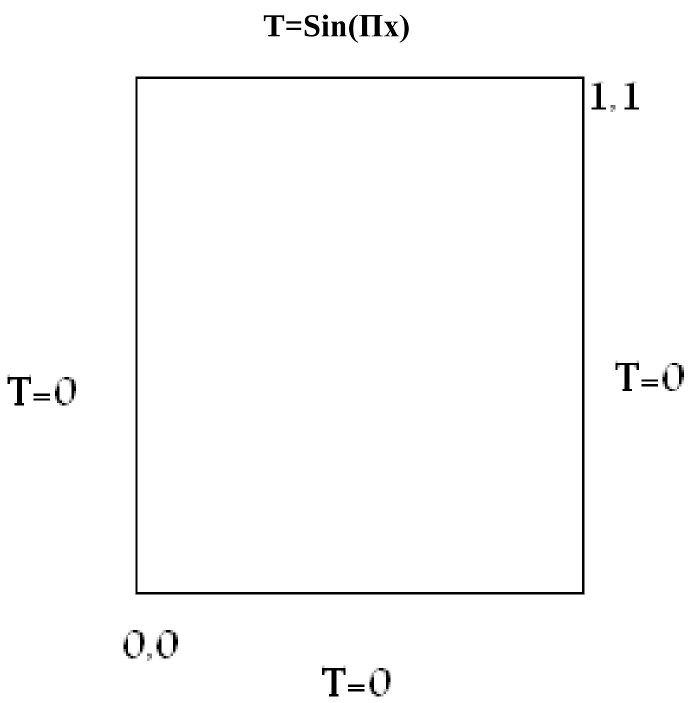
**CFD I, Project 1: Laplace Equation**

**مهلت تحویل :20 فروردین**

In this CFD programming project you solve the Laplace equation which is a mathematical model for diffusion and heat conduction problems. You are supposed to use 2$^{nd}$-order finite volume discretization for solving the Laplace equation on a unit square domain.




**Exact Solution: $T(x,y)=Sin(\Pi x)*Sinh(\Pi y)/Sinh(\Pi)$**

What you need to do:

1-Write your program in Fortran or C (not in Matlab).

2-Write your program in a Subroutine (or function) format, i.e.
SUBROUTINE FluxIntegral(), BoundaryCondition(), Jacobi (), ...

3-Check your boundary condition subroutine and make sure it does what you want it to do. More than 95% of the CFD codes problems are originated or related somehow to the boundary condition implementation.

4-Run your program for Mesh-1 10*10 using Gauss-Seidel routine using initial condition zero everywhere and let it converge to L2=1e-10 (norm of the residual or flux integral) . Compare your solution with the exact solution and make sure your solution trend is alright and the boundary condition is imposed correctly.

5-Run your program for Mesh-2: 20*20 up to Mesh-4: 40*40 using Gauss-Seidel and let them converge to L2=1e-12. Report the convergence history (both in terms of CPU time and iteration number) for M-1→M-4.

6-Verify the accuracy of your solution against the exact solution and show your code is a $2^{nd}$-order code by plotting the error norms versus mesh. Report L1, L2 and Linf for M1→ M4. Now, manipulate your the boundary condition subroutine in such a way that for only one boundary cell of your choice the boundary condition is implemented by a first-order discretization. Do the accuracy analysis again and draw a conclusion about your result. Visualize the solution in both cases in different plots and compare them together.

7- Use SOR for Mesh-4 with w=1.1, 1.3, 1.5 as over relaxation factors and compare your convergence history with Jacobi and Gauss-Seidel (both in terms of CPU time and iteration number). What is your preferred choice for convergence method and why it is so?

8- Now that you have tried the explicit version of your code, it is time to try the semi-implicit or line version and the fully implicit version. Write the coefficient matrix of the Laplace eq. including its boundary condition terms for the specified geometry in the current project.

8-1) Solve the system for line SOR with w=1.1 for M1, M2 & M4. You can use Tri-Diagonal solver for line iteration. Make sure that your solution still is correct and is similar to the explicit solver result for the same mesh. Report the CPU-time and compare it with the CPU-time of explicit SOR for the same mesh and over relaxation factor on one graph. Draw your own conclusion at the end.

8-2) Solve the coefficient matrix for the problem model over M1, M2 & M4 using a direct matrix solver method such as Gaussian elimination. Make sure that your solution still is correct and is similar to the explicit solver result for the same mesh. Report the CPU-time and compare it with the CPU-time of explicit solvers (L2=1e-12) for the same mesh on one graph. Draw your own conclusion at the end. You may have to implement sparse programing for M4 case, Why ?

8-3) Compare the CPU-time of convergence (L2=1e-12) for explicit, line and fully implicit solvers for all meshes (M1, M2 & M4) in one table. What do you conclude at the end of this project.

Courtesy of Professor Carl Ollivier-Gooch