Main Code:

The grid with necessary boundary conditions is initialised and copies are made which is sent as input to different functions to calculate using different methods

Point_Jacobi.py

This code block defines a function which takes initial temperature guess with boundary conditions and returns the no. of iterations and errors at each iteration using Point Jacobi

Point_Gauss_Siedel.py

This code block defines a function which takes initial temperature guess with boundary conditions and returns the no. of iterations and errors at each iteration using Point Gauss Seidel

Point_SOR.py

This code block defines a function which takes initial temperature guess with boundary conditions and returns the no. of iterations and errors at each iteration using Point Successive Over Relaxation

Main Code:

Once the different methods return the data, we have errors at respective iterations and temperature array for each iteration. So we now find the Exact solution using the Temp.py codeblock

Temp.py

This code block defines a function which takes x and y co-ordinate and returns the exact solution using the summation formula given. I have taken up to n=60 iterations whereas the actual theoretical n goes to infinity. The code returns a single temperature value

Main Code:

Now we have all the required data to make the plots. First we make the plot of error vs iterations for different methods. Then we plot the temperature variation and its deviation from exact solution for different methods and finally we make the contour plots for different methods along with the plot for exact temperature solution.