FEAP - - A Finite Element Analysis Program

Version 8.6 CFD Manual

Robert L. Taylor & Sanjay Govindjee
Department of Civil and Environmental Engineering
University of California at Berkeley
Berkeley, California 94720-1710

June 2020

Contents

List of Figures

List of Tables

Chapter 1

Computational Fluid Dynamics

The solution of fluid dynamics problems involves dealing with complex non-linear problems. The only form considered here is the Navier-Stokes theory using velocity and pressure dependent variables. The numerical solution is generally called *Computational Fluid Dynamics* or merely *CFD*.

In this chapter we summarize the basic theory and describe some discretization forms for the velocity and pressure. Chapter 2 describes the input of the material set data for solution using *FEAP*. The description of other aspects of mesh input is described in the *FEAP* User Manual [1].

Chapter 3 summarizaes the use of the *split algorithm* to solve problems formulated by the Chorin and Donea formulations.

1.1 Basic equations and weak form

1.1.1 Basic equations in conserving form

The governing equations for fluid dynamics may be written as

$$\frac{\partial \boldsymbol{\Phi}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \frac{\partial \mathbf{G}_i}{\partial x_i} - \mathbf{Q} = \mathbf{0} \tag{1.1}$$

where the individual terms are given by

$$\boldsymbol{\Phi} = \begin{Bmatrix} \rho\, u_a \\ c^{-2} p \end{Bmatrix} \tag{1.2a}$$

with $c^2 = K/\rho$, u_a are the velocity components for a range over the spatial dimension of the problem. The other relations are

$$\mathbf{F}_i = \begin{Bmatrix} \rho\, u_a\, u_i \\ \rho\, u_i \end{Bmatrix} \tag{1.2b}$$

$$\mathbf{G}_i = \begin{Bmatrix} -\delta_{ai}\, p + \tau_{ai} \\ 0 \end{Bmatrix} \tag{1.2c}$$

and

$$\mathbf{Q} = \begin{Bmatrix} \rho\, g_a \\ 0 \end{Bmatrix} \tag{1.2d}$$

In the above the terms

$$u_a\, u_i \rightarrow \begin{Bmatrix} u_1\, u_i \\ u_2\, u_i \\ u_3\, u_i \end{Bmatrix} \tag{1.2e}$$

and similar for other terms involving the a subscript.

For cases of constant density, the expansion for $\mathbf{F}_i$ leads to

$$\frac{\partial \mathbf{F}_i}{\partial x_i} = \begin{Bmatrix} \rho\, \left(u_a\, u_{i,i} + u_{a,i}\, u_i\right) \\ \rho\, u_{i,i} \end{Bmatrix} \tag{1.3}$$

and for incompressible cases where $c \rightarrow \infty$ (1.3) may be simplified to

$$\frac{\partial \mathbf{F}_i}{\partial x_i} = \begin{Bmatrix} \rho\, u_{a,i}\, u_i \\ \rho\, u_{i,i} \end{Bmatrix} \tag{1.4}$$

This sometimes referred to as the *non-conservative form*. In the sequel the derivations are generally given in the full, or *conservative* form of the equations. However, in discretizing we shall make approximations that are equivalent to those of the non-conservative form.

1.1.2 Weak forms

The weak form form (1.1) is given by

$$G(u_i, p) = \int_V \begin{bmatrix} \delta u_a & \delta p \end{bmatrix} \left[\frac{\partial \mathbf{\Phi}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \frac{\partial \mathbf{G}_i}{\partial x_i} - \mathbf{Q} \right] \mathrm{d}V = 0 \tag{1.5}$$

for which the $\mathbf{G}_i$ term is integrated by parts to give the from from which C_0 functions may be used to approximate the velocities u_a, giving

$$\begin{aligned} G(u_i, p) = &\int_V \begin{bmatrix} \delta u_a & \delta p \end{bmatrix} \left[\frac{\partial \mathbf{\Phi}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \mathbf{Q} \right] \mathrm{d}V \\ &+ \int_V \delta \epsilon_{ai} \left[-\delta_{ai}\, p + \tau_{ai} \right] \mathrm{d}V - \int_{\Gamma_t} \delta u_a \bar{t}_a \, \mathrm{d}A = 0 \end{aligned} \tag{1.6a}$$

where

$$\epsilon_{ij} = \tfrac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{1.6b}$$

with a similar expression for its variation and

$$\tau_{ij} = 2\,\mu\,\left(\epsilon_{ij} - \tfrac{1}{3}\,\delta_{ij}\epsilon_{kk}\right) \tag{1.6c}$$

defines the viscous deviatoric stresses for a linear Newtonian fluid.

