Mech 358 – Computer Lab 5

Problem 1

A comparison of the methods used to solve this problem is shown below in Figure 1. We can clearly see from the figure that the using the SOR method significantly increases the accuracy of the solution with the same number of iterations. The m-file used to solve this problem is shown below the figure.

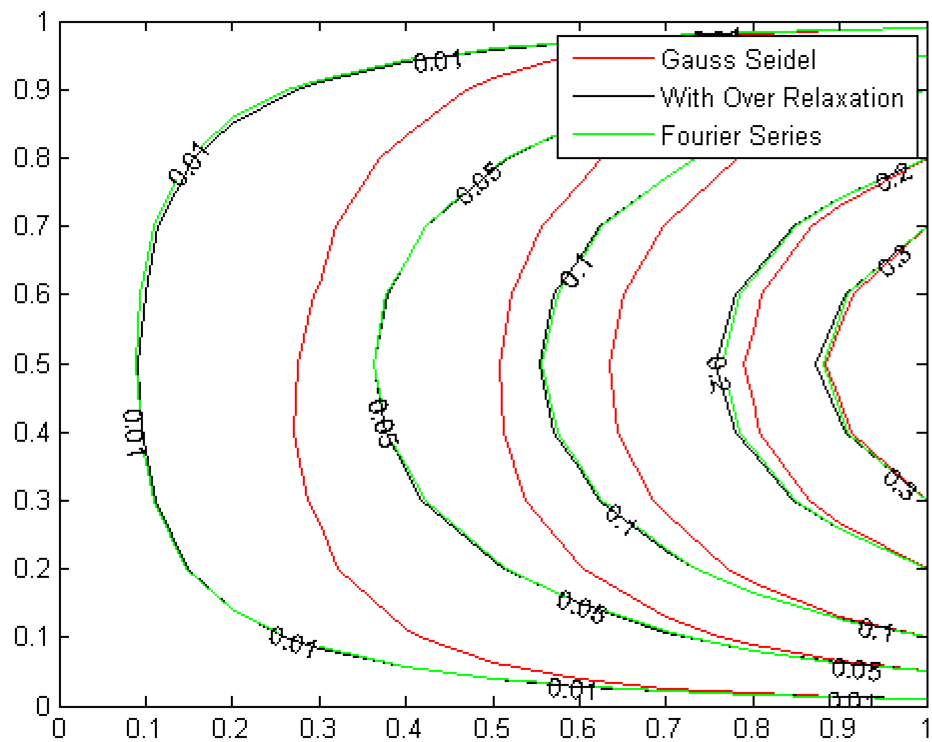

Figure 1 Comparison of the SOR method and the Gauss Seidel method

```
%this m-file will create the matrix that will solve the finite difference
%using Gauss seidel methods, it will also solve the problem with over
%relaxation and finally, solve the problem using the exact solution
%produced by forming a fourier series solution
clear all
close all
clc

%define the number of points in each row 9 interior points + 2 exterior
N = 11;
```

```
%set up our mesh
x = linspace(0,1,N);
y = linspace(0,1,N);




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%now for the guass seidel method

%create the phi matrix
phi_gauss = zeros(N,N);

%apply boundary conditions
%only non-zero terms occur at x = 1
%f(y) = y for 0< y <.5
phi_gauss(N-1,N) = y(2);
phi_gauss(N-2,N) = y(3);
phi_gauss(N-3,N) = y(4);
phi_gauss(N-4,N) = y(5);
phi_gauss(N-5,N) = y(6);

%f(y) = 1-y
phi_gauss(N-6,N) = 1-y(7);
phi_gauss(N-7,N) = 1-y(8);
phi_gauss(N-8,N) = 1-y(9);
phi_gauss(N-9,N) = 1-y(10);
phi_gauss(N-10,N) = 1-y(11);


%apply the gauss seidel method 15 times
for guass = 1:15

    %sweep from top to bottom
    for i = 1:N-2

        %sweep from left to right
        for j = 2:N-1

           phi_gauss(N-i,j) = 1/4*( phi_gauss(N-(i+1),j) + phi_gauss(N-(i-
1),j) + phi_gauss(N-i,j+1) + phi_gauss(N-i,j-1)   );

        end

    end




end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%now for the over relaxation method

%create the phi matrix
phi_relaxation = zeros(N,N);

%define a scalar to hold the value of phi to use in the relaxation method
phi_temp = 0;

%define a parameter for relaxation
w=1.5;

%apply boundary conditions
%only non-zero terms occur at x = 1
%f(y) = y for 0< y <.5
phi_relaxation(N-1,N) = y(2);
phi_relaxation(N-2,N) = y(3);
phi_relaxation(N-3,N) = y(4);
phi_relaxation(N-4,N) = y(5);
phi_relaxation(N-5,N) = y(6);

%f(y) = 1-y
phi_relaxation(N-6,N) = 1-y(7);
phi_relaxation(N-7,N) = 1-y(8);
