HOMEWORK-3

HAZIRLAYAN

ADI SOYADI : ENES ŞAHINER

ÖĞRENCI NUMARASI : 030200018

TESLIM TARIHI : 15.04.2024

DERS ADI : COMPUTATIONAL FLUID DYNAMICS

DERS YÜRÜTÜCÜSÜ : SERTAÇ ÇADIRCI

Table of Contents

Part-1, Unsteady Heat Conduction

This part deals with the two-dimensional transient heat conduction problem in a 3x1 rectangular plate. The plate is initially assumed to be at room temperature (18°C). At time t=0, the temperature of the bottom boundary is suddenly raised to 150°C. The aim is to estimate the time required for the center of the block to reach a temperature of 30°C.Also, the temperature distribution should be shown in color at equilibrium condition. Different computational domains such as 30x10, 90x30 and 180x60 should be used for a cell size of b=10 cm and the results should be compared for each grid. The implicit ADI method should be used to obtain numerical solutions for the multidimensional parabolic PDE.

Preparing the Equation

$$\frac{\partial T}{\partial t}=\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}$$

If the equation is discrete,

$$\frac{\partial T}{\partial t}=\frac{T_{i,j}^{n+\frac{1}{2}}-T_{i,j}^{n}}{\Delta t}$$

$$\frac{\partial^2 T}{\partial x^2}=\frac{T_{i+1,j}^{n+\frac{1}{2}}-2*T_{i,j}^{n+\frac{1}{2}}+T_{i-1,j}^{n+\frac{1}{2}}}{(\Delta x)^2}$$

$$\frac{\partial^2 T}{\partial y^2}=\frac{T_{i,j+1}^{n}-2*T_{i,j}^{n}+T_{i,j-1}^{n}}{(\Delta y)^2}$$

If they are substituted into the equation,

$$\frac{T_{i,j}^{n+\frac{1}{2}}-T_{i,j}^{n}}{\Delta t}=\frac{T_{i+1,j}^{n+\frac{1}{2}}-2*T_{i,j}^{n+\frac{1}{2}}+T_{i-1,j}^{n+\frac{1}{2}}}{(\Delta x)^2}+\frac{T_{i,j+1}^{n}-2*T_{i,j}^{n}+T_{i,j-1}^{n}}{(\Delta y)^2}$$

Step-1

If the equation is put into TDMA format,

$$b_i * T_{i+1,j}^{n+\frac{1}{2}} + a_i * T_{i,j}^{n+\frac{1}{2}} + c_i * T_{i-1,j}^{n+\frac{1}{2}} = d_i$$

Here,

$$b_i = c_i = \frac{\Delta t}{2 * (\Delta x)^2}$$

$$a_i = -\left(1 + \frac{\Delta t}{(\Delta x)^2}\right)$$

$$d_i = -T_{i,j}^n - \frac{\Delta t}{2 * (\Delta y)^2} * \left(T_{i,j+1}^n - 2 * T_{i,j}^n + T_{i,j-1}^n\right)$$

Here, the prosses is: first put j =1 and i =1, 2, …., N giving a system of N equations which are solved by any method. The same is continued with j =2, j =3, …., j=M. Thus a system of N equations is solved M times.

j = 1	i = 1, 2, 3, …., N
j = 2	i = 1, 2, 3, …., N
…	…
…	…
j = M	i = 1, 2, 3, …., N

Step-2

If the equation is put into TDMA format,

$$b_j * T_{i,j+1}^{n+\frac{1}{2}} + a_j * T_{i,j}^{n+\frac{1}{2}} + c_j * T_{i,j-1}^{n+\frac{1}{2}} = d_j$$

Here,

$$b_j = c_j = \frac{\Delta t}{2 * (\Delta x)^2}$$

$$a_j = -\left(1 + \frac{\Delta t}{(\Delta x)^2}\right)$$

$$d_j = -T_{i,j}^{n+\frac{1}{2}} - \frac{\Delta t}{2 * (\Delta y)^2} * \left(T_{i+1,j}^{n+\frac{1}{2}} - 2 * T_{i,j}^{n+\frac{1}{2}} + T_{i-1,j}^{n+\frac{1}{2}}\right)$$

Here, in this stage the solution proceeds vertically i.e. first i =1, j =1, 2, …, M; i =2, j =1, 2, …, M and so on.

i = 1	j = 1, 2, 3, …., M
i = 2	j = 1, 2, 3, …., M
…	…
…	…
i = N	j = 1, 2, 3, …., M

Estimating Time for Block Center to Reach 30°C

In the table below, for each domain the time required for the temperature of the center of the block to reach 30°C.

