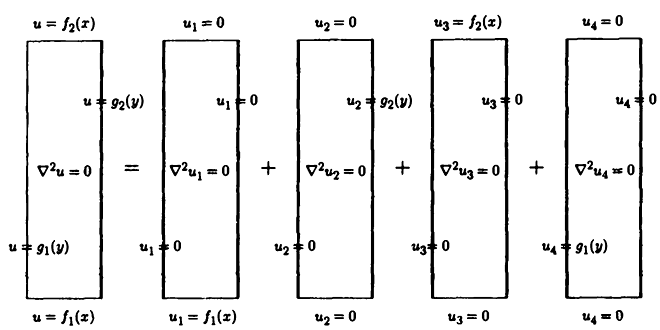


Figure 36: Dividing the boundary-value problem into smaller ones. For this problem, the domain is D and the boundaries are: $g_1(y) = 1$, $g_2(y) = 1$, $f_1(x) = -1$, $f_2(x) = -1$. Image Credit: Haberman.

$$\frac{d^2 f}{dx^2} = \lambda h,$$
$$\frac{d^2 g}{dy^2} = -\lambda g,$$
$$u(-a, y) = g_1(y), \tag{i}$$
$$u(a, y) = 0, \tag{ii}$$
$$u(x, -a) = 0, \tag{iii}$$
$$u(x, a) = 0, \tag{iv}$$

where the solutions are given by

$$f(x) = ae^{\sqrt{\lambda}x} + be^{-\sqrt{\lambda}x} = a\cosh\left(\sqrt{\lambda}x\right) + b\sinh\left(\sqrt{\lambda}x\right),$$
$$g(y) = c\cos\left(\sqrt{\lambda}y\right) + d\sin(\sqrt{\lambda}y).$$

The way in which the eigenvalues were defined allows us to compute them easily due to the boundary conditions (iii) and (iv). The eigenvalues are $\frac{n^2pi^2}{a^2}$, as was determined before. Further, to satisfy the boundary conditions, this implies that $c = 0$, getting rid of the cosine terms, so the general solution is given by

$$u(x, y) = \sum_{n=1}^{\infty}\left[a_n\cosh\left(\frac{n\pi x}{a}\right) + b_n\sinh\left(\frac{n\pi x}{a}\right)\right]\sin\left(\frac{n\pi}{a}\right).$$

To satisfy (ii), we must apply a translation by a. This can be performed because the Laplace equation has constant coefficients. Thus, translations don't affect the overall solution. The new solution is

$$u(x, y) = \sum_{n=1}^{\infty}\left[a_n\cosh\left(\frac{n\pi(x-a)}{a}\right) + b_n\sinh\left(\frac{n\pi(x-a)}{a}\right)\right]\sin\left(\frac{n\pi y}{a}\right),$$

and in satisfying (ii), we find that $a_n = 0$ because $\cosh(0) = 1$ and $\sinh(0) = 0$. Lastly, to satisfy (i), we have

$$u(-a, y) = \sum_{n=1}^{\infty} A_n\sinh(-2\pi n)\sin\left(\frac{n\pi y}{a}\right) = g_1(y),$$

where we perform the usual orthogonal projection to find that

$$A_n = \frac{1}{a\sinh(-2\pi n)}\int_{-a}^{a} g_1(y)\sin\left(\frac{n\pi y}{a}\right)dy.$$

Thus, the overall solution for u_4 is

$$u_4(x,y) = \sum_{n=1}^{\infty} \left[\frac{1}{a\sinh(-2\pi n)} \int_{-a}^{a} g_1(y) \sin\left(\frac{n\pi y}{a}\right) dy \right] \sinh\left(\frac{n\pi(x-a)}{a}\right) \sin\left(\frac{n\pi y}{a}\right).$$

To obtain the full solution, we apply the same analysis to the other three BVP's and sum all four individual solutions. The full solution is thus given by

$$\begin{aligned} u(x,y) = &\sum_{n=1}^{\infty} \left[\frac{1}{a\sinh(-2\pi n)} \int_{-a}^{a} g_1(y) \sin\left(\frac{n\pi y}{a}\right) dy \right] \sinh\left(\frac{n\pi(x-a)}{a}\right) \sin\left(\frac{n\pi y}{a}\right) \\ &+ \sum_{n=1}^{\infty} \left[\frac{1}{a\sinh(2\pi n)} \int_{-a}^{a} f_1(x) \sin\left(\frac{n\pi x}{a}\right) dy \right] \sinh\left(\frac{n\pi(y+a)}{a}\right) \sin\left(\frac{n\pi x}{a}\right) \\ &+ \sum_{n=1}^{\infty} \left[\frac{1}{a\sinh(2\pi n)} \int_{-a}^{a} g_2(y) \sin\left(\frac{n\pi y}{a}\right) dy \right] \sinh\left(\frac{n\pi(x+a)}{a}\right) \sin\left(\frac{n\pi y}{a}\right) \\ &+ \sum_{n=1}^{\infty} \left[\frac{1}{a\sinh(-2\pi n)} \int_{-a}^{a} f_1(x) \sin\left(\frac{n\pi x}{a}\right) dy \right] \sinh\left(\frac{n\pi(y-a)}{a}\right) \sin\left(\frac{n\pi x}{a}\right). \end{aligned}$$

Even though this is the general formula for the solution to the Laplacian on the square, because the function value on the boundary has values of -1 (horizontal sides) and 1 (vertical sides), then the integrals return a value of zero, suggesting that $u(x,y) = 0$. At first I was extremely skeptical of this conclusion, but after testing out my code with different boundary conditions, I was convinced. Because I used the numerical solver, I was able to generated different plots based on the number of terms used in the sum. Here is the code and the resulting plots.

```
%% Exercise 2

a = 2;
N = 100;
[x,y] = meshgrid(linspace(-a,a,N));

u1 = zeros(size(x));
u2 = zeros(size(x));
u3 = zeros(size(x));
u4 = zeros(size(x));


for n = 1:128
    u1 = u1 + quadgk(@(yy) ...
        sin(n*pi*yy/a),-a,a).*sinh(n*pi*(x-a)/a).*sin(n*pi*y/a)...
         /(a*sinh(-2*pi*n));
    u2 = u2 + quadgk(@(xx) ...
        -sin(n*pi*xx/a),-a,a).*sinh(n*pi*(y+a)/a).*sin(n*pi*x/a)...
```

```
        /(a*sinh(2*pi*n));
    u3 = u3 + quadgk(@(yy) ...
       sin(n*pi*yy/a),-a,a).*sinh(n*pi*(x+a)/a).*sin(n*pi*y/a)...
        /(a*sinh(2*pi*n));
    u4 = u4 + quadgk(@(xx) ...
       -sin(n*pi*xx/a),-a,a).*sinh(n*pi*(y-a)/a).*sin(n*pi*x/a)...
        /(a*sinh(-2*pi*n));
end

uu = u1 + u2 + u3 + u4;

figure
mesh(x,y,uu);
xlabel('$x$','interpreter','latex');
ylabel('$y$','interpreter','latex');
zlabel('$u(x,y)$','interpreter','latex');
title('Solution to Laplace''s Equation','interpreter','latex');
set(gca,'fontsize',18);
axis square
```