1.1.3 Linearization for a Newton solution

To construct a Newton solution to the above it is necessary to linearize (1.6a). Accordingly, the result gives

$$dG = \int_V \begin{bmatrix}\delta u_a & \delta p\end{bmatrix} \left[\frac{\partial d\mathbf{\Phi}}{\partial t} + \frac{\partial d\mathbf{F}_i}{\partial x_i}\right] \mathrm{d}V + \int_V \delta\epsilon_{ai}\left[-\delta_{ai}\,dp + d\tau_{ai}\right]\mathrm{d}V \tag{1.7a}$$

where

$$\begin{aligned} d\mathbf{\Phi} &= \begin{Bmatrix}\rho\,du_a \\ c^{-2}\,dp\end{Bmatrix} \\ d\mathbf{G}_i &= \begin{Bmatrix}\delta_{ai}\,dp + d\tau_{ai} \\ 0\end{Bmatrix} \\ d\mathbf{F}_i &= \begin{Bmatrix}\rho\,(du_a\,u_i + u_a\,du_i) \\ \rho u_i\end{Bmatrix}\end{aligned} \tag{1.7b}$$

Expanding (1.6a) for the pairs of equations yields for the velocity weak form

$$\begin{aligned} G_u &= \int_V \delta u_a\left[(\rho\,\dot{u}_a - g_a) + \rho\,\left(u_a\,u_i\right)_{,i}\right]\,\mathrm{d}V \\ &\quad + \int_V \delta\epsilon_{ai}\left(-\delta_{ai}p + \tau_{ai}\right)\,\mathrm{d}V - \int_{\Gamma_t} \delta u_a\,\bar{t}_a\,\mathrm{d}A = 0 \end{aligned} \tag{1.8a}$$

and the continuity weak form

$$G_p = \int_V \delta p\left[K^{-1}\,\dot{p} + u_{i,i}\right]\,\mathrm{d}V = 0 \tag{1.8b}$$

Similarly, splitting the linearizations in (1.7a) gives

$$\begin{aligned} dG_u &= \int_V \delta u_a\left[\rho\,d\dot{u}_a + \rho\,d\left(u_a\,u_i\right)_{,i}\right]\,\mathrm{d}V \\ &\quad + \int_V \delta d\epsilon_{ai}\left(-\delta_{ai}p + \tau_{ai}\right)\,\mathrm{d}V \end{aligned} \tag{1.9a}$$

and

$$dG_p = \int_V \delta p\left[K^{-1}\,d\dot{p} + du_{i,i}\right]\,\mathrm{d}V \tag{1.9b}$$

1.1.4 Incompressible Navier-Stokes equations

For an incompressible material $K \to \infty$ (and thus $c \to \infty$) and the pressure rate term may be dropped. Changing the sign on (1.8b) and (1.9b) results in symmetry of the pressure-volume rate term. In this case the pair of weak forms may be written as:

$$\begin{aligned} G_u = & \int_V \delta u_a \left[(\rho\, \dot{u}_a - g_a) + \rho\, (u_a\, u_i)_{,i} \right] \mathrm{d}V \\ & + \int_V \delta\epsilon_{ai}\, (-\delta_{ai} p + \tau_{ai})\, \mathrm{d}V - \int_{\Gamma_t} \delta u_a\, \bar{t}_a\, \mathrm{d}A = 0 \end{aligned} \tag{1.10a}$$

and

$$G_p = - \int_V \delta p\, u_{i,i}\, \mathrm{d}V = 0 \tag{1.10b}$$

1.2 Finite element discretization

The velocity u_a and its variation are interpolated using an isoparametric approximation with approximations expressed in direct (matrix) notation as

$$\begin{aligned} \mathbf{x} &= N_\alpha^{(u)}(\boldsymbol{\xi})\, \tilde{\mathbf{x}}_\alpha \\ \mathbf{u} &= N_\alpha^{(u)}(\boldsymbol{\xi})\, \tilde{\mathbf{u}}_\alpha \\ \delta\mathbf{u} &= N_\alpha^{(u)}(\boldsymbol{\xi})\, \delta\tilde{\mathbf{u}}_\alpha \end{aligned} \tag{1.11a}$$

