TASK

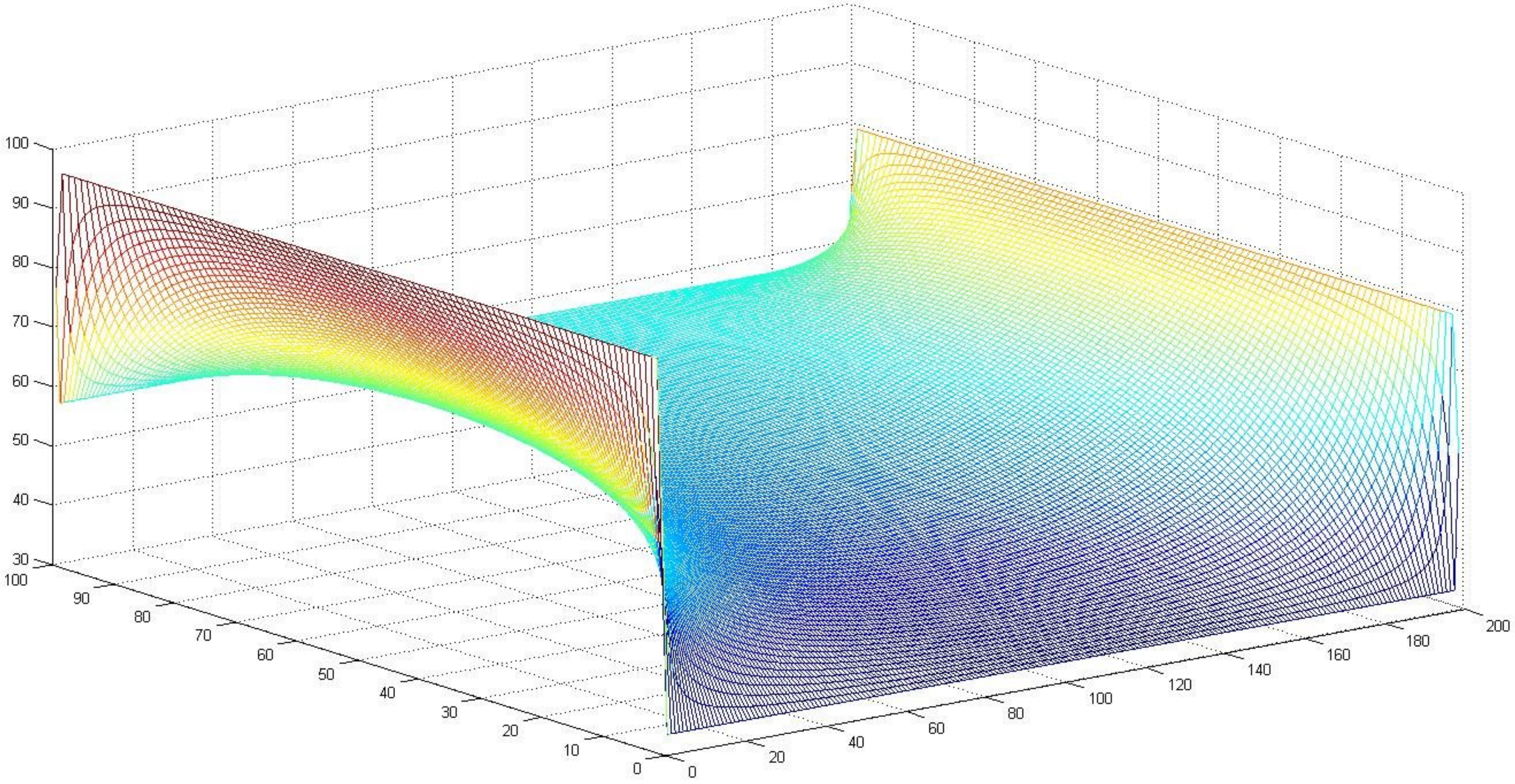
The temperature distribution in a thin metal plate with constant temperatures on each side can be modeled using a two-dimensional grid. The number of points in the grid is specified, as are the constant temperatures on the four sides. The temperature of an interior point equals the average of the four adjacent temperatures. Each time the temperature of an interior point changes, the temperatures of the adjacent points change. These changes continue until a thermal equilibrium is achieved and all temperatures become constant... Implement the calculation of the equilibrium temperature in the grid.