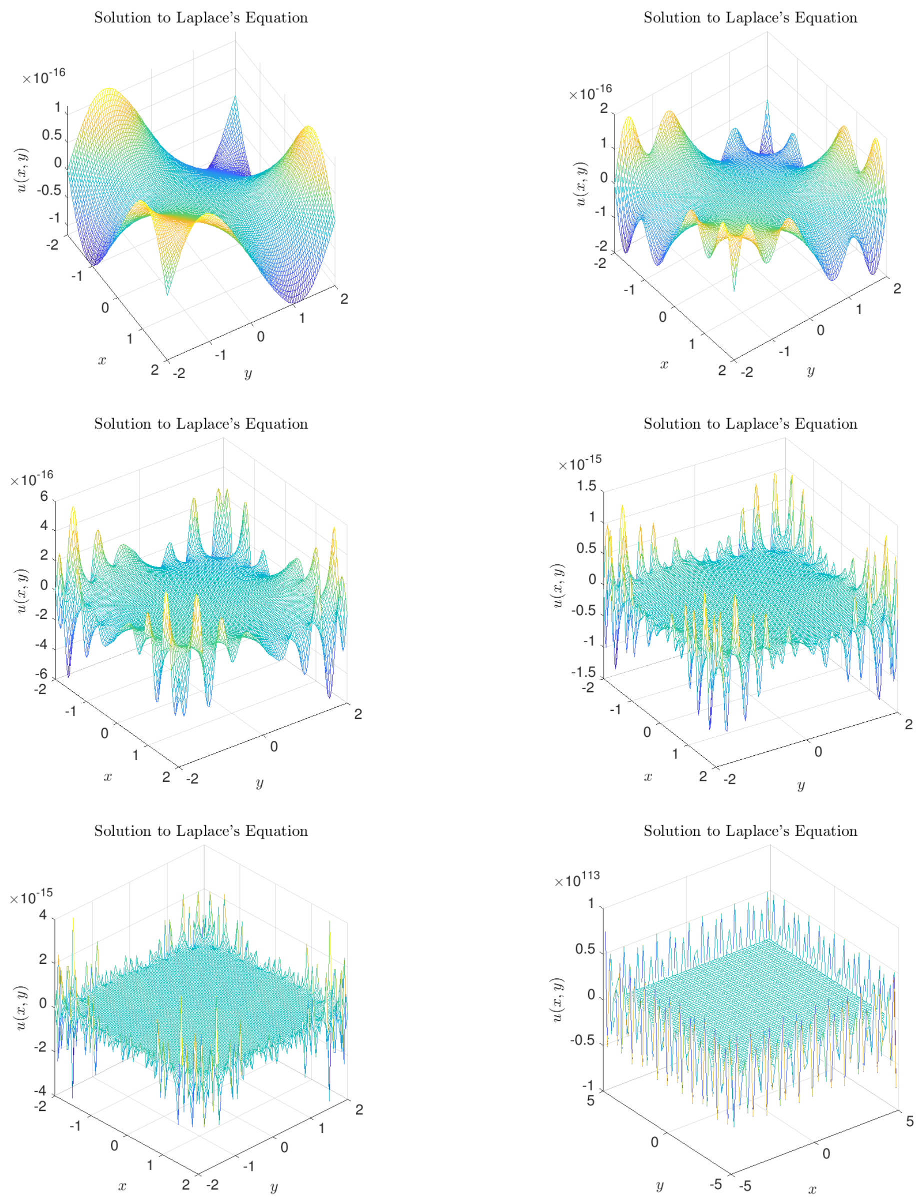


Figure 37: From left to right: $N = 2$, $N = 4$, $N = 8$, $N = 16$, $N = 32$, $N = 64$.

Interestingly enough, because numerical solvers were used to approximate the solution, we can see how increasing the number of terms affects what the solution looks like. Of course, the ideal plot is a flat plane, but it is interesting to see how ill-conditioned as N increases, but the flat plane is eventually seen, as expected.

EXERCISE 3

3. Solve the following initial value problem:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - u, \quad x \in \mathbb{R}, \quad u\big|_{t=0} = e^{-x^2}. \tag{13}$$

You can think of Equation (3) as heat equation for an imperfectly insulated infinite rod. Plot you solution for $t = 0.1$, $t = 0.5$, and $t = 1$ using $-5 \leq x \leq 5$ as the horizontal range.

Solution: Because the domain is the entire real line, and consequently there are no boundary conditions, we must use the Fourier Transform. In doing so, equation (13) becomes

$$\frac{d\widehat{u}}{dt} = -(\xi^2 + 1)\widehat{u}, \quad \widehat{u}\big|_{t=0} = \widehat{e^{-x^2}}. \tag{14}$$

The solution to equation (14) is

$$\widehat{u} = e^{-(\xi^2+1)t}\widehat{e^{-x^2}}.$$

The explicit expression for u is

$$\boxed{u(x,t) = \left(e^{-(\xi^2+1)t}\widehat{e^{-x^2}}\right)^{\vee} = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\xi x}e^{-(\xi^2+1)t}\left[\int_{-\infty}^{\infty} e^{-i\xi x}e^{-y^2}\,dy\right]d\xi}.$$

Here is the code and the resulting plot.

```
%% Exercise 3

t = [0 0.1 0.5 1];

x = linspace(-5,5,100);

u = zeros(length(x),1);

for tt = 1:length(t)
    for j = 1:length(x)
        u(j,:) = integral2(@(w,y) exp(1i*w.*(x(j)-y))...
            .*exp(-(w.^2 + 1)*tt).*exp(-y.^2),-5,5,-5,5)/(2*pi);
    end
    hold on
    plot(x,u);
    xlabel('Rod Position','interpreter','latex');
    ylabel('$u(x,t)$','interpreter','latex');
    legend(sprintf('$t$ = %1.1f',t(1)),sprintf('$t$ = %1.1f',t(2))...
        ,sprintf('$t$ = %1.1f',t(3)),sprintf('$t$ = ...
            %1.1f',t(4)),'interpreter','latex');
    title('Temperature Over Time','interpreter','latex');
    set(gca,'fontsize',18)
```

```
end
```

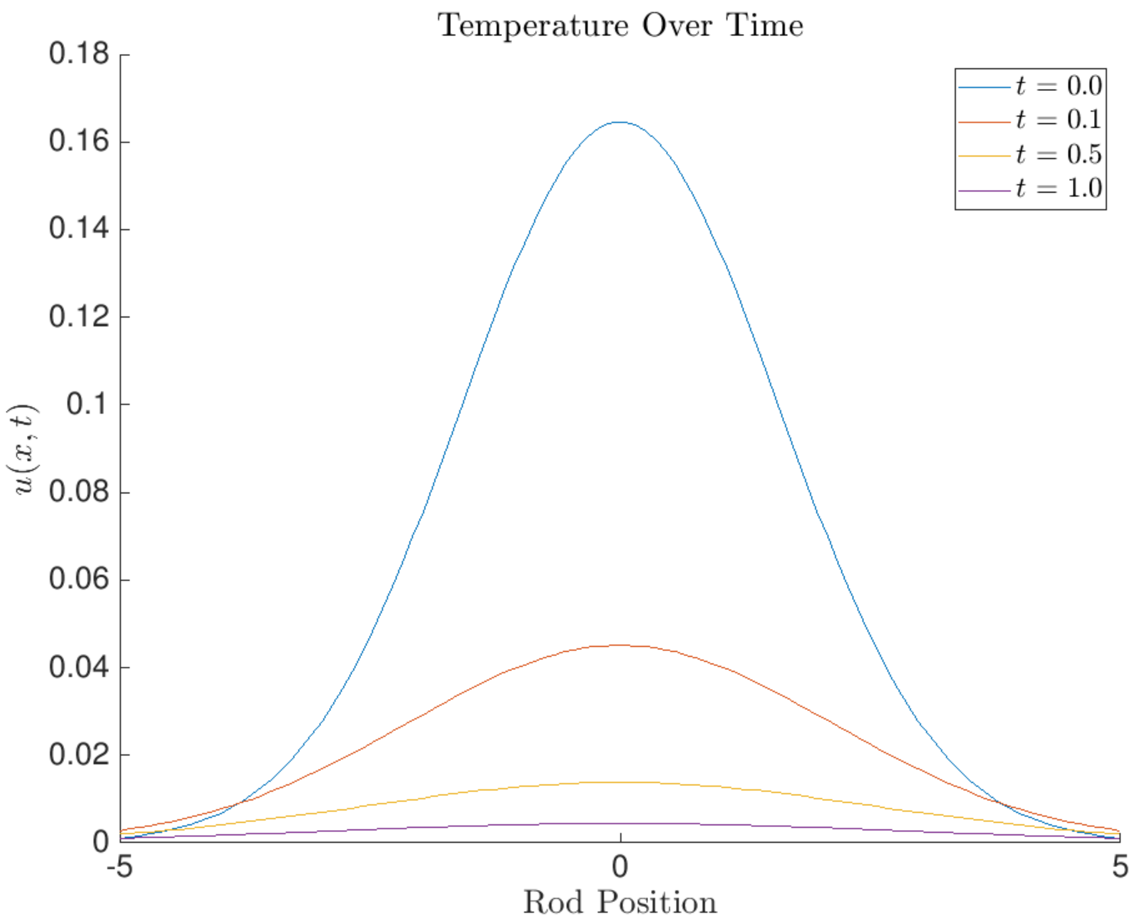


Figure 38: Plot of the temperature in the metal on the specified interval for the specified times.

HOMEWORK 7

EXERCISE 1

Derive the closed form of the solution to Laplace's Equation on the Disk, i.e., find the Poisson Kernel,

$$\begin{aligned} &\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = 0, \quad u\big|_{\partial D} = h \\ &0 \le r < 1, \quad 0 < \theta \le 2\pi. \end{aligned} \tag{1}$$

Solution: To begin, let us use the method of separation of variables: define $u(r,\theta) = g(r)f(\theta)$. Now, in substituting u into equation (1), we obtain the following:

$$\frac{d^2 f}{d\theta^2} = -n^2 f, \tag{2}$$

$$r^2\frac{d^2 g}{dr^2} + r\frac{dg}{dr} - n^2 g = 0, \tag{3}$$

where we define $\lambda = n^2$. Equation (3) is known as Euler's equation. The solution to equation (2) is

$$f(\theta) = Ae^{in\theta} + Be^{-in\theta}. \tag{4}$$

The solution to Euler's equation is not as obvious, but we use the *ansatz* $g(r) = r^\lambda$, i.e., a guess that is a power. In substituting our guess into equation (3) and further simplifying, we find that $\lambda = \pm n$, making our solution a linear combination of powers:

$$g(r) = ar^n + br^{-n}. \tag{5}$$

The full solution is thus

$$u(r,\theta) = \sum_{n=0}^{\infty} \left(A_n e^{in\theta} + B_n e^{-in\theta}\right)\left(a_n r^n + b_n r^{-n}\right). \tag{6}$$