30x10	4.6 second
90x30	6.1 second
180x60	6.5 second

Temperature Distributions

For 30x10 domain

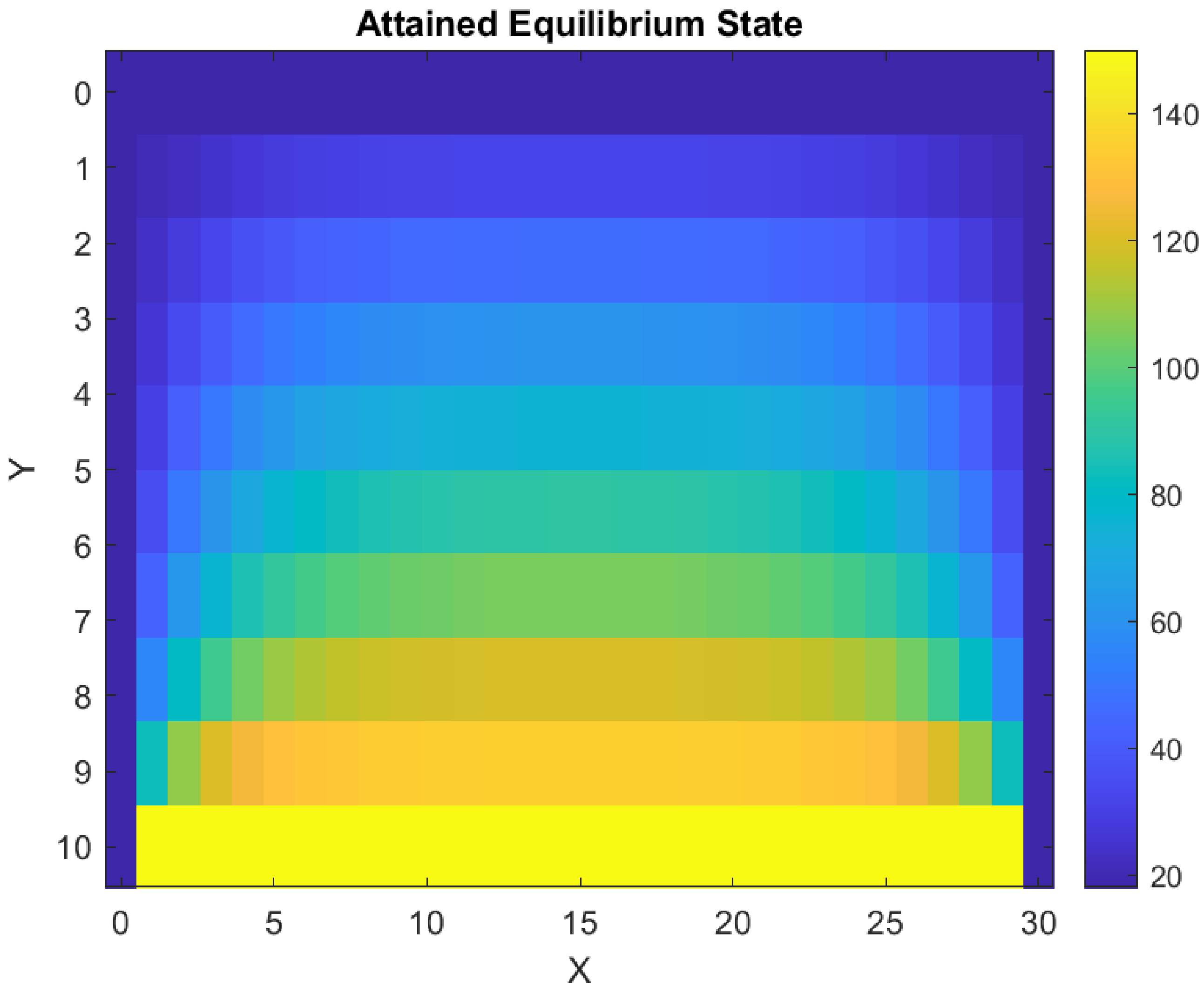


Figure 1. 30x10 domain temperature distribution.

