

Chapter 7

Incompressible Flow Solutions

Incompressible flows are by far the most common type of flows encountered in engineering problems. They are different than compressible flows mainly due to the missing equation of state. Density is not an unknown and pressure does not have any thermodynamic meaning. In an incompressible flow the role of pressure is to adjust itself immediately to the changes in a flow field so that the velocity is divergence free at all times. These differences make the numerical solution of incompressible flows more challenging compared to compressible flows, not only for FEM, but for other numerical techniques as well.

In CFD literature mass and momentum conservation equations together are called Navier-Stokes (N-S) equations. N-S equations are simplified into Stokes equations when the inertia effects are negligible as in the case of creeping flows. In this chapter first the FE formulation of Stokes equations will be presented, followed by the extension to N-S equations. Formulation of heat transfer problems for which the energy equation also needs to be solved will be done in the next chapter.

7.1 Primitive Variable Formulation of Incompressible Flows

Velocity components and pressure are known as the primitive variables. Although they are not the only choice of variables that can be used to formulate incompressible flows, they are the most commonly used ones. N-S equations in primitive formulation are given as

$$\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \vec{V} + \vec{f} \quad (7.1)$$

$$\nabla \cdot \vec{V} = 0 \quad (7.2)$$

where p is the pressure, \vec{V} is the velocity vector, ρ and μ are the constant density and dynamic viscosity of the fluid and \vec{f} is the body force per unit mass. This velocity and pressure based formulation is also known as mixed formulation.

The most common alternative for primitive variable formulation is the stream function-vorticity formulation, in which the pressure is no longer an unknown. Although it has computational advantages over primitive formulation in 2D, its extension to 3D problems and specification of the BCs is problematic.

N-S equations are nonlinear due to the inertial term. For very low Reynolds number cases (low speed flows and/or highly viscous fluids) this term is negligibly small compared to the viscous term and it drops from the equation. The resulting set of equations is linear and called Stokes Equations. In the

coming sections we'll first consider GFEM formulation of linear Stokes equations and then we'll include the nonlinear term.

Equations (7.1) and (7.2) should be supported by initial and boundary conditions. A divergence free velocity distribution should be provided as an initial condition. Boundary conditions can be of two types, specifying velocity components or specifying boundary traction as given below

$$\text{Velocity (Dirichlet) BC} \quad : \quad \vec{V} = \vec{V}_0 \quad \text{on} \quad \Gamma_D \quad (7.3a)$$

$$\text{Traction (Neumann) BC} \quad : \quad \vec{n} \cdot \vec{\sigma} = \vec{t} \quad \text{on} \quad \Gamma_N \quad (7.3b)$$

where \vec{n} is the unit outward normal vector of the boundary, $\vec{\sigma}$ is the stress tensor which is the sum of normal and shear stresses and \vec{t} is the traction force applied by the boundary on the fluid.

For fluid flows boundary condition at a solid wall is known as no slip BC, i.e. normal and tangential velocity components of the fluid are equated to those of the solid wall. For the common case of a stationary wall, both velocity components are equated to zero. No pressure BC is specified at solid walls. Inflow boundaries are treated in a similar way as solid walls, i.e. EBCs for velocity components are specified and no BC for pressure is necessary. Specification of BCs at outflow boundaries is not a completely resolved issue and it is possible to see different practices in the literature. Traction type BC should be specified at outflow boundaries, but the difficulty is that the required tractions are usually not known at an outflow boundary. It is common to see the use of simpler BCs at outflow boundaries, such as the specification of a constant pressure or the “do nothing” approach.

7.2 GFEM of 2D Stokes Equations in Primitive Variables

N-S equations are nonlinear due to the convective term of Eqn (7.1). It is easier to start with the Stokes equation for which this term drops. 2D, steady Stokes equations written in the Cartesian coordinate system are as follows

$$\text{x-Momentum} \quad : \quad -\mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial p}{\partial x} - \rho f_x = 0 \quad (7.4a)$$

$$\text{y-Momentum} \quad : \quad -\mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{\partial p}{\partial y} - \rho f_y = 0 \quad (7.4b)$$

$$\text{Continuity} \quad : \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (7.4c)$$

where u and v are the x - and y - components of the velocity vector and f_x and f_y are the components of the body force vector. It is possible to put Eqns (7.4a) and (7.4b) into the following form by the help of the continuity equation

$$\text{x-Momentum} \quad : \quad -\frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \frac{\partial p}{\partial x} - \rho f_x = 0 \quad (7.5a)$$

$$\text{y-Momentum : } -\frac{\partial}{\partial y}\left(2\mu\frac{\partial v}{\partial x}\right) - \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + \frac{\partial p}{\partial y} - \rho f_y = 0 \quad (7.5b)$$

which is more suitable to derive the weak form of the Stokes equations, as far as the physical meaning of the SVs are considered. To obtain the weak form of the Stokes equations, we first form the weighted integral forms of Eqns (7.5a), (7.5b) and (7.4c) and apply integration by parts to the terms of the momentum equation with second derivatives as well as the pressure terms so that physically meaningful SVs can be obtained. The resulting elemental weak form is

$$\int_{\Omega^e} \left[2\mu \frac{\partial w_x}{\partial x} \frac{\partial u}{\partial x} + \mu \frac{\partial w_x}{\partial y} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) - \frac{\partial w_x}{\partial x} p \right] d\Omega = \int_{\Omega^e} \rho w_x f_x d\Omega + \int_{\Gamma^e} w_x t_x d\Gamma \quad (7.6a)$$

$$\int_{\Omega^e} \left[2\mu \frac{\partial w_y}{\partial y} \frac{\partial v}{\partial y} + \mu \frac{\partial w_y}{\partial x} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) - \frac{\partial w_y}{\partial y} p \right] d\Omega = \int_{\Omega^e} \rho w_y f_y d\Omega + \int_{\Gamma^e} w_y t_y d\Gamma \quad (7.6b)$$

$$\int_{\Omega^e} -w_c \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) d\Omega = 0 \quad (7.6c)$$

Minus sign is added to the continuity equation on purpose, in order to get a symmetric stiffness matrix at the end. To make the formulation more general different weight functions (w_c , w_x and w_y) are used for each equation. Boundary integrals that are the by-products of integration by parts include the following traction terms

$$t_x = \left(2\mu \frac{\partial u}{\partial x} - p \right) n_x + \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) n_y \quad (7.7a)$$

$$t_y = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) n_x + \left(2\mu \frac{\partial v}{\partial y} - p \right) n_y \quad (7.7b)$$

where n_x and n_y are the Cartesian components of the unit outward normal vector at a boundary. These tractions are the secondary variables of the N-S equations. Corresponding primary variables are the two velocity components, u and v , which can be associated with the x - and y - components of the momentum equation. The remaining unknown, which is pressure, can be associated with the continuity equation, however it does not even appear in this equation. No integration by parts is applied to the continuity equation and there is no boundary integral for it. For incompressible Stokes equations pressure is neither a primary nor a secondary variable by itself, but it appears in the SVs associated with the momentum equations. This behavior of pressure creates a major challenge in the numerical solution of incompressible flows.

Again for the generality of the formulation we'll assume that different order of polynomials are used to approximate velocity and pressure unknowns, i.e.

$$\begin{aligned}
 u^h(x, y) &= \sum_{j=1}^{NENv} u_j S_j(x, y) \\
 v^h(x, y) &= \sum_{j=1}^{NENv} v_j S_j(x, y) \\
 p^h(x, y) &= \sum_{j=1}^{NENp} p_j \hat{S}_j(x, y)
 \end{aligned} \tag{7.8}$$

where $NENv$ and $NENp$ are the number of velocity and pressure nodes over an element, which can be different as shown in Figure 7.1.

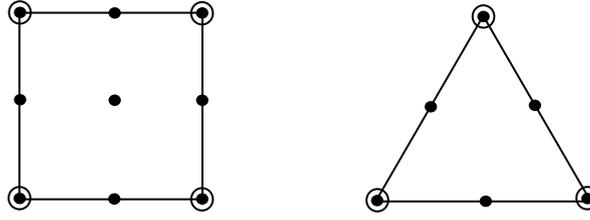


Figure 7.1 Typical quadrilateral and triangular elements with $NENv > NENp$. Circles and dots represent the points at which velocity components and pressure are stored, respectively.

If $NENv$ and $NENp$ are different, then different shape functions need to be used for velocity components and pressure, and they are denoted by S and \hat{S} . In GFEM formulation weight functions are selected to be the same as the shape functions as shown below

$$w_c = \hat{S}_i \quad , \quad w_x = w_y = S_i \tag{7.9}$$

Substituing Eqns (7.8) and (7.9) into (7.6) we get

$$\begin{aligned}
 \sum_{j=1}^{NENv} \left[\int_{\Omega^e} \left(2\mu \frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial x} + \mu \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial y} \right) d\Omega \right] u_j + \sum_{j=1}^{NENv} \left(\int_{\Omega^e} \mu \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial x} d\Omega \right) v_j \\
 + \sum_{j=1}^{NENp} \left(\int_{\Omega^e} -\frac{\partial S_i}{\partial x} \hat{S}_j d\Omega \right) p_j = \int_{\Omega^e} \rho S_i f_x d\Omega + \int_{\Gamma^e} S_i t_x d\Gamma
 \end{aligned} \tag{7.10a}$$

$$\begin{aligned}
 \sum_{j=1}^{NENv} \left(\int_{\Omega^e} \mu \frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial y} d\Omega \right) u_j + \sum_{j=1}^{NENv} \left[\int_{\Omega^e} \left(2\mu \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial y} + \mu \frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial x} \right) d\Omega \right] v_j \\
 + \sum_{j=1}^{NENp} \left(\int_{\Omega^e} -\frac{\partial S_i}{\partial y} \hat{S}_j d\Omega \right) p_j = \int_{\Omega^e} \rho S_i f_y d\Omega + \int_{\Gamma^e} S_i t_y d\Gamma
 \end{aligned} \tag{7.10b}$$

$$\sum_{j=1}^{NENv} \left(\int_{\Omega^e} -\hat{S}_i \frac{\partial S_j}{\partial x} d\Omega \right) u_j + \sum_{j=1}^{NENv} \left(\int_{\Omega^e} -\hat{S}_i \frac{\partial S_j}{\partial x} d\Omega \right) v_j = 0 \tag{7.10c}$$

These elemental equations can be written in the following compact form

$$[K]^e \{U\}^e = \{F\}^e \quad (7.11)$$

which can be detailed as follows

$$\begin{bmatrix} [K^{11}] & [K^{12}] & [K^{13}] \\ [K^{21}] & [K^{22}] & [K^{23}] \\ [K^{31}] & [K^{32}] & [K^{33}] \end{bmatrix}^e \begin{Bmatrix} \{u\} \\ \{v\} \\ \{p\} \end{Bmatrix}^e = \begin{Bmatrix} \{F^1\} \\ \{F^2\} \\ \{F^3\} \end{Bmatrix}^e \quad (7.12)$$

where

$$\begin{aligned} K_{ij}^{11} &= \int_{\Omega^e} \left[2\mu \frac{\partial S_i^e}{\partial x} \frac{\partial S_j^e}{\partial x} + \mu \frac{\partial S_i^e}{\partial y} \frac{\partial S_j^e}{\partial y} \right] d\Omega & K_{ij}^{31} &= K_{ji}^{13} \\ K_{ij}^{12} &= \int_{\Omega^e} \mu \frac{\partial S_i^e}{\partial y} \frac{\partial S_j^e}{\partial x} d\Omega & K_{ij}^{32} &= K_{ji}^{23} \\ K_{ij}^{13} &= \int_{\Omega^e} -\frac{\partial S_i^e}{\partial x} \hat{S}_j^e d\Omega & K_{ij}^{33} &= 0 \\ K_{ij}^{21} &= K_{ji}^{12} & F_i^1 &= \int_{\Omega^e} \rho S_i^e f_x d\Omega + \int_{\Gamma^e} S_i^e t_x d\Gamma \\ K_{ij}^{22} &= \int_{\Omega^e} \left[\mu \frac{\partial S_i^e}{\partial x} \frac{\partial S_j^e}{\partial x} + 2\mu \frac{\partial S_i^e}{\partial y} \frac{\partial S_j^e}{\partial y} \right] d\Omega & F_i^2 &= \int_{\Omega^e} \rho S_i^e f_y d\Omega + \int_{\Gamma^e} S_i^e t_y d\Gamma \\ K_{ij}^{23} &= \int_{\Omega^e} -\frac{\partial S_i^e}{\partial y} \hat{S}_j^e d\Omega & F_i^3 &= 0 \end{aligned} \quad (7.13)$$

At this point it is important to consider the following remarks

- For an element with $NENv$ velocity nodes and $NENp$ pressure nodes, there are totally $NEU = 2*NENv + NENp$ elemental unknowns. Therefore the size of $[K]^e$ is $NEU \times NEU$. For such an element, sizes of elemental sub-matrices and sub-vectors are

$$\begin{array}{ll} \{u\}, \{v\} & : NENv \times 1 \\ \{p\} & : NENp \times 1 \\ [K^{11}], [K^{12}], [K^{21}], [K^{22}] & : NENv \times NENv \\ [K^{13}], [K^{23}] & : NENv \times NENp \\ [K^{31}], [K^{32}] & : NENp \times NENv \\ [K^{33}] & : NENp \times NENp \end{array}$$

- $[K^{33}]$ is a zero matrix. Therefore the elemental system has zero entries on its main diagonal. Similarly the assembled stiffness matrix will also have zeros on its main diagonal.

- The elemental system, and therefore the assembled global system is symmetric (This symmetry will be lost when nonlinear terms are added in the N-S equations).
- If all boundary conditions are of Dirichlet type, i.e. only velocity components are specified, pressure can be determined only up to an arbitrary constant because in the DEs (Eqn (7.1)) it is only present by its gradient. In such a case, in order to define the pressure uniquely, the usual practice is to provide the value of pressure at one point of the flow domain or impose an average pressure value for the whole flow domain.
- Pressure is not a primary variable and its derivatives do not appear in the in the weak form of the problem (Eqn (7.6)). Therefore it is possible to approximate pressure to be discontinuous across neighboring elements (using a single pressure node at the center of the element), which is a common choice.
- **Unsteady terms can be included using semi-discrete formulation discussed in Chapter 4. Resultant mass matrix will be (This will be discussed later. Remove it from here)**

$$[M]^e = \begin{bmatrix} [M^{11}] & [0] & [0] \\ [0] & [M^{22}] & [0] \\ [0] & [0] & [0] \end{bmatrix}^e \quad \text{where} \quad M^{11} = M^{22} = \int_{\Omega^e} \rho S_i^e S_j^e d\Omega \quad (7.14)$$

7.3 Numerical Challenges of Solving Incompressible Flow Equations

Three main difficulties for the numerical solution of incompressible flow equations are

- Mixed formulation presented in the previous section resulted in a global stiffness matrix (and global mass matrix for unsteady problems) with zeros on the main diagonal. Although these global systems can be solved using direct methods (such as Gauss Elimination or LU decomposition) with proper pivoting, iterative methods are preferred for large problems due to efficiency concerns. Unfortunately, iterative techniques become ineffective due to the presence of zeros on the diagonal.
- Similar to the advection diffusion equation discussed in the previous chapter, for convection dominated flows, i.e. high Reynolds number flows, GFEM may provide results that have unphysical (spurious) node-to-node oscillations, especially when used with a not fine enough mesh. Literature is full of various stabilization techniques such as Streamline Upwind Petrov Galerkin (SUPG), Galerkin Least Squares (GLS), Pressure Stabilized Petrov Galerkin (PSPG), etc. to overcome this. These stabilization techniques in general modify GFEM so that there are no longer zero entries on the main diagonal of the algebraic system of equations, which is an additional benefit of using them.
- When primitive variables are used, the algebraic system obtained from GFEM should satisfy an extra compatibility condition, known as Ladyzhenskaya-Babuska-Brezzi (LBB) condition or

the inf-sup condition so that unphysical node-to-node pressure oscillations (checkerboard pressure distribution) can be avoided. According to the LBB condition pressure approximation must be at least one order lower than the velocity approximation over an element (NEN_p should be less than NEN_v). There are only a limited number of LBB-stable elements, i.e. elements that are known to satisfy the LBB condition.