SOLUTION

The Matlab implementation and its output are displayed below.

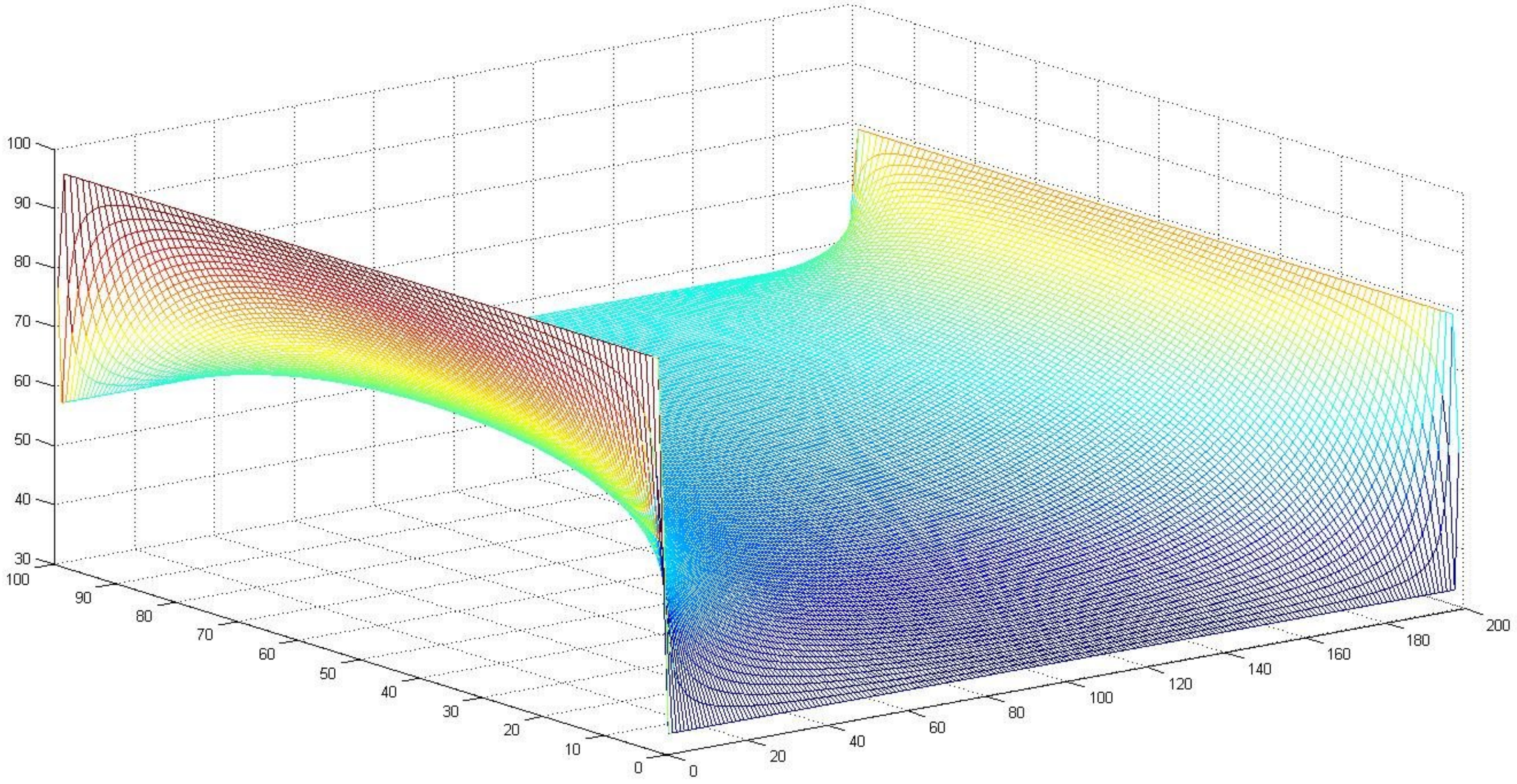
```
function [IterationsFirstMethod, IterationsSecondMethod, IterationsThirdMethod] = Temperature()

% There are three basic methods by which the temperature for each grid can
% be updated.
%
% Method 1: Use one grid to store the temperatures. When a point is updated,
% its new value is used to update the next point. Continue updating the
% points, moving along the rows until the temperature difference for all
% updates is less than a user-entered tolerance value.
%
% Method 2: Similar to method 1, except the values are updated by moving
% down the columns instead of along the rows.
%
% Method 3: Use two grids: one that calculates the temperatures based on
% the current values and one that stores the temperature values as they
% are calculated. Once the new temperatures are calculated completely for
% the grid, they replace the “old” temperatures in one swift move. Thus,
% all the temperatures are updated using one set of grid values. Stop
% calculating when the difference between the new and old temperatures is
% less than the user-defined tolerance

% Inputs.
x           = input('enter the width of the plate:        ');
y           = input('enter the height of the plate:       ');
c1          = input('enter the upper side temperature:    ');
c2          = input('enter the bottom side temperature:   ');
c3          = input('enter the right side temperature:    ');
```