phi_relaxation(N-8,N) = 1-y(9);
phi_relaxation(N-9,N) = 1-y(10);
phi_relaxation(N-10,N) = 1-y(11);


%apply the gauss seidel method 15 times
for guass = 1:15

    %sweep from top to bottom
    for i = 1:N-2

        %sweep from left to right
        for j = 2:N-1

           %find the next value and save it as a temporary value
           phi_temp = 1/4*( phi_relaxation(N-(i+1),j) + phi_relaxation(N-(i-
1),j) + phi_relaxation(N-i,j+1) + phi_relaxation(N-i,j-1)   );

           %now use the over-relaxtion method to find the next estimate
           phi_relaxation(N-i,j) = w*phi_temp +(1-w)*phi_relaxation(N-i,j)

        end

    end




end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%now for the exact solution using a fourier series

%create a meshgrid for x and y
[X,Y] = meshgrid(x,y)

%define a vector to hold the partial sums
sum = zeros(N,N)

%define a vector to temporarily hold the current partial sum
phi_fourier = zeros(N,N)

%sum the first 20 terms of the fourier series
for n = 1:20

    bn = 4/( (n^2*pi^2)*sinh(n*pi) ) * sin(n*pi/2);

    phi_fourier = bn*sin(n*pi*Y).*sinh(n*pi*X);

    sum = sum + phi_fourier;
end


%Plotting the contours
C=[0.01 0.05 0.1 0.2 0.3];

%create a for loop to plot it at each interval
for i = 1:5

    %plot the gauss seidel method
    contour(x,y,phi_gauss,C(i),'r');

    %keep the plot open
    hold on

    %plot the relaxation method
    contour(x,y,phi_relaxation,C(i),'k')

    %plot the fourier method in gree stars
    [a,b] = contour(x,y,sum,C(i),'g');

    %add the labels
     clabel(a,b);

end

%add a legend
legend('Gauss Seidel','With Over Relaxation','Fourier Series')
```

Problem 2

Two m-files were used to solve this problem. The first found the minimum temperature in the plate vs. T1 and plotted it. The second found the value of T1 that gave a minimum temperature in the plate of 200. It then created a contour plot of the plate. Figure 2 shows a plot of the minimum temperature vs. T1. Figure 3 shows the contour plot of the temperature in the plate. Figure 4 shows a surface plot of the temperature within the plate. All these figures show the solution when T1 = 286.5 (the final value of T1). The m-files are shown below the figures. The value found for T1 is shown below in Table 1.

