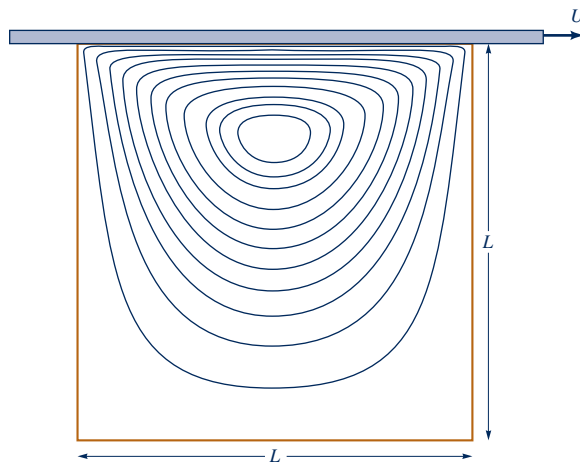


CFD Driven Cavity Example

I.1 Problem

One of the classic benchmarks in CFD is the driven cavity problem. Consider steady, incompressible, viscous flow in a square cavity; the fluid fills the cavity and the top moves at a finite velocity as shown in Fig. I.1, which causes the fluid to rotate within the cavity. This two-dimensional cavity is of length and width L and the top moves with speed U . In dimensionless form the length and width of the cavity are 1 and the speed of the top is 1. Find the low Reynolds number solution for flow in this two-dimensional cavity.



■ **Figure I.1** Driven cavity problem geometry.

I.2 Physics

For steady, incompressible, viscous flow, the Navier-Stokes equations (Eqs. 6.127) can be used to solve for the flow field. With the Stokes flow assumption (i.e., negligible convective terms), the Reynolds number can be taken as less than or equal to 1. This implies a very viscous, slow-moving fluid and hence the convective terms are negligible. That is, all of the nonlinear $u\partial u/\partial x$, $v\partial u/\partial y$, $w\partial u/\partial z$, $\dots$, $v\partial w/\partial y$, $w\partial w/\partial z$ convective acceleration terms in Eqs. 6.127 are negligible. Other results of the assumptions include dropping the $\partial/\partial t$ term from the Navier-Stokes equations for steady flow and setting the density, ρ , equal to a constant value for incompressible flow. The two-dimensional geometry also simplifies the governing equations by dropping the $\partial/\partial z$ terms and the w velocity terms. Therefore the final governing equations include the two-dimensional continuity equation, and the x and y momentum equations.

I-2 Appendix I ■ CFD Driven Cavity Example

One advantage in solving a two-dimensional, incompressible flow problem is that the typical governing equations can be replaced with a stream function-vorticity formulation (see Roache¹ for details). The three governing equations are replaced with two equations: the stream function equation and the vorticity transport equation. In addition, instead of having to solve for the three primitive variables, u , v , and p , you only need to solve for stream function, ψ , and vorticity, Ω , which reduces the coding effort and the necessary computer resources. The velocity and pressure can be found from the stream function and vorticity once a solution is obtained. For example, if the stream function values are known at all grid points, the velocity can be calculated from Eqn. I.3 shown below. This approach complicates the implementation of the boundary conditions for the problem, but overall it simplifies the numerical solution.

Neglecting the convective terms due to the Stokes flow (low Reynolds number) assumption, the non-dimensional governing equations for the cavity flow are given as (see Roache¹)

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\Omega \quad (\text{I.1})$$

$$\text{Re}^{-1} \left(\frac{\partial^2 \Omega}{\partial x^2} + \frac{\partial^2 \Omega}{\partial y^2} \right) = 0 \quad (\text{I.2})$$

For this example, a Reynolds number of unity will be used. These equations are coupled partial differential equations and need to be solved numerically to obtain the flow solution.

The boundary conditions for this problem are the standard no-slip condition for the four walls. When using primitive variables, applying the no-slip condition entails setting u and v equal to zero on the stationary walls. Since the governing equations use the stream function-vorticity formulation, the no-slip condition must be transformed into ψ and Ω terms. By definition, the stream function relates to u and v velocity components with the following equations:

$$u = \frac{\partial \psi}{\partial y} = 0 \quad \text{and} \quad v = -\frac{\partial \psi}{\partial x} = 0, \quad (\text{I.3})$$

at the wall. Therefore, if the wall is horizontal (aligned with x), then setting ψ to a constant value satisfies the $v = 0$ boundary condition. The same holds true for a vertical wall, where $\psi = \text{constant}$ satisfies the $u = 0$ equation. The standard convention is to set $\psi = 0$ which gives the boundary condition for the stream function. A boundary condition is also needed for the vorticity variable, Ω .