Some commonly used elements that satisfy the LBB condition, i.e. LBB-stable elements, are shown in Table 7.1. Although they provide good results, the use of these elements bring complications to programming. Researchers have spent a lot of effort to come up with formulations that allow the use of equal order approximation for both velocity and pressure. Interestingly, stabilization techniques that are used to obtain oscillation free results for highly convective flows are also known to be useful in circumventing the LBB condition (being able to use elements that do not satisfy the LBB condition), which can be seen as the third motivation for using them.

Note that the checkerboard pressure distribution problem is also seen in Finite Difference and Finite Volume Methods, for which people commonly seek solutions by using staggered (not colocated) grids, i.e. grids where pressures and velocities are calculated at different points.

7.4 GLS Stabilization of Stokes Equations for Linear Triangles and Quadrilaterals

Similar to AD equation discussed in previous chapters, Galerkin Least Squares (GLS) or similar residual based stabilization techniques can be applied to Stokes and N-S equations, too [2]. These stabilized formulations have three benefits; i) they enable non-oscillatory solution of high Reynolds number problems with reasonably fine meshes, ii) they enable the use of equal order interpolation for velocity and pressure, i.e. NEN_v = NEN_p, without a checkerboard pressure field, iii) they get rid of the zero diagonal entries of the global system.

As discussed in Chapter 6, GLS adds the following stabilization term to the weak form of the problem

$$\text{GLS Stabilization : } \int_{\Omega^e} \tau \mathcal{L}(w) \cdot R(\vec{V}, p) d\Omega \quad (7.15)$$

where the residual R is based on the momentum equation given by

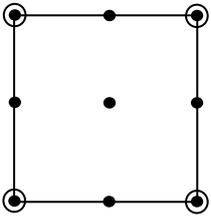
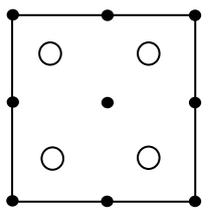
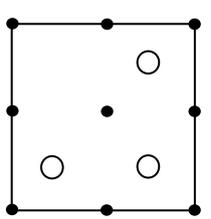
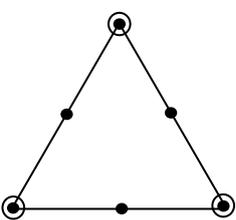
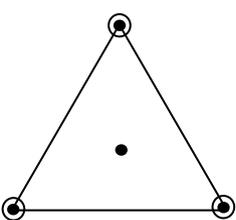
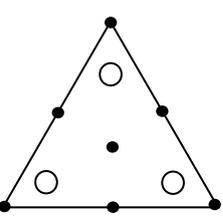
$$R = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \vec{V} + \vec{f} \quad (7.16)$$

and the \mathcal{L} operator for Stokes equation is

$$\mathcal{L}(w) = -\frac{1}{\rho} \nabla w_c + \frac{\mu}{\rho} \nabla^2 \mathbf{w} \quad (7.17)$$

where \mathbf{w} is a vector combination of weight functions w_x and w_y .

Table 7.1 Incomplete list of LBB-stable quadrilateral and triangular elements. Black circles represent velocity nodes, white circles represent pressure nodes [1].

	<p>Q_2Q_1 element (Taylor Hood element) Continuous biquadratic velocity Continuous bilinear pressure $NEN_v = 9, NEN_p = 4$</p>
	<p>Q_2Q_{-1} element Continuous biquadratic velocity Discontinuous pressure $NEN_v = 9, NEN_p = 4$</p>
	<p>Q_2P_{-1} element Continuous biquadratic velocity Discontinuous pressure $NEN_v = 9, NEN_p = 3$</p>
	<p>P_2P_1 (Taylor Hood) element Continuous quadratic velocity Continuous linear pressure $NEN_v = 6, NEN_p = 3$</p>
	<p>$P_1^+P_1$ (Mini) element Continuous quadratic velocity + cubic bubble Discontinuous pressure $NEN_v = 4, NEN_p = 3$</p>
	<p>$P_2^+P_{-1}$ (Crouzeix-Raviart) element Continuous quadratic velocity + cubic bubble Discontinuous linear pressure $NEN_v = 7, NEN_p = 3$</p>

Using Eqns (7.16) and (7.17) in Eqn (7.15) the contribution of the GLS stabilization integral to the momentum and continuity equations will be as follows

$$\text{GLS contribution to x-momentum : } \int_{\Omega^e} (-\mu \nabla^2 w_x) \tau \left(-\mu \nabla^2 u + \frac{\partial p}{\partial x} - \rho f_x \right) d\Omega \quad (7.18a)$$

$$\text{GLS contribution to y-momentum : } \int_{\Omega^e} (-\mu \nabla^2 w_y) \tau \left(-\mu \nabla^2 v + \frac{\partial p}{\partial y} - \rho f_y \right) d\Omega \quad (7.18b)$$

$$\text{GLS contribution to continuity : } \int_{\Omega^e} \tau (\nabla w_c) \cdot (-\mu \nabla^2 \vec{V} + \nabla p - \rho \vec{f}) d\Omega \quad (7.18c)$$

If the velocity approximation over the element is linear, i.e. if 3-node triangular or 4-node quadrilaterals are used for velocity approximation, then the GLS contribution to momentum equations given by (7.18a) and (7.18b) will vanish because the terms with second derivatives, i.e. $\nabla^2 w_x$ and $\nabla^2 w_y$ will be zero. Also $\nabla^2 \vec{V}$ term will be zero for the GLS contribution to the continuity equation and this integral will simplify to

$$\text{GLS contribution to continuity : } \int_{\Omega^e} \tau (\nabla w_c) \cdot (\nabla p - \rho \vec{f}) d\Omega \quad (7.19)$$

Subtracting (not adding because the continuity equation is already multiplied with a minus sign) Eqn (7.19) from Eqn (7.6c) we get the following modified weak form of the continuity equation

$$\int_{\Omega^e} \left[-w_c \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) - \tau \left(\frac{\partial w_c}{\partial x} \frac{\partial p}{\partial x} + \frac{\partial w_c}{\partial y} \frac{\partial p}{\partial y} \right) \right] d\Omega = \int_{\Omega^e} -\tau w_c \left(\frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} \right) d\Omega \quad (7.20)$$

where GLS contributions are shown in red. These extra terms changes K_{33} and F_3 of Eqn (7.13) as follows (both of which were equal to zero for GFEM)

$$K_{ij}^{33} = \int_{\Omega^e} -\tau \left(\frac{\partial \hat{S}_i}{\partial x} \frac{\partial \hat{S}_j}{\partial x} + \frac{\partial \hat{S}_i}{\partial y} \frac{\partial \hat{S}_j}{\partial y} \right) d\Omega \quad (7.21a)$$

$$F_i^3 = \int_{\Omega^e} -\tau \hat{S}_i \left(\frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} \right) d\Omega \quad (7.21b)$$

As mentioned previously the main contribution of GLS is the one to K_{33} , which is useful in getting rid of the zero diagonal entries of the global system. Although there is no unique way of selecting the stabilization parameter τ , it is usually taken to be

$$\tau = \frac{1}{3} \frac{h_e^2}{4\nu} \quad (7.22)$$

where h_e is a measure of element length, which can be taken as the diameter of the circumcircle for triangular elements or the largest face or diagonal length for quadrilateral elements.

It is very important to note that GLS is a residual based stabilization technique, i.e. if the original GFEM solution is already good enough (free from unphysical oscillations) than the contribution of

GLS will be negligible. It is also important to note that the stabilization parameter τ is a function of element size and as the mesh is refined it tends to zero, i.e. the effect of GLS stabilization disappears as the mesh is refined.

7.5 Sample Stokes Solution – Lid-Driven Cavity Problem

The classical “lid-driven cavity” benchmark problem is sketched in Figure 7.2. The flow is inside a square domain (Cavity) with a side length of $L = 1$. Left, right and bottom walls of the cavity are stationary, whereas the top wall (lid) is moving to the right with a speed of U_o . Motion of the lid puts the fluid inside the cavity into motion and a large clockwise rotating vortex forms, with possible smaller vortices at the bottom and top left corners. Governing similarity parameter is the Reynolds number ($\rho U_o L / \mu$) and by selecting proper values for the lid velocity, density and viscosity, the problem can be studied at any desired Reynolds number. Stokes solutions of this section are performed using $L = 1$, $\rho = 1$ and $\mu = 1$, which corresponds to $Re = 1$ of a Navier-Stokes solution (Actually when solving the Stokes equations it is not appropriate to mention about the Reynolds number, because Stokes equations correspond to the limiting case of zero Reynolds number, which is only an idealization, but a valid one for creeping flows).

This is a popular benchmark problem for flow solvers due its simple geometry and BCs. However it has one inherent difficulty; the singularities at the top corners, where boundaries with different velocities meet. Fine enough elements should be used at the top corners, and actually close to all the walls, to obtain accurate results.

As discussed before, since all BCs are of Dirichlet type, pressure at one point of the flow domain, in this case the lower left corner, is fixed to be zero. Simulations are performed using a mesh of 400 equi-sized, square elements, as shown in Figure 7.2. Four point Gauss Quadrature integration is used for all solutions.

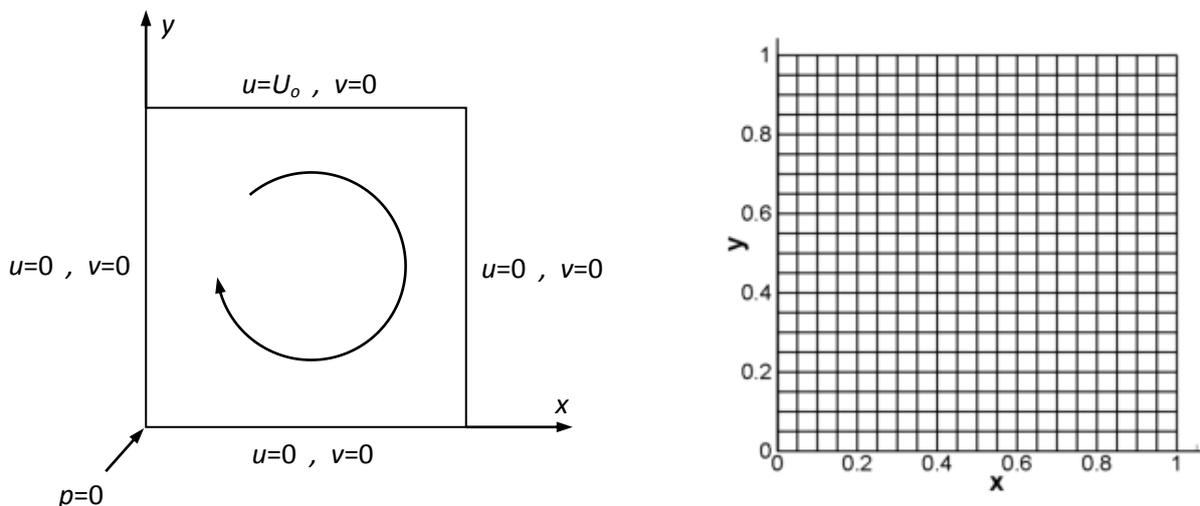


Figure 7.2 Definition of the lid-driven cavity problem and the uniform mesh of 400 quadrilateral elements

The first solution is obtained with LBB-unstable Q_1Q_1 quadratic elements, which use linear approximations for both velocity and pressure, i.e. $NENv = NENp = 4$. Solving the global system of equations using the backslash operator of MATLAB, which uses a direct solver, gives a warning of “Matrix is close to singular or badly scaled. Results may be inaccurate”, but a solution can still be obtained. Streamlines and three-dimensional pressure contours of this first solution are shown in Figure 7.3. Although the velocity distribution and the streamlines are reasonable, small oscillations in the velocity field can be seen, especially close to the bottom corners. But the pressure distribution has large unphysical oscillations, known as checkerboard pressure field in the literature.

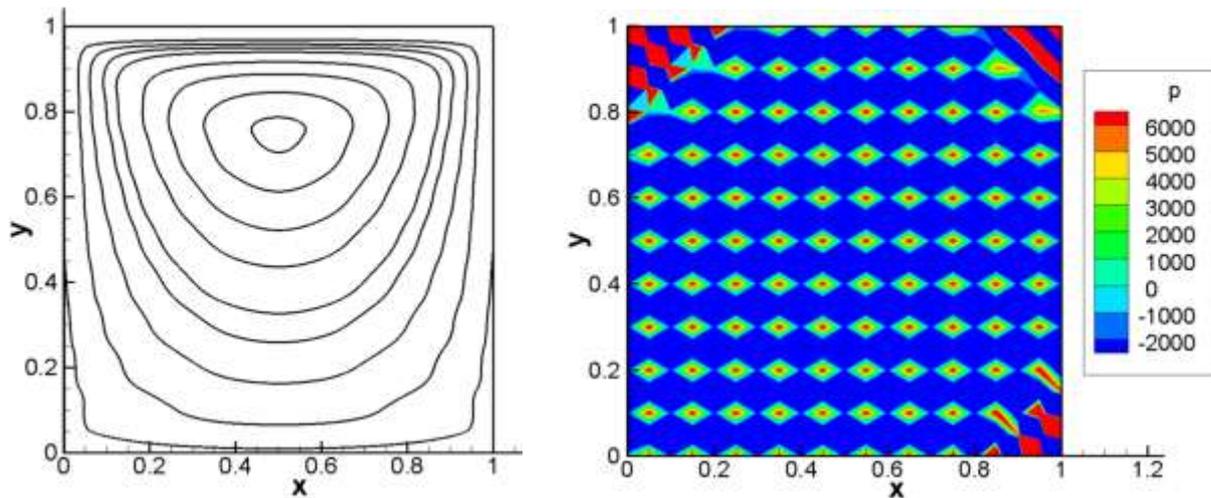


Figure 7.3 Streamlines and pressure contours obtained using GFEM with LBB-unstable Q_1Q_1 (linear) quadrilateral elements

Second solution, shown in Figure 7.4 is obtained with the same 400 linear element mesh. However, this time GLS stabilization is applied. As seen, streamlines are smoother compared to the previous solution and the pressure field is not oscillatory. Maximum and minimum pressure values occur at the right and left top corners.

Third solution, shown in Figure 7.5 is obtained using 400 LBB-stable Q_2Q_1 elements that use 9 velocity nodes and 4 pressure nodes. Although the element number is the same as the previous solutions total number of unknowns is larger. Without GLS stabilization, MATLAB can solve the system of equations using the backslash operator without any singularity warning. Streamlines are almost identical to the ones obtained by the previous solution and the pressure variation is not oscillatory. Note that during the post-processing step, only the velocity values corresponding to element corners are used for simplicity, which introduces a certain amount of visualization error.