The pressure has no derivatives and may be approximated by either a piecewise continuous or continuous approximation where

$$\begin{aligned} p &= N_\gamma^{(p)}\, \tilde{p}_\gamma \\ \delta p &= N_\gamma^{(p)}\, \delta\tilde{p}_\gamma \end{aligned} \tag{1.11b}$$

Using these in (1.10a) and introducing Voigt notation yields[1] Details on shape functions and basic finite element schemes uses may be found in References [2] and [3].

$$\begin{aligned} G_u = & \delta\tilde{\mathbf{u}}_\alpha^T \int_V N_\alpha^{(u)} \left[\left(\rho\, N_\beta^{(u)}\, \dot{\tilde{\mathbf{u}}}_\beta - \mathbf{g} \right) + \rho\, (\boldsymbol{\nabla}\mathbf{u})\, \mathbf{u} \right] \mathrm{d}V \\ & + \delta\tilde{\mathbf{u}}_\alpha^T \int_V \mathbf{B}_\alpha^T\, (-\mathbf{m}\, p + \boldsymbol{\tau})\, \mathrm{d}V - \delta\tilde{\mathbf{u}}_\alpha^T \int_{\Gamma_2} N_\alpha^{(u)}\, \bar{\mathbf{t}}\, \mathrm{d}A = 0 \end{aligned} \tag{1.12a}$$

Similarly in (1.10b) one obtains

$$G_p = -\delta p_\gamma \int_V N_\gamma^{(p)}\, \mathbf{m}^T \mathbf{B}_\beta \tilde{\mathbf{u}}_\beta\, \mathrm{d}V = 0 \tag{1.12b}$$

[1] The direct notation $(\boldsymbol{\nabla}\mathbf{u})\,\mathbf{u}$ equals $u_i\, u_{a,i}$ in direction a using index notation.

These equations are supplemented by the velocity boundary condition

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on} \quad \Gamma_1 \tag{1.12c}$$

The evaluation of the integrals leads to the following arrays:

$$\begin{aligned}
\mathbf{M}_{\alpha\beta} &= \int_V N_\alpha^{(u)}\, \rho\, N_\beta^{(u)}\, \mathrm{d}V\, \mathbf{I} \\
\mathbf{A}_\alpha &= \int_V N_\alpha^{(u)}\, \boldsymbol{\nabla}\mathbf{u}\mathbf{u}\, \mathrm{d}V \\
\mathbf{P}_\alpha &= \int_V \mathbf{B}_\alpha^T \boldsymbol{\tau}\, \mathrm{d}V \\
\mathbf{C}_{\alpha\gamma} &= \int_V \mathbf{m}^T \mathbf{B}_\alpha\, N_\gamma^{(p)}\, \mathrm{d}V = \int_V \mathbf{b}_\alpha\, N_\gamma^{(p)}\, \mathrm{d}V \\
\mathbf{f}_\alpha &= \int_V N_\alpha^{(u)}\, \mathbf{g}\, \mathrm{d}V + \int_{\Gamma_2} N_\alpha^{(u)}\, \bar{\mathbf{t}}\, \mathrm{d}A
\end{aligned} \tag{1.13}$$

where $\mathbf{b}_\alpha = [N_{\alpha,1}^{(u)}, N_{\alpha,2}^{(u)}, N_{\alpha,3}^{(u)}]^T$, the volume change derivatives. This allows the equations to be written as

$$\begin{aligned}
\mathbf{M}_{\alpha\beta}\dot{\tilde{\mathbf{u}}}_\beta + \mathbf{A}_\alpha(\mathbf{u}) - \mathbf{C}_{\alpha\gamma}\, \tilde{p}_\gamma + \mathbf{P}_\alpha(\mathbf{u}) &= \mathbf{f}_\alpha \\
-\mathbf{C}_{\gamma\beta}^T\, \tilde{\mathbf{u}}_\beta &= \mathbf{0}
\end{aligned} \tag{1.14}$$

with the additional requirement $\mathbf{u} = \bar{u}$ on Γ_1. Note that the equations also use $\delta\mathbf{u} = \mathbf{0}$ on Γ_1.