Let us further simplify this expression. The first thing to notice is that when $r = 0$, u becomes discontinuous. This implies that, for a continuous solution, the coefficients b_n must be equal to zero. Further, we can change the index of summation so that we only have one exponential term; this also means that in doing so, we must have $r^{|n|}$ for continuous solutions. Lastly, in collecting all of the terms, we obtain the following simplified solution:

$$u(r,\theta) = \sum_{n=-\infty}^{\infty} A_n r^{|n|} e^{in\theta}. \tag{7}$$

To satisfy the boundary condition, we have

$$u(1,\theta) = \sum_{n=-\infty}^{\infty} A_n e^{in\theta} = h(\theta).$$

We thus project the initial condition onto the classical Fourier Basis and ultimately perform the classic Fourier Transform on the disk to produce the coefficients

$$A_n = \frac{1}{2\pi}\int_0^{2\pi} e^{-in\theta} h(\theta) d\theta,$$

where the full solution is

$$u(r,\theta) = \sum_{n=-\infty}^{\infty} \left[\frac{1}{2\pi}\int_0^{2\pi} e^{-int} h(t)dt\right] r^{|n|} e^{in\theta} \tag{8}$$

where we changed the variable of integration to avoid obfuscating the independent variable θ. To obtain the closed form of the solution, we first rearrange the order of operations to obtain

$$u(r,\theta) = \frac{1}{2\pi}\int_0^{2\pi} \left[\sum_{n=-\infty}^{\infty} r^{|n|} e^{in(\theta-t)}\right] h(t)dt. \tag{9}$$

Now, the Poisson Kernel is

$$K_n(\theta - t) = \sum_{n=-\infty}^{\infty} r^{|n|} e^{in(\theta-t)}.$$

So, the solution to the Laplace equation on a disk can be expressed as the convolution of the Poisson Kernel and the initial condition, which is reminiscent of the solution to the Heat equation on the real line:

$$u(r,\theta) = K_p \star h = \frac{1}{2\pi}\int_0^{2\pi} K_n(\theta - t) h(t) dt. \tag{10}$$

To derive an exact expression for the Poisson Kernel, first we rewrite the sum, collect terms, and simplify:

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} r^{|n|} e^{in(\theta-t)} &= \sum_{n=0}^{\infty} r^n e^{in(\theta-t)} + \sum_{n=-\infty}^{-1} r^{|n|} e^{in(\theta-t)} \\
&= \sum_{n=0}^{\infty} r^n e^{in(\theta-t)} + \sum_{n=1}^{\infty} r^n e^{-in(\theta-t)} \\
&= \sum_{n=0}^{\infty} r^n e^{in(\theta-t)} + \sum_{n=0}^{\infty} r^n e^{-in(\theta-t)} - 1.
\end{aligned}$$

Next, we apply the formula for geometric series with infinite sums,

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}.$$

In doing so, we rewrite the sums, find the common denominator, and simplify:

$$\begin{aligned}\sum_{n=0}^{\infty} r^n e^{in(\theta-t)} + \sum_{n=0}^{\infty} r^n e^{-in(\theta-t)} - 1 &= \frac{1}{1-re^{i(\theta-t)}} + \frac{1}{1-re^{-i(\theta-t)}} - 1 \\ &= \frac{1-re^{-i(\theta-t)} + 1 - re^{i(\theta-t)} - \left(1-re^{i(\theta-t)}\right)\left(1-re^{-i(\theta-t)}\right)}{\left(1-re^{i(\theta-t)}\right)\left(1-re^{-i(\theta-t)}\right)} \\ &= \boxed{\frac{1-r^2}{1-2r\cos(\theta-t)+r^2}}.\end{aligned}$$

In substituting the closed form of the Poisson Kernel into equation (10), we have the full solution to Laplace's Equation on a disk:

$$\boxed{u(r,\theta) = \frac{1}{2\pi}\int_0^{2\pi}\left[\frac{1-r^2}{1-2r\cos(\theta-t)+r^2}\right]h(t)dt.} \tag{11}$$

Here is some code that utilizes different initial conditions. The first plot has the initial condition $g(\theta) = \theta\log\theta$ and the second $g(\theta) = \cos 4\theta + \sin 2\theta$, respectively.

```
%% Poisson Kernel and Solution to Laplace Equation on a Disk

% Generating the grid

N = 100;
[r,theta] = meshgrid(linspace(0,1,N),linspace(0,2*pi,N));

% Initial Conditions

%g = @(p) log(p).*p;
g = @(p) cos(4*p) + sin(2*p);

% Solution

u = zeros(size(r));

for n = 0:N
    u = u + (1/pi)*integral(@(t) ...
        g(t).*cos(n*t),0,2*pi).*cos(n*theta).*r.^(abs(n))...
         + (1/pi)*integral(@(t) ...
             g(t).*sin(n*t),0,2*pi).*sin(n*theta).*r.^(abs(n));
```

```
end

% Plotting in Cartesian Coordinates

figure
mesh(r.*cos(theta),r.*sin(theta),real(u));
xlabel('$x$','Interpreter',"latex");
ylabel('$y$','Interpreter',"latex");
zlabel('$u(x,y)$','Interpreter',"latex");
title('Laplacian on the Unit Disk');
set(gca,'fontsize',18);
```

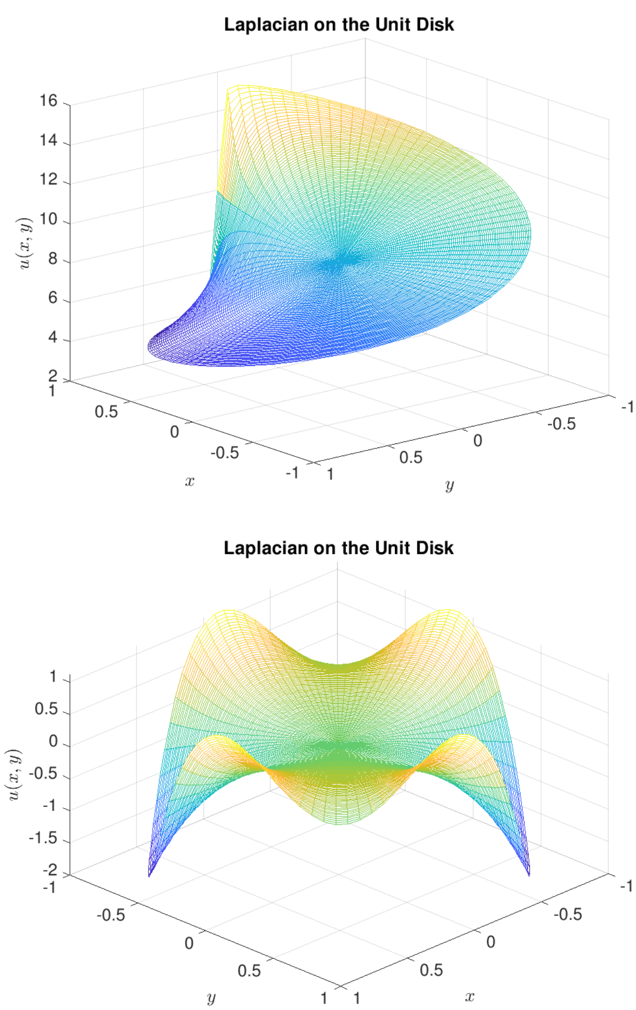


Figure 39

HOMEWORK 8

EXERCISE 1

Reconstruct the initial data for the Heat Equation given the following

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial t^2}, \quad x \in \mathbb{R}, \quad t > 0$$
$$u\Big|_{t=0} = f,$$
$$\{u(x_n, 1)\}_{n=1}^{N}$$

for the following initial conditions:

(a) $f(x) = 3x$

(b) $f(x) = \cos(x) + \sin(x)$

(c) $f(x) = e^{-x^2}$

(d) $f(x) = sinc(x)$

(e) $f(x) = humps\ function$

Further, for each initial condition analyze what happens as you vary the the number of grid points for $N = 10, 20, 40, 80$. Comment on what you observe.

Solution: In order to reconstruct the initial data for the Heat Equation for any case, we need the minimum-norm solution. However, to consider such solutions, it would be useful to briefly discuss optimization and the difference between over-determined and under-determined linear systems.

1 *Optimization*

We first encountered optimization problems in Calculus I, and then again in Calculus III. Optimization is essential because there are many real world problems where the optimum solution is required. Using the least amount of material during construction, utilizing the least amount of material used in an organic synthesis, or using Lagrange multipliers for constrained optimization[6] are all examples of optimization. However, optimization is best discussed in the context of linear algebra.

1.1 *Optimization in Linear Algebra*

[6] In context, we can use Lagrange multipliers in optimizing the action, the difference between the potential and kinetic energies, from which the equations of motion can be derived. I've finally found a connection. . .

Optimization in the context of linear algebra is the best tool to use in order to reconstruct the initial data. To begin, let us define the space that we are working with, $(C^\infty[-a,a], L^2)$ $(a = \pi)$, which is a Hilbert Space[7]. The importance of specifying that it is a Hilbert Space allows us to use the *Riesz Representation Theorem*, which is as follows:

> **Theorem 1.1.1.** If T is a bounded linear functional on a Hilbert space H then there exists some $a \in H$ such that for every $x \in H$ we have
>
> $$T(x) = \langle a, x \rangle.$$
>
> Moreover, $|T| = |a|$, where $|T|$ is the operator norm of T and $|a|$ is the L^2-norm of a.