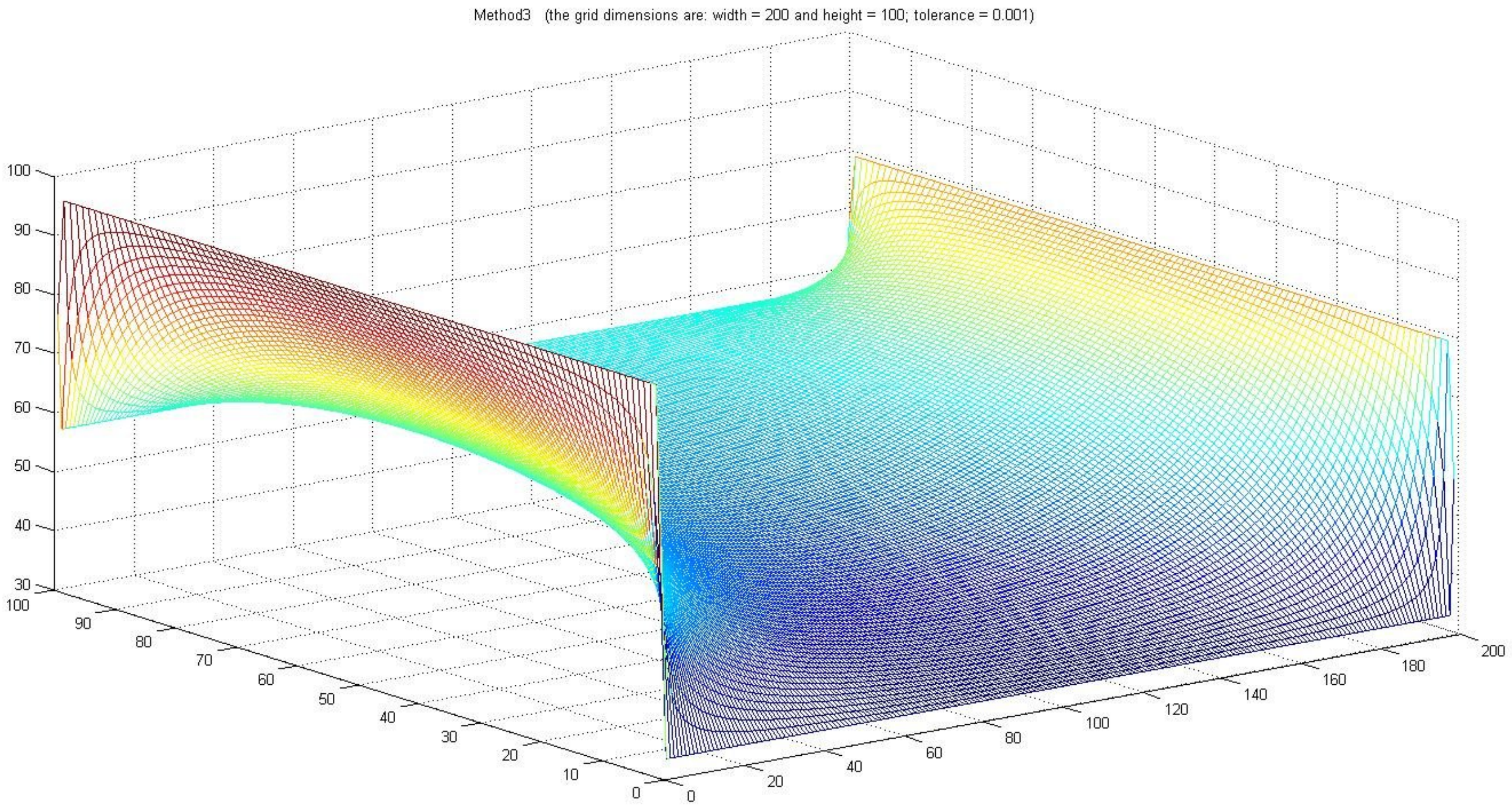
```
c4          = input('enter the left side temperature:     ');
tolerance   = input('enter the tolerance:                 ');

% Chosing one of the availabe methods.
method                  = input('Chose one: Method1,Method2,Method3:   ','s');
[IterationsFirstMethod] = TemperatureBalance(x,y,c1,c2,c3,c4,tolerance,method);

% Chosing another method.
method                  = input('Chose one: Method1,Method2,Method3:   ','s');
[IterationsSecondMethod]= TemperatureBalance(x,y,c1,c2,c3,c4,tolerance,method);

% Chosing another method.
method                  = input('Chose one: Method1,Method2,Method3:   ','s');
[IterationsThirdMethod] = TemperatureBalance(x,y,c1,c2,c3,c4,tolerance,method);

----------------------------------------------------------------------------------------------------------------------------------------------------------------

function [Iter,Temp] = TemperatureBalance(x,y,c1,c2,c3,c4,tolerance,method)

vx              = 1:x;
vy              = 1:y;
[XMat,YMat]     = meshgrid(vx,vy);
Temp            = XMat * 0;
OldTemp         = XMat * 0;

% Initialization
Temp(1,:)       = c1;
Temp(y,:)       = c2;
Temp(:,x)       = c3;
Temp(:,1)       = c4;
Temp(1,1)       = (c1 + c4)/2;
Temp(1,x)       = (c1 + c3)/2;
Temp(y,1)       = (c2 + c4)/2;
Temp(y,x)       = (c2 + c3)/2;

% Iterations
Iter = 0;
