Chapter 12 Numerical Simulation

The classical exact solutions of the Navier-Stokes equations are valuable for the insight they give and as a exact result that any approximate method should be able to duplicate with to an acceptable precision. However, if we limit ourselves to exact solutions or even approximate perturbation or series solutions, a vast number of important engineering problems will be beyond reach. To meet this need, a number of computational fluid mechanics (CFM) codes have become commercially available. A student could use the CFM code as a virtual experiment to learn about fluid mechanics. While this may be satisfactory for someone who will become a design engineer, this is not sufficient for the researcher who may wish to solve a problem that has not been solved before. The user usually has to treat the CFM code as a "black box" and accept the result of the simulation as correct. The researcher must always be questioning if any computed result is valid and if not, what must be done to make it valid. Therefore I believe Ph.D. students should know how to develop their own numerical simulators rather than be just a user.

We will start with a working FORTAN and MATLAB code for solving very simple, generic problems in 2-D. The student should be able to examine each part of the code and understand everything with the exception of the algorithm for solving the linear system of equations. Then the student will change boundary conditions, include transient and nonlinear capabilities, include curvilinear coordinates, and compute pressure and stresses.

The Stream Function – Vorticity Method

Two-dimensional flow of an incompressible, Newtonian fluid can be formulated with the stream function and vorticity as dependent variables.

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=-w$$

$$\frac{dw}{dt}=\frac{\partial w}{\partial t}+v_x\frac{\partial w}{\partial x}+v_y\frac{\partial w}{\partial y}=\nu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}\right)$$

where

$$\mathbf{v}=(v_x,v_y,0)=\nabla\times\mathbf{A}=\left(\frac{\partial\psi}{\partial y},-\frac{\partial\psi}{\partial x},0\right),\quad \mathbf{A}=\left(0,0,\psi\right)$$

$$\mathbf{w}=(0,0,w)=\left(0,0,\frac{\partial v_y}{\partial x}-\frac{\partial v_x}{\partial y}\right)$$

We will begin developing a generic code by assuming steady, creeping (zero Reynolds number) flow with specified velocity on the boundaries in a rectangular domain. The PDE are now,

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -w$$

$$, \qquad 0 < x < L_x, \quad 0 < y < L_y, \quad \text{Re} = 0$$

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

This is a pair of linear, elliptic PDEs. The boundary conditions with specified velocity are

$$\begin{aligned}
&\mathbf{v}_{normal} = \text{specified} \\
&\mathbf{v}_{tangential} = \text{specified} = \quad O(U) \\
&\psi_{BC} = \int\limits_{boundary} \mathbf{v} \bullet \mathbf{n}\, ds \\
&w_{BC} = \nabla \times \mathbf{v}\big|_{boundary}
\end{aligned}$$

The boundary conditions at a plane of symmetry are

$$\begin{aligned}
&\mathbf{n} \bullet \mathbf{v} = 0 \\
&n \bullet \nabla \mathbf{v} = 0 \\
&\psi\text{=constant} \\
&w = 0
\end{aligned}$$

The PDE and definitions are made dimensionless with respect to reference quantities such that the variables are of the order of unity.

$$x* = \frac{x}{L_x}, \quad y* = \frac{y}{L_y}, \quad \alpha = \frac{L_x}{L_y}$$

$$\mathbf{v}* = \frac{\mathbf{v}}{U}, \quad \psi* = \frac{\psi}{\psi_o}, \quad w* = \frac{w}{w_o}$$

$$\left[\frac{\psi_o}{L_x^2\, w_o}\right] \frac{\partial^2 \psi *}{\partial x*^2} + \left[\frac{\psi_o}{L_y^2\, w_o}\right] \frac{\partial^2 \psi *}{\partial y*^2} = -w*$$

$$, \qquad 0 < x* < 1, \quad 0 < y* < 1$$

$$\frac{\partial^2 w*}{\partial x*^2} + \alpha^2 \frac{\partial^2 w*}{\partial y*^2} = 0$$

$$v_x* = \left[\frac{\psi_o}{U\, L_y}\right] \frac{\partial \psi *}{\partial y*}, \quad v_y* = -\left[\frac{\psi_o}{U\, L_x}\right] \frac{\partial \psi *}{\partial x*}$$

We have four dimensionless groups and two reference quantities. We can specify two of the groups to equal unity and the other two are equal to the aspect ratio or its square.

$$\left[\frac{\psi_o}{L_x^2\, w_o}\right]=1, \quad \left[\frac{\psi_o}{U\, L_x}\right]=1$$

$$\Rightarrow \quad w_o=\frac{U}{L_x}, \quad \psi_o=L_x\, U$$

The PDE and definitions with the * dropped are now as follows.

$$\frac{\partial^2\psi}{\partial x^2}+\alpha^2\frac{\partial^2\psi}{\partial y^2}=-w$$

$$, \qquad 0<x<1, \quad 0<y<1$$

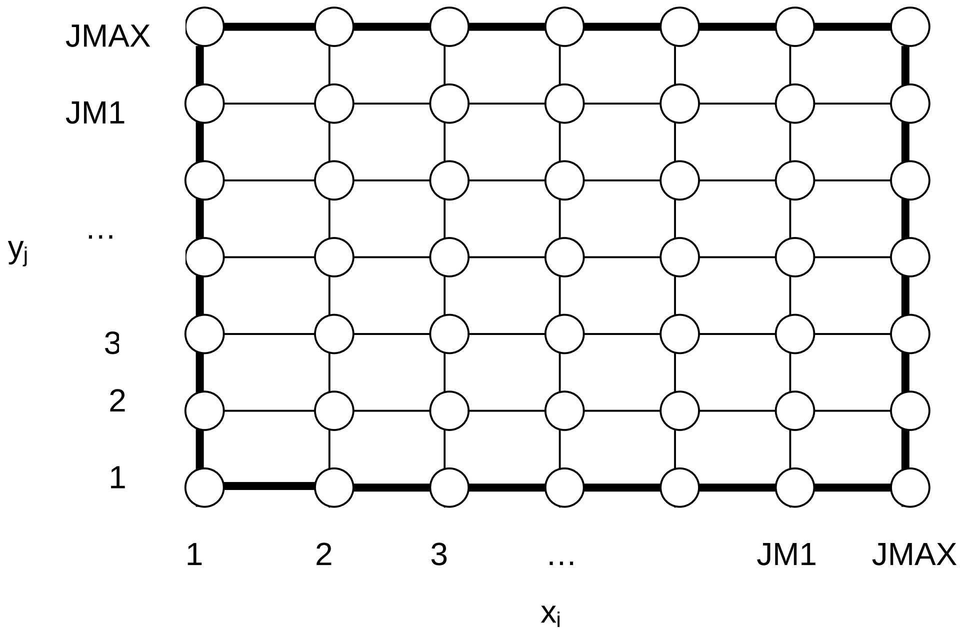
$$\frac{\partial^2 w}{\partial x^2}+\alpha^2\frac{\partial^2 w}{\partial y^2}=0$$