Before working with $(C^\infty[-a,a], L^2)$, which is infinite-dimensional, let us consider the finite-dimensional case, specifically $\mathbb{R}^2$. Let us define a linear functional $F : \mathbb{R}^2 \to \mathbb{R}$, such that $F(x) = \langle a, x \rangle$, where $a, x \in \mathbb{R}^2$, and a is called the *representer*. Linear systems can be over-determined or under-determined, which conveys how many solutions there are to the system[8]. Over-determined systems have more equations than unknowns and for under-determined systems, there are more unknowns than equations. In either case, as long as the linear system is consistent, then for both over- and under-determined cases, the minimum-norm solution can be obtained. To find the minimum-norm solution of $F(x) = \langle a, x \rangle = a_1x_1 + a_2x_2 = c$, where $c \in \mathbb{R}$, which is under-determined, we know that we can represent the solution x as $x = \alpha a + \beta a^\perp$, the superposition of a scalar multiple of the representer, in the span of the subspace $S \subset \mathbb{R}^2$, and a scalar multiple of an element from the null-space of S, denoted $S^\perp$. Thus, to determine the minimum-norm solution, we apply the linear-functional F to our solution and obtain

$$\begin{aligned}
F(x) &= \langle a, x \rangle \\
&= \langle a, \alpha a + \beta a^\perp \rangle \\
&= \alpha \langle a, a \rangle + \beta \langle a, a^\perp \rangle \\
&= \alpha \langle a, a \rangle \\
&= c, \quad \text{where} \quad \alpha = \frac{c}{|a|^2}.
\end{aligned}$$

Now, we can take the norm of x and determine the conditions for which the minimum-norm solution is obtained:

[7] Specifically, we have a real Hilbert Space. Hilbert spaces are inner-product spaces that are complete with respect to a metric, which, in this case is the L^2-norm.

[8] For the case where there is full rank, there is exactly one solution

$$
\begin{aligned}
|x|^2 &= \langle x, x\rangle \\
&= \left\langle \frac{c}{|a|^2}a + \beta a^{\perp}, \frac{c}{|a|^2}a + \beta a^{\perp} \right\rangle \\
&= \frac{c^2}{|a|^2} + \beta^2 \langle a^{\perp}, a^{\perp}\rangle .
\end{aligned}
$$

Thus, our minimum-norm solution is obtained by setting $\beta = 0$ to obtain $x = \frac{c}{|a|^2}a$. In extending this to $\mathbb{R}^n$, we take a set of linearly-independent representers $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, and define our linear functional $F : \mathbb{R}^n \to \mathbb{R}$, such that $F(x) = \langle \alpha, x\rangle$, where $\alpha, x \in \mathbb{R}^n$. To obtain the minimum-norm solution of $F_i(x) = \langle \alpha_i, x\rangle = c_i$, we apply the same logic. We assume that the solution is a linear-combination of the representers $x = \sum_{i=1}^n d_i \alpha_i$. In performing this inner-product, for the first representer, we obtain

$$
\begin{aligned}
F_1(x) &= \langle \alpha_1, x\rangle \\
&= \left\langle \alpha_1, \sum_{j=1}^n d_j \alpha_j \right\rangle \\
&= \sum_{j=1}^n d_j \langle \alpha_1, \alpha_j\rangle = c_1 .
\end{aligned}
$$

In performing this for $i = 2 \ldots n$ the remaining representers α_i, we obtain a matrix-vector system $Gd = c$, where G is the Gram Matrix, d are the unknown coefficients, and c is the vector of known data points. The following system is

$$
\begin{bmatrix}
\langle \alpha_1, \alpha_1\rangle & \langle \alpha_1, \alpha_2\rangle & \cdots & \langle \alpha_1, \alpha_n\rangle \\
\langle \alpha_2, \alpha_1\rangle & \langle \alpha_2, \alpha_2\rangle & \cdots & \langle \alpha_2, \alpha_n\rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle \alpha_n, \alpha_1\rangle & \langle \alpha_n, \alpha_2\rangle & \cdots & \langle \alpha_n, \alpha_n\rangle
\end{bmatrix}
\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix},
$$

which has an exact solution due to the linear-independence of the representers, and hence the positive-definiteness of G. Thus, G^{-1} exists and the system is exactly solved for d. The solution x omits the perpendicular parts because the coefficients for those elements is zero for the minimum-norm solution. This is also true for the infinite-dimensional case. However, as such, infinite-dimensional cases lead to under-determined systems (infinitely many unknowns) and there are many factors that can affect whether or not the reconstruction of the initial data is successful or not, which are to be duly explored.

1.2 *The Heat Equation Solution and Reconstruction of Initial Data*

The solution of the Heat Equation on the real line can be expressed as the convolution of the initial data f with the Heat Kernel

$$u(x,t) = \int_{-\infty}^{\infty} \frac{e^{\frac{-(x-y)^2}{4t}}}{\sqrt{4\pi t}} f(y)\ dy.$$

Thus, at $t = 1$, we have

$$u(x,1) = \int_{-\infty}^{\infty} \frac{e^{\frac{-(x-y)^2}{4}}}{\sqrt{4\pi}} f(y)\ dy.$$

For a set of discrete points $\{u(x_n, 1)\}_{n=1}^{n=N}$, we can write

$$u(x_n,1) = \int_{-\infty}^{\infty} \frac{e^{\frac{-(x-y)^2}{4}}}{\sqrt{4\pi}} f(y)\ dy = \left\langle \frac{1}{\sqrt{4\pi}} e^{\frac{-(x_n-x)^2}{4}}, f(x) \right\rangle = u_n,$$

where the representer is the Gaussian and our unknown is f. Further, from this point on, it will be useful to denote $g(x_n) = g_n$ to be the Gaussian evaluated at x_n. To determine the minimum-norm solution, we form the following Gram matrix-vector system

$$\begin{bmatrix} \langle g_1, g_1\rangle & \langle g_1, g_2\rangle & \cdots & \langle g_1, g_n\rangle \\ \langle g_2, g_1\rangle & \langle g_2, g_2\rangle & \cdots & \langle g_2, g_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle g_n, g_1\rangle & \langle g_n, g_2\rangle & \cdots & \langle g_n, g_n\rangle \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix},$$

where solving the system exactly produces the coefficients such that the approximation of the initial condition becomes the linear combination $f^{\star}(x) = \sum_{n=1}^{N} c_n g_n$. Here is the code and the plots.

```
%% Inverse Problem Heat Equation

% Forming a Symmetric Grid

L = pi;
N = 10;
h = (2*L)/N;
xx = -L:h:L;

% Heat Equation Solution on Grid

%f = @(x) 3*x;
%f = @(x) cos(x) + sin(x);
%f = @(x) exp(-x.^2); % Initial Data
f = @sinc;
%f = @humps;
t = 1;

u = zeros(size(xx));

```

```
for k = 1:length(xx)
    h = @(y) exp(-((xx(k)-y).^2)/(4*t))/sqrt(4*pi*t); % Representer
    u(k) = quadgk(@(y) h(y).*f(y),-L,L); % Coefficients (solution at t=1)
end

% Gram matrix

gram = zeros(length(xx),length(xx));

for ii = 1:length(xx)
    for jj = 1:length(xx)
        gram(ii,jj) = quadgk(@(y) exp(-((xx(ii)-y).^2)/(4*t))...
            .*exp((-(xx(jj)-y).^2)/(4*t)),-L,L)/(4*pi);
    end
end

% Minimum-norm solution

reps = zeros(length(xx),length(xx));

for n = 1:length(xx)
    reps(:,n) = exp((-(xx(n)-xx).^2)/(4*t))/sqrt(4*pi*t); % Matrix of ...
        representers as columns
end

c = linsolve(gram,u');

ff = reps*c;

figure
hold on
plot(xx,ff,'b-');
plot(xx,f(xx),'ro');
xlabel('x','interpreter','latex');
ylabel('y','interpreter','latex');
legend('Minimum-Norm Solution','Actual Initial ...
    Condition','interpreter','latex');
title('Inverse Problem for the Heat Equation: $f = ...
    3x$','interpreter','latex');
set(gca,'fontsize',18);
```

(a) $f = 3x$

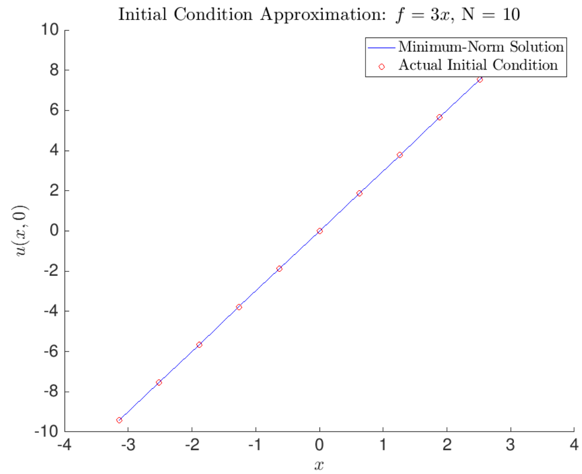
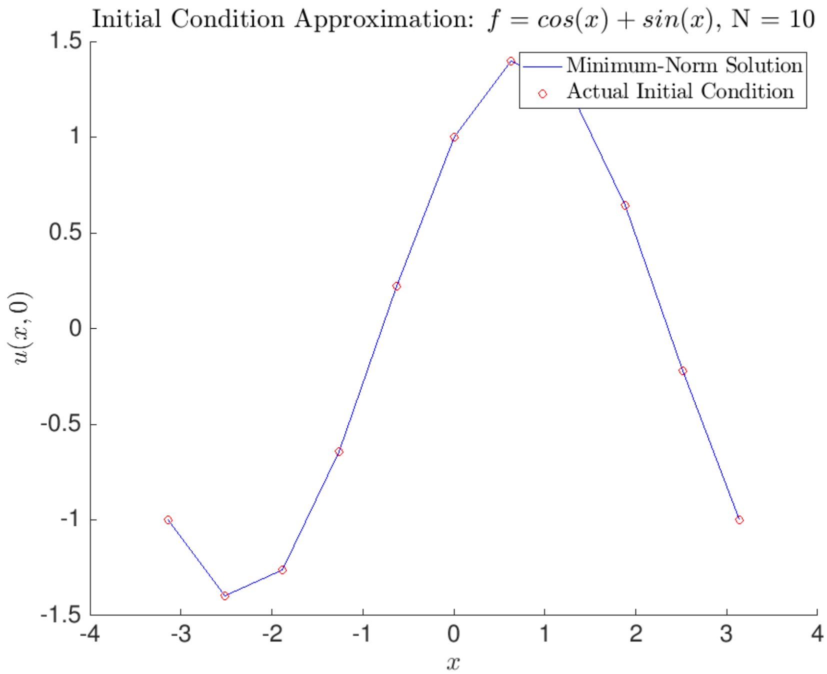


Figure 40: The graph for this initial data is the same for all values of N.