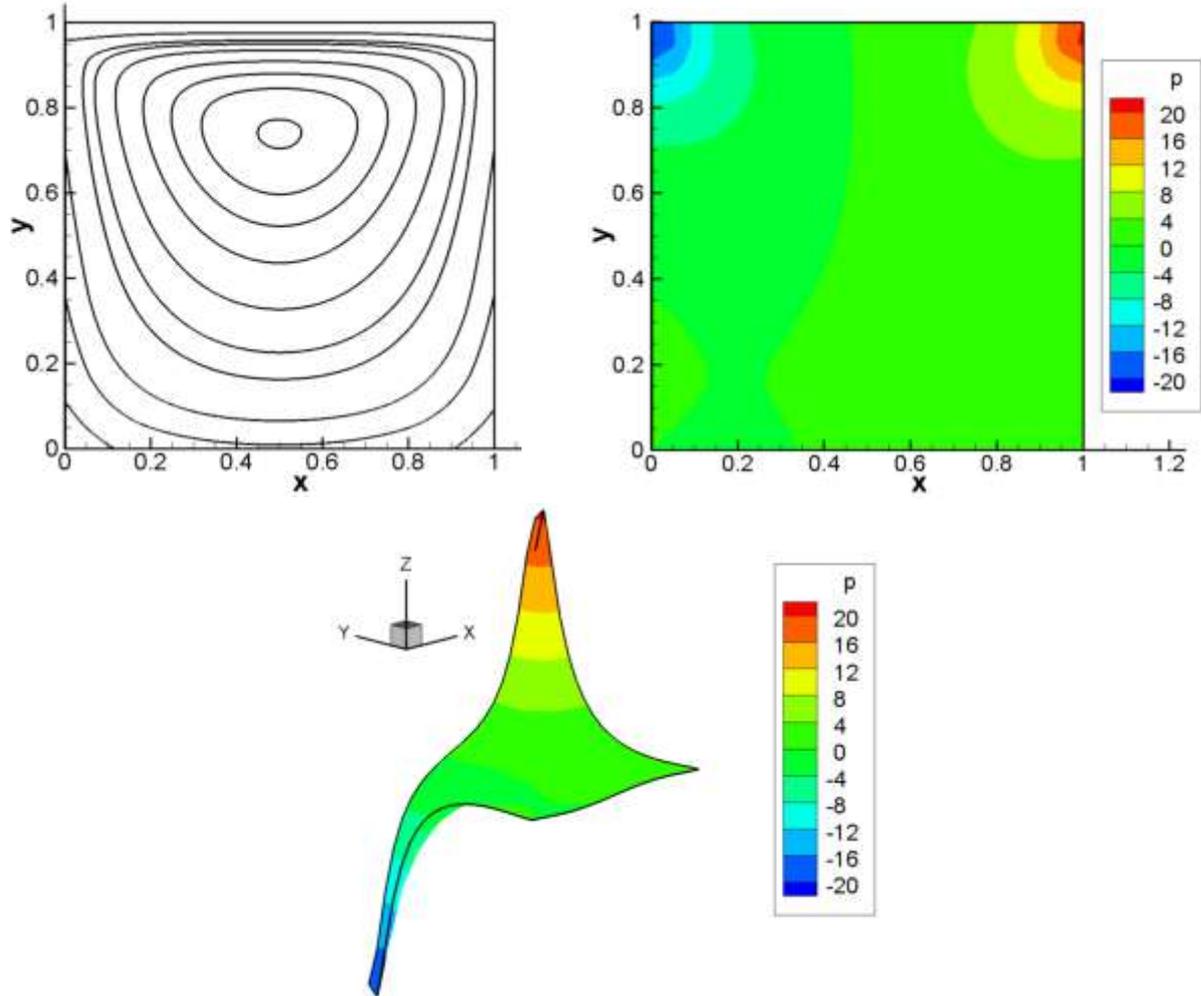


Figure 7.4 Streamlines, pressure contours and 3D plot of pressure field obtained using LBB-unstable Q_1Q_1 quadrilateral elements with GLS stabilization

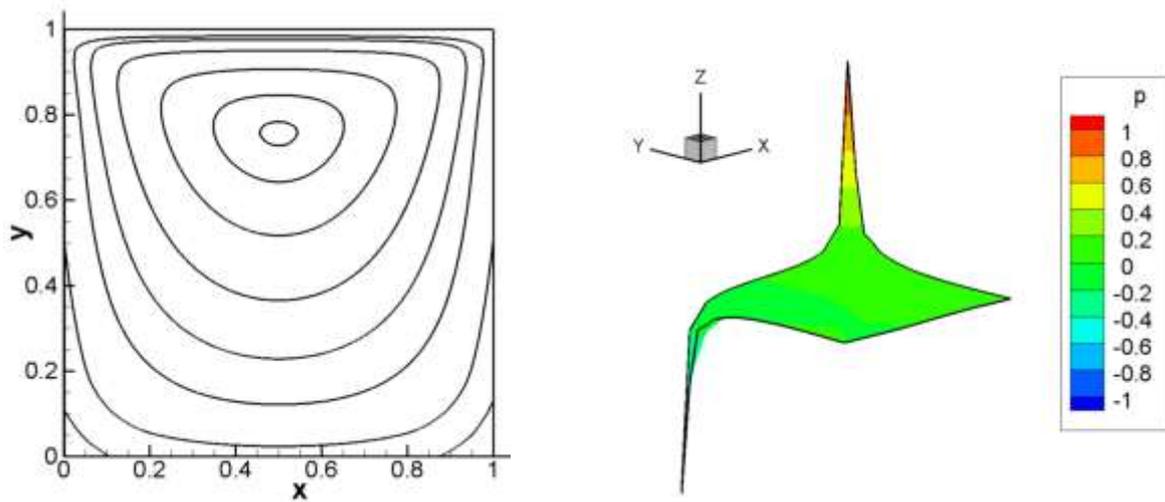


Figure 7.5 Streamlines and 3D pressure field obtained for using GFEM with LBB-stable Q_2Q_1 quadrilateral elements

A final solution for the lid-driven cavity problem is obtained using triangular elements. A non-uniform mesh of 864 triangles, generated by mesh2d code, is shown in Figure 7.6. This solution uses linear approximation for both pressure and velocity, i.e., $NENv = NENp = 3$ together with GLS stabilization. Without GLS, this LBB-unstable element gives an oscillatory pressure field. For the same mesh of 864 triangles, but with quadratic velocity approximation ($NENv = 6$), pressure oscillations can be avoided except for the regions very close to the top corners.

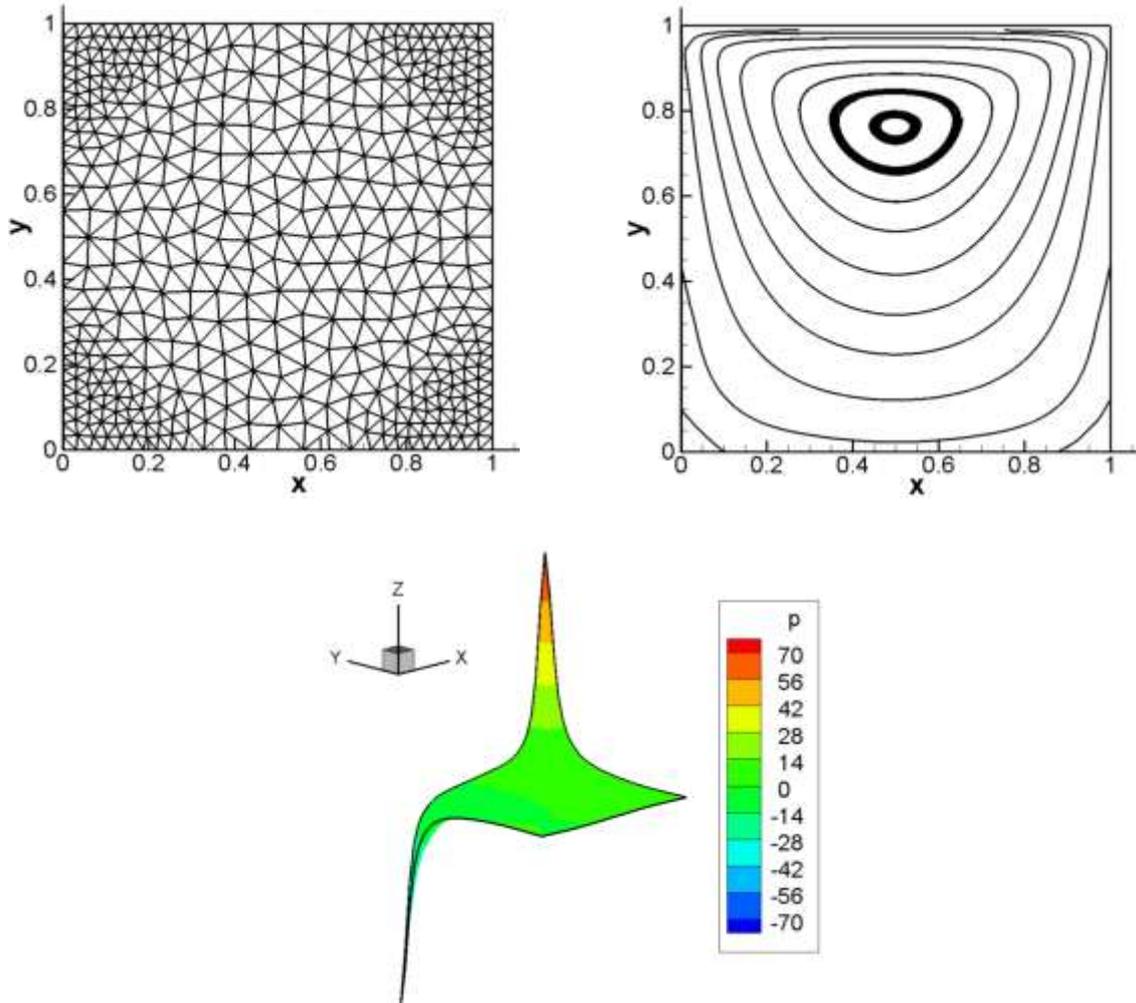


Figure 7.6 Streamlines and pressure contours obtained for Stokes solution using LBB-unstable linear triangular elements ($NE = 864$) with GLS stabilization

Following comments can be made about the Stokes solutions of the Cavity problem:

- LBB-unstable elements (linear triangles and quadrilaterals) provide oscillatory results with GFEM. It is important to test GFEM with LBB-unstable elements of finer grids to make a better conclusion.
- Acceptable solutions can be obtained with LBB-unstable elements if GLS stabilization is used.

- LBB-stable elements with quadratic velocity and linear pressure approximation provide non-oscillatory solutions without stabilization.
- Cavity problem is a challenging benchmark problem due to the singularities at the top corners where fixed sides walls meet with the moving top wall. At these corners there are sharp velocity and pressure changes. This difficulty can be seen clearly by the different pressure scales obtained in different solutions. Each solution provides quite different minimum and maximum pressure values, but these extreme values occur only very close to the top corners of the cavity. Away from these corners pressure is almost constant and this behavior can be captured correctly by all acceptable solutions. However, extreme pressure values at the top corners can only be captured at various orders of precision with different solution techniques.
- 2D, steady Stokes solutions are not very demanding computationally. For meshes of less than 1000 elements that are tried, solution can be obtained in less than 10 seconds on a standard PC.
- Comparing the results of different solutions just by looking at the contour plots or streamlines may be misleading. It is better to make more quantitative comparisons, like velocity and/or pressure distributions across different cross sections of the flow field. Comparisons should also be made against credible solutions of the literature. These kind of detailed comparions will be done in the following sections for N-S solutions.

7.6 Sample Stokes Solution – Flow Over a Cylinder

Consider the flow over a 2D cylinder with the problem geometry and BCs shown in Figure 7.7. The outside box has a size of 20x10 units. Cylinder has a diameter of 1 and its origin is at (5,5).

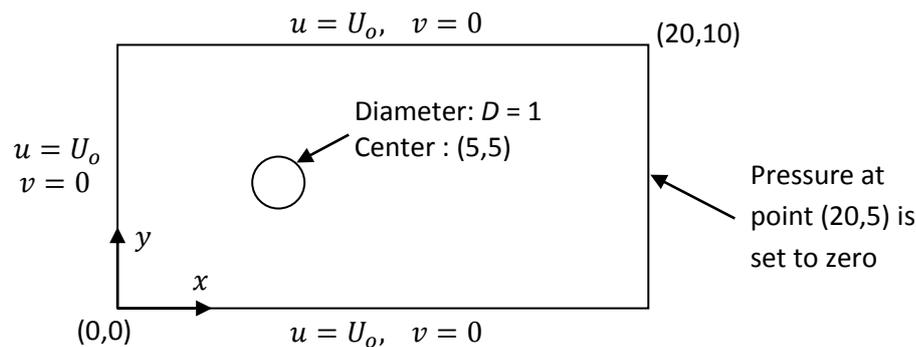


Figure 7.7 Geometry and BCs for the flow over a cylinder problem

Uniform flow of magnitude U_o is specified at the left inlet. Top and bottom walls are specified to be sliding with the inlet velocity to minimize the effect of walls on the flow field. No slip BC is provided on the cylinder surface by fixing both velocity components to zero. Finally the pressure is fixed to be zero at the center point of the outlet.

Size of the box and the placement of the cylinder within the box are important. The box should be wide enough to simulate external uniform flow passing over the cylinder. Also the distance between the cylinder and the inlet and outlet boundaries should be large enough so that uniform inlet and fully developed outlet BCs can be applied correctly.

The task is to obtain the velocity and pressure field around the cylinder. Solutions will be done for three different U_o values of 0.1, 1 and 10. The mesh shown in Figure 7.8 is used for all simulations. It is created using mesh2d software and has 2090 elements. Elements are clustered around the cylinder, with smaller elements used in the wake region.

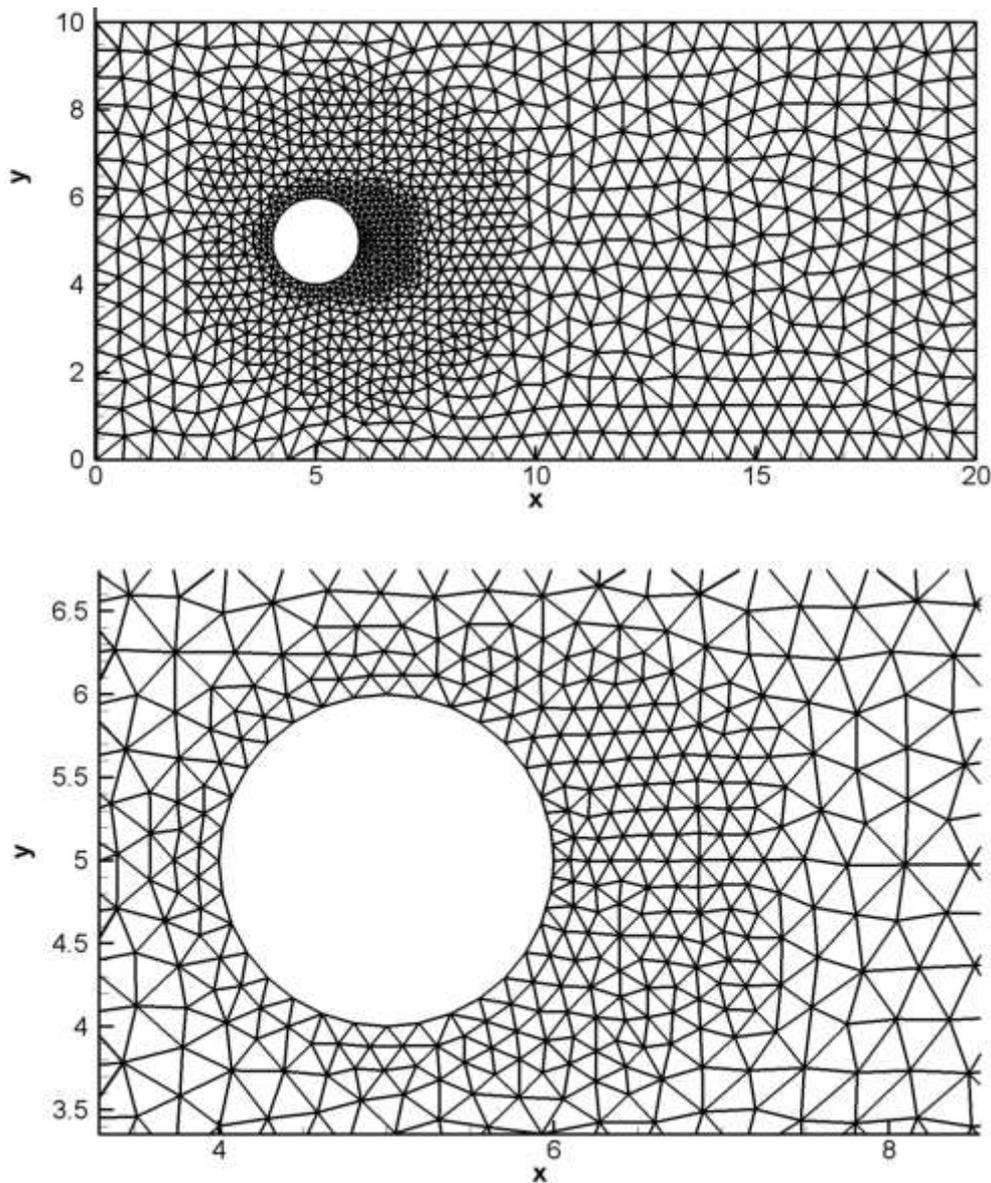


Figure 7.8 Triangular mesh of 2090 elements used for the flow around a 2D cylinder. Bottom plot shows the close up view around the cylinder.