$$v_x=\alpha\frac{\partial\psi}{\partial y}, \quad v_y=-\frac{\partial\psi}{\partial x}$$

$$w=\frac{\partial v_y}{\partial x}-\alpha\frac{\partial v_x}{\partial y}$$

Finite Difference Approximation

The PDE and BC will be solved using a finite difference method. A grid point formulation rather than a grid block formulation will be used since the dependent variables are specified at the boundaries rather than their flux or normal derivative. The computation domain will be discretized such that the boundary conditions and dependent variables are evaluated at x_1, x_2, …, x_{JM1}, x_{JMAX} and y_1, y_2, …, y_{JM1}, y_{JMAX}. The finite difference grid appears as follows.



Finite difference grid for discretizing PDE (grid point formulation)

The unit square is now discretized into *JMAX* by *JMAX* grid points where the boundary conditions and dependent variables will be evaluated. The grid spacing is $\delta = 1/(JMAX - 1)$. The first and last row and column are boundary values.

The partial derivatives will be approximated by finite differences. For example, the second derivative of vorticity is discretized by a Taylor's series.

$$w_{i+1} = w_i + \delta\left(\frac{\partial w}{\partial x}\right)_i + \frac{\delta^2}{2}\left(\frac{\partial^2 w}{\partial x^2}\right)_i + \frac{\delta^3}{6}\left(\frac{\partial^3 w}{\partial x^3}\right)_i + \frac{\delta^4}{24}\left(\frac{\partial^4 w}{\partial x^4}\right)_{\bar{x}}$$

$$w_{i-1} = w_i - \delta\left(\frac{\partial w}{\partial x}\right)_i + \frac{\delta^2}{2}\left(\frac{\partial^2 w}{\partial x^2}\right)_i - \frac{\delta^3}{6}\left(\frac{\partial^3 w}{\partial x^3}\right)_i + \frac{\delta^4}{24}\left(\frac{\partial^4 w}{\partial x^4}\right)_{\bar{\bar{x}}}$$

$$\left(\frac{\partial^2 w}{\partial x^2}\right)_i = \frac{w_{i-1} - 2\,w_i + w_{i+1}}{\delta^2} - \frac{\delta^2}{24}\left[\left(\frac{\partial^4 w}{\partial x^4}\right)_{\bar{x}} + \left(\frac{\partial^4 w}{\partial x^4}\right)_{\bar{\bar{x}}}\right]$$

$$= \frac{w_{i-1} - 2\,w_i + w_{i+1}}{\delta^2} + O\left(\delta^2\right)$$

The finite difference approximation to the PDE at the interior points results in the following set of equations.

$$
\begin{aligned}
&\psi_{i+1,j}+\psi_{i-1,j}+\alpha^2\psi_{i,j+1}+\alpha^2\psi_{i,j-1}-2(1+\alpha^2)\psi_{i,j}+\delta^2 w_{i,j}=0\\
&w_{i+1,j}+w_{i-1,j}+\alpha^2 w_{i,j+1}+\alpha^2 w_{i,j-1} \qquad\qquad -2(1+\alpha^2)w_{i,j}=0\\
&i,j=2,3,\cdots JMAX-1
\end{aligned}
$$

The vorticity at the boundary is discretized and expressed in terms of the components of velocity at the boundary, the stream function values on the boundary and a stream function value in the interior grid. (A greater accuracy is possible by using two interior points.) The stream function at the first interior point (*i*=2) from the *x* boundary is written with a Taylor's series as follows.

$$
\psi_2=\psi_1\pm\delta\left(\frac{\partial\psi}{\partial x}\right)_1+\frac{\delta^2}{2}\left(\frac{\partial^2\psi}{\partial x^2}\right)_1+O(\delta^3)
$$

$$
\begin{aligned}
\left(\frac{\partial^2\psi}{\partial x^2}\right)_1&=\frac{2(\psi_2-\psi_1)}{\delta^2}\mp\frac{2}{\delta}\left(\frac{\partial\psi}{\partial x}\right)_1+O(\delta)\\
&=\frac{2(\psi_2-\psi_1)}{\delta^2}\pm\frac{2}{\delta}v_y^{BC}+O(\delta)\\
w_1^{BC}&\equiv\left(\frac{\partial v_y}{\partial x}-\alpha\frac{\partial v_x}{\partial y}\right)^{BC}\\
&=-\left(\frac{\partial^2\psi}{\partial x^2}\right)_1-\alpha\frac{\partial v_x^{BC}}{\partial y}\\
&=-\frac{2(\psi_2-\psi_1^{BC})}{\delta^2}\mp\frac{2}{\delta}v_y^{BC}-\alpha\frac{\partial v_x^{BC}}{\partial y}+O(\delta)
\end{aligned}
$$

The choice of sign depends on whether *x* is increasing or decreasing at the boundary. Similarly, on a *y* boundary,

$$
w_1^{BC}=-\frac{2\alpha^2(\psi_2-\psi_1^{BC})}{\delta^2}\pm\frac{2\alpha v_x^{BC}}{\delta}+\frac{\partial v_y^{BC}}{\partial x}+O(\delta)
$$

The boundary condition on the stream function is specified by the normal component of velocity at the boundaries. Since we have assumed zero normal component of velocity, the stream function is a constant on the boundary, which we specify to be zero.