(b) $f = \cos(x) + \sin(x)$


Initial Condition Approximation: $f = cos(x) + sin(x)$, N = 10
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4

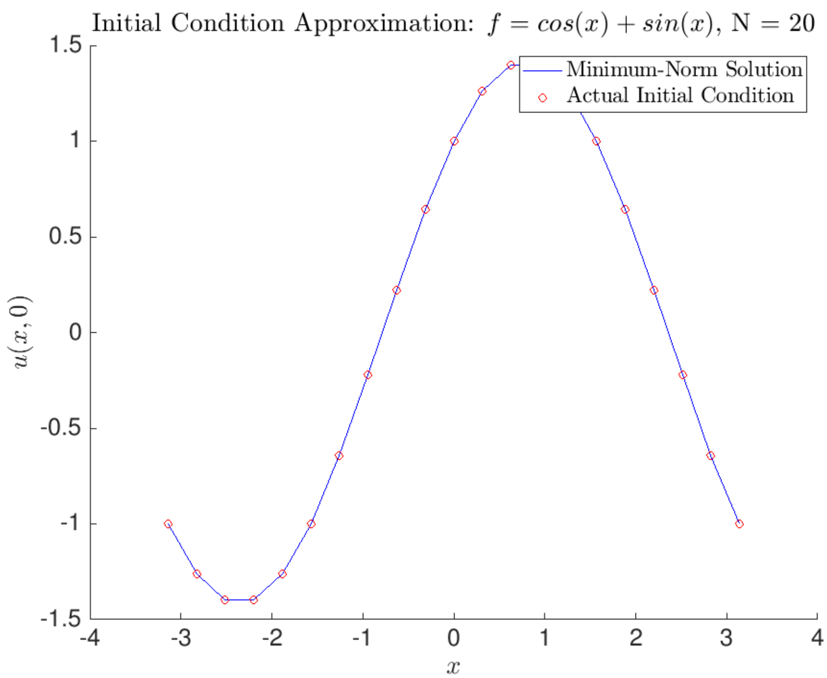

Initial Condition Approximation: $f = cos(x) + sin(x)$, N = 20
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x
1.5
1
0.5
0
-0.5
-1
-1.5
-4
-3
-2
-1
0
1
2
3
4

Figure 41

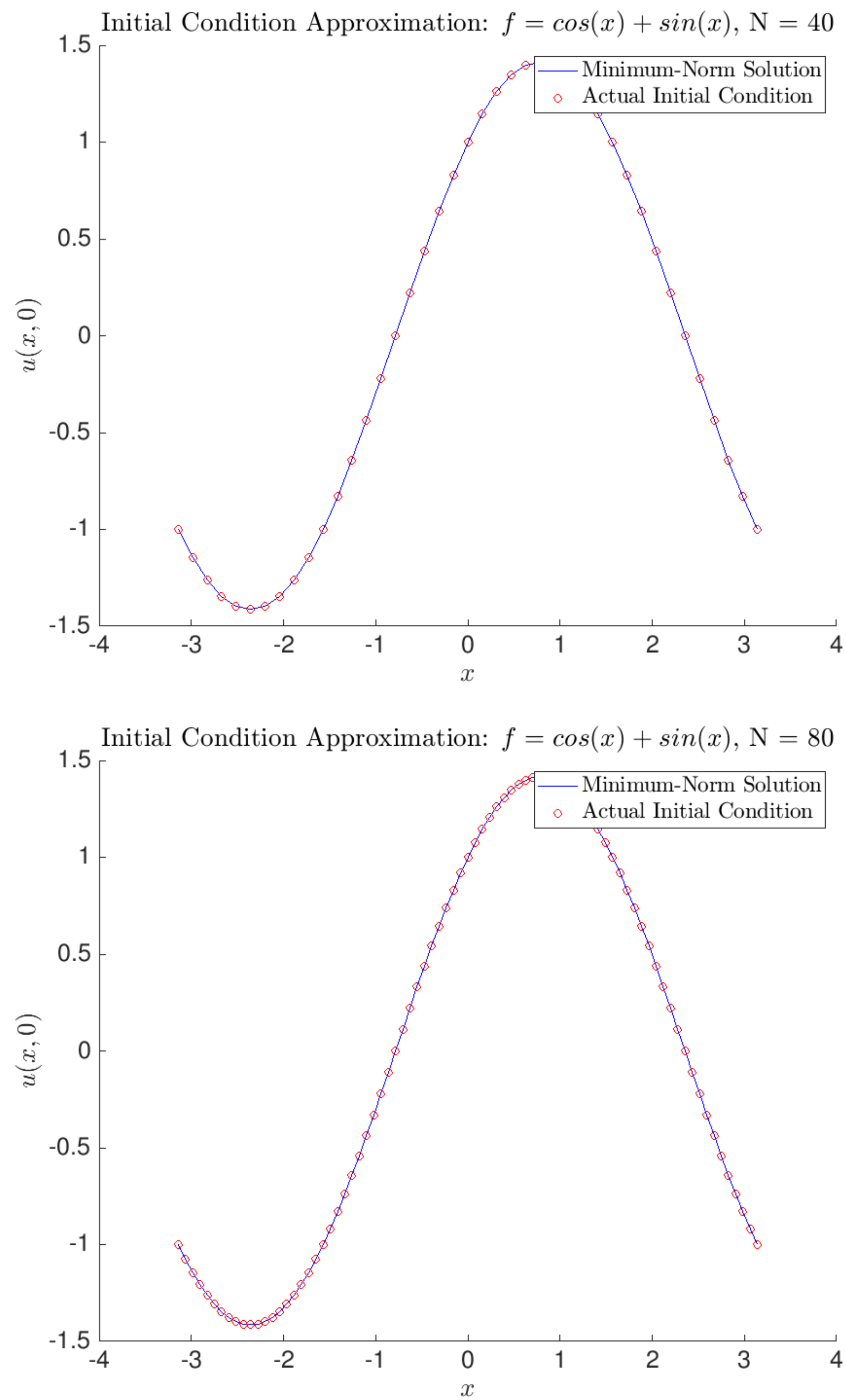
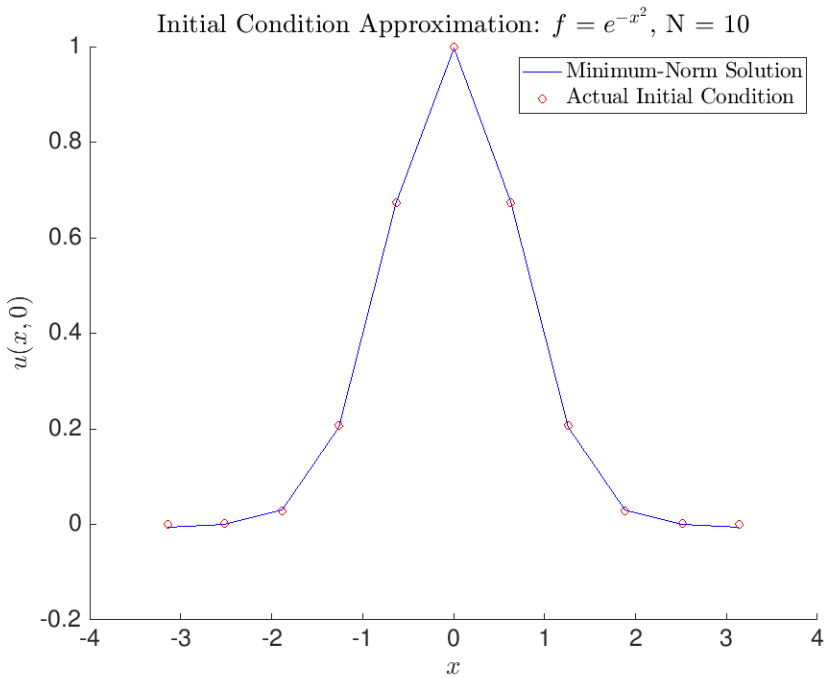