Table 1 T1 vs. the Minimum Temperature in the Plate and the Percentage Error

T1	Minimum Temp	Error %
286.5	200.0249	0.0125

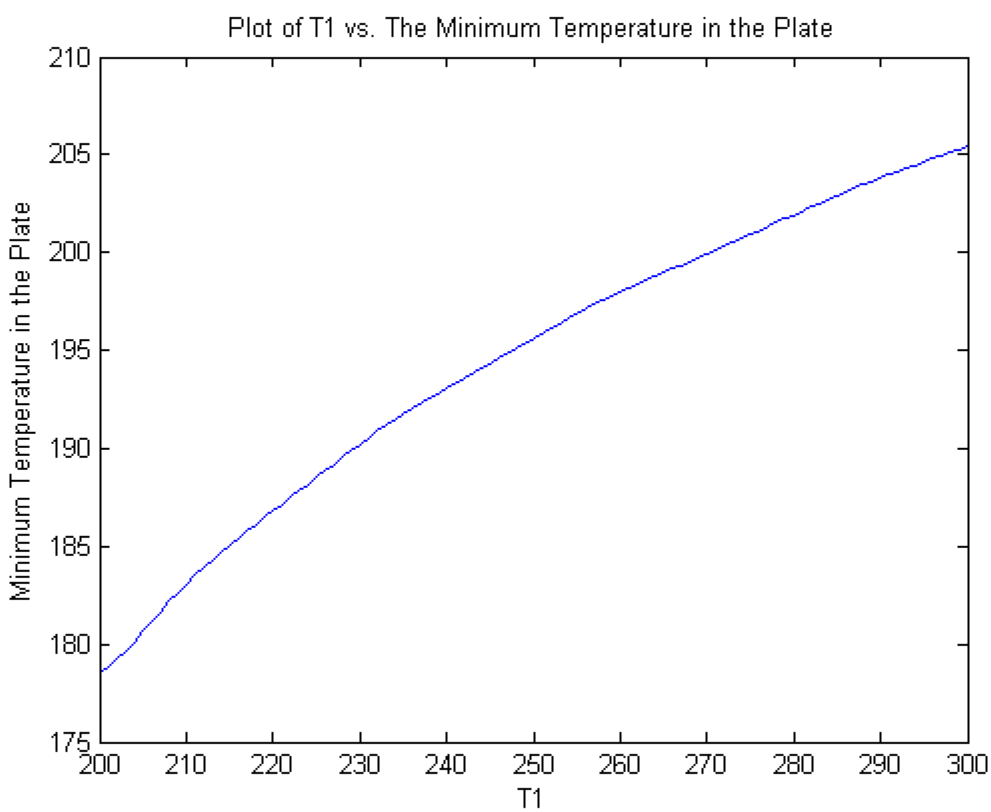

Figure 2 Plot of T1 vs. the Minimum Temperature in the Plate

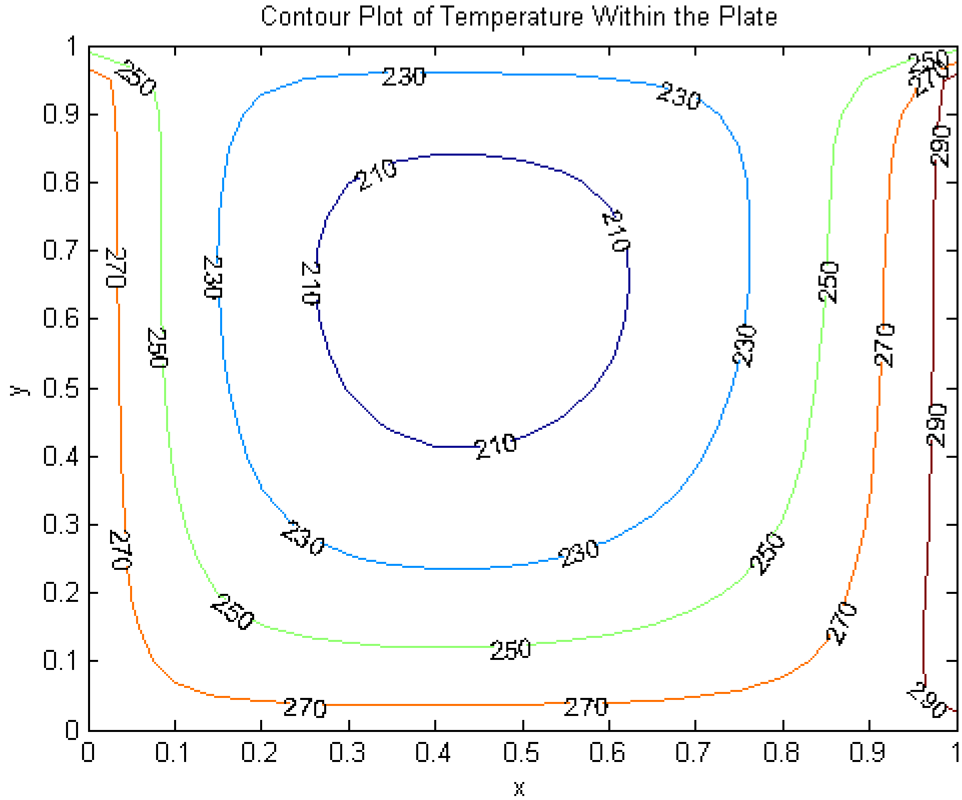

Figure 3 Contour Plot of the Temperature in the Plate

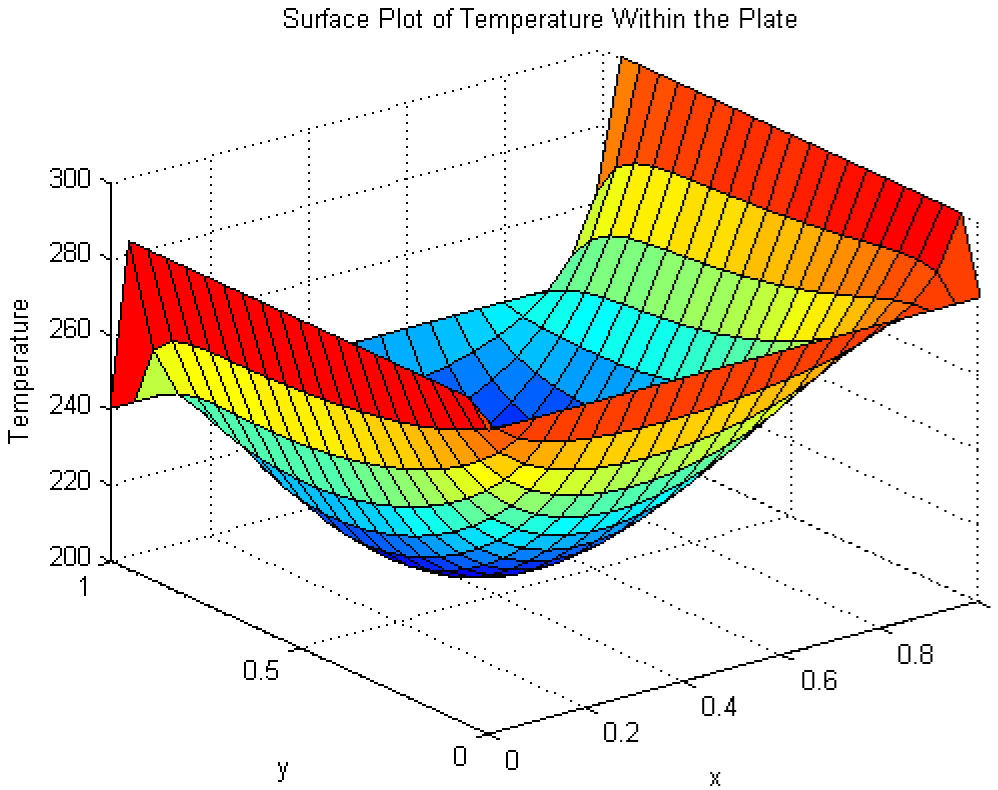

Figure 4 Surface plot of the Temperature in the plate

The first m-file plotted T1 vs. Tmin

```
%this m-file will create the matrix that will solve the finite difference
%using Gauss seidel methods and plot T1 vs Tmin
clear all
close all
clc

%define the number of points in each row 19 interior points + 2 exterior
N = 21;

%set up our mesh
x = linspace(0,1,N);
y = linspace(0,1,N);
h = 0.05;

%define a vector to save the minimum temperature
Tmin = zeros(1,101);

%define a counter i that will be used to increment the Tmin
counter = 1;

%define constants
```

```
Tinf = 100;
s = 10^-8;

%define heat transfer coeff.
U = 5.0./sqrt(0.1+x);

%Boundary conditions
T2 = 300;
T3 = 240;
T4 = 280;

%create 2 phi matrices to solve
phi_1 = zeros(N,N);

%apply know BC's
%T2
phi_1(1:N,N) = T2;

%T3
phi_1(N,1:N) = T3;

%T4
phi_1(1,1:N) = T4;



%run a for loop that finds the minimum temperature from T1=200 to T1=300
%guess a values for T1
for T1_guess1 = 200:300;

%apply b.c at T1
phi_1(1:N,1) = T1_guess1;

%solve each using the gauss seidel method
%initialize the relative error as 1 since it is checked after completing
%the first loop
rel_error1 = 1;


%apply the gauss seidel until sucessive rel error is less than 0.1%
while rel_error1 > 0.1/100