The stream function at the boundary is calculated from the normal component of velocity by numerical integration using the trapezodial rule, e.g.,

$$
\psi_j=\psi_{j-1}+\frac{\delta}{2}\left[\left(v_x\right)_{j-1}+\left(v_x\right)_j\right]/\alpha
$$

Solution of Linear Equations

The finite difference equations for the PDE and the boundary conditions are a linear system of equations with two dependent variables. The dependent variables at a x_i, y_j grid point will be represented as a two component vector of dependent variables,

$$\mathbf{u}_{i,j} = \begin{bmatrix} \psi_{i,j} \\ w_{i,j} \end{bmatrix}$$

The pair of equations for each grid point can be represented in the following form

$$\begin{aligned} &\mathbf{a}_{i,j}\,\mathbf{u}_{i+1,j} + \mathbf{b}_{i,j}\,\mathbf{u}_{i-1,j} + \mathbf{c}_{i,j}\,\mathbf{u}_{i,j+1} + \mathbf{d}_{i,j}\,\mathbf{u}_{i,j-1} + \mathbf{e}_{i,j}\,\mathbf{u}_{i,j} = \mathbf{f}_{i,j} \\ &i, j = 2, 3, \cdots JMAX - 1 \end{aligned}$$

Each coefficient is a 2×2 matrix. For example,

$$\mathbf{e}_{i,j}\,\mathbf{u}_{i,j} = \begin{bmatrix} e_{i,j,1,1}\,\psi_{i,j} + e_{i,j,1,2}\,w_{i,j} \\ e_{i,j,2,1}\,\psi_{i,j} + e_{i,j,2,2}\,w_{i,j} \end{bmatrix}$$

The components of the 2×2 coefficient matrix are the coefficients from the difference equations. The first row is coefficients for the stream function equation and the second row is coefficients for the vorticity equation. The first column is coefficients for the stream function variable and the second column is the coefficients for the vorticity variable. For example, at interior points not affected by the boundary conditions,

$$\mathbf{a}_{i,j} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{b}_{i,j} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{c}_{i,j} = \begin{bmatrix} \alpha^2 & 0 \\ 0 & \alpha^2 \end{bmatrix}$$

$$\mathbf{d}_{i,j} = \begin{bmatrix} \alpha^2 & 0 \\ 0 & \alpha^2 \end{bmatrix}$$

$$\mathbf{e}_{i,j} = \begin{bmatrix} -2(1+\alpha^2) & \delta^2 \\ 0 & -2(1+\alpha^2) \end{bmatrix}$$

$$\mathbf{f}_{i,j} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The coefficients for the interior grid points adjacent to a boundary are modified as a result of substitution the boundary value of stream function or the linear equation for the boundary vorticity into the difference equations. The stream function equation is coupled to the vorticity with the **e**$_{ij}$ coefficient and the vorticity equation is coupled to the stream function through the boundary conditions. For example, at a *x* = 0 boundary, the coefficients will be modified as follows.

$$
\begin{aligned}
&i = 2\\
&f(i,j,1) = f(i,j,1) - b(i,j,1,1) * \psi^{BC}_{1,j}\\
&f(i,j,2) = f(i,j,2) - b(i,j,2,2) * (2 * \psi^{BC}_{1,j} / \delta^2 - 2 * v^{BC}_{v}(j) / \delta - \alpha \frac{\partial v^{BC}_{x}}{\partial y})\\
&e(i,j,2,1) = e(i,j,2,1) - b(i,j,2,2) * 2 / \delta^2
\end{aligned}
$$

If the boundary condition is that of zero vorticity as on a surface of symmetry, it is necessary to only set to zero the coefficient that multiplies the boundary value of vorticity, i.e.,

$$
\begin{aligned}
&i = 2\\
&b(i,j,2,2) = 0, \quad \text{when } w(1,j) = 0
\end{aligned}
$$

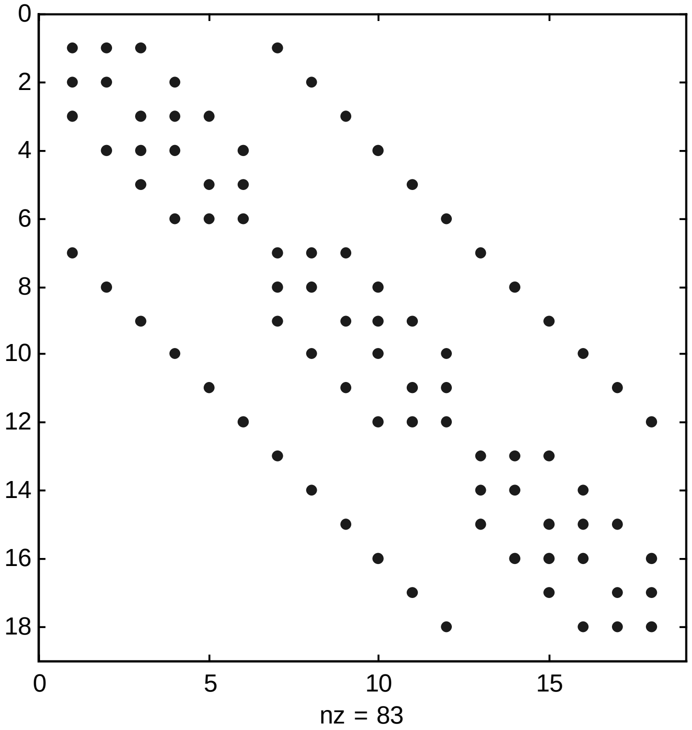
The linear system of equations and unknowns can be written in matrix form by ordering the equations with the *i* index changing the fastest, i.e., *i* = 2, 3, …, *JMAX*-1. The linear system of equations can now be expressed in matrix notation as follows.

$$\mathbf{A} \bullet \mathbf{x} = \mathbf{F}$$

where

$$
\mathbf{x} = \begin{bmatrix} \mathbf{u}_{2,2} \\ \mathbf{u}_{3,2} \\ \vdots \\ \mathbf{u}_{JM1,JM1} \end{bmatrix} = \begin{bmatrix} \psi_{2,2} \\ w_{2,2} \\ \psi_{3,2} \\ w_{3,2} \\ \vdots \\ \psi_{JM1,JM1} \\ w_{JM1,JM1} \end{bmatrix}
$$

The non-zero coefficients of coefficient matrix **A** for *JMAX* = 5 is illustrated below. Since the first and last grid points of each row and column of grid points are boundary conditions, the number of grid with pairs of unknowns is only 3×3. Only one grid point is isolated from boundaries. The *i,j* location of the coefficients become clearer if a box is drawn to enclose each 2×2 coefficient.