At a no-slip boundary, vorticity is produced. In the driven cavity for this example, it is the vorticity production at the wall that is propagated into the flow field through the vorticity transport equation (Eq. I.2) and which drives the problem. The boundary condition for Ω can be derived from the no-slip conditions. It can be shown from Eqn. (6.17) that the vorticity for two-dimensional flow in standard x - y Cartesian coordinates is given by

$$-\Omega = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \quad (\text{I.4})$$

Along a horizontal wall, the v velocity is a constant value, $v = 0$. Therefore, $\partial v / \partial x = 0$ at the wall. Substituting this into the definition of vorticity (Eq. I.4), a wall boundary condition is produced: $-\Omega_{\text{wall}} = \partial u / \partial y|_{\text{wall}} = \partial^2 \psi / \partial y^2|_{\text{wall}}$. Along a vertical wall, a similar boundary condition can be derived based on the same arguments: $-\Omega_{\text{wall}} = -\partial v / \partial x|_{\text{wall}} = \partial^2 \psi / \partial x^2$.

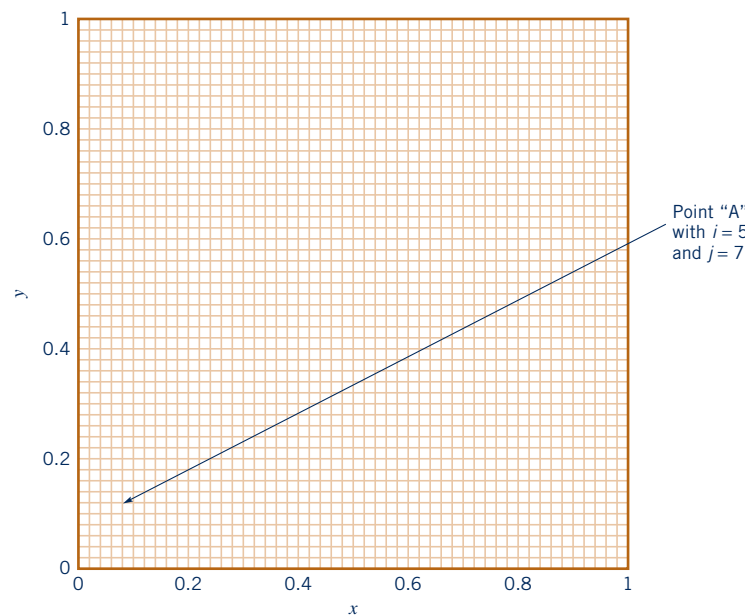
For a beginning student of CFD, these transformed boundary conditions may be difficult to understand. However, this only slightly detracts from the academic value of the example as a demonstration of numerically solving a simple viscous flow. The reader should consult Roache¹ for more details.

¹Roache, P.J., "Fundamentals of Computational Fluid Dynamics," Hermosa Publishers, Albuquerque, NM, 1998.

I.3 Grid

The geometry for the problem is a square, two-dimensional cavity with a non-dimensional size of 1. The top and bottom walls are aligned with the x axis, while the side walls are aligned with the y axis. The grid used in this example is a uniform structured grid shown in Figure I.1. This particular grid has 51 grid points in both the x and y directions. Using the standard CFD notation of M being the maximum number of grid points in the x direction and N being the maximum number of y grid points, the grid size is written as $M \times N$, or 51×51 .

When simulating a viscous flow, it is standard to use some type of grid stretching to cluster grid points in viscous regions where high gradients are expected. In this case, viscous effects are dominant throughout the entire cavity, so a uniform grid is appropriate. (Recall from Sect. 7.6 that $Re = \text{inertial force}/\text{viscous force}$, so that with $Re \leq 1$ as in this example, viscous forces are important throughout the flow.)



■ **Figure I.2** Uniform structured grid for the driven cavity simulation.