Solutions obtained for three different free stream speeds are given in Figure 7.9. They are obtained using LBB-stable elements with 6 velocity and 3 pressure nodes without any stabilization. As seen all three solutions have the same streamline pattern. Pressure contours are also identical, but the pressure scales are different. As the free stream velocity is increased by a factor of 10, scale of the viscous terms increase by a factor of 10, and therefore pressure values that should balance viscous forces in the momentum equations also increase by a factor of 10.

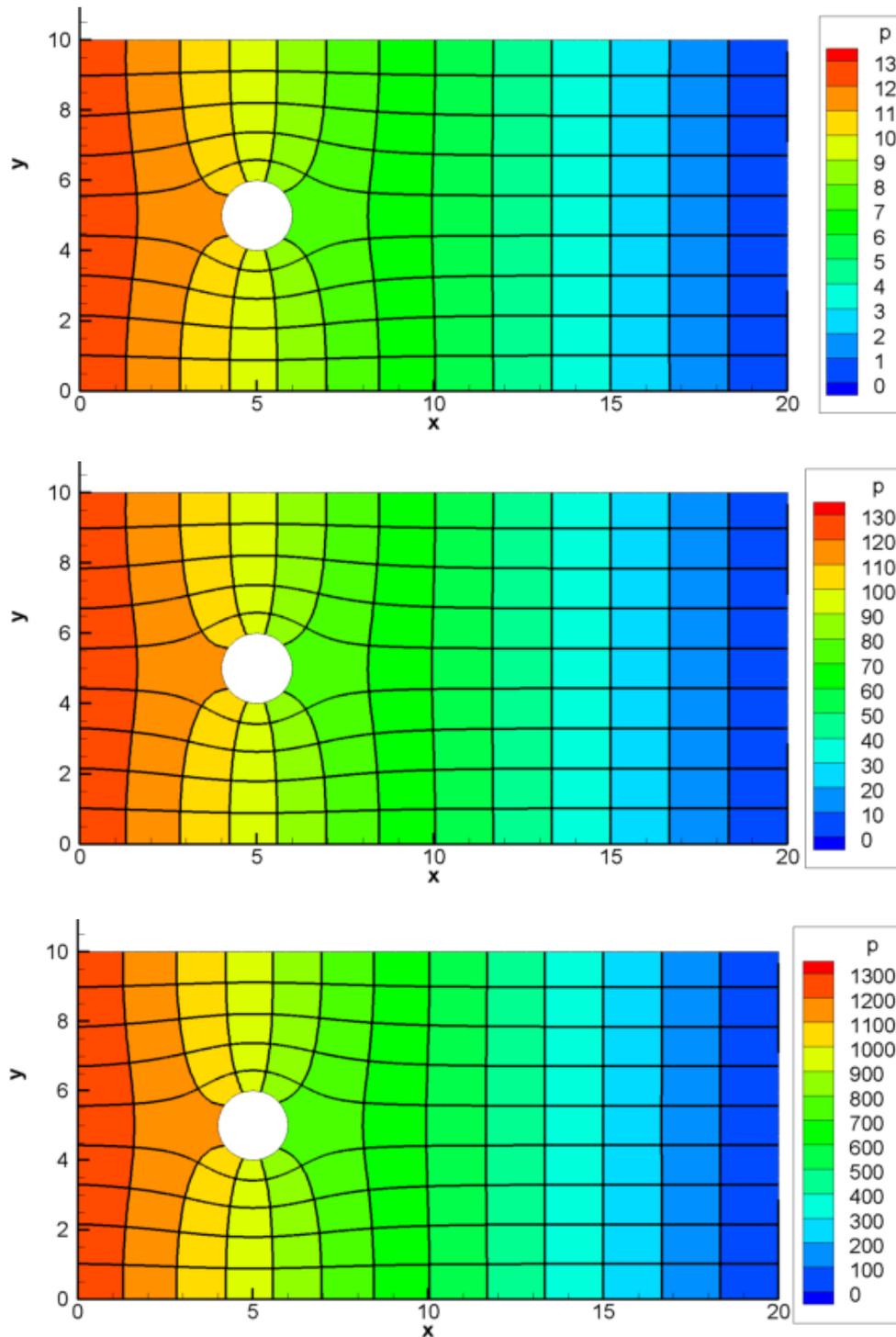


Figure 7.9 Streamlines and constant pressure lines for Stokes solution of flow over a cylinder. Freestream velocities are 0.1 (top), 1 (middle) and 10 (bottom).

It is of course not physical to get the same pressure and velocity field patterns no matter what we use for the free stream velocity, but this is a drawback of the Stokes formulation. Flow fields look similar to those of potential flow theory and no matter what we use for the free stream velocity it is not possible to see flow separation. Later we'll perform Navier-Stokes simulations with the parameters that we used here and will obtain different and more physical results.

7.7 GFEM of 2D N-S Equations in Primitive Variables

Two-dimensional, steady, incompressible, viscous flows are governed by the following Navier-Stokes equations written in the Cartesian coordinate system

$$\text{x-Momentum: } \rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) - \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial p}{\partial x} - \rho f_x = 0 \quad (7.23a)$$

$$\text{y-Momentum: } \rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) - \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{\partial p}{\partial y} - \rho f_y = 0 \quad (7.23b)$$

$$\text{Continuity: } \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (7.23c)$$

Similar to what we did for Stokes equations, momentum equations can be put into the following form by the help of the continuity equation.

$$\underbrace{\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right)}_{\text{Nonlinear}} - \frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \frac{\partial p}{\partial x} - \rho f_x = 0 \quad (7.24a)$$

$$\underbrace{\rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right)}_{\text{Nonlinear}} - \frac{\partial}{\partial y} \left(2\mu \frac{\partial v}{\partial y} \right) - \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \frac{\partial p}{\partial y} - \rho f_y = 0 \quad (7.24b)$$

The difference between these equations and the Stokes equations, studied previously in Section 7.2, are the nonlinear terms. Due to the nonlinearity, the set of algebraic equations that will be obtained at the end of this section can not be solved in a single shot, but an iterative solution is necessary (similar to finding roots of a nonlinear function using an iterative method such as Newton-Raphson). In such an iterative solution nonlinear terms can be linearized in a number of different ways. The simplest possibility, which will be used in this section, is known as Picard linearization, in which the nonlinear terms are replaced by

$$\rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) \rightarrow \rho \left(u_0 \frac{\partial u}{\partial x} + v_0 \frac{\partial u}{\partial y} \right) \quad (7.25a)$$

$$\rho \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) \rightarrow \rho \left(u_0 \frac{\partial v}{\partial x} + v_0 \frac{\partial v}{\partial y} \right) \quad (7.25b)$$

New variables u_0 and v_0 are known velocity component values from the previous iteration. Although this linearization is simple to implement, its convergence is slower than the more commonly used Newton linearization, which will be discussed later.

Obtaining weak form of the N-S equations is very similar to the procedure followed for the Stokes equations. We first form the weighted integral forms of the equations and apply integration by parts to the terms of the momentum equation with second derivatives and the pressure terms. The resulting elemental weak form is

$$\int_{\Omega^e} \left[\rho w_x \left(u_0 \frac{\partial u}{\partial x} + v_0 \frac{\partial u}{\partial y} \right) + 2\mu \frac{\partial w_x}{\partial x} \frac{\partial u}{\partial x} + \mu \frac{\partial w_x}{\partial y} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) - \frac{\partial w_x}{\partial x} p \right] d\Omega = \int_{\Omega^e} \rho w_x f_x d\Omega + \int_{\Gamma^e} w_x t_x d\Gamma \quad (7.26a)$$

$$\int_{\Omega^e} \left[\rho w_y \left(u_0 \frac{\partial v}{\partial x} + v_0 \frac{\partial v}{\partial y} \right) + 2\mu \frac{\partial w_y}{\partial y} \frac{\partial v}{\partial y} + \mu \frac{\partial w_y}{\partial x} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) - \frac{\partial w_y}{\partial y} p \right] d\Omega = \int_{\Omega^e} \rho w_y f_y d\Omega + \int_{\Gamma^e} w_y t_y d\Gamma \quad (7.26b)$$

$$\int_{\Omega^e} -w_c \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) d\Omega = 0 \quad (7.26c)$$

To be consistent with the formulation used for the Stokes equations, continuity equation is multiplied with minus one. Substituting the approximate forms of the unknowns as summations of unknown nodal values multiplied with shape functions and replacing the weight functions by proper shape functions we get the following GFEM formulation

$$\sum_{j=1}^{NENv} \left\{ \int_{\Omega^e} \left[\rho S_i \left(u_0 \frac{\partial S_j}{\partial x} + v_0 \frac{\partial S_j}{\partial y} \right) + 2\mu \frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial x} + \mu \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial y} \right] d\Omega \right\} u_j + \sum_{j=1}^{NENv} \left(\int_{\Omega^e} \mu \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial x} d\Omega \right) v_j + \sum_{j=1}^{NENp} \left(\int_{\Omega^e} -\frac{\partial S_i}{\partial x} \hat{S}_j d\Omega \right) p_j = \int_{\Omega^e} \rho S_i f_x d\Omega + \int_{\Gamma^e} S_i t_x d\Gamma \quad (7.27a)$$

$$\sum_{j=1}^{NENv} \left(\int_{\Omega^e} \mu \frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial y} d\Omega \right) u_j + \sum_{j=1}^{NENv} \left\{ \int_{\Omega^e} \left[\rho S_i \left(u_0 \frac{\partial S_j}{\partial x} + v_0 \frac{\partial S_j}{\partial y} \right) + 2\mu \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial y} + \mu \frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial x} \right] d\Omega \right\} v_j + \sum_{j=1}^{NENp} \left(\int_{\Omega^e} -\frac{\partial S_i}{\partial y} \hat{S}_j d\Omega \right) p_j = \int_{\Omega^e} \rho S_i f_y d\Omega + \int_{\Gamma^e} S_i t_y d\Gamma \quad (7.27b)$$

$$\sum_{j=1}^{NENv} \left(\int_{\Omega^e} -\hat{S}_i \frac{\partial S_j}{\partial x} d\Omega \right) u_j + \sum_{j=1}^{NENv} \left(\int_{\Omega^e} -\hat{S}_i \frac{\partial S_j}{\partial x} d\Omega \right) v_j = 0 \quad (7.27c)$$

which can be written in the following form

$$\begin{bmatrix} [K^{11}] & [K^{12}] & [K^{13}] \\ [K^{21}] & [K^{22}] & [K^{23}] \\ [K^{31}] & [K^{32}] & [K^{33}] \end{bmatrix}^e \begin{Bmatrix} \{u\} \\ \{v\} \\ \{p\} \end{Bmatrix}^e = \begin{Bmatrix} \{F^1\} \\ \{F^2\} \\ \{F^3\} \end{Bmatrix}^e \quad (7.28)$$

where the submatrices except K^{11} and K^{22} are the same as the ones derived previously for the Stokes equations (See Eqn (7.13)). K^{11} and K^{22} now include the nonlinear terms as shown below

$$K_{ij}^{11} = \int_{\Omega^e} \left[2\mu \frac{\partial S_i^e}{\partial x} \frac{\partial S_j^e}{\partial x} + \mu \frac{\partial S_i^e}{\partial y} \frac{\partial S_j^e}{\partial y} + \rho u_0 S_i^e \frac{\partial S_j^e}{\partial x} + \rho v_0 S_i^e \frac{\partial S_j^e}{\partial y} \right] d\Omega \quad (7.29a)$$

$$K_{ij}^{22} = \int_{\Omega^e} \left[\mu \frac{\partial S_i^e}{\partial x} \frac{\partial S_j^e}{\partial x} + 2\mu \frac{\partial S_i^e}{\partial y} \frac{\partial S_j^e}{\partial y} + \rho u_0 S_i^e \frac{\partial S_j^e}{\partial x} + \rho v_0 S_i^e \frac{\partial S_j^e}{\partial y} \right] d\Omega \quad (7.29b)$$

Due to the existence of the nonlinear terms, elemental system and therefore assembled global system becomes unsymmetric.

7.8 Newton Linearization

GFEM formulation provided in the previous section made use of the Picard linearization for the nonlinear terms. Picard linearization is simple but it has a slower convergence rate, compared to the Newton linearization that will be discussed in this section.

In Newton linearization nonlinear terms of the momentum equation are linearized in the following way

$$\frac{\partial \vec{V}}{\partial t} + \vec{V}_0 \cdot \nabla \vec{V} + \vec{V} \cdot \nabla \vec{V}_0 = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \vec{V} + \vec{f} + \vec{V}_0 \cdot \nabla \vec{V}_0 \quad (7.30)$$

Upon convergence, i.e. as \vec{V} approaches to \vec{V}_0 , extra terms on the left and right hand sides will cancel out and the original N-S equations will be obtained. Obtaining the elemental system of this equation coupled with the continuity equation is left as an exercise for you.

7.9 GLS Stabilization of N-S Equations for Linear Triangles and Quadrilaterals

This is similar to the one discussed in Section 7.4 for Stokes equations. GLS stabilization will bring the following extra integrals

$$\text{GLS terms of x-mom : } \int_{\Omega^e} [\rho(\vec{V}_0 \cdot \nabla) w_x - \mu \nabla^2 w_x] \tau \left[\rho(\vec{V}_0 \cdot \nabla) u - \mu \nabla^2 u + \frac{\partial p}{\partial x} - \rho f_x \right] d\Omega \quad (7.31a)$$

$$\text{GLS terms of y-mom : } \int_{\Omega^e} [\rho(\vec{V}_0 \cdot \nabla) w_y - \mu \nabla^2 w_y] \tau \left[\rho(\vec{V}_0 \cdot \nabla) v - \mu \nabla^2 v + \frac{\partial p}{\partial y} - \rho f_y \right] d\Omega \quad (7.31b)$$

$$\text{GLS terms of continuity : } \int_{\Omega^e} \tau (\nabla w_c) \cdot [\rho(\vec{V}_0 \cdot \nabla) \vec{V} - \mu \nabla^2 \vec{V} + \nabla p - \rho \vec{f}] d\Omega \quad (7.31c)$$

Here we made use of \vec{V}_0 as the velocity vector of the previous iteration in the context of Picard linearization. Similar to the Stokes equations case, for the use of 3-node triangular or 4-node quadrilateral approximation of velocity second order velocity derivatives of viscous terms vanish and GLS contributions simplify as follows

$$\text{GLS terms of x-momentum : } \int_{\Omega^e} [\rho(\vec{V}_0 \cdot \nabla)w_x] \tau \left[\rho(\vec{V}_0 \cdot \nabla)u + \frac{\partial p}{\partial x} - \rho f_x \right] d\Omega \quad (7.32a)$$

$$\text{GLS terms of y-momentum : } \int_{\Omega^e} [\rho(\vec{V}_0 \cdot \nabla)w_y] \tau \left[\rho(\vec{V}_0 \cdot \nabla)v + \frac{\partial p}{\partial y} - \rho f_y \right] d\Omega \quad (7.32b)$$

$$\text{GLS terms of continuity : } \int_{\Omega^e} \tau (\nabla w_c) \cdot [\rho(\vec{V}_0 \cdot \nabla)\vec{V} + \nabla p - \rho \vec{f}] d\Omega \quad (7.32c)$$