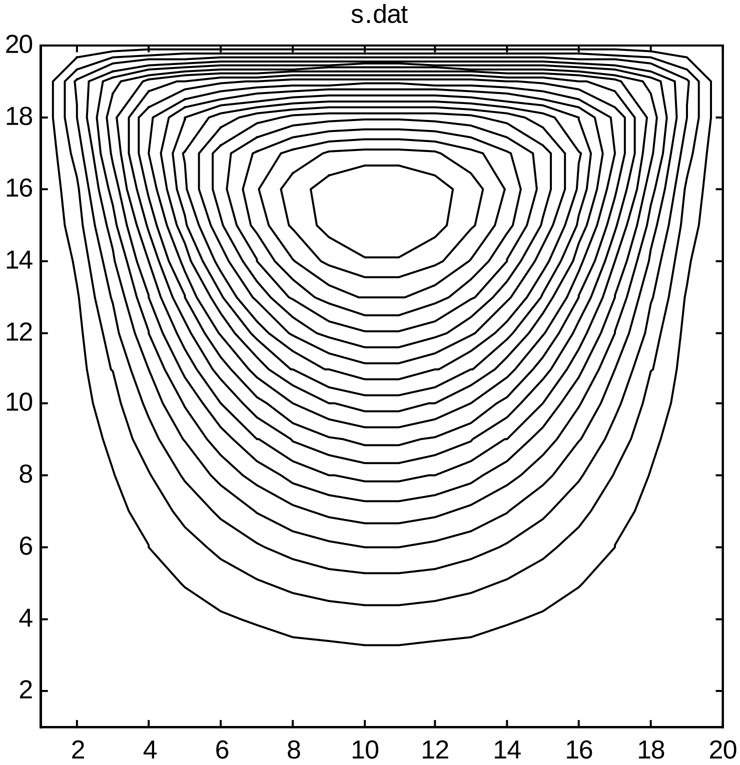


Most of the elements of the coefficient matrix are zero. In fact, the coefficient matrix is a block pentadiagonal sparse matrix and only the nonzero coefficients need to be stored and processed for solving the linear system of equations. The non-zero coefficients of the coefficient matrix are stored with the *row-indexed sparse storage mode* and the linear system of equations is solved by preconditioned biconjugate gradient method (Numerical Recipes, 1992).

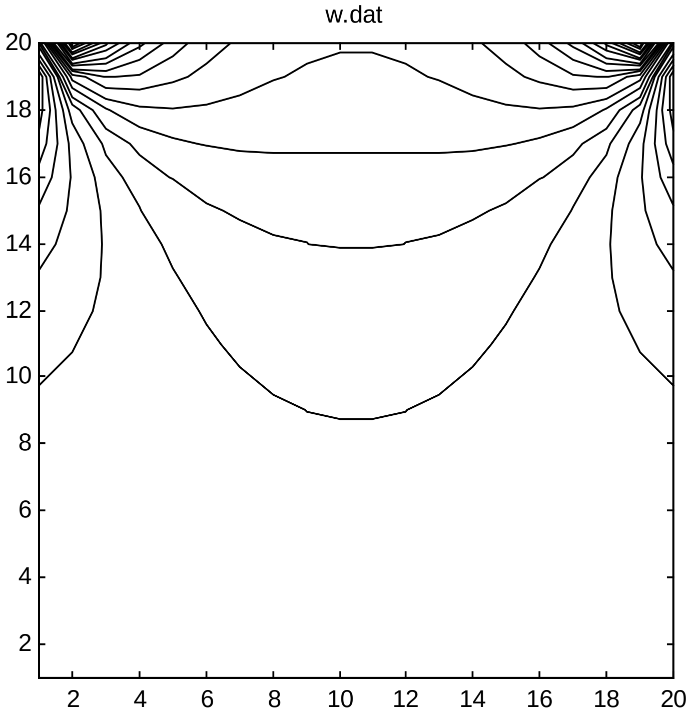
The row-indexed sparse matrix mode requires storing the nonzero coefficients in the array, *sa(k),* and the column number of the coefficient in the coefficient matrix in the array, *ija(k),* where k = 1, 2, …. The indices that are needed to identify the coefficients are the *i,j* grid location (*i,j* = 2, 3, …, *JMAX*-1), the m equation and dependent variable identifier (m = 1 for stream function; = 2 for vorticity), and the *IA* row index of the coefficient matrix (*IA* = 1, 2, 3, 4, …, 2(*JMAX*-2)^2). The coefficient matrix has two equations for each grid point, the first for the stream function and the second for vorticity. The total number of equations is *NN* = 2(*JMAX*-2)^2.

The diagonal elements of the coefficient array are first stored in the *sa(k)* array. The first element of *ija(k), ija(1)* = *NN*+2 and can be used to determine the size of the coefficient matrix. The algorithm then cycles over the pairs of rows in the coefficient matrix while keeping track of the *i,j* locations of the grid point and cycling over equation m = 1 and 2. The off-diagonal coefficients are stored in *sa(k)* and the column number in the coefficient matrix stored in *ija(k).* As each row is completed, the k index for the next row is stored in the first *NN* elements of *ija*.

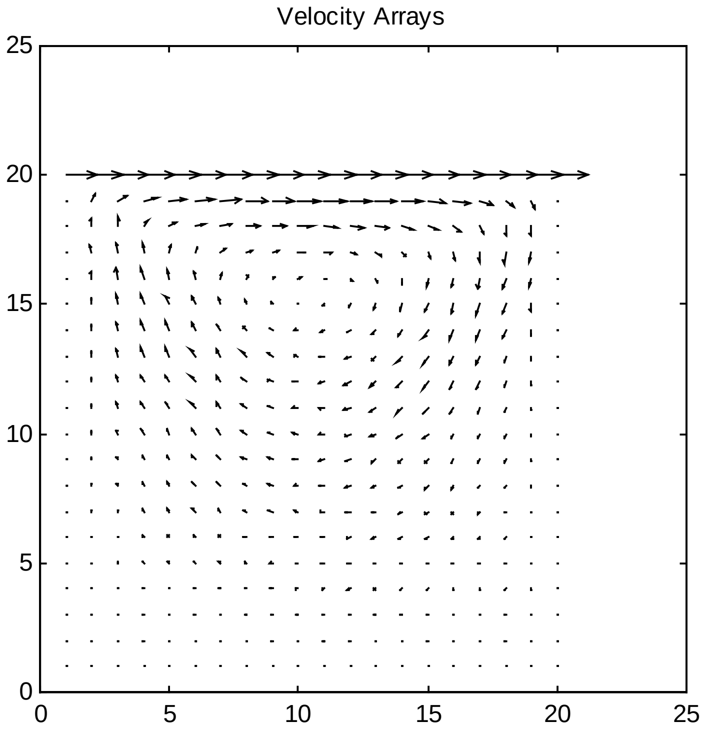
A test problem for this code is a square box that has one side sliding as to impart a unit tangential component of velocity along this side. All other walls are stationary. The stream function, vorticity, and velocity contours for this problem with a 20×20 grid is illustrated below.



Stream function contours for square with top side sliding



Vorticity contours for square with top side sliding



Velocity vectors for square with top side sliding

Assignment 12.1 Use the code in the */MatlabScripts/numer/* directory of the CENG501 web site as a starting point. Add the capability to include arbitrarily specified normal component of velocity at the boundaries. The stream function at the boundary should be numerically calculated from the specified velocity at the boundary. Test the code with Couette and Plane-Poiseuille flows first in the *x* and then in the *y* directions. Display vector plot of velocity and contour plots of stream function and vorticity. Show you analysis and code.