I.4 Discretize

To perform a numerical simulation for the flow field, the equations governing the continuum fluid need to be discretized into algebraic equations. In this example, we will use the finite difference formulation to perform the discretization. Using a basic second-order accurate central difference scheme for the derivatives, the governing equations (Eqs. I.1 and I.2) become:

$$\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{(\Delta x)^2} + \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{(\Delta y)^2} = -\Omega_{i,j} \quad (\text{I.5})$$

$$\frac{\Omega_{i+1,j} - 2\Omega_{i,j} + \Omega_{i-1,j}}{(\Delta x)^2} + \frac{\Omega_{i,j+1} - 2\Omega_{i,j} + \Omega_{i,j-1}}{(\Delta y)^2} = 0 \quad (\text{I.6})$$

The i,j refers to the grid index in the x and y directions, respectively. For example, the grid point at $i,j = (1,1)$ is at $x = y = 0$ in Fig. I.2. For a grid location of $i,j = (3,4)$, move three grid lines in the positive x direction from $i = 1$, and four grid lines up in the positive y direction from $j = 1$. The partial differential equations have now been transformed to algebraic equations involving the numerous $\psi_{i,j}$ and $\Omega_{i,j}$ variables that the computer can numerically handle. The equations are second-order accurate ($O(\Delta x^2)$) in space.

I-4 Appendix I ■ CFD Driven Cavity Example

With the stream function-vorticity formulation, some of the required boundary conditions are of the gradient or partial differential form (see above). Therefore, those equations will also need to be discretized in order for the computer algorithm to incorporate them. The wall boundary conditions are similar, with the stream function equal to zero and the vorticity at the wall depending on a normal velocity gradient at the wall. The $\psi = 0$ boundary condition is straightforward to implement within the code. To handle the gradient boundary conditions, we first write a Taylor series expansion for ψ values at one grid point off the wall. To illustrate this method, the expansion will be written for the bottom horizontal wall of the cavity, where the wall is located at $j = 1$.

$$\psi_{i,2} = \psi_{i,1} + \left. \frac{\partial \psi}{\partial y} \right|_{i,1} \Delta y + \left. \frac{\partial^2 \psi}{\partial y^2} \right|_{i,1} \frac{\Delta y^2}{2} + \left. \frac{\partial^3 \psi}{\partial y^3} \right|_{i,1} \frac{\Delta y^3}{6} + \dots \quad (\text{I.7})$$

According to the boundary conditions discussed above, $\psi_{i,1} = 0$, which drops out the first term on the right-hand side. In addition, $\partial \psi / \partial y|_{i,1} = u_{i,1} = 0$ from the no-slip condition, which drops the second term. Substituting the vorticity boundary condition for the horizontal wall into Eq. (I.7), and solving for Ω at the wall gives $\Omega_{i,1} = -2\psi_{i,2}/\Delta y^2$. The boundary condition has an order of accuracy $O(\Delta y)$. The boundary conditions for the four walls can now be written as:

$$\text{Left wall: } \psi_{1,j} = 0 \text{ and } \Omega_{1,j} = -2\psi_{2,j}/(\Delta x)^2 \quad (\text{I.8})$$

$$\text{Right wall: } \psi_{M,j} = 0 \text{ and } \Omega_{M,j} = -2\psi_{M-1,j}/(\Delta x)^2 \quad (\text{I.9})$$

$$\text{Bottom wall: } \psi_{i,1} = 0 \text{ and } \Omega_{i,1} = -2\psi_{i,2}/(\Delta y)^2 \quad (\text{I.10})$$

$$\text{Top wall: } \psi_{i,N} = 0 \text{ and } \Omega_{i,N} = -(2\Delta y U + 2\psi_{i,N-1})/(\Delta y)^2 \quad (\text{I.11})$$

Recall that the variables M and N represent the maximum number of grid points in the x and y directions, respectively. Note that the vorticity at the top wall is different than the other three walls, which is due to the fact that the top wall is moving at a constant velocity, $u = U = 1$ (non-dimensional). The first term within the brackets of Eq. I.11 (i.e., $2\Delta y U$), which is absent in the boundary conditions for the stationary surfaces (Eqs. I.8, I.9, and I.10), accounts for the motion of the top wall. The stream function values are fixed at the wall and will not change during the solution procedure. The vorticity boundary conditions are dependent on the nearest interior stream function value, which gets updated during the simulation.

For this particular problem, a simple explicit scheme can be applied for the solution method. Solving for the stream function and vorticity values at the i,j grid location from Eqs. I.5 and I.6, and assuming that Δx is equal to Δy (as is the case in Fig. I.1), the explicit discretized equations become:

$$\psi_{i,j}^{n+1} = \frac{1}{4} (\psi_{i+1,j}^n + \psi_{i-1,j}^n + \psi_{i,j+1}^n + \psi_{i,j-1}^n + (\Delta x)^2 \Omega_{i,j}^n) \quad (\text{I.12})$$

$$\Omega_{i,j}^{n+1} = \frac{1}{4} (\Omega_{i+1,j}^n + \Omega_{i-1,j}^n + \Omega_{i,j+1}^n + \Omega_{i,j-1}^n) \quad (\text{I.13})$$

The superscript $n + 1$ represents the new iteration value that is being calculated at the i,j grid point and the values at n represent the ψ and Ω values at the previous iteration. If the algorithm is at the first iteration, then the n value represents the initial guess imposed on all interior grid points. Equations I.12 and I.13 are valid for all interior grid points, while Equations I.8–I.11 are used for the boundary grid points.