1.2.1 Taylor-Hood solution

The Taylor-Hood approach uses a continuous interpolation for both the velocity and the pressure [4]. The continuous pressure is assumed one order lower than that for the velocity, thus, the lowest order is a quadratic order velocity with a linear order pressure. The method may be used for either steady state or transient solutions. The method is monolithic and commonly uses a Newton method to solve the non-linear equations. Accordingly, a linearization of (1.14) about the current solution yields

$$\begin{aligned}
\mathbf{M}_{\alpha\beta} d\dot{\tilde{\mathbf{u}}}_\beta - \mathbf{C}_{\alpha\gamma}\, \tilde{p}_\gamma + \mathbf{K}_{\alpha\beta}\, d\tilde{\mathbf{u}}_\beta &= \mathbf{R}_\alpha \\
-\mathbf{C}_{\gamma\beta}^T\, d\tilde{\mathbf{u}}_\beta &= \mathbf{r}_\gamma
\end{aligned} \tag{1.15}$$

A discrete time integration method is introduced that permits

$$d\dot{\tilde{\mathbf{u}}}_\beta = c_1\, d\tilde{\mathbf{u}}_\beta \tag{1.16}$$

where $c_1 = \mathrm{O}(1/\Delta\mathrm{t})$ the discrete time increment. The residuals $\mathbf{R}_\alpha$ and $\mathbf{r}_\gamma$ result from moving all theterms in (1.14) to the right hand side of the equal sign. The set of matrix equations for the solution may then be written as

$$\begin{bmatrix} (c_1\,\mathbf{M}_{\alpha\beta} + \mathbf{K}_{\alpha\beta}) & -\mathbf{C}_{\alpha\gamma} \\ -\mathbf{C}^T_{\delta\beta} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} d\tilde{\mathbf{u}}_\beta \\ d\tilde{p}_\gamma \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_\alpha \\ \mathbf{r}_\delta \end{Bmatrix} \tag{1.17}$$

1.2.2 Chorin split

The incompressible Navier-Stokes equations are given by

$$\begin{aligned} &\rho \dot{u}_a + \rho\,(u_a u_i)_{,i} - \tau_{ia,i} + p_{,a} = g_a \\ &u_{i,i} = 0 \end{aligned} \tag{1.18}$$

The Chorin split consists of discretizing in time and spitting the first of (1.18) as (see Chorin [5])

$$\begin{aligned} \frac{\rho}{\Delta t}(u^\star_a - u^n_a) &= \tau^n_{ia,i} - \rho(u^n_a u^n_i)_{,i} \\ \frac{\rho}{\Delta t}(u^{n+1}_a - u^\star_a) &= -p^{n+1}_{,a} + g_a \end{aligned} \tag{1.19}$$

where $(\cdot)^n$ denotes an approximation at time t_n. The second of (1.19) may be solve once the pressure is known. Taking the divergence of this equation and using the second of (1.18) yields

$$-\frac{\rho}{\Delta t} u^\star_{a,a} = -p^{n+1}_{,aa} + g_a \tag{1.20}$$

which yields a Poisson equation for the pressure at time t_{n+1}.

The above system may be discretized to obtain a three step solution scheme similar to the Donea *et al.* scheme above, however, the split is performed on the strong form of the time discretized equations. Thus one has the weak forms

$$\begin{aligned} \int_V \delta u_a \left[\frac{\rho}{\Delta t}\,(u^\star_a - u^n_a) + \rho(u^n_a u^n_i)_{,i}\right] \mathrm{d}V \\ + \int_V \delta\epsilon_{ai}\,\tau^n_{ai}\,\mathrm{d}V - \int_{\Gamma_t} \delta u_a\,\bar{t}_a \mathrm{d}A = 0 \end{aligned} \tag{1.21a}$$

$$\int_V \delta p_{,a}\,p^{n+1}_{,a}\,\mathrm{d}V + \int_V \delta p\,\frac{\rho}{\Delta t} u^\star_{a,a}\,\mathrm{d}V - \int_{\Gamma_q} \delta p\,n_a\,\bar{p}_{,a}\,\mathrm{d}A = 0 \tag{1.21b}$$

and

$$\begin{aligned} \int_V \delta u_a \left[\frac{\rho}{\Delta t}\left(u^{n+1}_a - u^\star_a\right) - g_a\right] \mathrm{d}V - \int_V \delta u_{a,a}\,p^{n+1}\,\mathrm{d}V \\ + \int_{\Gamma_p} \delta u_a\,n_a\,\bar{p}\,\mathrm{d}A = 0 \end{aligned} \tag{1.21c}$$