Assignment 12.2 Form teams of two or three and do one of the following additions to the code. Find suitable problems to test code.

1. Calculate pressure and stress. Test problems: Couette and Plane-Poiseuille flows.
2. Calculate transient and finite Reynolds number flows. Test problems: Plate suddenly set in motion and flow in cavity.
3. Add option for cylindrical polar coordinates. Use the coordinate transformation, $z = \ln r$. Test problems: line source and annular rotational Couette flow.

Form new teams of two or three with one member that developed each part of the code. Include all features into the code. Test the combined features. Work together as a team to do the following simulation assignments.

Assignment 12.3 Boundary layer on flat plate. Compute boundary layer velocity profiles and drag coefficient and compare with the Blasius solution. What assumption is made about the value of the Reynolds in the Blasius solution?

Assignment 12.4 Flow around cylinder. Assume potential flow far from the cylinder. Calculate drag coefficient as a function of Reynolds number and compare with literature.

Assignment 12.5 Flow around cylinder. Find the Reynolds number at which boundary layer separation occurs.

Assignment 12.7 Flow around cylinder. Find the Reynolds number for the onset of vortex shedding.

FORTRAN on Owlnet

Modern FORTRAN compilers in the MSWindows are very user friendly. They have built in project workspace routines, documentation, and integrated editor and debugger. If you use the FORTRAN compiler on Owlnet, it is useful to know how to use *make* files to compile and link FORTRAN code. The following is an example *make* file. You may give it the name, *makefile*, with no extension.

```
f77 -c streamf1.f;
f77 -c bc1.f;
f77 -c coef1.f;
f77 -c asolve.f;
f77 -c atimes.f;
f77 -c dsprsax.f;
f77 -c dsprstx.f;
f77 -c linbcg.f;
f77 -c snrm.f;
f77 -o exe streamf1.o bc1.o coef1.o asolve.o atimes.o
dsprsax.o dsprstx.o linbcg.o snrm.o
```

You have to give yourself permission to execute the make file with the one-time command,

```
chmod +x makefile
```

The line with the *–c* option compiles the source code and makes an object code file with the extension of .o. The line with the *–o* option links the object files into the executable file called *exe*. This executable file can be executed by typing its name, *exe*, on the command line.

This example *makefile* is not very efficient because it will compile every source code listed. There are instructions to recompile only the source code that has been modified or "touched" since the last time the *makefile* was invoked. However, I do not recall the instructions.

If you are not familiar with FORTRAN, you should get a paperback book on FORTRAN-77 or FORTRAN-90. An example is D. M. Etter, *Structured FORTRAN 77 for Engineers and Scientists*. The following is a tutorial from the CAAM 211 webpage.
http://www.caam.rice.edu/~caam211/JZoss/project1.html

Computer facilities that have compilers usually have an interactive debugger. However, I have not found how to access the debugger on *owlnet*. The debugger with the Microsoft FORTRAN Powerstation (now Compaq FORTRAN) is very easy to use.

Transients and Finite Reynolds Number

The development of numerical simulators should start with a problem for which exact solutions exist so that the numerical algorithm can be verified. Steady, creeping flow has many exact solutions and thus it was used as the starting point for developing a numerical simulator for the Navier-Stokes equations. The student should verify that their code is able to duplicate exact solutions to any desired degree of accuracy.

The next stage in the development of the code is to add transients and finite Reynolds numbers. Finite Reynolds number results in nonlinear terms in the Navier-Stokes equation. Algorithms for solving linear systems of equations such as the conjugate gradient methods solves linear systems in one call to the routine. Nonlinear terms need to be iterated or lagged during the transient solution. Thus nonlinear steady state flow may be solved as the evolution of a transient solution from initial conditions to steady state.

The transient and nonlinear terms now need to be included when making the vorticity equation dimensionless.

$$\frac{dw}{dt} = \frac{\partial w}{\partial t} + v_x \frac{\partial w}{\partial x} + v_y \frac{\partial w}{\partial y} = \frac{\mu}{\rho}\left[\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right]$$

$$t^* = \frac{t}{t_o}$$

$$\frac{\partial w^*}{\partial t^*} + \left[\frac{U\, t_o}{L_x}\right] v_x^* \frac{\partial w^*}{\partial x^*} + \alpha \left[\frac{U\, t_o}{L_x}\right] v_y^* \frac{\partial w^*}{\partial y^*} = \left[\frac{\mu\, t_o}{\rho\, L_x^2}\right]\left[\frac{\partial^2 w^*}{\partial x^{*2}} + \alpha^2 \frac{\partial^2 w^*}{\partial y^{*2}}\right]$$

Two of the dimensionless groups can be set to unity by expressing the reference time in terms of the ratio of the characteristic velocity and characteristic length. The remaining dimensionless group is the Reynolds number.

$$\left[\frac{U\, t_o}{L_x}\right] = 1 \quad \Rightarrow t_o = \frac{L_x}{U}$$

$$\left[\frac{\mu\, t_o}{\rho\, L_x^2}\right] = \left[\frac{\mu}{\rho U\, L_x}\right] = \frac{1}{\mathrm{Re}}$$

The dimensionless equation for vorticity with the * dropped is

$$Re\left[\frac{\partial w}{\partial t} + v_x \frac{\partial w}{\partial x} + \alpha\, v_y \frac{\partial w}{\partial y}\right] = \frac{\partial^2 w}{\partial x^2} + \alpha^2 \frac{\partial^2 w}{\partial y^2}$$

The convective derivative can be written in conservative form by use of the continuity equation.

$$\left(v_i\, w_j\right)_{,i} = v_{i,i}\, w_j + v_i\, w_{j,i}$$

but

$$v_{i,i} = 0, \quad \text{for incompressible flow}$$

thus

$$v_i\, w_{j,i} = \left(v_i\, w_j\right)_{,i}$$

or

$$\mathbf{v}\,\nabla\mathbf{w} = \nabla \bullet \mathbf{v}\,\mathbf{w}$$

$$Re\left[\frac{\partial w}{\partial t} + \frac{\partial v_x w}{\partial x} + \alpha\frac{\partial v_y w}{\partial y}\right] = \frac{\partial^2 w}{\partial x^2} + \alpha^2\,\frac{\partial^2 w}{\partial y^2}$$

We will first illustrate how to compute the transient, linear problem before tackling the nonlinear terms. Stability of the transient finite difference equations is greatly improved if the spatial differences are evaluated at the new, *n+1*, time level while the time derivative is evaluated with a backward-difference method.