Initial Condition Approximation: $f = cos(x) + sin(x)$, N = 40
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x
Initial Condition Approximation: $f = cos(x) + sin(x)$, N = 80
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x

Figure 42

(c) $f = e^{-x^2}$



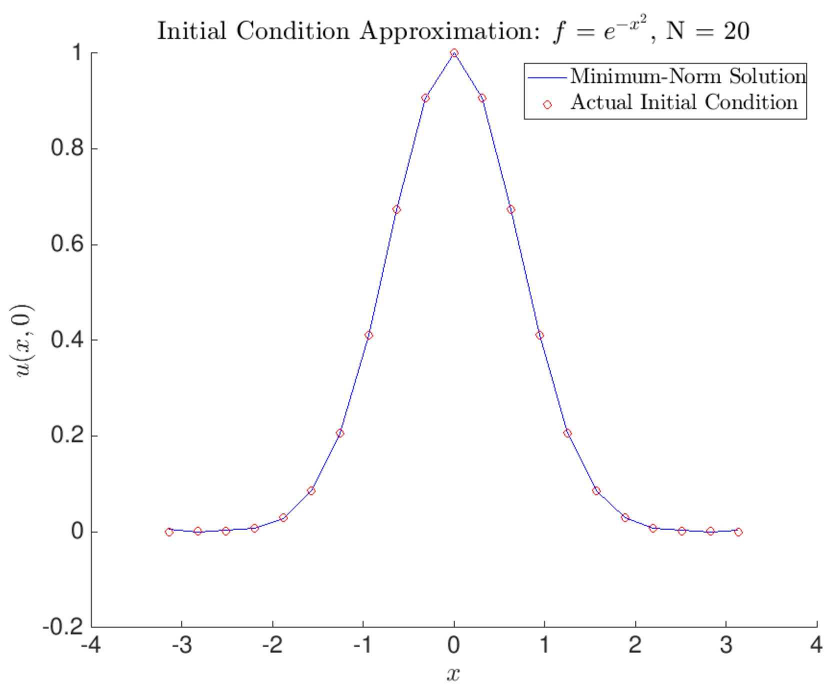
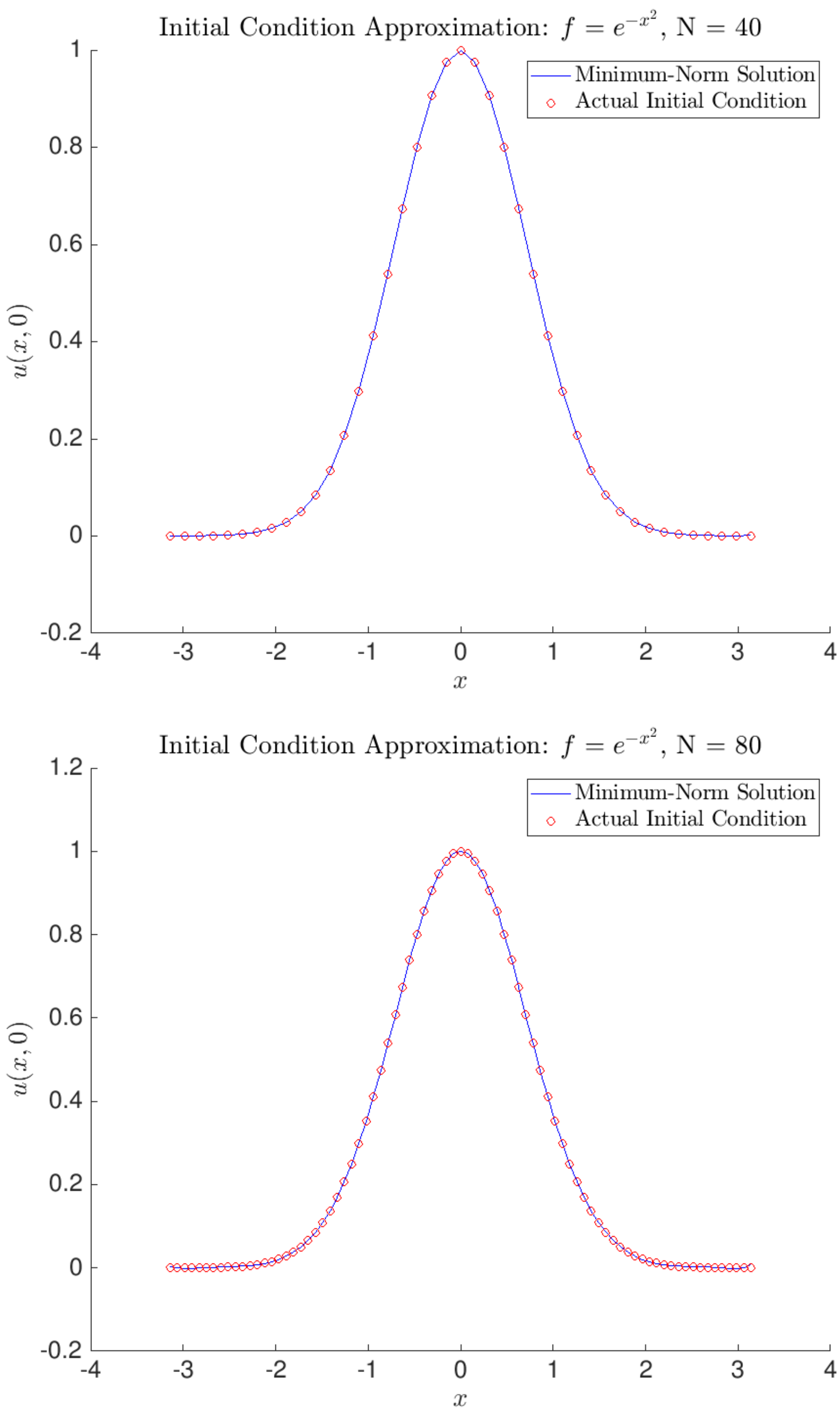


Figure 43



Initial Condition Approximation: $f = e^{-x^2}$, N = 40
Minimum-Norm Solution
Actual Initial Condition
$u(x, 0)$
x
Initial Condition Approximation: $f = e^{-x^2}$, N = 80
Minimum-Norm Solution
Actual Initial Condition
$u(x, 0)$
x

Figure 44

(d) $f = \text{sinc}(x)$

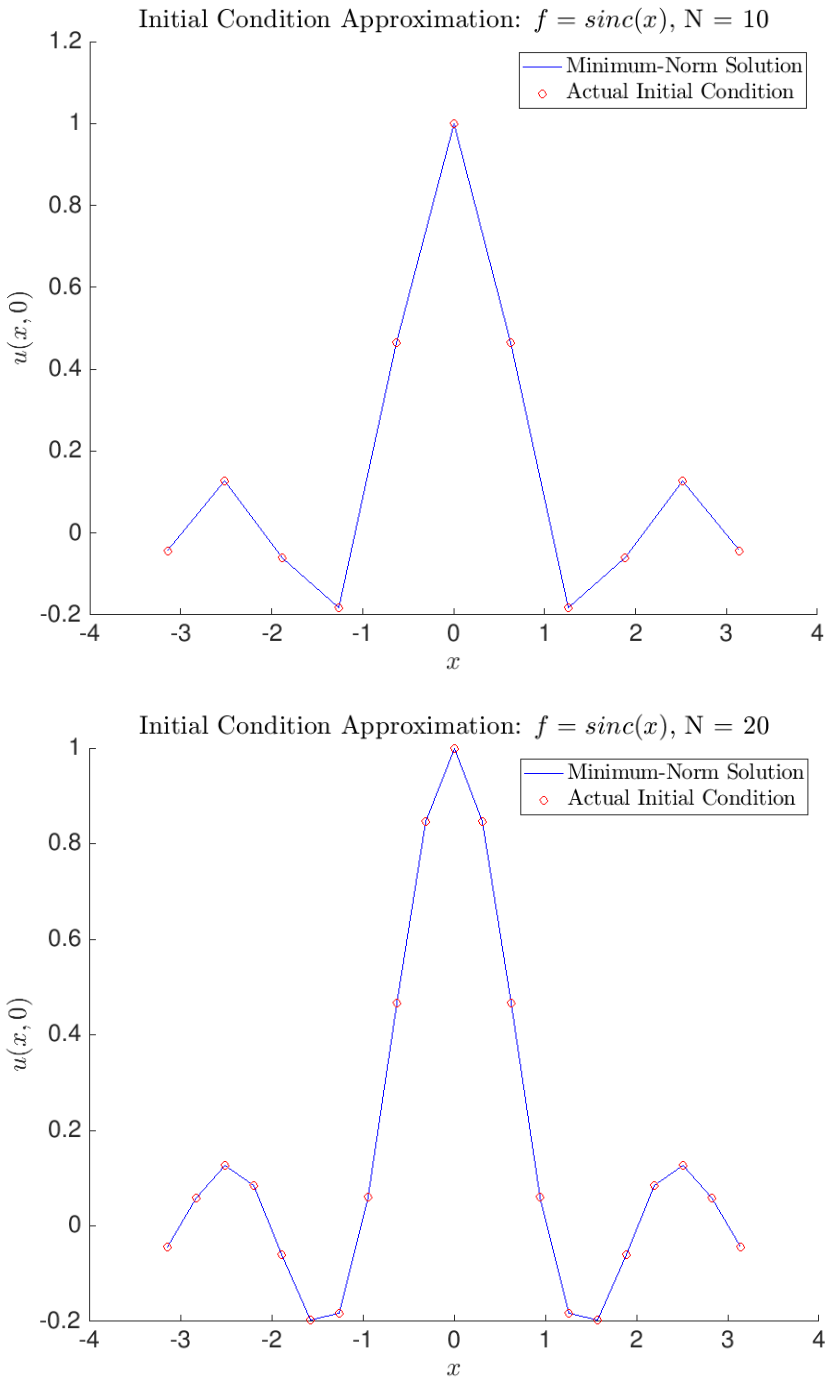
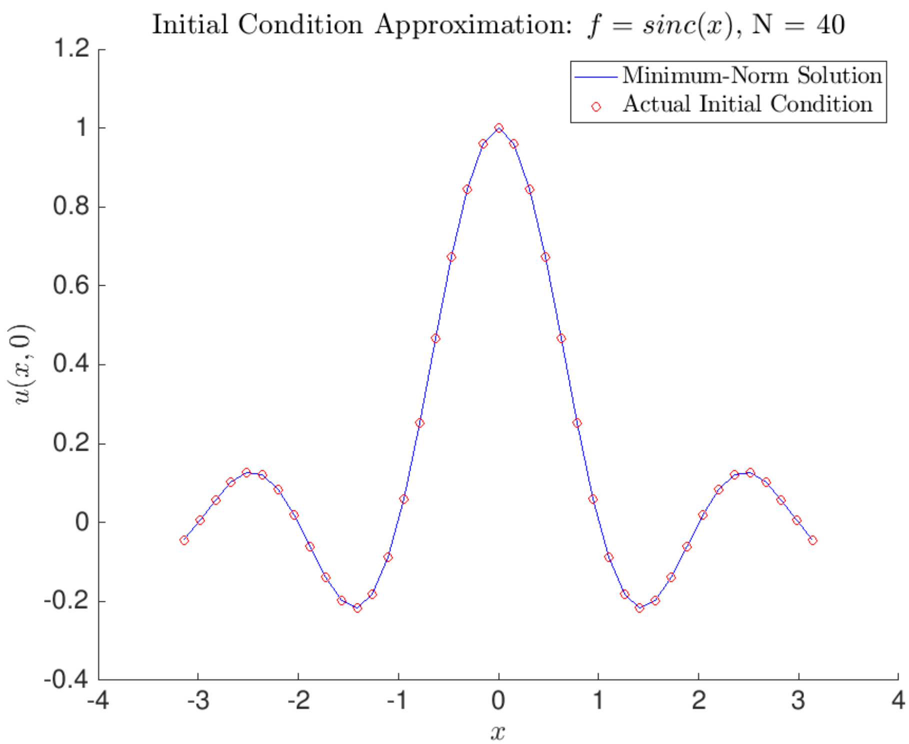