For 90x30 domain

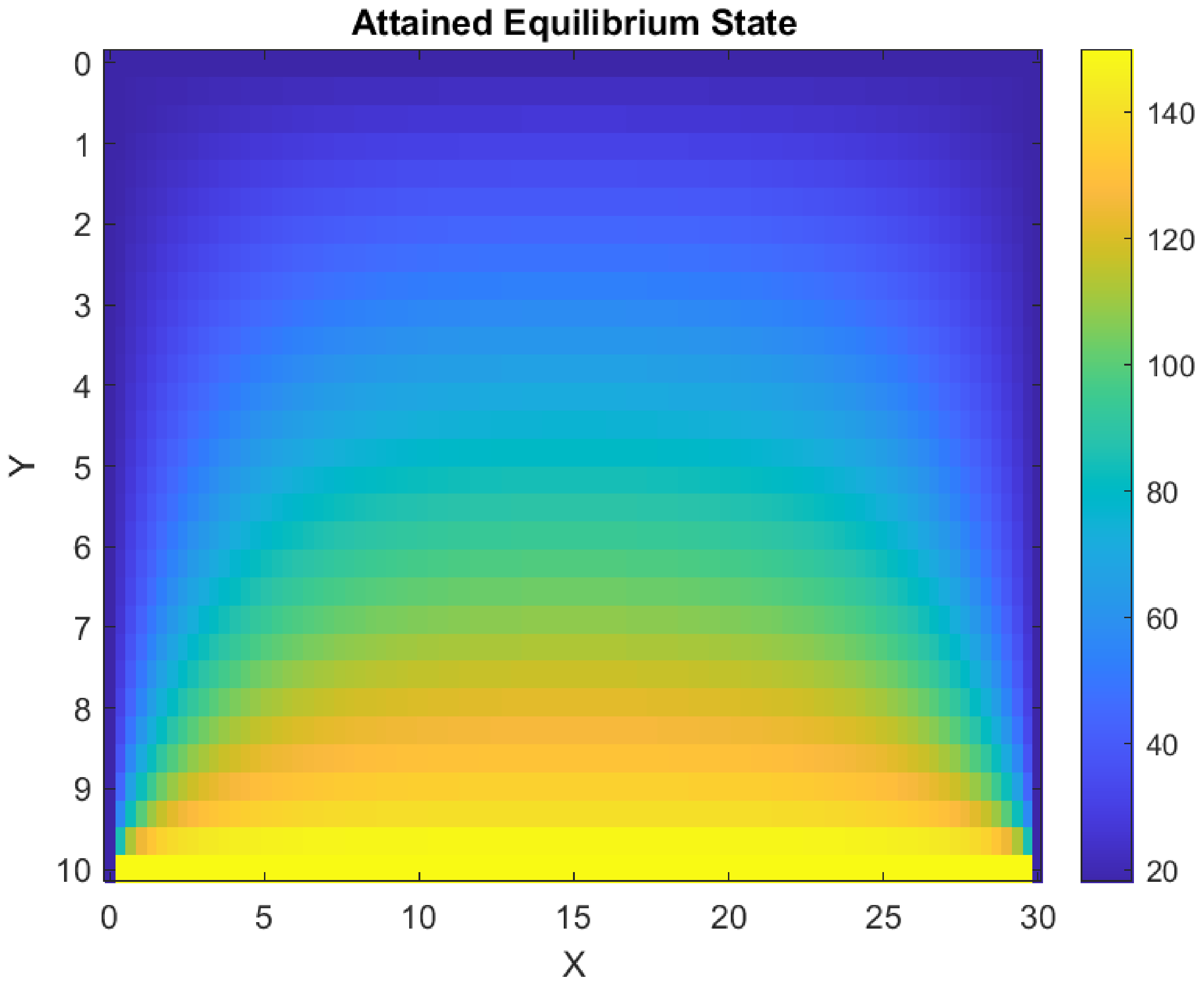


Figure 2. 90x30 domain temperature distribution.