$$\frac{\partial w}{\partial t} \approx \frac{w_{i,j}^{n+1} - w_{i,j}^{n}}{\Delta t} + O(\Delta t) = \frac{\Delta w_{i,j}}{\Delta t} + O(\Delta t)$$
$$\Delta w_{i,j} = w_{i,j}^{n+1} - w_{i,j}^{n}$$

Recall that the finite difference form of the steady state equations were expressed as a system of equations with coefficients, *a, b*, , ect. The vorticity equation for the *i,j* grid point will be rewritten but now with the transient term included. It is convenient to solve for Δw rather than w^{n+1} at each time step.

$$a\,\Delta w_{i+1,j}^{n+1} + b\Delta\, w_{i-1,j}^{n+1} + c\,\Delta\, w_{i,j+1}^{n+1} + d\,\Delta w_{i,j-1}^{n+1} + e\,\Delta w_{i,j}^{n+1} - \frac{\delta^2\,\mathrm{Re}}{\Delta t}\Delta w_{i,j}^{n+1}$$
$$= f - a\, w_{i+1,j}^{n} - b\, w_{i-1,j}^{n} - c\;\, w_{i,j+1}^{n} - d\; w_{i,j-1}^{n} - e\, w_{i,j}^{n}$$

We can see that this equation is of the same form as before with only the coefficients *e* and *f* of the vorticity equation modified.

$$e' = e - \frac{\delta^2 Re}{\Delta t}$$
$$f' = f - a\, w_{i+1,j}^{n} - b\, w_{i-1,j}^{n} - c\;\, w_{i,j+1}^{n} - d\; w_{i,j-1}^{n} - e\, w_{i,j}^{n}$$

Since the variable in the linear system of equations is now Δw rather than *w*, the equations for the stream function also need to be modified.

$$a_{1,1}\,\psi_{i+1,j} + b_{1,1}\,\psi_{i-1,j} + c_{1,1}\,\psi_{i,j+1} + d_{1,1}\,\psi_{i,j+1} + e_{1,1}\,\psi_{i,j} + e_{1,2}\,\Delta w_{i,j}$$
$$= f_{1,1} - e_{1,2}\,w_{i,j}^{n}$$
thus
$$f' = f_{1,1} - e_{1,2}\,w_{i,j}^{n}$$

These modifications to the coefficients should be made after the coefficients are updated for convection and boundary conditions.

The numerical calculations will start from some initial value of vorticity and proceed to steady state. The size of the time step is important if accuracy of the transient solution is of interest. The truncation error of the time finite difference expression is approximately the product $\Delta t\,\dfrac{\partial^2 w}{\partial t^2}$. The time step size can be chosen as to keep this value approximately constant. Thus the time step size will be chosen as to limit the maximum change in the magnitude of *w* over a time step. Let Δw_{max} be the maximum change in w over the previous time step and Δw_{spec} be the specified value of the desired maximum change in *w*. The new time step can be estimated from the following expression.

$$\Delta t_{new} = \Delta t_{old}\,\frac{\Delta w_{spec}}{\Delta w_{\max}}$$

The initial time step size needs to be specified, and the new time step may be averaged with the old or constrained to increase by not greater than some factor.

Now that we have a means of calculating the transients solution, this approach can be used to treat the nonlinear terms. Recall that when the equations were linear, the system of equations had constant coefficients and the conjugate-gradient linear solver solved the linear system of equations.

$$\mathbf{A}\,\mathbf{x} = \mathbf{F}$$

The convective derivatives have a product of velocity and vorticity, which can be expressed as a product of the derivative of stream function and vorticity. One approximation would be to use the value of the stream function from the old time step to calculate the coefficients for the new time step. An alternative is use a predictor-corrector approach, similar to the Runge-Kutta solution for quasilinear ordinary differential equations. The "predictor" step estimates the solution at the new time step using the coefficients calculated from the stream function of the previous time step. This gives an estimate of the stream function at the new time step. This estimated stream function is then used to evaluate the coefficients for the "corrector" step. The steps in the calculations are as follows.

$$\mathbf{A}(\mathbf{x}^{n})\,\mathbf{x}^{n+1*} = \mathbf{F}$$
$$\mathbf{A}(\mathbf{x}^{n+1*})\,\mathbf{x}^{n+1} = \mathbf{F}$$

The choice of the finite difference expression for the convective derivative is very important for the stability of the solution. We will illustrate here treatment of only the *x* derivative. Suppose *i-1*, *i*, and *i*+1 are the grid points where the stream function and vorticity are evaluated and *i* – ½ and *i*+ ½ are the mid-points where the velocity-vorticity products are evaluated.

$$\frac{\partial u\,w}{\partial x} \approx \frac{\left(u\,w\right)_{i+1/2} - \left(u\,w\right)_{i-1/2}}{\Delta x}$$

The dependent variables are known only at the grid points and interpolation will be needed to evaluate in between location. If the product is evaluated at *i* – ½ and *i*+ ½ by using the arithmetic average of the values at the grid points on either side, the numerical solution will tend to oscillate if the convective transport dominates the diffusive (viscous) transport. It is preferable to determine the upstream direction from the sign of $u_{i+1/2}$ or $u_{i-1/2}$ and use the vorticity from the upstream grid point. This is illustrated below.

```
$u_{i-1/2}$ = 0.5*(u(i,j)+u(i-1,j))
if $u_{i-1/2}$ >0 then
        $(uw)_{i-1/2} = u_{i-1/2}\, w_{i-1}$ ;          $b'=b+\delta$ Re $u_{i-1/2}$
else
        $(uw)_{i-1/2} = u_{i-1/2}\, w_{i}$ ;            $e'=e+\delta$ Re $u_{i-1/2}$
endif

$u_{i+1/2}$ = 0.5*(u(i,j)+u(i+1,j))
if $u_{l+1/2}$ >0 then
        $(uw)_{i+1/2} = u_{i+1/2}\, w_{i}$ ;            $e'=e - \delta$ Re $u_{i+1/2}$
else
        $(uw)_{i+1/2} = u_{i+1/2}\, w_{i+1}$ ;          $a'=a - \delta$ Re $u_{i+1/2}$
endif
```

The y-direction will be similar except the aspect ratio, α, must be included. These modifications to the coefficients must be made before the coefficients are updated for the boundary conditions.