MaxDicrep = max(max(Temp - OldTemp));
while MaxDicrep > tolerance
    OldTemp = Temp;
    Iter    = Iter + 1;
    switch method
        case 'Method1'
            for i = 2:(y-1)
                for j = 2:(x-1)
                    Temp(i,j) = ( Temp(i,j-1) + Temp(i,j+1) + Temp(i-1,j) + Temp(i+1,j) ) / 4;
                end
            end

        case 'Method2'
            for j = 2:(x-1)
                for i = 2:(y-1)
                    Temp(i,j) = ( Temp(i,j-1) + Temp(i,j+1) + Temp(i-1,j) + Temp(i+1,j) ) / 4;
                end
            end

        case 'Method3'
            for i = 2:(y-1)
                for j = 2:(x-1)
                    Temp(i,j) = ( OldTemp(i,j-1) + OldTemp(i,j+1) + OldTemp(i-1,j) + OldTemp(i+1,j) ) / 4;
                end
            end
    end

    MaxDicrep = max(max(Temp - OldTemp));
end

mesh(XMat,YMat,Temp);
title([method,'   (the grid dimensions are: width = ', num2str(x), ' and height = ', num2str(y), '; tolerance = ',
num2str(tolerance), ')']);
```



Method1 (the grid dimensions are: width = 200 and height = 100; tolerance = 0.001)

Method2 (the grid dimensions are: width = 200 and height = 100; tolerance = 0.001)




Method3 (the grid dimensions are: width = 200 and height = 100; tolerance = 0.001)
100
90
80
70
60
50
40
30
100
90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100
120
140
160
180
200