For 180x60 domain

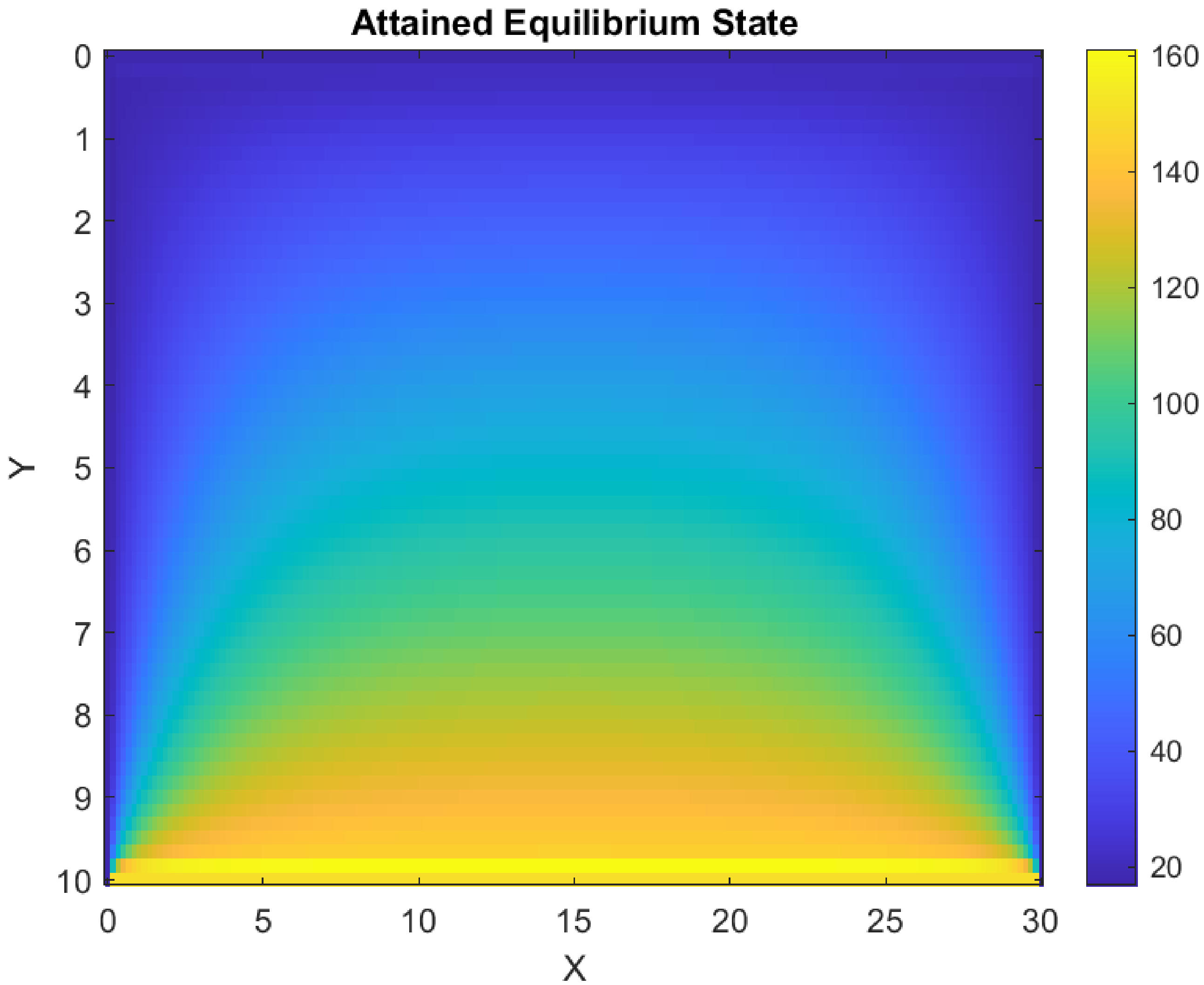


Figure 3. 180x60 domain temperature distribution.

Here, the time had to be reduced because the computer's RAM was not enough for the calculations. Therefore, there is an error at the bottom of the chart. Based on the experience obtained from the previous graphics, it can be said that this error will not occur when there is enough time.

Comments

As can be seen from the graphs, as the domain size increases, the resulting graph becomes smoother. Additionally, as the domain grows, it is necessary to make the time step smaller, otherwise some errors will appear on the graph.

Part-2, Steady Heat Conduction

This part deals with the steady-state two-dimensional heat conduction problem in a rectangular plate. The plate is assumed to have reached equilibrium, and the equation is an elliptic PDE. The bottom wall is subject to a temperature of 150°C and the other walls are at 18°C. The aim is to compare the convergence speed of three iterative schemes: Jacobi, Gauss-Seidel and Gauss-Seidel SOR (ω=1.6) for different computational domains. The 2D temperature surfaces should be shown for each scheme. The temperature of the center point of the block is used as an indicator to compare the convergence speed.

Preparing the Equation

$$0 = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

If the equation is discrete,

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1}^{j} - 2 * T_i^j + T_{i-1}^{j}}{h^2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_i^{j+1} - 2 * T_i^j + T_i^{j-1}}{k^2}$$

If they are substituted into the equation,

$$\frac{T_{i+1}^{j} - 2 * T_i^j + T_{i-1}^{j}}{h^2} + \frac{T_i^{j+1} - 2 * T_i^j + T_i^{j-1}}{k^2} = 0$$

Here,

$$\gamma = \left(\frac{h}{k}\right)^2$$

So, the equation is

$$T_{i+1}^{j} - 2 * T_{i}^{j} + T_{i-1}^{j} + \gamma * \left(T_{i}^{j+1} - 2 * T_{i}^{j} + T_{i}^{j-1}\right) = h^2 * f_{i,j}$$

$$T_{i+1}^{j} + T_{i-1}^{j} + \gamma * \left(T_{i}^{j+1} + T_{i}^{j-1}\right) + T_{i}^{j} * (-2 - 2 * \gamma) = h^2 * f_{i,j}$$