Calculation of Pressure

The vorticity-stream function method does not require calculation of pressure to determine the flow field. However, if the force or drag on a body or a conduit is of interest, the pressure must be computed to determine the stress field. Here we will derive the Poisson equation for pressure and determine the boundary conditions using the equations of motion. If the pressure is desired only at the boundary, it may be possible to integrate the pressure gradient determined from the equations of motion. The dimensionless equations of motion for incompressible flow of a Newtonian fluid are as follows.

$$\frac{d\mathbf{v}*}{dt*} = \frac{\partial \mathbf{v}*}{\partial t*} + \mathbf{v}*\bullet\nabla*\mathbf{v}* = \frac{\partial \mathbf{v}*}{\partial t*} + \nabla*\bullet(\mathbf{v}*\mathbf{v}*) =$$

$$-\nabla*P*+\frac{1}{Re}\nabla*^2\,\mathbf{v}*$$

where

$$\mathbf{v}* = \frac{\mathbf{v}}{U},\quad \mathbf{x}* = \frac{\mathbf{x}}{L},\quad t* = \frac{U}{L}t,\quad P* = \frac{P}{\rho U^2},\quad \nabla* = L\nabla$$

$$P = p - \rho\, g\, z,\quad Re = \frac{\rho U\, L}{\mu}$$

Here all coordinates were scaled with respect to the same length scale. The aspect ratio must be included in the final equation if the coordinates are scaled with respect to different length scales. We will drop the * and use *x* and *y* for the coordinates and *u* and *v* as the components of velocity.

The following derivation follows that of Hoffmann and Chiang (1993). The equations of motion in 2-D in conservative form is as follows.

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (u\,v)}{\partial y} = -\frac{\partial P}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + \frac{\partial (u\,v)}{\partial x} + \frac{\partial (v^2)}{\partial y} = -\frac{\partial P}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

The Laplacian of pressure is determined by taking the divergence of the equations of motion. We will carry out the derivation step wise by first taking the *x*-derivative of the *x*-component of the equations of motion.

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) + 2\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + 2u\frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + u\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + v\frac{\partial^2 u}{\partial x \partial y} = -\frac{\partial^2 P}{\partial x^2} + \frac{1}{Re}\frac{\partial}{\partial x}\left(\nabla^2 u\right)$$

Two pair of terms cancels because of the equation of continuity for incompressible flow.

$$\frac{\partial u}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0$$

$$u\frac{\partial^2 v}{\partial x \partial y} + u\frac{\partial^2 u}{\partial x^2} = u\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0$$

Thus

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) + \left(\frac{\partial u}{\partial x}\right)^2 + u\frac{\partial^2 u}{\partial x^2} + \frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + v\frac{\partial^2 u}{\partial x \partial y} = -\frac{\partial^2 P}{\partial x^2} + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x}\left(\nabla^2 u\right)$$

Similarly, the y-component of the equations of motion become

$$\frac{\partial}{\partial t}\left(\frac{\partial v}{\partial y}\right)+\left(\frac{\partial v}{\partial y}\right)^2+v\frac{\partial^2 v}{\partial y^2}+\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}+u\frac{\partial^2 v}{\partial x\partial y}=-\frac{\partial^2 P}{\partial y^2}+\frac{1}{Re}\frac{\partial}{\partial y}\left(\nabla^2 v\right)$$

The *x* and *y*-components of the above equations are now added together and several pairs of terms cancels pair-wise.

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0$$

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x\partial y}=\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0$$

$$\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 v}{\partial y^2}=\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0$$

$$\frac{\partial}{\partial x}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{\partial}{\partial y}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)=\frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+\frac{\partial^2}{\partial y^2}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)=0$$

Therefore the equations reduce to

$$\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}=-\left(\frac{\partial^2 P}{\partial x^2}+\frac{\partial^2 P}{\partial y^2}\right)$$

The left-hand side can be further reduced by consideration of the continuity equation as follows.

$$\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)^2=\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+2\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}=0$$

from which

$$\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2=-2\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}$$

Upon substitution, we have

$$-\left(\frac{\partial^2 P}{\partial x^2}+\frac{\partial^2 P}{\partial y^2}\right)=2\left(\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}-\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right)$$

This equation can be written in terms of the stream function.

$$\frac{\partial^2 P}{\partial x^2}+\frac{\partial^2 P}{\partial y^2}=2\left[\frac{\partial^2\psi}{\partial x^2}\frac{\partial^2\psi}{\partial y^2}-\left(\frac{\partial^2\psi}{\partial x\partial y}\right)^2\right]$$

This is the equation when the coordinates have the same reference length. If the coordinates are normalized with respect to different lengths, then the equation is as follows.

$$\frac{\partial^2 P}{\partial x^2}+\alpha^2\frac{\partial^2 P}{\partial y^2}=2\alpha^2\left[\frac{\partial^2\psi}{\partial x^2}\frac{\partial^2\psi}{\partial y^2}-\left(\frac{\partial^2\psi}{\partial x\partial y}\right)^2\right]$$

The Poisson equation for pressure needs to have boundary conditions for solution. The equations of motion give an expression for the pressure gradient. The normal derivative of pressure at the boundary can be determined for the Neumann-type boundary condition. The equations of motion usually simplify at boundaries if the boundary has no slip BC or if it has parallel flow in the direction either parallel or normal to the boundary. For example, for no-slip

$$\frac{\partial P}{\partial x}=\frac{\alpha^2}{Re}\frac{\partial^2 v}{\partial y^2},\quad \text{at } y=c, u=v=0$$

$$\alpha\frac{\partial P}{\partial y}=\frac{1}{Re}\frac{\partial^2 u}{\partial x^2},\quad \text{at } x=c, u=v=0$$

If the flow is parallel and in the direction normal to the boundary

$$\frac{\partial P}{\partial x}=\frac{\alpha^2}{Re}\frac{\partial^2 u}{\partial y^2},\quad y=c, v=0, u=u(y)$$

$$\alpha\frac{\partial P}{\partial y}=\frac{1}{Re}\frac{\partial^2 v}{\partial x^2},\quad x=c, u=0, v=v(x)$$

Cylindrical-Polar Coordinates

The code for the numerical solution of the Navier-Stokes equations in Cartesian coordinates can be easily modified to cylindrical-polar coordinates. The coordinate transformation is first illustrated for the Laplacian operator.

$$\nabla^2\psi=\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\psi}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2\psi}{\partial\theta^2},\qquad r_1\le r\le r_2,\quad 0\le\theta\le\theta_o$$

The independent variables are made dimensionless with respect to the boundary parameters.

$$r^* = \frac{r}{r_1}, \quad \theta^* = \frac{\theta}{\theta_o}, \qquad 1 \le r^* \le \frac{r_2}{r_1} = \beta, \quad 0 \le \theta^* \le 1$$