I.5 Solve

The first item in the “solve” portion of the CFD flowchart (Fig. A.6) is algorithm development. When using commercial code, this step can be bypassed. For this example problem, a pseudo code is provided that gives an outline of the code in standard sentence form to show the sequential progression of the solution. A pseudo code can be translated to whatever computer language

is appropriate. It should be noted that some variation in the solution process is possible; the following pseudo code offers one possibility.

1. Setup constants for the problem or solicit input from the user
 - Maximum number of grid points
 - Grid spacing
 - Convergence criterion
 - Etc.
2. Initialize stream function and vorticity matrices (i.e., the values of $\psi_{i,j}$ and $\Omega_{i,j}$ variables for $1 \leq i \leq M$ and $1 \leq j \leq N$) to zero (initial guess values) at all grid points
3. Solve for the vorticity boundary conditions
4. Solve governing equations on interior grid points and update variables. For example, the stream function and vorticity equations (I.12 and I.13) at grid point $i = 5$ and $j = 7$ (point “A” in Fig. I.2) for the first iteration would be written as:

$$\psi_{5,7}^1 = \frac{1}{4}(\psi_{6,7}^0 + \psi_{4,7}^0 + \psi_{5,8}^0 + \psi_{5,6}^0 + (\Delta x^2)\Omega_{5,7}^0)$$

$$\Omega_{5,7}^1 = \frac{1}{4}(\Omega_{6,7}^0 + \Omega_{4,7}^0 + \Omega_{5,8}^0 + \Omega_{5,6}^0)$$

The subscripts represent the grid location while the superscripts represent the iteration level. In the above example, values at the “0” iteration level are known and values at the new iteration level of “1” are being calculated.

5. Check convergence criterion at each grid point for both ψ and Ω
6. Is the solution converged at all grid points for both variables?
 - No: return to Step 3 and loop through steps again
 - Yes: continue to next step
7. Post-process data if needed (e.g. velocity values from stream function)
8. Write data to file(s)
9. Stop

The solution method requires initial conditions for the code to start. In this case, values for ψ and Ω at all grid points were initialized to zero as a starting solution. Once iterations begin, the boundary condition on the top wall (finite velocity) starts to propagate throughout the flow field, along with the no-slip boundary conditions. Generally, some type of initial condition, or starting solution, is required to begin iterating in a CFD code. For most algorithms, all that is required is a reasonable guess for the starting values. For some codes, a more refined starting solution is needed.

To update the ψ and Ω values for the interior grid points, the code needs to be able to sweep through (e.g. left-to-right and bottom-to-top) all of the grid points in a systematic fashion. For example, the code could start at grid point (1,1), sweep in the x -direction to (1,M), then move to the next row, or j value and repeat. Once the code has updated values at all grid points, it has completed one iteration loop. For this particular problem, the choice on the starting sweep point and the sweep direction do not affect the solution; for some CFD problems, sweep direction is important.

Convergence is checked by comparing ψ and Ω values from previous and current iteration levels at each grid point. If convergence is not achieved, the new values are now labeled as previous and new values are again calculated. Iterations continue until all grid points meet the convergence criterion.

It should be noted that even for this simple CFD example and relatively coarse grid, the solution process requires over 5000 algebraic equations to be solved for each iteration step. Therefore, some type of computer algorithm is necessary to conduct the flow simulation in a structured and timely fashion. Current CFD codes investigating real-life flow problems are using grids with millions of points and solving multiple equations at each grid location.