Introducing the finite element discretization, the solution of (1.21a) and (1.21c) are identical to (1.25) and (1.30), respectively. A discretization of (1.21b) yields

$$\delta\tilde{p}_\gamma \left[\int_V N^{(p)}_{\gamma,a} N^{(p)}_{\delta,a} \, \mathrm{d}V \, \tilde{p}^{\delta,n+1} + \frac{\rho}{\Delta t} \int_V N^{(p)}_\gamma N^{(u)}_{\beta,a} \, \mathrm{d}V \, \tilde{u}^{\star\beta}_a \right] = 0 \tag{1.22}$$

which (ignoring the gradient of g_a) yields the discrete equations

$$\mathbf{H}_{\gamma\delta} \, \tilde{\mathbf{p}}^{n+1}_\delta = -\frac{\rho}{\Delta t} \mathbf{C}_{\gamma\beta} \, \tilde{\mathbf{u}}^\star_\beta \tag{1.23}$$

1.2.3 Donea *et al.* solution

The Donea *et al* [6, 7] approach uses the Chorin split [5] to solve the transient problem. The solution starts with a predictor step ignoring both the momentum pressure and all boundary conditions except the prescribed velocity on Γ_1. Accordingly, the first of (1.14) is written as

$$\begin{gathered} \frac{1}{\Delta t} \mathbf{M}_{\alpha\beta} \left(\tilde{\mathbf{u}}^\star_\beta - \tilde{\mathbf{u}}^n_\beta \right) = -\mathbf{P}_\alpha(\mathbf{u}^n) - \mathbf{A}_\alpha(\mathbf{u}^n) \\ \tilde{\mathbf{u}}^\star_\beta = \bar{\mathbf{u}} \quad \text{on} \quad \Gamma_1 \end{gathered} \tag{1.24}$$

The solution is given by

$$\begin{aligned} \tilde{\mathbf{u}}^\star_\beta &= \tilde{\mathbf{u}}^n_\beta - \Delta t \, \mathbf{M}^{-1}_{\alpha\beta} \left[\mathbf{P}_\alpha(\mathbf{u}^n) + \mathbf{A}_\alpha(\mathbf{u}^n) \right] \\ \tilde{\mathbf{u}}^\star_\beta &= \bar{\mathbf{u}} \quad \text{on} \quad \Gamma_1 \end{aligned} \tag{1.25}$$

The corrector step uses the remaining momentum terms and needs to compute

$$\begin{gathered} \frac{1}{\Delta t} \mathbf{M}_{\alpha\beta} \left(\tilde{\mathbf{u}}^{n+1}_\beta - \tilde{\mathbf{u}}^\star_\beta \right) = \mathbf{f}_\alpha + \mathbf{C}_{\alpha\gamma} \tilde{p}^{n+1}_\gamma \\ \mathbf{C}^T_{\delta\beta} \tilde{\mathbf{u}}^{n+1}_\beta = \mathbf{E}^{n+1}_\delta \end{gathered} \tag{1.26a}$$

where the last equation ensures global conservation[2] with

$$\mathbf{E}^{n+1}_\delta = -\int_{\Gamma_1} N^{(p)}_\delta \, \mathbf{n}^T \bar{\mathbf{u}} \, \mathrm{d}A \tag{1.26b}$$

where $\mathbf{n}$ denotes the outward unit normal to the boundary. The solution requires knowledge of the pressure in order to obtain the final velocity. An equation for the pressure may be obtained by rewriting the first of (1.26a) as

$$\mathbf{C}_{\alpha\gamma} \tilde{p}^{n+1}_\gamma = \frac{1}{\Delta t} \mathbf{M}_{\alpha\beta} \left(\tilde{\mathbf{u}}^{n+1}_\beta - \tilde{\mathbf{u}}^\star_\beta \right) - \mathbf{f}_\alpha \tag{1.27}$$

[2] See [6] or [7] for additional comments on using the form of the last equation.