These GLS integrals will bring modifications to the following elemental sub-matrices and sub-vectors, where the matrices and vectors with subscript GFEM are the ones given in Eqns (7.13) and (7.29)

$$K_{ij}^{11} = K_{ij}^{11}{}_{GFEM} + \int_{\Omega^e} \tau \rho^2 \left(u_0 \frac{\partial S_i^e}{\partial x} + v_0 \frac{\partial S_i^e}{\partial y} \right) \left(u_0 \frac{\partial S_j^e}{\partial x} + v_0 \frac{\partial S_j^e}{\partial y} \right) d\Omega$$

$$K_{ij}^{13} = K_{ij}^{13}{}_{GFEM} + \int_{\Omega^e} \tau \rho \left(u_0 \frac{\partial S_i^e}{\partial x} + v_0 \frac{\partial S_i^e}{\partial y} \right) \frac{\partial \hat{S}_j^e}{\partial x} d\Omega$$

$$K_{ij}^{22} = K_{ij}^{22}{}_{GFEM} + \int_{\Omega^e} \tau \rho^2 \left(u_0 \frac{\partial S_i^e}{\partial x} + v_0 \frac{\partial S_i^e}{\partial y} \right) \left(u_0 \frac{\partial S_j^e}{\partial x} + v_0 \frac{\partial S_j^e}{\partial y} \right) d\Omega$$

$$K_{ij}^{23} = K_{ij}^{23}{}_{GFEM} + \int_{\Omega^e} \tau \rho \left(u_0 \frac{\partial S_i^e}{\partial x} + v_0 \frac{\partial S_i^e}{\partial y} \right) \frac{\partial \hat{S}_j^e}{\partial y} d\Omega \quad (7.30)$$

$$K_{ij}^{31} = K_{ij}^{31}{}_{GFEM} - \int_{\Omega^e} \tau \rho \frac{\partial \hat{S}_i^e}{\partial x} \left(u_0 \frac{\partial S_j^e}{\partial x} + v_0 \frac{\partial S_j^e}{\partial y} \right) d\Omega$$

$$K_{ij}^{32} = K_{ij}^{32}{}_{GFEM} - \int_{\Omega^e} \tau \rho \frac{\partial \hat{S}_i^e}{\partial y} \left(u_0 \frac{\partial S_j^e}{\partial x} + v_0 \frac{\partial S_j^e}{\partial y} \right) d\Omega$$

$$K_{ij}^{33} = K_{ij}^{33}{}_{GFEM} - \int_{\Omega^e} \tau \left(\frac{\partial \hat{S}_i^e}{\partial x} \frac{\partial \hat{S}_j^e}{\partial x} + \frac{\partial \hat{S}_i^e}{\partial y} \frac{\partial \hat{S}_j^e}{\partial y} \right) d\Omega$$

$$F_i^1 = F_i^1{}_{GFEM} + \int_{\Omega^e} \tau \rho^2 \left(u_0 \frac{\partial S_i^e}{\partial x} + v_0 \frac{\partial S_i^e}{\partial y} \right) f_x d\Omega$$

$$F_i^2 = F_i^2{}_{GFEM} + \int_{\Omega^e} \tau \rho^2 \left(u_0 \frac{\partial S_i^e}{\partial x} + v_0 \frac{\partial S_i^e}{\partial y} \right) f_y d\Omega$$

$$F_i^3 = F_i^3{}_{GFEM} - \int_{\Omega^e} \tau \rho \left(\frac{\partial \hat{S}_i^e}{\partial x} f_x + \frac{\partial \hat{S}_i^e}{\partial y} f_y \right) d\Omega$$

Stabilization parameter τ is the same as the one defined previously in Eqn (7.22)

7.10 Sample N-S Solution – Lid-Driven Cavity Problem

Again we'll start with the lid-driven cavity problem solved previously using the Stokes equations. This problem will be solved for various different Reynolds numbers by changing the viscosity of the fluid. Size of the cavity, lid velocity and density of the fluid will always be kept as zero.

First, $Re = 100$ solution shown in Figure 7.10 is obtained using the mesh shown in Figure 7.6. LBB-stable P_2P_1 element is used without any stabilization. Solution matches perfectly with Ghia's benchmark results [3]. Two small vortices are observed at the bottom corners of the cavity.

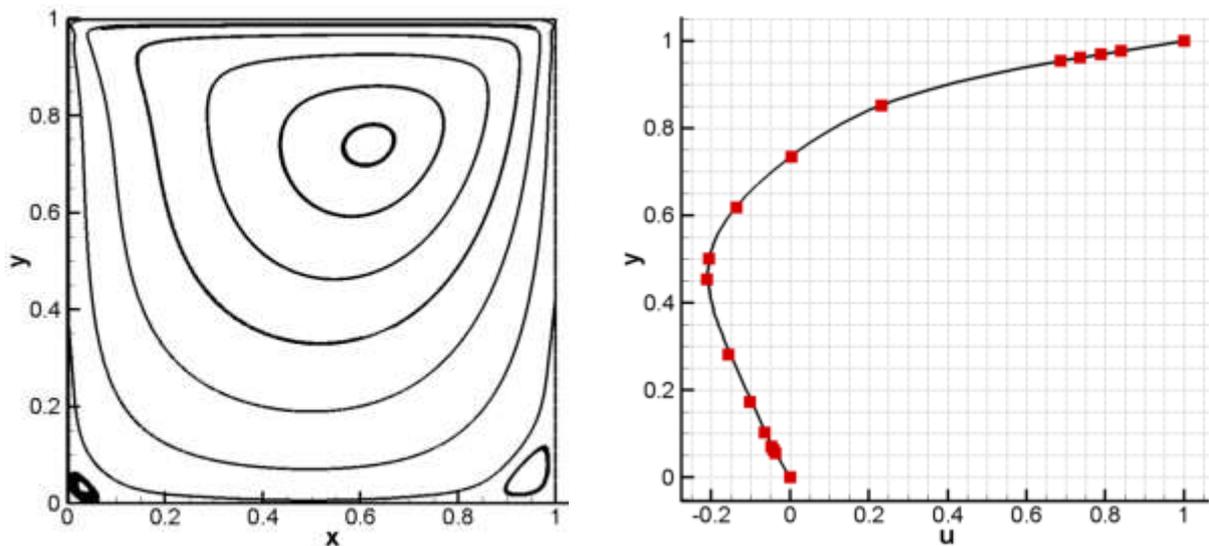


Figure 7.10 Streamlines and u velocity component profile along $x = 0.5$ for $Re = 100$. Mesh has 864 P_2P_1 triangles. Red squares are reference results of Ghia [3].

When the same mesh is used to obtain a solution for $Re = 1000$, the comparison with the reference solution was not perfect as seen in Figure 7.11. A finer mesh of 1320 triangular elements is used to get an improved solution. Compared to the previous lower Re solution, it is seen that as the Reynolds number increases vortices at the bottom corners get larger and the core of the centerline vortex moves towards the center of the cavity.

Finally cavity solution for $Re = 3200$, shown in Figure 7.12 is obtained. Similar to the previous $Re = 1000$ case, 864 element mesh provided unacceptable results. A new mesh of 4840 elements is created and it provided results with acceptable accuracy. As seen from Figure 7.12, at $Re = 3200$ a new vortex is formed close to the upper left corner of the domain.

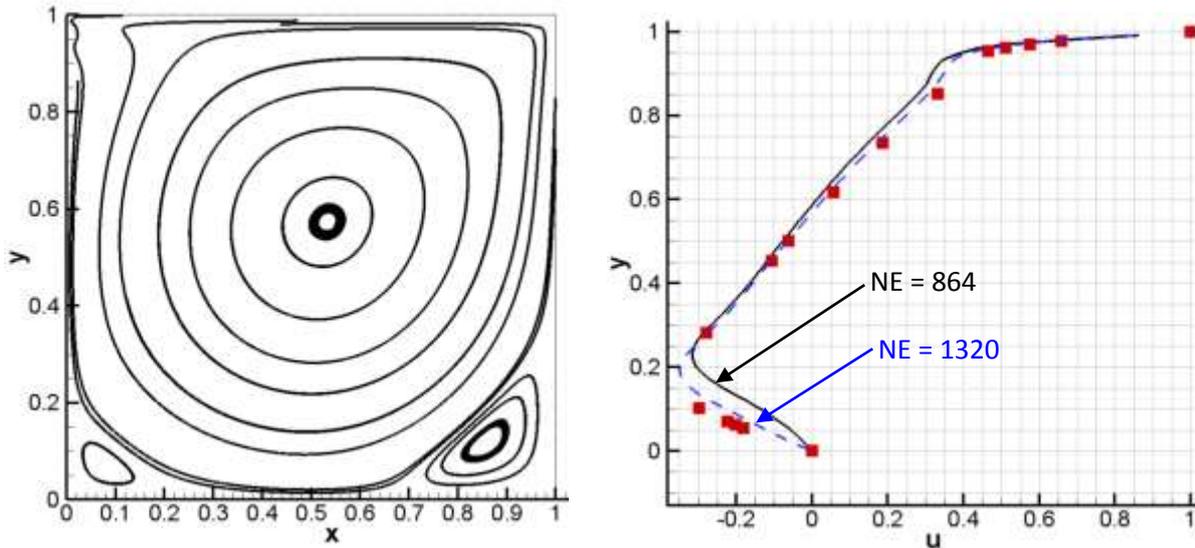


Figure 7.11 Streamlines and u velocity component profile along $x = 0.5$ for $Re = 1000$. Streamline plot is obtained with a mesh of 1320 P_2P_1 triangles. Red squares are reference results of Ghia [3].

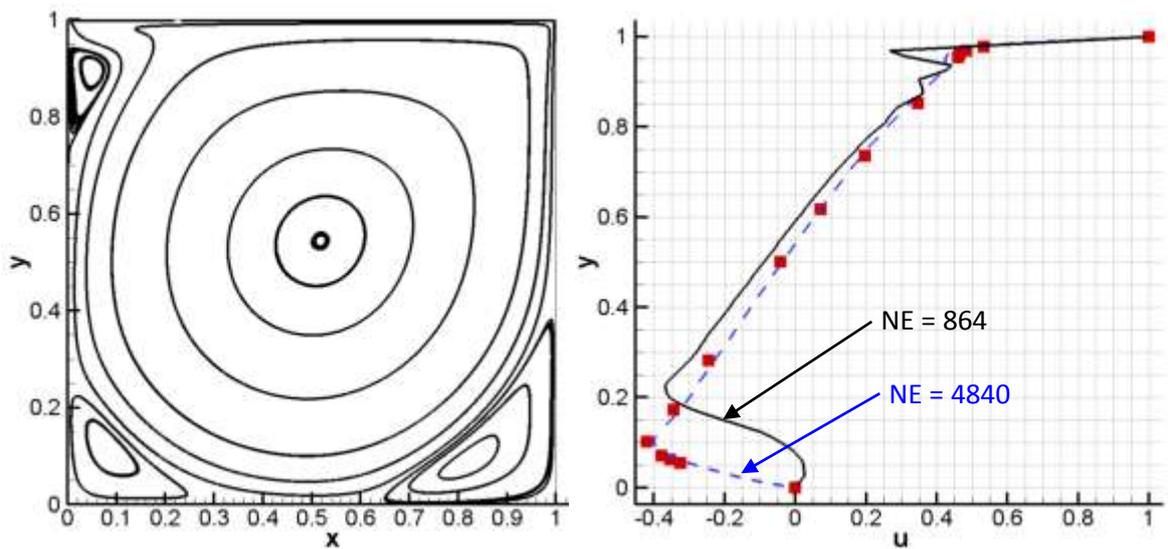


Figure 7.12 Streamlines and u velocity component profile along $x = 0.5$ for $Re = 3200$. Streamline plot is obtained with a mesh of 4840 P_2P_1 triangles. Red squares are reference results of Ghia [3].

In order to get a better feeling of the flow fields, Figure 7.13 show 3D plots of u and v velocity components for all three Reynolds numbers simulated in this part. Oscillations are seen close to the top corners of the cavity due to sudden velocity and pressure changes. Note that no stabilization is applied to any of these solutions.

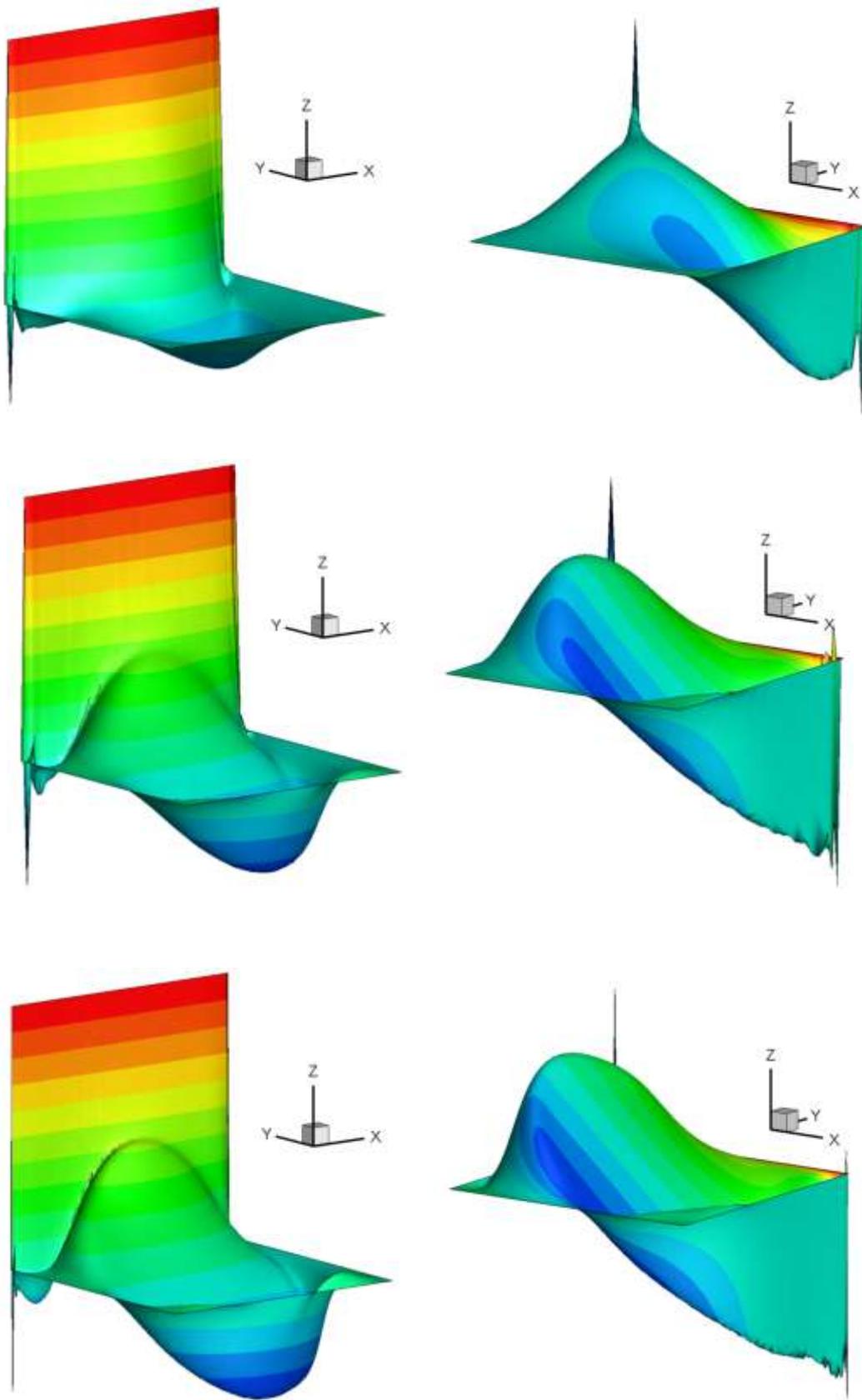


Figure 7.13 3D views of u (left) and v velocity fields (right) for $Re = 100$ (top), $Re = 1000$ (middle) and $Re = 3200$ (bottom)

N-S solutions of the lid-driven cavity problem can be summarized as follows

- With fine enough meshes LBB-stable elements without any stabilization provide acceptable solutions.
- Accurate solution of higher Reynolds number flows require a finer mesh compared to lower Re flows.
- The finest mesh used in these solutions has 4840 elements with 6 velocity and 3 pressure nodes. This mesh has a total of 2573 pressure and 9985 velocity nodes, which makes a total of 22,543 unknowns. Global stiffness matrix is a square matrix of size $22,543 \times 22,543$. If stored in full form this matrix requires close to 4 GB of memory. With a standard PC it is difficult to make such a memory allocation. Instead of storing this matrix in full form a sparse matrix storage technique is utilized. Global stiffness matrix mentioned above has 603,762 nonzero entries, i.e. only 0.12 % of all its entries are nonzero as shown in Figure 7.14 obtained with MATLAB's `spy` command. When stored in sparse form global stiffness matrix requires less than 10 MB of memory. For sparse storage MATLAB's built-in coordinate storage format is used.
- Although provides extensive memory savings, the use of sparse storage makes coding more complicated. An simpler alternative is to change the global node numbering such that the bandwidth of the stiffness matrix reduces. This reordering can automatically be done using MATLAB's `symrcm` command. The resulting banded matrix shown in Figure 7.14 has a bandwidth of 1078, and its storage requires 185 MB of memory, which is manageable for this 2D problem, but this number can be very high for large 3D problems.