Figure 45


Initial Condition Approximation: $f = sinc(x)$, N = 40
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-4
-3
-2
-1
0
1
2
3
4

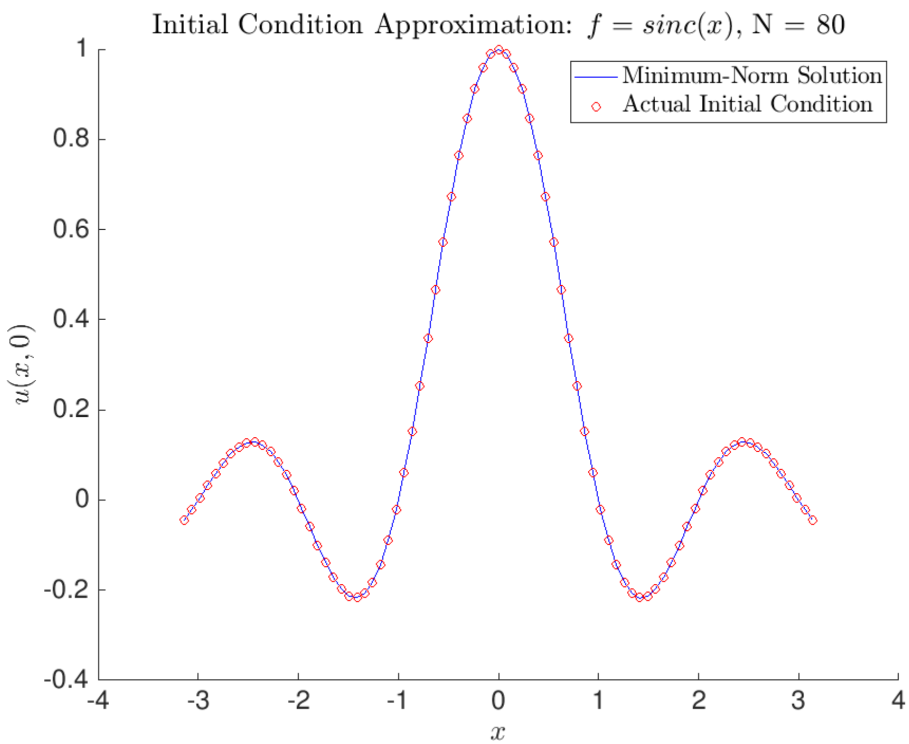
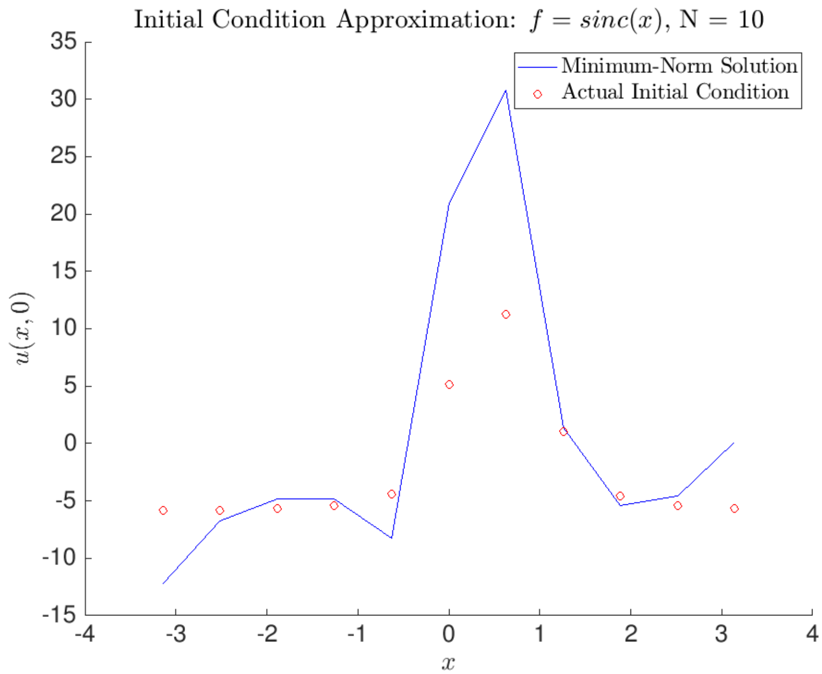

Initial Condition Approximation: $f = sinc(x)$, N = 80
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-4
-3
-2
-1
0
1
2
3
4