Discrete form of elliptic PDE which gives the unknown in the diagonal:

$$T_{i}^{j} = \frac{1}{2 + 2 * \gamma} * \left[T_{i+1}^{j} + T_{i-1}^{j} + \gamma * \left(T_{i}^{j+1} + T_{i}^{j-1}\right) - h^2 * f_{i,j}\right]$$

For Jacobi iteration:

$$T_{i,j}^{(k+1)} = \frac{1}{2 + 2 * \gamma} * \left[T_{i+1,j}^{(k)} + T_{i-1,j}^{(k)} + \gamma * \left(T_{i,j+1}^{(k)} + T_{i,j-1}^{(k)}\right) - h^2 * f_{i,j}\right]$$

For Gauss-Seidel iteration:

$$T_{i,j}^{(k+1)} = \frac{1}{2 + 2 * \gamma} * \left[T_{i+1,j}^{(k)} + T_{i-1,j}^{(k+1)} + \gamma * \left(T_{i,j+1}^{(k)} + T_{i,j-1}^{(k+1)}\right) - h^2 * f_{i,j}\right]$$

For Gauss-Seidel SOR iteration:

$$T_{i,j}^{(k+1)} = (1 - w) * T_{i,j}^{(k)} + \frac{w}{2 + 2 * \gamma} * \left[T_{i+1,j}^{(k)} + T_{i-1,j}^{(k+1)} + \gamma * \left(T_{i,j+1}^{(k)} + T_{i,j-1}^{(k+1)}\right) - h^2 * f_{i,j}\right]$$

In homework file, "w" equals to 1.6. If w is between 1 and 2, it means over relaxation. It is best for linear problems.

Temperature Distribution

30x10 domain for each method

<u>Jacobi iteration:</u>

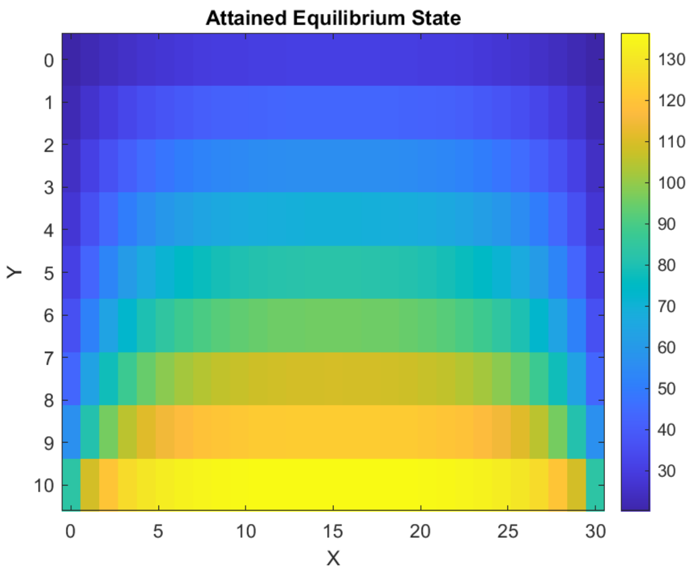


Figure 4. 30x10 domain, Jacobi Iteration Temperature Distribution

```
Temperature at center node converged to: 82.368323
Number of iterations: 1
```

Figure 5. 30x10 domain, Jacobi Iteration, converged temperature

<u>Gauss-Seidel iteration:</u>

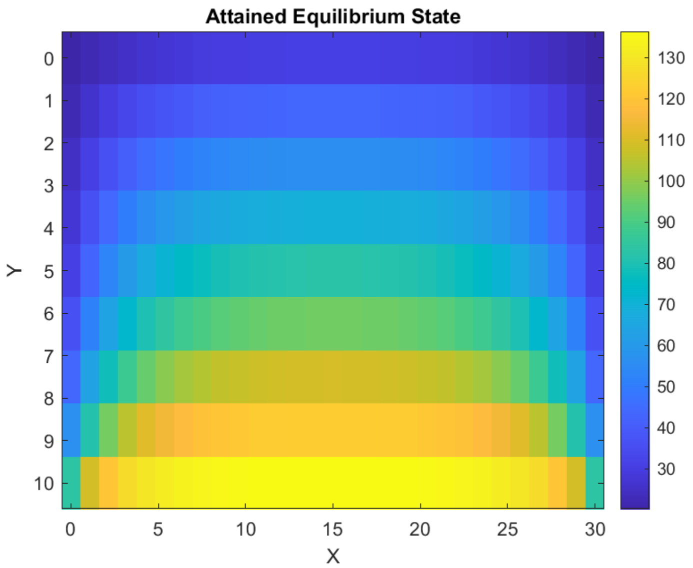


Figure 7.30x10 domain, Gauss-Seidel Iteration Temperature Distribution

```
Temperature at center node converged to: 82.368325
Number of iterations: 1
```