I-6 Appendix I ■ CFD Driven Cavity Example

I.6 Analyze

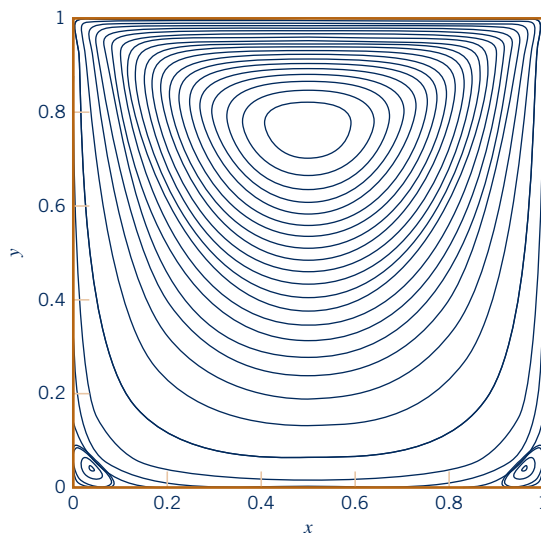
The first item under the “Analyze” portion of the CFD Methodology is “Verification and Validation”. This is an extremely important step and can be quite an in-depth topic. As mentioned in Appendix A, the CFD code/simulation cannot be treated as a magic black box; just because the code converges and produces a solution does not guarantee an accurate or physically-correct solution. There are a few common techniques that are applicable to all CFD simulations and are discussed below.

The required convergence limit is problem-dependent. Each different problem may require different convergence criteria to produce a reasonably accurate solution. At minimum, the dependence on the convergence limit needs to be checked when using CFD to solve a flow field. The convergence limit should be decreased several times over several orders-of-magnitude to ensure that the solution is not changing appreciably when the limit is decreased. For example, the initial convergence criterion may be 0.001. Another simulation should then be conducted with the limit reduced to 0.0001 and the results compared to the previous limit. The driven cavity example was converged to 0.0001 for both stream function and vorticity. Typically, some variables converge faster than others (for a variety of reasons). Stream function values converged faster than vorticity for this example.

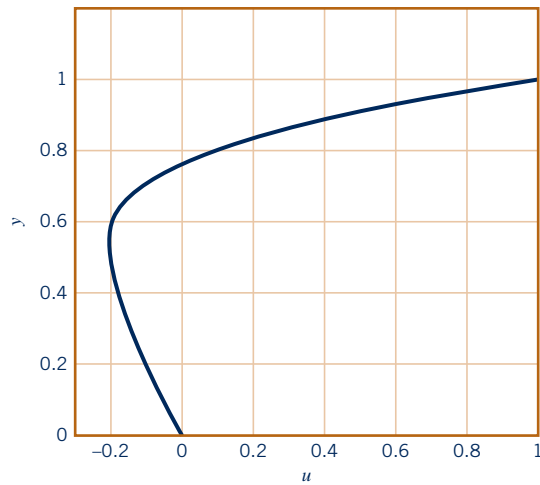
The second basic validation tool for all CFD simulations is to ensure a grid-independent solution. As discussed in Appendix A, the solution should not change as the grid is refined. The grids used for the driven cavity included 51×51 , 101×101 , and 151×151 . Using the centerline velocity results to make the comparison, it was found that 101×101 produced a grid-independent solution. The centerline velocity changed slightly from 51×51 to 101×101 . The results for 101×101 and 151×151 were in good agreement.

Several representative results from the driven cavity simulation are presented below. Figure I.3 shows the stream function contours (i.e. lines of constant ψ) for the cavity. The fluid is contained within the cavity so as the top plate moves at a finite velocity (speed $u = 1$ in the dimensionless representation used), it creates a large rotating motion, displayed by the contours. Also note the smaller recirculating regions in the two lower corners. The fluid cannot maneuver in the sharp corner, so a separation bubble is created.

Figure I.4 shows a vertical centerline velocity profile for the u velocity component. As expected, the velocity is zero at $y = 0$ (wall) with the no-slip boundary condition. The top plate is moving in the positive x direction at a non-dimensional value of 1. In the interior, the plot reflects the fact that the upper portion of fluid is moving in the same direction as the top plate, or in the positive direction. The lower portion is moving in the negative direction, providing the rotating motion within the cavity. The velocity profile plot is a good candidate for use as a measure for grid independence for the different grid resolutions.



■ **Figure I.3** Stream function contours for Stokes flow in a driven cavity.



■ **Figure I.4** x-component of velocity along a vertical centerline in the driven cavity.

When interpreting the CFD results, it is always a good idea to ask two questions: 1) Do the results make physical sense?, and 2) Is there experimental data (same or similar) to compare baseline values? It is important to have a strong background in fluid mechanics when using CFD as an engineering tool. This experience helps to answer the first question. Available experimental results of a similar nature help to verify that the governing equations are appropriate, the equations were implemented correctly, and that there are no major bugs in your code. Remember that CFD is a complex topic; a strong background and a great deal of experience are needed to be proficient in the subject.