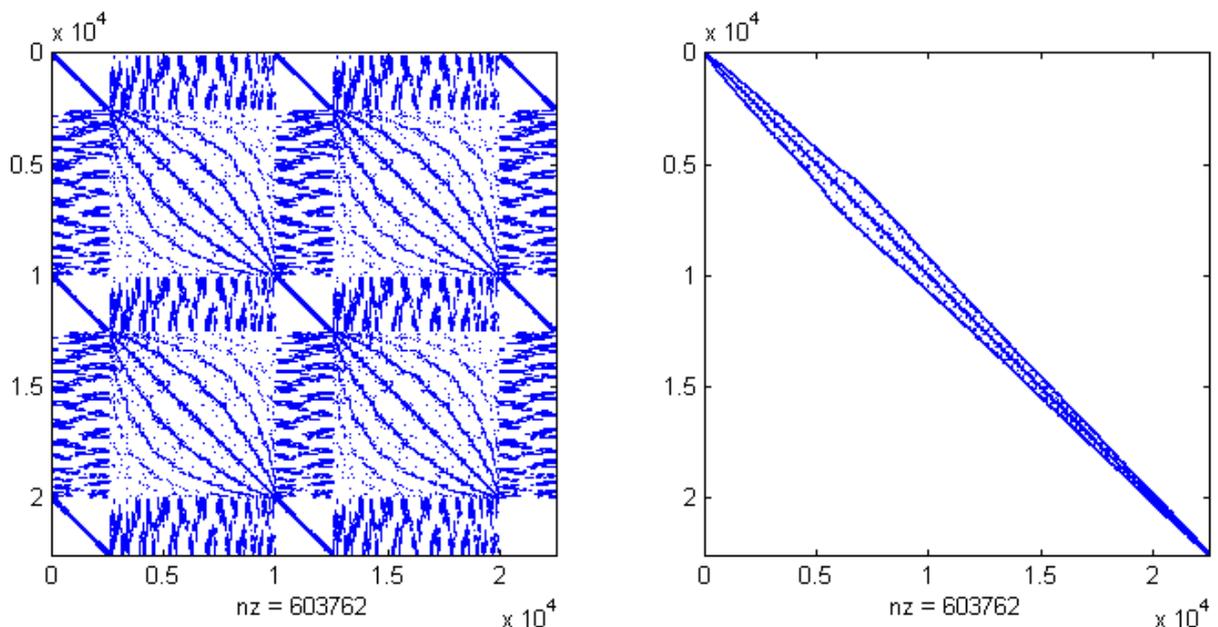


Figure 7.14 Sparsity pattern of the original (left) stiffness matrix for a mesh with $NE=4840$, $NN=9985$. Sparsity pattern of the banded (right) stiffness matrix obtained by renumbering the global nodes using MATLAB's `symrcm` command.

- Iteration numbers and solution times spent for Picard iterations of all the simulations performed for the lid-driven cavity problem are given in Table 7.2. Convergence check is done by comparing velocity components of the current iteration with the values of the previous iteration and the simulation is stopped if the absolute difference for both velocity components are less than 10^{-6} .

Table 7.2 Computational performance of five simulations performed for the cavity problem

Re	NE (P_2P_1)	Ndof	Time Spent for Picard Iterations (s)	Number of Iterations	Time Per Iteration (s)
100	864	4151	7.6	11	0.7
1000	864	4152	19.8	29	0.7
1000	1320	6263	27.3	26	1.1
3200	864	4151	95	134	0.7
3200	4840	22543	277	68	4.1

As seen from this table, time required per iteration increases as the mesh gets finer, as expected. An interesting observation is that for high Reynolds numbers coarse mesh solutions require more iterations to converge. Even the finest mesh of this 2D problem took less than 5 minutes to converge on a standart PC.

7.11 Sample N-S Solution – Flow Over a Cylinder

Let's now revisit the flow over a cylinder problem that we previously solved using Stokes equations. Using the same parameters as we used previously, i.e. keeping cylinder diameter, density and viscosity constant at 1 and changing freestream speed U_o as 0.1, 1, 10, we'll simulate flows with Reynolds numbers of 0.1, 1 and 10. Same mesh of 2090 triangular elements, shown in Figure 7.8, is used for all the runs. Pressure at the center of the exit boundary is equated to zero.

Results are shown in Figure 7.15 as pressure contours and streamlines. Comparing these results with the ones obtained previously by solving Stokes equations (Figure 7.9) reveal important differences. Lowest Reynolds number case provides a pressure contour pattern that is the most similar to those obtained by Stokes equations. But as the Reynolds number gets larger pressure contours become totally different than the ones obtained for Stokes equations. For $Re = 10$ case flow separates and circulation bubbles form in the wake of the cylinder, which were not observed for the Stokes solution.

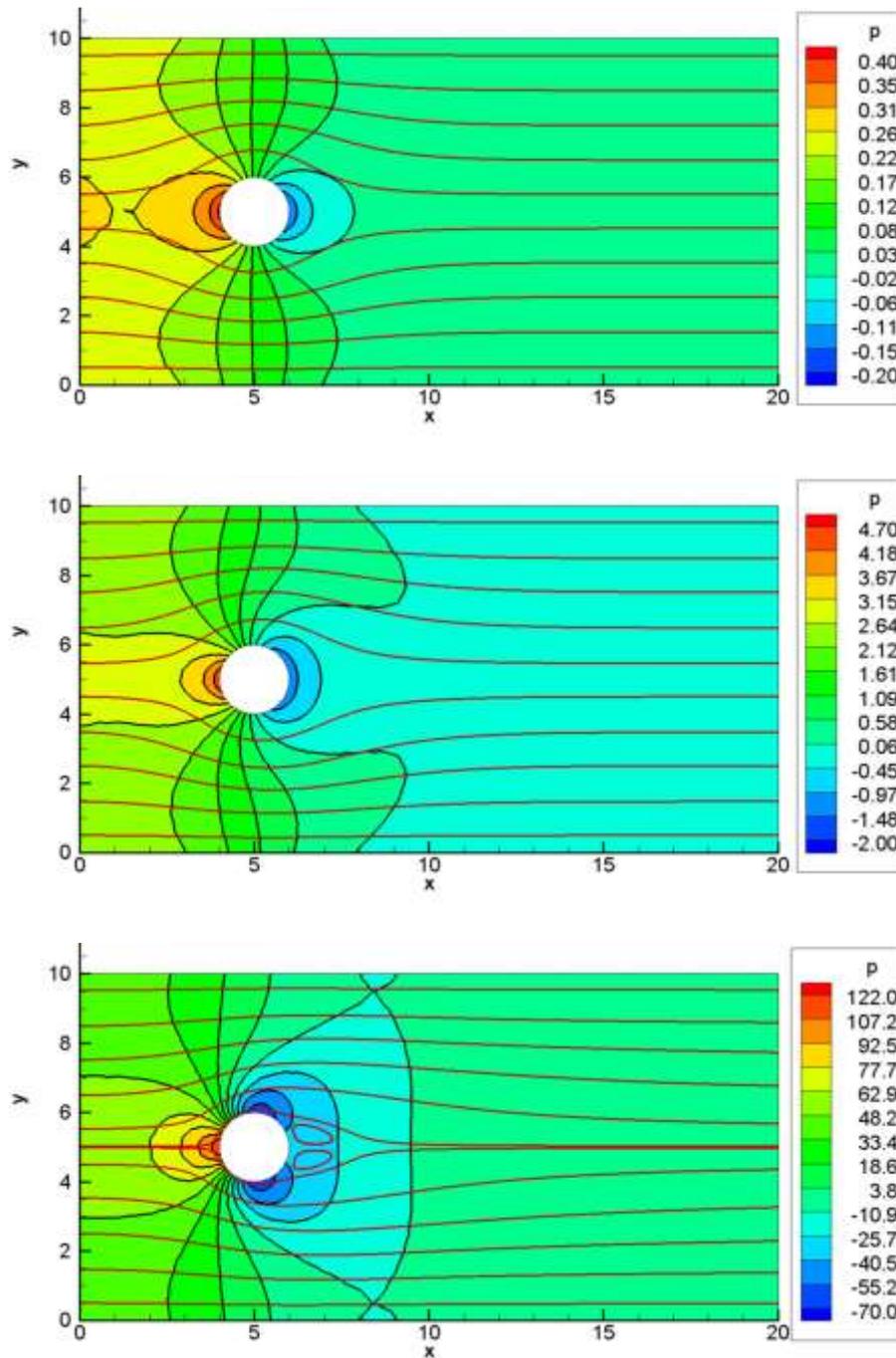


Figure 7.15 Pressure contours and streamlines for flow over a cylinder for Reynolds number of 0.1 (top), 1 (middle), 10 (bottom)

It is known that this flow field is steady and symmetric with respect to the horizontal centerline of the cylinder upto a Reynolds number of about 50. For higher Reynolds numbers flow becomes unsteady (actually time periodic) with the well known von Karman vortex street forming in the wake of the cylinder.

As mentioned previously size of the box is important in specifying uniform inflow and fully developed outflow BCs properly. Let's perform a new simulation, but this time use a twice larger box around the

cylinder. The width and length of the new box is 20x50 and the cylinder is located 20 and 30 units away from the inlet and outlet, respectively.

We simulated only $Re = 10$ case with this new geometry and obtained the results shown in Figure 7.16. This result should be compared with the last plot of Figure 7.15. As seen pressure scale changed considerably. This is an indication that previous domain was not large enough for proper specification of selected BCs. Is the new one large enough? To be sure we need to generate an even larger one and compare the results of it with the one given below.

According to Figure 7.16 pressure at the exit boundary seems to be not constant. But actually the range of pressure at the exit boundary is $-0.16 < p_{exit} < 0.08$, which is a small change compared to the actual range of pressure seen in the legend of Figure 7.16. Of course it is always possible to set the pressure of all the nodes on the exit boundary to zero.

As a conclusion, for the proper simulation of external uniform flow passing over an object we need to locate the outside box far enough away from the object.

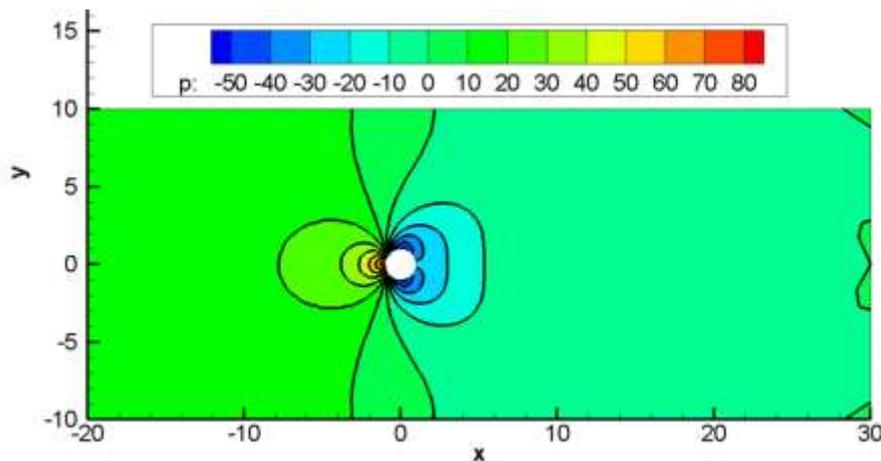


Figure 7.16 Pressure contours for $Re = 10$ obtained with a larger (20x50) box around the cylinder.

Pressure distribution around the cylinder is important for the estimation of drag force. For the solution given in Figure 7.16 pressure variation around the cylinder is given in Figure 7.17. As seen pressure is maximum at the front side stagnation point. It drops up to $\theta = 100^\circ$ and then starts to recover. After this point pressure gradient in the main flow direction is positive (adverse pressure gradient) and the flow separates. Overall pressure on the back side of the cylinder is much smaller compared to that of on the front side, creating a net pressure drag. The data can also be expressed as the non-dimensional $c_p = (p - p_\infty)/(0.5\rho U^2)$, where $p_\infty = p(-20,0) = 17.1$, $\rho = 1$ and $U = 10$.

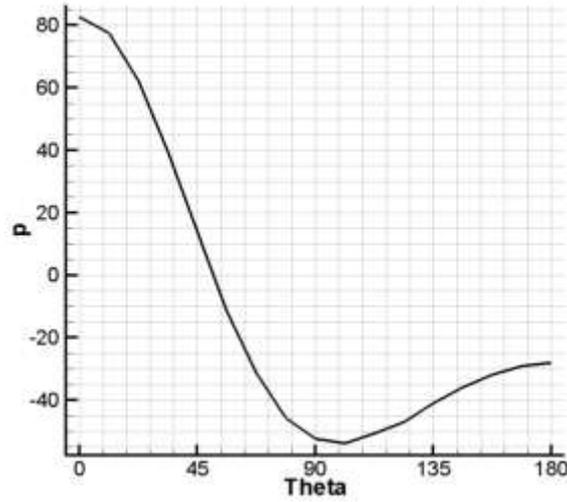


Figure 7.17 Pressure distribution over the upper half of the cylinder for $Re = 10$. $\theta = 0^\circ$ and $\theta = 180^\circ$ correspond to front side and back side stagnation points of the cylinder.

7.12 FE Formulation of the Energy Equation

In Section 7.7 we obtained GFEM formulation of continuity and linear momentum equations, i.e. Navier-Stokes equations. In this chapter we'll add the following energy equation to this set.

$$\text{Conservation of Energy : } \rho c_p (\vec{V} \cdot \nabla) T = k \nabla^2 T + \phi \quad (7.31)$$

where c_p is the specific heat at constant pressure, k is the conductivity of the fluid and ϕ is the viscous dissipation term, which is negligibly small in many practical applications. It'll also be neglected in the current FEM formulation.

When we consider the viscosity of the fluid to be constant, i.e. not a function of temperature, Navier-Stokes equations become decoupled from the energy equation. In other words for incompressible flows it is possible to first solve the Navier-Stokes equations and obtain the velocity field, and then use it in the energy equation to get the temperature distribution. It is worth to note that the energy conservation equation is the same as the advection-diffusion equation that we studied in detail in Chapter 6.

For temperature storage one can use the nodes where velocity components are stored. Weight function w_x used previously for the x -component of the momentum equation can be used for the discretization of the energy equation. After obtaining the weighted residual statement and applying integration by parts to the diffusion term we get the following elemental weak form

$$\int_{\Omega^e} \left[\rho c_p w_x \left(u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right) + k \left(\frac{\partial w_x}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial w_x}{\partial y} \frac{\partial T}{\partial y} \right) \right] d\Omega = \int_{\Gamma^e} w_x q_n d\Gamma \quad (7.32)$$

where u and v are the already known velocity field components obtained by the solution of N-S equations. q_n is the secondary variable given as

$$q_n = k \left(\frac{\partial T}{\partial x} n_x + \frac{\partial T}{\partial y} n_y \right) \quad (7.33)$$

which corresponds to the heat flux crossing the boundaries. For the approximation of the scalar temperature field we can make use of the shape functions S_j already derived and used for velocity components

$$T^h(x, y) = \sum_{j=1}^{NENp} T_j S_j(x, y) \quad (7.34)$$

Using this approximate temperature field in the elemental weak form we get the following elemental stiffness matrix.

$$K^e = \int_{\Omega^e} \left[\rho c_p S_i \left(u \frac{\partial S_j}{\partial x} + v \frac{\partial S_j}{\partial y} \right) + k \left(\frac{\partial S_i}{\partial x} \frac{\partial S_j}{\partial x} + \frac{\partial S_i}{\partial y} \frac{\partial S_j}{\partial y} \right) \right] d\Omega \quad (7.35)$$

Elemental force vector is identically equal to zero since we neglected the viscous dissipation term and internal heat generation.