Figure 6. 30x10 domain, Gauss-Seidel Iteration, converged temperature

<u>Gauss-Seidel SOR iteration:</u>

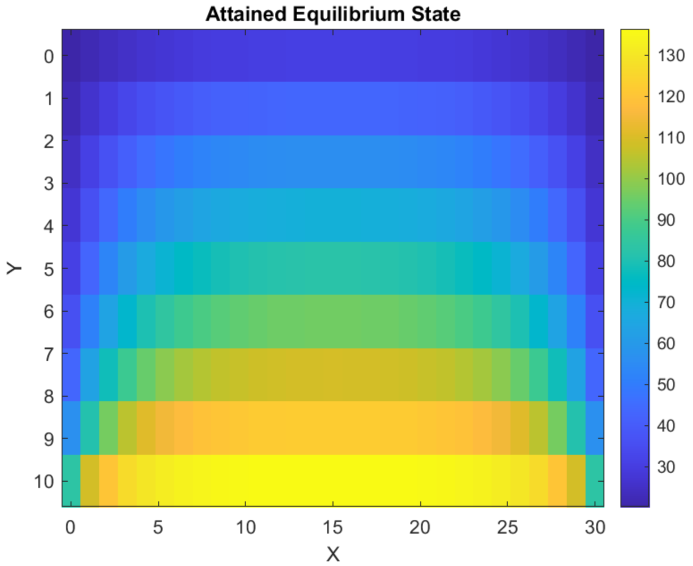


Figure 9.30x10 domain, Gauss-Seidel SOR Iteration Temperature Distribution

```
Temperature at center node converged to: 82.368326
Number of iterations: 1
```

Figure 8.30x10 domain, Gauss-Seidel SOR Iteration, converged temperature

90x30 domain for each method

<u>Jacobi iteration:</u>

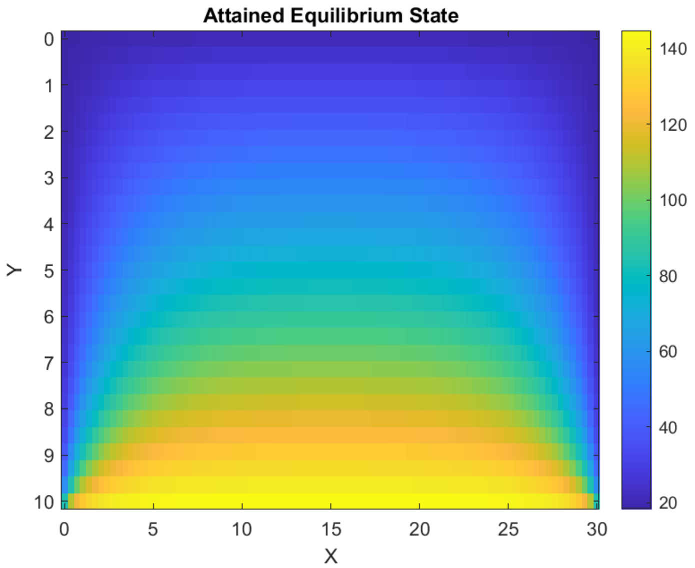
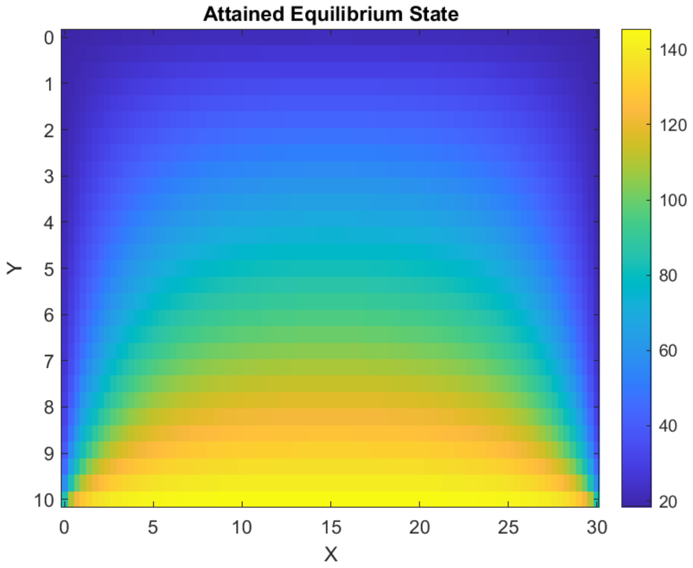