Figure 46

(e) $f = humps(x)$



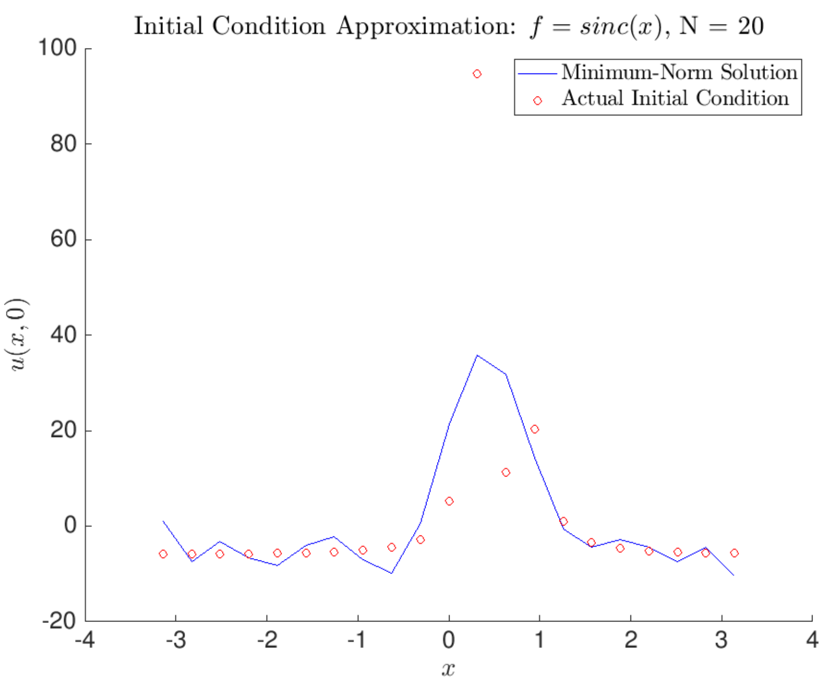


Figure 47

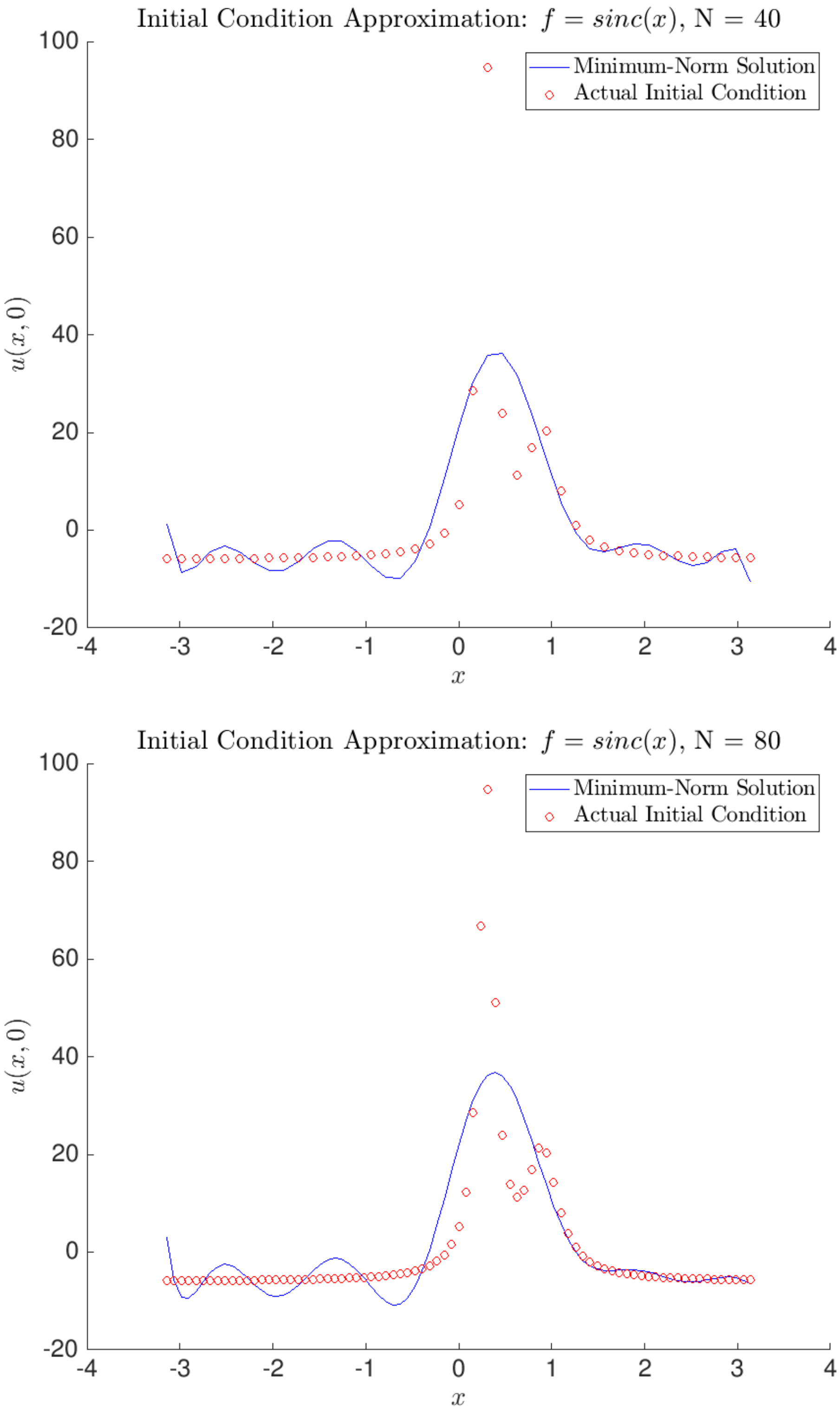

Initial Condition Approximation: $f = sinc(x)$, N = 40
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x
Initial Condition Approximation: $f = sinc(x)$, N = 80
Minimum-Norm Solution
Actual Initial Condition
$u(x,0)$
x

Figure 48

For the first four functions, the initial data is constructed exactly with the minimum-norm solution. However, for the last function, MATLAB's built-in `humps` function, the construction is not very good compared to the other functions. This is why I chose the humps function. Initially, I thought that my code wasn't working, but this case is one of the cases where the initial data can't be reconstructed accurately. I did some investigating to determine why. First, I thought that the magnitude of humps might have been to large, so I modified (b) and used $f = 100\cos(3x) - 90\sin(2x)$ as the initial data. The results are below.

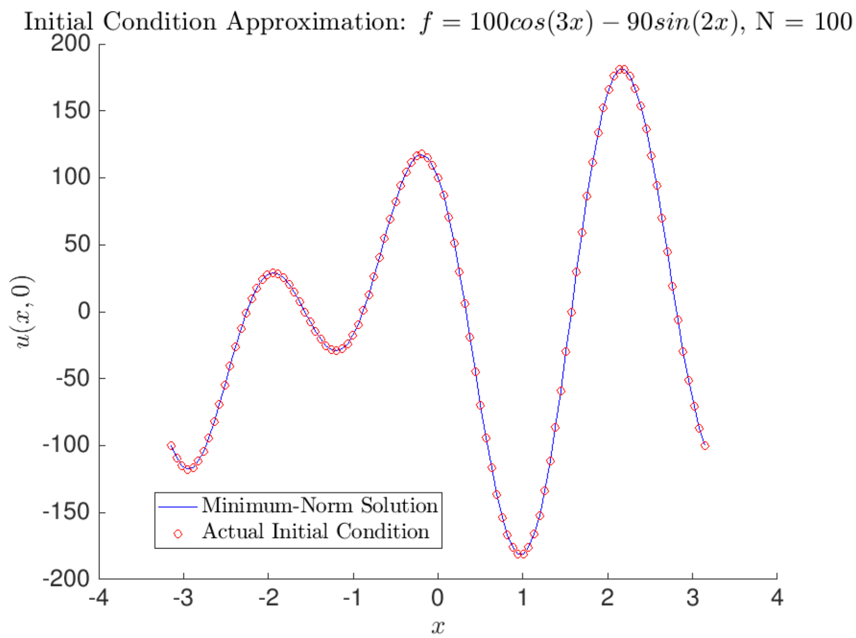


Figure 49

As we can see, the minimum-norm solution still perfectly matches the original solution, despite the large magnitude of the initial data. I further tried varying other parameters, like increasing the number of data points, varying the length of the interval, which did not make much of a difference. I feel this is an important result because it conveys how linear systems are not always perfectly solved, even with the best solution. I feel this is especially true for infinite-dimensional spaces. I also think that part of the reason is that we are using Gaussians to interpolate the initial data. Perhaps if other functions were used, the interpolation would be much bett.er Just for fun, I also plotted the minimum-norm solution for the `sign` and `sawtooth` built-in MATLAB functions. Their plots are on the following page.

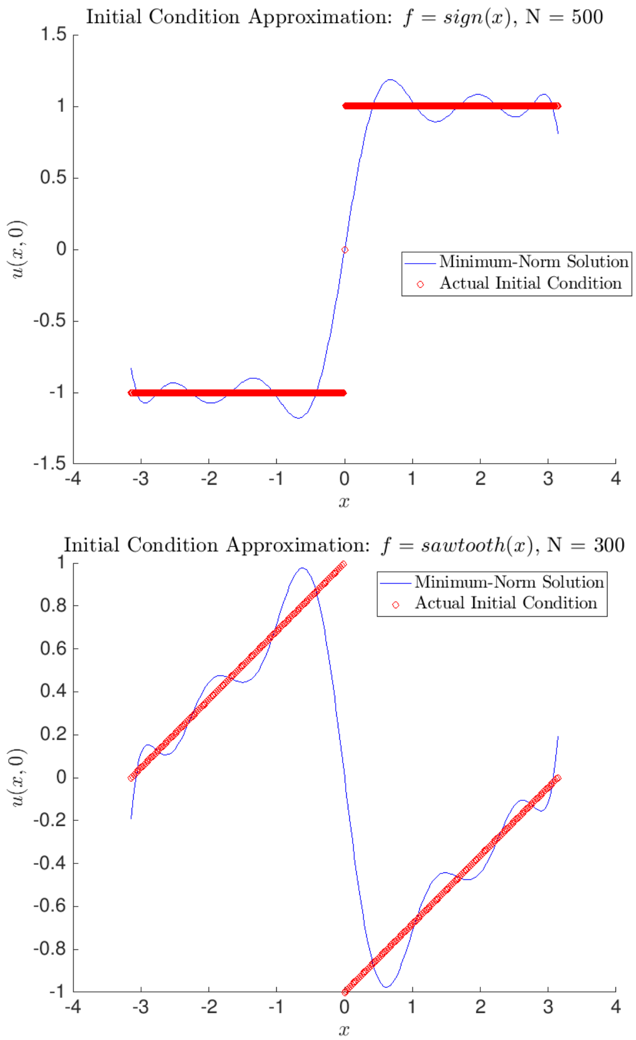


Figure 50

What's interesting to note about these minimum-norm solutions is that they exhibit Gibbs Phenomenon, which I find pretty intriguing. Even though the approximations of these initial conditions aren't great, they are still much better than the humps approximation.

Commentary: After the recent discussions that we've had, in using `pinv`, the results surprisingly stayed the same, despite using `linsolve` and `mldivide`. Of course I shouldn't assume that they would normally work and should default to using `pinv` for ill-condtioned matrices.