    %save the value at all the mesh points
    error_check_phi_1 = phi_1;

    %sweep from top to bottom
    for i = 1:N-2

        %sweep from left to right
        for j = 2:N-1

           %solve for phi 1
```

```
            phi_1(N-i,j) = 1/4*( phi_1(N-(i+1),j) + phi_1(N-(i-1),j) +
phi_1(N-i,j+1) + phi_1(N-i,j-1) ) - h^2/4*( U(j)*( phi_1(i,j) - Tinf) +
s*phi_1(i,j)^4 );


        end

    end

    %check the error at all the points and find the maximum
    error1 = abs((phi_1 -  error_check_phi_1)./(phi_1));
    rel_error1 = max(max(error1));


end

    %check the minimum values of each
    min_phi_1 = min(min(phi_1));

    %save this minimum temperature
    Tmin(counter) = min_phi_1;

    %increment 1 to i
    counter = counter+1;



end





%plot T1 vs the Minimum Temperature
T1 = 200:300;
plot(T1,Tmin)
hold on;

%add labels to the plot
xlabel('T1')
ylabel('Minimum Temperature in the Plate')
title('Plot of T1 vs. The Minimum Temperature in the Plate')
```

The second M-file found the minimum value of T1 using trial and error

```
%this m-file will create the matrix that will solve the finite difference
%using Gauss seidel methods
clear all
close all
clc
```

```
%define the number of points in each row 19 interior points + 2 exterior
N = 21;

%set up our mesh
x = linspace(0,1,N);
y = linspace(0,1,N);
h = 0.05;

[X,Y] = meshgrid(x,y);

%define constants
Tinf = 100;
s = 10^-8

%define heat transfer coeff.
U = 5.0./sqrt(0.1+x);

%Boundary conditions
T2 = 300;
T3 = 240;
T4 = 280;

%guess a value for T1 and use trial and error to find T1
T1_guess1 = 286.5;