Figure 11. 90x30 domain, Jacobi Iteration Temperature Distribution

```
Temperature at center node converged to: 82.463573
Number of iterations: 1000
```

Figure 10. 90x30 domain, Jacobi Iteration, converged temperature

<u>Gauss-Seidel iteration:</u>

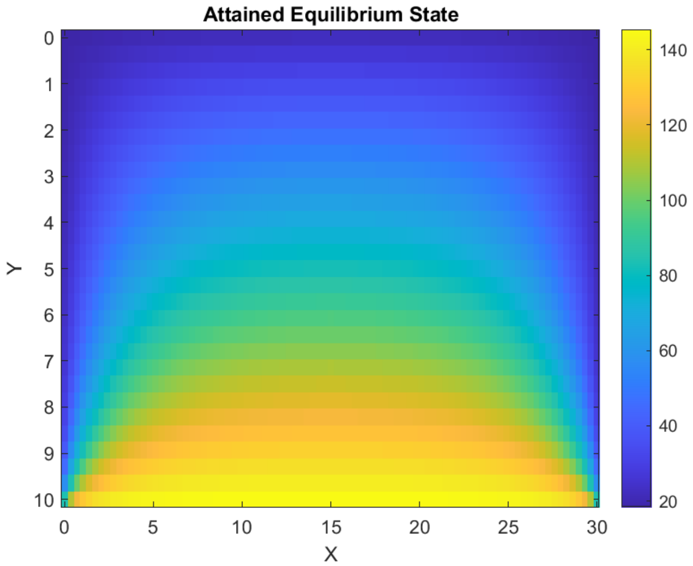



```
Temperature at center node converged to: 82.476295
Number of iterations: 1000
```

Figure 12. 90x30 domain, Gauss-Seidel Iteration, converged temperature

Figure 13. 90x30 domain, Gauss-Seidel Iteration Temperature Distribution

<u>Gauss-Seidel SOR iteration:</u>



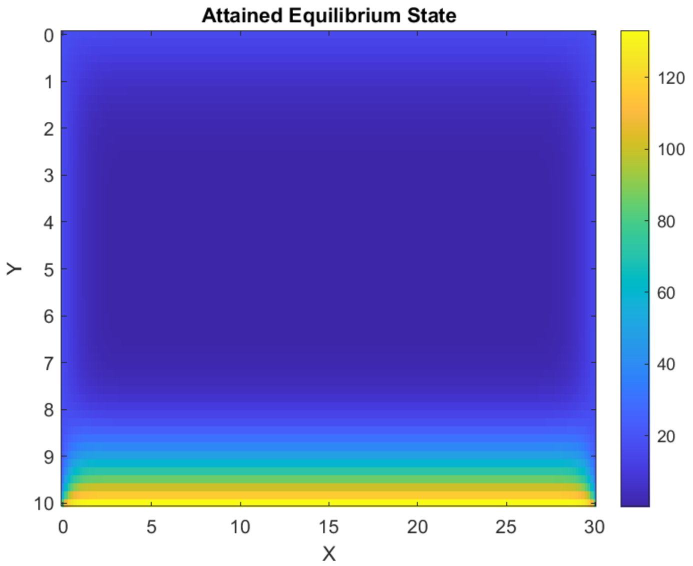

```
Temperature at center node converged to: 82.476948
Number of iterations: 1
```

Figure 14. 90x30 domain, Gauss-Seidel SOR Iteration, converged temperature

Figure 15. 90x30 domain, Gauss-Seidel SOR Iteration Temperature Distribution

180x60 domain for each method

Jacobi iteration:



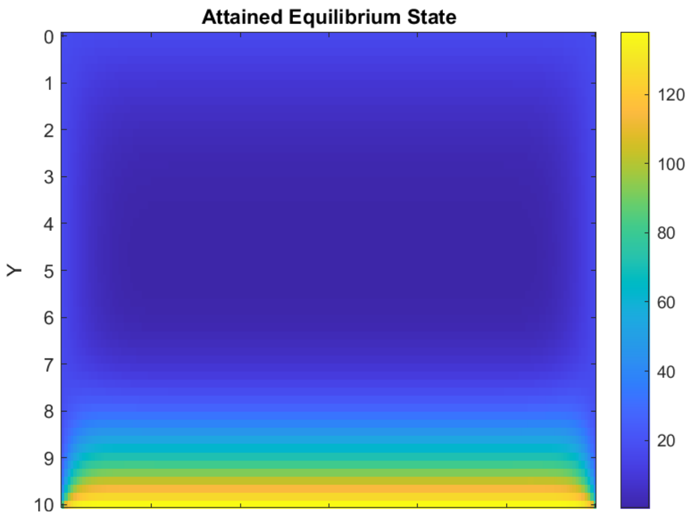

```
Temperature at center node converged to: 2.077336
Number of iterations: 100
```