The Lapacian operator with the dimensionless coordinates after dropping the * is now,

$$\nabla^2 \psi = \frac{1}{r_1^2}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \psi}{\partial r}\right) + \frac{1}{\theta_o^2}\frac{1}{r^2}\frac{\partial^2 \psi}{\partial \theta^2}\right], \qquad 1 \le r \le \beta, \quad 0 \le \theta \le 1$$

The radial coordinate is transformed to the logarithm of the radius.

$$z = \frac{\ln r}{\ln \beta} = \gamma \ln r, \qquad 0 \le z \le 1, \quad \gamma = \frac{1}{\ln\left(r_2 / r_1\right)}$$

$$r = \exp\left(z / \gamma\right)$$

$$\frac{\partial}{\partial r} = \frac{dz}{dr}\frac{\partial}{\partial z} = \frac{\gamma}{r}\frac{\partial}{\partial z}$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \psi}{\partial r}\right) = \frac{\gamma^2}{r^2}\frac{\partial^2 \psi}{\partial z^2}$$

The Laplacian operator is now as follows,

$$\nabla^2 \psi = \frac{1}{r_1^2}\left[\frac{\gamma^2}{r^2}\frac{\partial^2 \psi}{\partial z^2} + \frac{\alpha^2}{r^2}\frac{\partial^2 \psi}{\partial \theta^2}\right], \quad 0 \le z \le 1, \quad 0 \le \theta \le 1, \quad \alpha = \frac{1}{\theta_o}$$

The finite difference expression for the Laplacian will be same as that for Cartesian coordinates with (z,θ) substituted for (*x*,*y*) except for γ^2 and r^2 factors.

$$0 \le z_i \le 1, \quad 0 \le \theta_j \le 1, \quad i, j = 1, 2, \ldots, JMAX$$

$$\delta = \frac{1}{JMAX - 1}$$

$$z_i = \delta\left(i - 1\right), \quad \theta_j = \delta\left(j - 1\right)$$

The curl operator is modified from that in Cartesian coordinates.

$$
\begin{aligned}
v_r &= \frac{\alpha}{r}\frac{\partial \psi}{\partial \theta} \\
v_\theta &= -\frac{\partial \psi}{\partial r} = -\frac{\gamma}{r}\frac{\partial \psi}{\partial z} \\
w &= \frac{1}{r}\frac{\partial \left(r\, v_\theta \right)}{\partial r} - \frac{\alpha}{r}\frac{\partial v_r}{\partial \theta} \\
&= \frac{\gamma}{r^2}\frac{\partial \left(r\, v_\theta \right)}{\partial z} - \frac{\alpha}{r}\frac{\partial v_r}{\partial \theta} \\
&= -\frac{\gamma^2}{r^2}\frac{\partial^2 \psi}{\partial z^2} - \frac{\alpha^2}{r^2}\frac{\partial^2 \psi}{\partial \theta^2}
\end{aligned}
$$

The vorticity boundary conditions with the transformed coordinates is for the *z* boundary,

$$
w_1^{BC} = -\frac{\gamma^2}{r^2}\frac{2}{\delta^2}\left(\psi_2 - \psi_1^{BC} \right) \mp \frac{\gamma}{r}\frac{2}{\delta} v_\theta^{BC} - \frac{\alpha}{r}\frac{\partial v_r^{BC}}{\partial \theta}
$$

and for the θ boundary,

$$
w_1^{BC} = -\frac{\alpha^2}{r^2}\frac{2}{\delta^2}\left(\psi_2 - \psi_1^{BC} \right) \pm \frac{\alpha}{r}\frac{2}{\delta} v_r^{BC} + \frac{\gamma}{r^2}\frac{\partial \left(r v_\theta^{BC} \right)}{\partial z}
$$

The stream function at the boundaries are expressed different from that in Cartesian coordinates.

$$
\begin{aligned}
d\psi &= \left(r\, v_r / \alpha \right) d\theta, \quad \text{at } z \text{ boundary} \\
d\psi &= -\left(r\, v_\theta / \gamma \right) dz, \quad \text{at } \theta \text{ boundary}
\end{aligned}
$$

The convective terms are expressed different from that in Cartesian coordinates.

$$\begin{aligned}\frac{\partial(u\,w)}{\partial x} &\Rightarrow \frac{1}{r}\frac{\partial(r\,u\,w)}{\partial r}\\ &= \frac{\gamma}{r^2}\frac{\partial(r\,u\,w)}{\partial z}\\ &\approx \frac{\gamma}{r^2\,\delta}\left[(r\,u\,w)_{i+1/2}-(r\,u\,w)_{i-1/2}\right]\\ \alpha\frac{\partial(v\,w)}{\partial y} &\Rightarrow \frac{\alpha}{r}\frac{\partial(v\,w)}{\partial\theta}\\ &\approx \frac{\alpha}{r\,\delta}\left[(v\,w)_{j+1/2}-(v\,w)_{j-1/2}\right]\end{aligned}$$

The code should be written so one can have either Cartesian coordinates or transformed cylindrical-polar coordinates. A parameter will be needed to identify the choice of coordinates, e.g. *icase*=1 for Cartesian coordinates and *icase*=2 for transformed cylindrical-polar coordinates. Also, another parameter should be specified to identify the choice of boundary conditions, e.g., *ibc*=1 for radial flow, *ibc*=2 for Couette flow, and *ibc*=3 for flow around a cylinder.

Test cases with known solutions should be used to verify the code. The first case is radial, potential flow from a line source and the second is Couette flow in the annular region between two cylindrical surfaces.

$$\left.\begin{aligned} v_r &= 1/r\\ v_\theta &= 0\end{aligned}\right\}\ \text{radial flow}$$

$$\left.\begin{aligned} v_r &= 0\\ v_\theta &= (r-1/r)/(\beta-1/\beta)\end{aligned}\right\}\ \text{Couette flow}$$

Flow around a cylinder needs a boundary condition for the flow far away from the cylinder. The flow very far away may be uniform translation. However, this condition may be so far away that it may result in loss of resolution near the cylinder. Another boundary condition that may be specified beyond the region of influence of the boundary layer is to use the potential flow past a cylinder. This boundary condition will not be correct in the wake of cylinder where the flow is disturbed by the convected boundary layer buts its influence may be minimized if the outer boundary is far enough.

$$\left.\begin{aligned} v_r &= \left(1-1/r^2\right)\,\cos\theta\\ v_\theta &= -\left(1+1/r^2\right)\,\sin\theta\end{aligned}\right\}\ \text{Potential flow past cylinder}$$

Potential flow will not be a valid approximation along the θ boundaries close to the cylinder. Here, one may assume a surface of symmetry, at least for the upstream side.