7.13 Sample Heat Transfer Problem – Developing Flow Between Parallel Plates

Consider the hydrodynamically developed, but thermally developing flow in a channel formed by two infinitely wide parallel plates as shown in Figure 7.18. Channel height and length are 1 and 10, respectively. Flow enters the channel with parabolic velocity profile and uniform temperature. It heats up by the plates that are kept at fixed temperature. Fluid properties are taken to be constant constant as shown in the figure. Three simulations are performed with three different conductivity values of 1, 0.02 and 0.01. Results are shown in Figures 7.19 and 7.20.

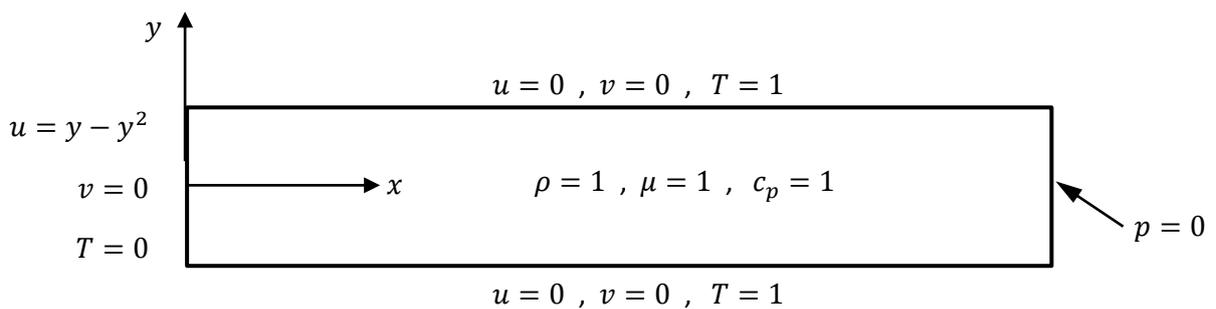


Figure 7.18 Thermally developing flow between parallel plates

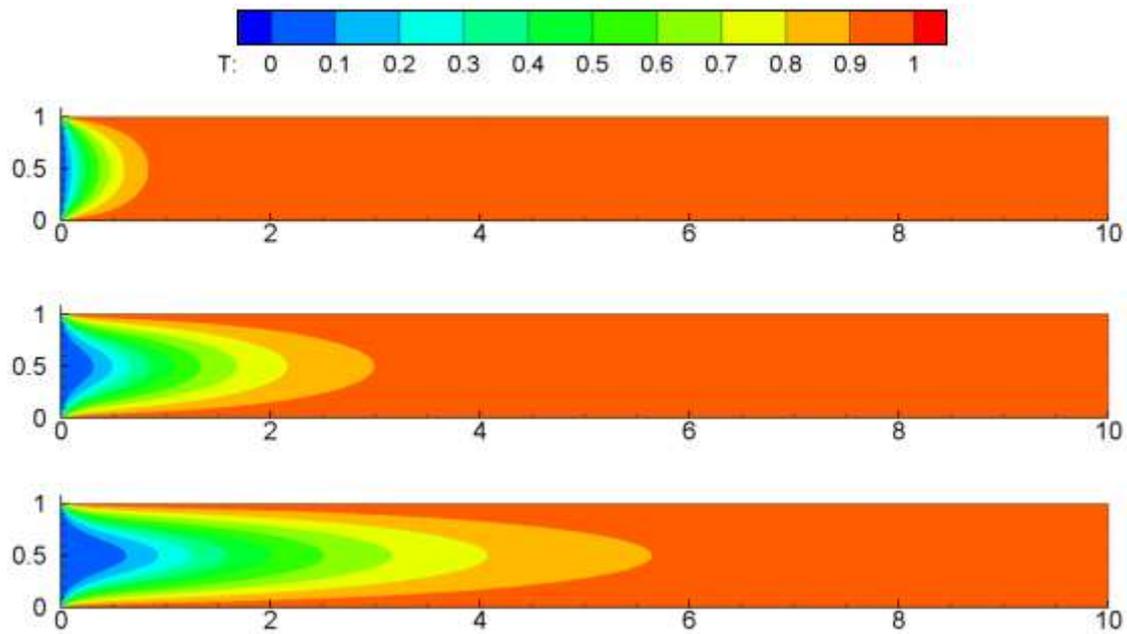


Figure 7.19 Temperature contours of thermally developing flow between parallel plates for $k = 1$ (Top), $k = 0.02$ (Middle) and $k = 0.01$ (Bottom)

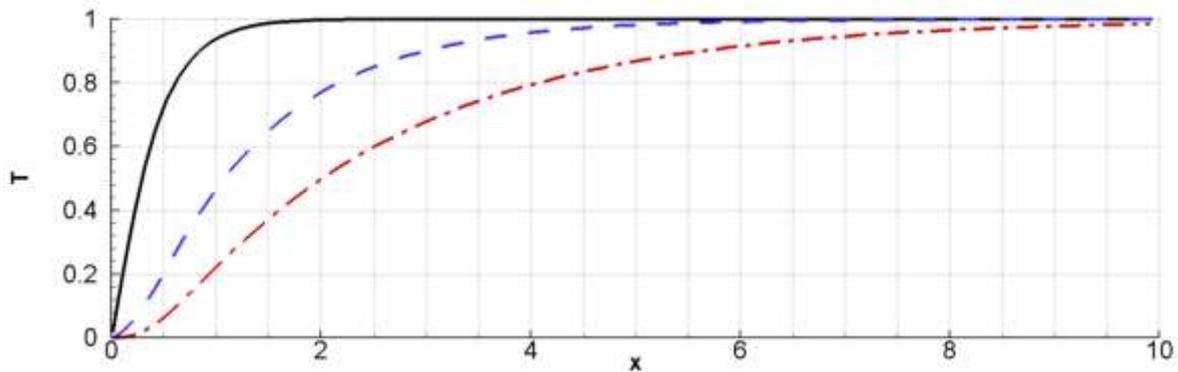


Figure 7.20 Temperature variation along the channel at $y = 0.5$ for $k = 1$ (Solid, black), $k = 0.02$ (Dashed, blue) and $k = 0.01$ (Dashed-dotted, red)

As seen in Figures 7.18 and 7.19, as the conductivity decreases, Prandtl number and therefore Peclet number increases. It means advection heat transfer becomes dominant compared to conduction heat transfer and thermal boundary layer develops in a slower rate.

7.14 Sample Heat Transfer Problem - Heat Exchanger

Consider the heat exchanger given in Figure 7.21, which is half of an actual exchanger. Water ($\rho = 1000 \text{ kg/m}^3$, $\mu = 1.08 \times 10^{-3} \text{ Pa} \cdot \text{s}$, $c_p = 4186 \text{ J/kgK}$, $k = 0.6 \text{ W/mK}$) enters the gray colored region from the left with a parabolic velocity profile and maximum velocity of 10^{-3} m/s .

Inlet water temperature is 20 °C. Water exits the domain from the right boundary. A warmer fluid at 50 °C passes through the circular tubes.

No slip BC is applied on the circular tubes with fixed temperature of 50 °C. Bottom boundary is the symmetry plane of the heat exchanger. $v = 0$ and $dT/dy = 0$ boundary conditions are applied at the bottom boundaries. Pressure is fixed to be zero on the exit plane. On the top surface of the heat exchanger no slip BC for velocity and insulated thermal BC are provided.

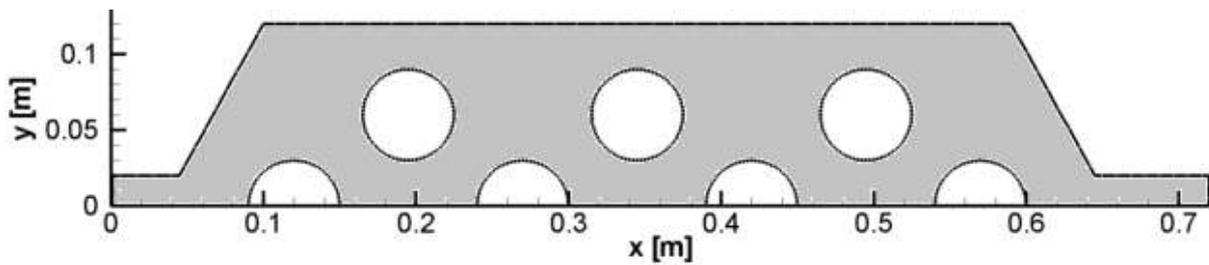


Figure 7.21 Geometry of the heat exchanger problem

A mesh of 2824 triangular elements is used for the simulation. Obtained solution is shown in Figure 7.22. Fluid heats up as it passes over the circular tubes and at the exit minimum and maximum temperatures are calculated to be 33.8 °C and 43 °C, respectively.

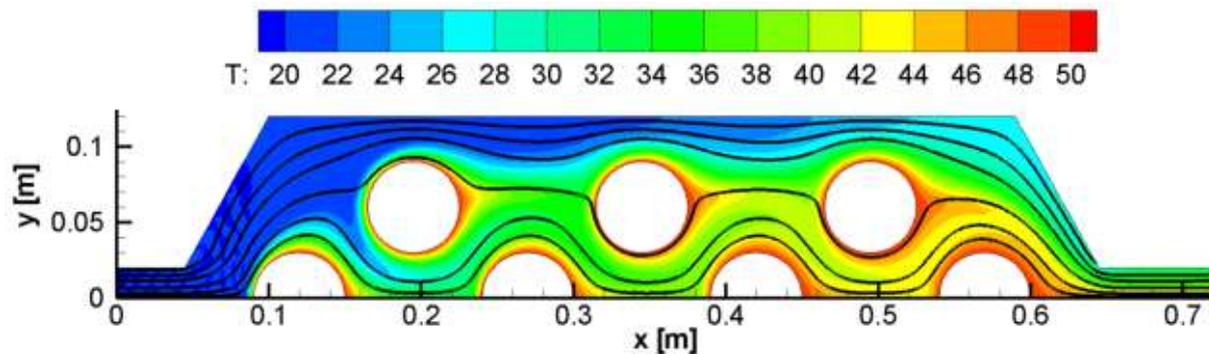


Figure 7.22 Streamlines and temperature contour for the heat exchanger problem.

Same problem is solved for a second time by increasing the flow rate by a factor of 10, i.e. maximum velocity of the inlet velocity profile is $10^{-2} m/s$. For this faster flow heat transfer from the circular tubes is less as seen in Figure 7.23. At the exit, minimum and maximum temperatures are calculated to be 19.1 °C and 36.6 °C, respectively. As seen, most of the incoming fluid follows a path close to the top wall, by-passing the circular tubes. For this faster flow a different tube arrangement could perform better. For example removing the tube closest to the inlet or putting tubes close to the top wall can be tried.

Here it is important to remember that after a certain critical speed flow over the tubes might become unsteady and a steady solution that we are performing here might result in erroneous solutions.

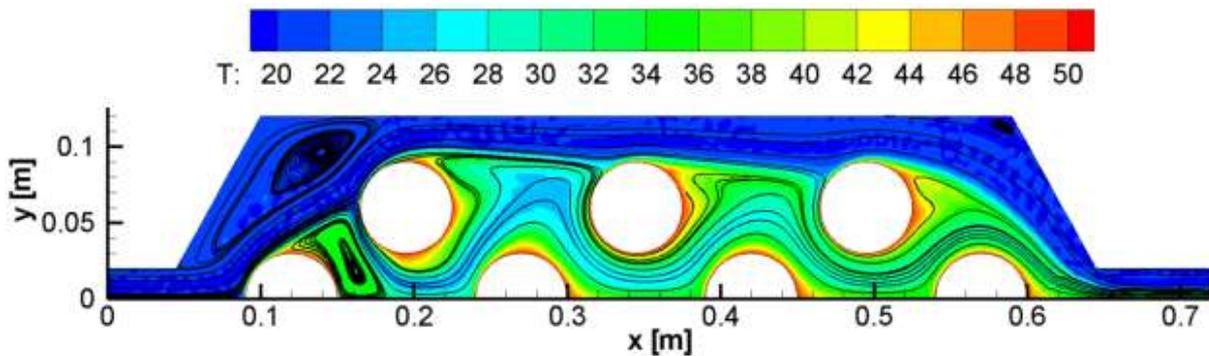


Figure 7.23 Streamlines and temperature contour for the heat exchanger problem with 10 times larger flow rate compared to the original solution.

7.15 Profiling the Code and Memory Usage

MATLAB comes with a tool called profiler. When you run a code using this tool a report will be generated showing how much time each function and each line of the code is taking. By this way it is possible to detect the time consuming bottlenecks and think about improvements.

When we profile the solver for the heat exchanger problem described in the previous section ($NE = 2824$, $NCN = 1652$, $NN = 6130$, $NENv = 6$, $NENp = 3$) we get the result shown in Figure 7.24. This figure shows the top part of a longer list. Profiling the code took a total of 58 seconds. This is longer than the actual run time of the code, because during profiling additional tasks are done.

According to Figure 7.24 `calcElemSys()` function is the most time consuming one. It took 14.2 s of 58 s. Second most time consuming function is `assemble()` with 12.4 s. Both of these functions are called 19,768 times. 3rd and 4th most time consuming functions are called only once and they are related to sparse storage. `solve()` function is the 5th on the list with 4.4 s, which is about 8 % of all run time. It is called 7 times because there were 7 Picard iterations in this solution. To conclude, constructing the elemental systems and assembling them are the most time consuming parts of the solution, not the actual solution of the global system. Note that there is also a function called `solveEnergy()` used for the linear algebraic system solution of the energy equation. It is not even in the list. It is called only once and it takes only 0.1 s.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
...yNavierStokesHeat2DStoarse>calcElemSys	19768	14.183 s	14.183 s	
steadyNavierStokesHeat2DStoarse>assemble	19768	12.351 s	12.351 s	
...rStokesHeat2DStoarse>setupKeKstospaceMap	1	7.114 s	7.114 s	
...NavierStokesHeat2DStoarse>setupKstospace	1	5.199 s	5.199 s	
steadyNavierStokesHeat2DStoarse>solve	7	4.434 s	4.434 s	
...avierStokesHeat2DStoarse>readInputFile	1	3.474 s	3.078 s	
...rStokesHeat2DStoarse>calcElemSysEnergy	2824	1.879 s	1.879 s	
...avierStokesHeat2DStoarse>createTecplot	1	1.844 s	1.844 s	
...sHeat2DStoarse>setupKeKstospaceEnergyMap	1	1.190 s	1.190 s	
steadyNavierStokesHeat2DStoarse>calcJacob	1	0.970 s	0.970 s	
...StokesHeat2DStoarse>setupKstospaceEnergy	1	0.958 s	0.958 s	

Figure 7.24 Profiling result of heat exchanger problem with NE = 2824, NN = 6130

`calcElemSys()` function takes most of the time. When you click on the function name more details about this function is presented, as seen in Figure 5.25. Top five most time consuming lines of the function are shown. It is possible to scroll down and see how much time other lines are taking.

Line Number	Code	Calls	Total Time	% Time	Time Plot
1426	<code>Ke22add(i,j) = Ke22add(i,j) + ...</code>	2846592	1.139 s	8.0%	
1418	<code>Ke11add(i,j) = Ke11add(i,j) + ...</code>	2846592	1.101 s	7.8%	
1424	<code>Ke12add(i,j) = Ke12add(i,j) + ...</code>	2846592	0.962 s	6.8%	
1431	<code>end</code>	2846592	0.878 s	6.2%	
1414	<code>gDSp(:, :) = elem(e).gDSp(:, :, k...</code>	79072	0.843 s	5.9%	
All other lines			9.261 s	65.3%	
Totals			14.183 s	100%	

Figure 5.25 Profiling result of the function `calcElemSys()` for the heat exchanger problem

Memory usage of the code is as important as the time it takes to run it. As mentioned earlier, by storing the global stiffness matrix of the N-S system and the energy equation system as sparse matrices we are able to run relatively big 2D problems. For the heat exchanger problem with $NE = 2824$, $NCN = 1652$, $NN = 6130$, $NENv = 6$, $NENp = 3$, there are 12,260 velocity unknowns, 1652 pressure unknowns and 6130 temperature unknowns. Totally there are 20,042 unknowns. Most memory demanding variables of this solution are listed below

<code>soln</code> structure	:	11,566,812 Bytes
<code>elem</code> structure	:	10,754,432 Bytes
A variable inside <code>solve()</code> function	:	4,110,908 Bytes
A variable inside <code>solveEnergy()</code> function	:	759,296 Bytes
<code>coord</code> variable	:	98,080 Bytes
<code>BC</code> structure	:	56,666 Bytes

Note that by default MATLAB stores all variables as double precision floating point numbers. To save memory, certain variables can be declared to be single precision numbers or integers. For example inside the most memory consuming `soln` structure there is another structure called `Ksparse`, which has arrays `row` and `col`. These arrays store integer values so they can be declared to be so. Another large variable is the `KeKsparseMap` matrix inside `elem` structure, which can also be defined as integer.