Figure 16. 180x60 domain, Jacobi Iteration, converged temperature

Figure 17. 180x60 domain, Jacobi Iteration Temperature Distribution

Here, when the maximum number of iterations was set to 1000, the program ran for about an hour and could not display any plot on the screen. Therefore, the maximum number of iterations was reduced to 100. However, as can be seen from the results, the results are not significant. It is obvious that with a better computer, the maximum number of iterations will be higher and meaningful results will be obtained.

Gauss-Seidel iteration:




```
Temperature at center node converged to: 4.877741
Number of iterations: 100
```

Figure 18. 180x60 domain, Gauss-Seidel Iteration, converged temperature

Figure 19.180x60 domain, Gauss-Seidel Iteration Temperature Distribution

180x60 domain, what is written in the Jacobi iteration section is also valid here.

Gauss-Seidel SOR iteration:

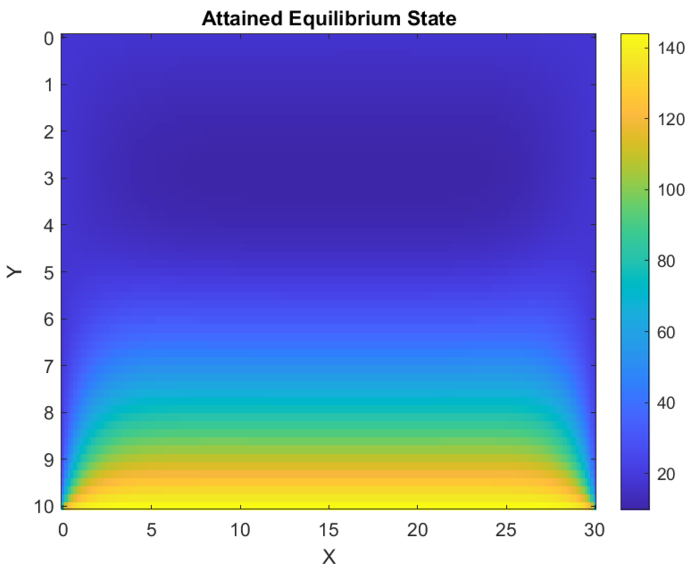


Figure 21. 180x60 domain, Gauss-Seidel SOR Iteration Temperature Distribution

```
Temperature at center node converged to: 26.967313
Number of iterations: 100
```

Figure 20. 180x60 domain, Gauss-Seidel SOR Iteration, converged temperature

Results

Below table shows the converged temperatures for each domain.

	30x10	90x30	180x60
Jacobi Iteration	82.368323	82.463573	2.077336
Gauss-Seidel	82.368325	82.476295	4.877741
Gauss-Seidel Sor	82.368326	82.476948	26.967313

As can be seen from the table, the highest temperatures were in the form of Gauss-Seidel Sor, Gauss-Seidel, and Jacobi. Additionally, keeping the maximum number of iterations low in the 180x60 domain was beneficial. In this way, it was observed which method worked faster. The temperature, which was approximately 27 degrees in the Gauss-Seidel SOR iteration, was 4.9 degrees in Gauss-Seidel and 2 degrees in Jacobi. The fact that there is such a difference between temperatures shows the working speed of the methods.

Below table shows the iteration number for each domain.

	30x10	90x30	180x60
Jacobi Iteration	1	1000	100
Gauss-Seidel	1	1000	100
Gauss-Seidel Sor	1	1	100

It seems a little difficult to make a meaningful comment on this table. However, when looking at the data in the 30x10 and 90x30 domains, the Gauss-Seidel SOR method made 1 iteration. In this form, it has made fewer iterations than others.

Resources

CFD Lecture Notes, Çadırcı, S.

Laplace_Equation_2D_Dirichlet_BCs, Engineering-Stream - Software for Engineering Applications

Poisson_Equation_2D_Dirichlet_BCs, Engineering-Stream - Software for Engineering Applications

S. R. (2018, January 20). *Solve 2D Transient Heat Conduction Problem Using ADI Finite Difference Method*. YouTube. https://www.youtube.com/watch?v=NB0J3ENdbKQ