%create 2 phi matrices to solve
phi_1 = zeros(N,N);

%apply boundary conditions
%T1
phi_1(1:N,1) = T1_guess1;

%T2
phi_1(1:N,N) = T2;

%T3
phi_1(N,1:N) = T3;

%T4
phi_1(1,1:N) = T4;

%solve using the gauss seidel method
%initialize the relative error as 1 since it is checked after completing
%the first loop
rel_error1 = 1;


%apply the gauss seidel until sucessive rel error is less than 0.1%
while rel_error1 > 0.1/100

    %save the value at all the mesh points
    error_check_phi_1 = phi_1;
```

```
    %sweep from top to bottom
    for i = 1:N-2

        %sweep from left to right
        for j = 2:N-1

           %solve for phi 1
           phi_1(N-i,j) = 1/4*( phi_1(N-(i+1),j) + phi_1(N-(i-1),j) +
phi_1(N-i,j+1) + phi_1(N-i,j-1) ) - h^2/4*( U(j)*( phi_1(i,j) - Tinf) +
s*phi_1(i,j)^4 );


        end

    end

    %check the error at all the points and find the maximum
    error1 = abs((phi_1 -  error_check_phi_1)./(phi_1));
    rel_error1 = max(max(error1));


end

%check the minimum value and display it
min_phi_1 = min(min(phi_1))


%Plotting the contours
C=[210 230 250 270 290];

%create a for loop to plot it at each interval
for i = 1:5

    %plot the gauss seidel method
    [a,b] = contour(X,Y,phi_1,[C(i) C(i)]);

    %keep the plot open
    hold on

    %add the labels
     clabel(a,b);

end


%add titles to the plot
xlabel('x')
ylabel('y')
title('Contour Plot of Temperature Within the Plate')
```

```
%create a surface plot as well in a new figure
figure
surf(X,Y,phi_1)

%add labels
xlabel('x')
ylabel('y')
zlabel('Temperature')
title('Surface Plot of Temperature Within the Plate')
```