7.16 Segregated Formulation (Pressure Correction / Projection Methods)

In the formulation of Navier-Stokes equations discussed up to this point, all primitive variables, u , v and p are solved together in a single global system, which can be called a fully-coupled solution. Due to the weak coupling of pressure and velocity in incompressible flows this mixed formulation has couple of challenges as mentioned earlier. Most important of them is the zero entries located on the main diagonal of the stiffness matrix. Due to these zeros, iterative solution techniques are known to converge very slowly and they are not preferred. However, it is also known that as the size of the problem (number of the unknowns) gets larger direct solvers turn out to be disadvantageous in terms of both computation time and memory requirements. Most practical 3D simulations are beyond the capabilities of direct solvers. If we want to make use of iterative linear system solution techniques (do not get confused with the iterations that we are doing due to the linearization of the nonlinear terms of the Navier-Stokes equations) we need an alternative formulation.

In the Finite Difference and Finite Volume community it is a quite common practice to solve incompressible Navier-Stokes equations in a segregated way, i.e. velocity components and pressure are not assembled into a single system of global equations. This type of solution is popularized by Patankar by the name of SIMPLE (Semi Implicit Method for Pressure Linked Equations) and many of today's Finite Volume based commercial CFD software make use of this formulation. There are

several variants of SIMPLE. These segregated solution methods are also known as pressure correction methods or projection methods.

Fully-coupled formulation of 2D steady, incompressible Navier-Stokes equations result in the following system of global equations

$$\begin{bmatrix} [K_{xx}] & [K_{xy}] & [-C_x] \\ [K_{yx}] & [K_{yy}] & [-C_y] \\ [C_x^T] & [C_y^T] & [0] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{v\} \\ \{p\} \end{Bmatrix} = \begin{Bmatrix} \{F_x\} \\ \{F_y\} \\ \{b_c\} \end{Bmatrix} \quad (7.36)$$

Here we assumed that nonlinear terms are already included in $[K_{xx}]$ and $[K_{yy}]$ matrices using for example the Picard iteration technique. Sizes of the submatrices and subvectors of Eqn (7.36) depend on the number of velocity and pressure unknowns of the problem. Right hand side of the third equation, which is the continuity equation, is not zero due to the possible contributions of non-zero boundary conditions for specified velocities. Such BC contributions may also exist in $\{F_x\}$ and $\{F_y\}$ vectors. Segregated solution technique described below is based on the discussion of reference [4].

Equation (7.36) can also be written in the following open form

$$[K_{xx}]\{u\} + [K_{xy}]\{v\} - [C_x]\{p\} = \{F_x\} \quad (7.37a)$$

$$[K_{yx}]\{u\} + [K_{yy}]\{v\} - [C_y]\{p\} = \{F_y\} \quad (7.37b)$$

$$[C_x^T]\{u\} + [C_y^T]\{v\} = \{b_c\} \quad (7.37c)$$

or in the following rearranged form

$$[K_{xx}]\{u\} - [C_x]\{p\} = \{F_x\} - [K_{xy}]\{v\} = \{f_x\} \quad (7.38a)$$

$$[K_{yy}]\{v\} - [C_y]\{p\} = \{F_y\} - [K_{yx}]\{u\} = \{f_y\} \quad (7.38b)$$

$$[C_x^T]\{u\} + [C_y^T]\{v\} = \{b_c\} \quad (7.38c)$$

Note that new defined $\{f_x\}$ and $\{f_y\}$ are different than the original $\{F_x\}$ and $\{F_y\}$. In a segregated formulation we think that the first equation can be used to solve for $\{u\}$, the second one can be used to solve for $\{v\}$ and the last one is used to solve for $\{p\}$. The problem is that pressure does not exist in the last equation. Therefore we need to combine and arrange these equations to get a new equation that can be solved for pressure. Also note that since all these three unknowns (u , v and p) depend on each other, segregated solutions are iterative by nature (again do not get confused with nonlinear iterations or iterations of an iterative linear system solver, they are all different things) and they start from an initial guessed solution.

To obtain the missing equation for pressure consider Eqn (7.38a) written for iteration i and $i + 1$ (Brackets are dropped for simplicity)

$$K_{xx}u^i + K_{xy}v^i - C_x p^i = f_x^i \quad \text{and} \quad K_{xx}u^{i+1} + K_{xy}v^{i+1} - C_x p^{i+1} = f_x^{i+1} \quad (7.39)$$

Left multiply Eqns (7.39) with $C_x^T K_{xx}^{-1}$ and subtract the one for iteration i from the one for iteration $i + 1$. What we get is

$$C_x^T (u^{i+1} - u^i) - C_x^T K_{xx}^{-1} C_x (p^{i+1} - p^i) = K_{xx}^{-1} (f_x^{i+1} - f_x^i) \quad (7.40a)$$

Now write equations similar to the ones in Eqn (5.24) for the y -momentum equation. Left multiply them with $C_y^T K_{yy}^{-1}$ and subtract from each other to get

$$C_y^T (v^{i+1} - v^i) - C_y^T K_{yy}^{-1} C_y (p^{i+1} - p^i) = K_{yy}^{-1} (f_y^{i+1} - f_y^i) \quad (7.40b)$$

Add Eqns (7.40a) and (7.40b)

$$\begin{aligned} & \underline{C_x^T u^{i+1} + C_y^T v^{i+1}} - (C_x^T u^i + C_y^T v^i) - (C_x^T K_{xx}^{-1} C_x + C_y^T K_{yy}^{-1} C_y) (p^{i+1} - p^i) \\ & = \underline{K_{xx}^{-1} (f_x^{i+1} - f_x^i)} + \underline{K_{yy}^{-1} (f_y^{i+1} - f_y^i)} \end{aligned} \quad (7.41)$$

Note that superscripts i and $i + 1$ denote current and next iteration levels. This equation is used in an iterative loop and upon convergence the underlined term will approach to $\{b_c\}$ due to the continuity equation. Similarly upon convergence double underlined terms will approach to zero. Using these facts Eqn (7.41) can be simplified to

$$(C_x^T K_{xx}^{-1} C_x + C_y^T K_{yy}^{-1} C_y) \Delta p^{i+1/2} = -C_x^T u^i - C_y^T v^i \quad (7.42a)$$

where $\Delta p^{i+1/2} = p^{i+1} - p^i$ is an intermediate pressure difference between two iterations. After obtaining $\Delta p^{i+1/2}$, we use it in Eqns (7.40a) and (7.40b) to obtain intermediate velocities

$$u^{i+1/2} = u^i + K_{xx}^{-1} C_x \Delta p^{i+1/2} \quad (7.42b)$$

$$v^{i+1/2} = v^i + K_{yy}^{-1} C_y \Delta p^{i+1/2} \quad (7.42c)$$

Step 1 : First step of the segregation solution process is the solution of Eqn (7.42a) to obtain intermediate pressure difference

Here it is important to note that calculating the inverse of K_{xx} and K_{yy} so that the left hand side matrix of Eqn (7.42a) can be formed is problematic, because inverse of a banded matrix is a full matrix, needing excessive storage. Remember the original motivation of seeking a segregated formulation; being able to use an iterative linear system solver so that run time and memory

requirements can be reduced. For this purpose instead of taking the inverses of original K_{xx} and K_{yy} matrices it is preferred to take the inverses of other representative matrices that are easier to invert. This simplification does not alter the nature of the formulation, but it results in slower convergence of the iterations. Usually diagonal matrices formed by the diagonal entries of K_{xx} and K_{yy} are used for this purpose

$$\tilde{K}_{xx} = \text{diag}(K_{xx}) \quad , \quad \tilde{K}_{yy} = \text{diag}(K_{yy}) \quad (7.43)$$

When these diagonal matrices are used in Eqns (5.27a) we get the following equation that should be solved in Step 1.

$$(C_x^T \tilde{K}_{xx}^{-1} C_x + C_y^T \tilde{K}_{yy}^{-1} C_y) \Delta p^{i+1/2} = -C_x^T u^i - C_y^T v^i \quad (7.44)$$

Step 2: Use Eqns (7.42b) and (7.42c) (but with diagonal matrices) to get $u^{i+1/2}$ and $v^{i+1/2}$.

$$u^{i+1/2} = u^i + \tilde{K}_{xx}^{-1} C_x \Delta p^{i+1/2} \quad (7.45a)$$

$$v^{i+1/2} = v^i + \tilde{K}_{yy}^{-1} C_y \Delta p^{i+1/2} \quad (7.45b)$$

Also update pressure as

$$p^{i+1} = p^i + \Delta p^{i+1/2}$$

Since segregated solution is an iterative process $\Delta p^{i+1/2}$ has considerable error in it and the above pressure updating is usually relaxed as follows to obtain a converging solution

$$p^{i+1} = p^i + (1 - \alpha_p) \Delta p^{i+1/2} \quad (7.46)$$

where $0 < \alpha_p < 1$ is the pressure relaxation factor. Values close to zero provide no relaxation and values close to 1 provide full relaxation, ie. no change from previous iteration.

Step 3: After obtaining the pressure for the new iteration we can now solve momentum equations to get new values for velocity components. Using Eqn (5.23a)

$$u^{i+1} = K_{xx}^{-1} (f_x - C_x p^{i+1})$$

which can also be relaxed as follows

$$u^{i+1} = \left[\left(\frac{\alpha_u}{1 - \alpha_u} \right) \tilde{K}_{xx} + K_{xx} \right]^{-1} \left[f_x - C_x p^{i+1} + \left(\frac{\alpha_u}{1 - \alpha_u} \right) \tilde{K}_{xx} u^{i+1/2} \right] \quad (7.47)$$

Upon convergence $u^{i+1/2}$ and u^{i+1} will be the same and underlined terms of the above equation will cancel out. Similar to pressure relaxation, the role of velocity relaxation is again to control the speed of change of variables the iterative solution process. Also the addition of $\left(\frac{\alpha_u}{1-\alpha_u}\right) \tilde{K}_{xx}$ makes K_{xx} more diagonally dominant and therefore makes this system more suitable to be solved with iterative solution techniques.

Step 4 : Repeat step 3 but this time use Eqn (7.38b) to obtain v velocities at new iteration level can be obtained.

$$v^{i+1} = \left[\left(\frac{\alpha_v}{1-\alpha_v} \right) \tilde{K}_{yy} + K_{yy} \right]^{-1} \left[f_y - C_y p^{i+1} + \left(\frac{\alpha_v}{1-\alpha_v} \right) \tilde{K}_{yy} v^{i+1/2} \right] \quad (7.48)$$

This 4 step process should be repeated until convergence for all variables is achieved. It is apparent that the solution should start with an initial guess. Note that Picard iteration that is used for fully-coupled solution of previous sections is already included in the iteration loop of this segregated solution approach, therefore we do not need an extra loop for linearization of nonlinear terms.

7.17 Matrix Free, Element By Element (EBE) Solution

In the segregated solution approach described in the previous section computationally intensive tasks are the solution of pressure equation, Eqn (7.44), and solution of momentum equations, Eqns (7.47) and (7.48). These equations can be solved using direct or iterative techniques. Segregated solvers are designed for this new iterative solver option, which is not possible in the fully-coupled solutions.

Pressure equation, Eqn (7.44), is symmetric and positive-semidefinite, allowing the use of a popular iterative method named Conjugate Gradient (CG). Momentum equations, Eqns (7.47) and (7.48), on the other hand are not symmetric and they can be solved using Conjugate Gradient Squared (CGS) or Generalized Minimum Residual (GMRES) methods.

For all these iterative system solution methods, main task is the multiplication of the square right-hand-side matrix with a vector. In Element By Element (EBE) solution approach this matrix vector multiplication is done using element level matrices and global matrices are not assembled at all, hence the name matrix-free solution. By avoiding the assembly of global systems EBE solutions result in enormous memory savings. fulfilling the initial motivation of using segregated solution technique.

7.18 Other FE Formulations of Incompressible N-S Equations

In previous sections we discussed mixed (fully-coupled) and segregated FE formulations of incompressible N-S equations. In practice the mixed formulation is not preferred due to its high

memory requirement and stability issues, and when used, almost always it is supported by a stabilization technique such as SUPG. Although the idea of segregated solution is very popular in finite volume solvers, it is also not used heavily by the finite element community. There are many other alternative formulations for the solution of incompressible flows. Some of them are penalty method [2], least-squares FEM [5], artificial compressibility method [6], fractional step method [7] Taylor-Galerkin method [6] and characteristic based split method [8].

7.19 Exercises

E-7.1. Implement the Newton linearization technique described in Section 7.8 to the 2D N-S code and compare its convergence speed with the Picard technique.

E-7.2. Solve the lid-driven cavity problem for $Re = 5000$ and compare the results with a reference solution.

E-7.3. For the flow over a cylinder problem studied in Section 7-11 one of the most important results is the drag coefficient C_D , which is a nondimensional drag force. In order to calculate it both the shear drag and the pressure drag over the cylinder needs to be calculate. Perform C_D calculation for $Re = 1, 10$ and 40 , and compare your results with the ones available in the literature.

E-7.4. A popular benchmark problem for incompressible N-S solvers is the flow over a backward facing step. Do a literature survey for this flow and find reference solutions, perform simulations and compare results with the available ones.

E-7.5. One simple suggestion to increase the heat transfer rate of the heat exchanger problem discussed in Section 7.14 is to use more tubes with smaller diameter. Perform new simulations with a number of alternative designs based on this concept. Compare exit temperatures. Also compare the pressure drop (pressure difference between the inlet and the exit) and evaluate new designs from this perspective.

E-7.6. Use the 2D N-S code to study the boundary layer development over a flat plate. Select a properly sized problem domain, boundary conditions and flow and fluid properties. Generate a good enough mesh and obtain the developing laminar boundary layer over the flat plate. Compare the growth rate of the boundary layer and the velocity profiles inside the boundary layer with the known analytical solution. Perform the solution for a range of (not for a single) Reynolds number.

E-7.7. In this chapter we only considered 2D flows in the Cartesian coordinate system. 2D, axisymmetric flows in the rz plane of the cylindrical coordinate system are also frequently encountered. Work on the formulation of axisymmetric flows and modify the N-S solver accordingly. Test the new code for the developing laminar flow in a constant diameter pipe problem. Consider a long enough pipe with uniform inlet velocity and let the flow develop into the analytically known parabolic profile. Calculate the constant rate of pressure drop in the fully developed region and compare with the known analytical value.

7.20 References

- [1] P. M. Gresho, R. L. Sani, *Incompressible Flow and the Finite Element Method*, John Wiley and Sons, 2000.
- [2] J. N. Reddy, D. K. Gartling, *The Finite element Method in Heat Transfer and Fluid Dynamics*, 2nd ed., CRC Press, 2000.
- [3] U. Ghia, K. N. Ghia, C. T. Shin, High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *J. Comput. Phys.*, 48, 387–411, 1982.
- [4] V. Haroutunian, M. S. Engelman, I. Hasbani, Segregated Finite element Algorithms for the Numerical Solution of Large-Scale Incompressible Flow Problems, *Int. J Numer. Meth. Fluids*, 17, 323-348, 1993.
- [5] B-N Jiang, *The Least-Squares Finite Element Method – Theory and Applications in Computational Fluid Dynamics and Electromagnetics*, Springer, 1998.
- [6] R. Löhner, *Applied CFD Techniques – An Introduction Based on the Finite Element Methods*, John Wiley and Sons, 2001.
- [7] J. Donea, S. Giuliani, H. Laval, L. Quartapelle, Finite element of the Unsteady Navier-Stokes Equations by a Fractional Step Method, *Comp. Meth Appl. Mech Engrg*, 30, 53-73, 1982.
- [8] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, *The Finite Element Method for Fluid Dynamics*, 6th ed., Elsevier, 2006.