and then premultiplying by $\mathbf{C}^T_{\delta\beta}\,\mathbf{M}^{-1}_{\beta\alpha}$ to obtain

$$\mathbf{C}^T_{\delta\beta}\mathbf{M}^{-1}_{\beta\alpha}\mathbf{C}_{\alpha\gamma}\tilde{p}^{n+1}_{\gamma} = \mathbf{H}_{\delta\gamma}\tilde{p}^{n+1}_{\gamma} = \frac{1}{\Delta t}\mathbf{C}^T_{\delta\beta}\left(\tilde{\mathbf{u}}^{n+1}_{\beta} - \tilde{\mathbf{u}}^{\star}_{\beta}\right) - \mathbf{C}^T_{\delta\beta}\mathbf{M}^{-1}_{\beta\alpha}\mathbf{f}_{\alpha} \tag{1.28}$$

then using the second of (1.26a) one obtains

$$\mathbf{H}_{\delta\gamma}\tilde{p}^{n+1}_{\gamma} = \frac{1}{\Delta t}\left(\mathbf{E}^{n+1}_{\delta} - \mathbf{C}^T_{\delta\beta}\tilde{\mathbf{u}}^{\star}_{\beta}\right) - \mathbf{C}^T_{\delta\beta}\mathbf{M}^{-1}_{\beta\alpha}\mathbf{f}_{\alpha} \tag{1.29}$$

Once the pressure is known (1.26a) may be solved for the velocity as

$$\tilde{\mathbf{u}}^{n+1}_{\beta} = \tilde{\mathbf{u}}^{\star}_{\beta} + \Delta t\mathbf{M}^{-1}_{\beta\alpha}\left(\mathbf{f}_{\alpha} + \mathbf{C}_{\alpha\gamma}\tilde{p}^{n+1}_{\gamma}\right) \tag{1.30}$$

In summary, for each time increment first solve (1.25) followed by solution of (1.28) and finally (1.30).

1.2.4 Stabilization: Equal order interpolation

The basic equations may be interpolated for the velocity $\mathbf{u}$ and pressure p by the same interpolation, however, analysis shows that this form fails the usual stability and leads to severe oscillations in the solution variables. A simple approach is to *stabilize* the approximation for pressure. Several schemes have been proposed, however, a simple and effective method is to modify the functional by adding a term [8]

$$\Pi_p(p,\bar{p}) = \frac{\alpha}{\mu}\int_{\Omega}\left(p-\bar{p}\right)^2\,\mathrm{d}V \tag{1.31}$$

where $\bar{p}$ is a projection of the pressure p and in a finite element context the approximations are given by

$$\begin{aligned} p &= N_a\,\tilde{p}_a \\ \bar{p} &= \phi(\mathbf{x})_{\alpha}\,\hat{p}_{\alpha} \end{aligned} \tag{1.32}$$

The approximations used in ϕ_b are one order lower than those in N_a.

The implementation is carried out by taking a variation with respect to its two arguments. The variation yields

$$\delta\Pi_p = \frac{\alpha}{\mu}\int_{\Omega}\left(\delta p - \delta\bar{p}\right)\left(p-\bar{p}\right)\,\mathrm{d}V \tag{1.33}$$

The term for $\delta\bar{p}$ may be restricted to single elements and using (1.32) yields the linear equations

$$H_{\alpha\beta}\,\hat{p}_{\beta} = G_{\alpha a}\,\tilde{p}_a \tag{1.34}$$

where

$$H_{\alpha\beta} = \int_\Omega \phi_\alpha \, \phi_\beta \, \mathrm{d}V \quad \text{and} \quad G_{\alpha a} = \int_\Omega \phi_\alpha \, N_a \, \mathrm{d}V \tag{1.35}$$

This may be inserted into the term with the variation of p and yields the term to be appended to the main functional

$$\delta p_a \, \left(M_{ab} - G_{\alpha a} H^{-1}_{\alpha\beta} G_{\beta b} \right) \, p_b \tag{1.36}$$

where

$$M_{ab} = \int_\Omega N_a \, N_b \, \mathrm{d}V \tag{1.37}$$

is a consistent mass like term. This term is multiplied by the scaling factor α/μ and *subtracted* from the diagonal terms of pressure variables.

1.2.5 Taylor-Galerkin formulation

Governing equation for convection-diffusion

Starting from the conservation form of the equations

$$\frac{\partial \boldsymbol{\phi}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} + \frac{\partial \mathbf{G}_i}{\partial x_i} + \mathbf{Q} = \mathbf{0} \tag{1.38}$$

we consider a scalar case where [9]

$$\begin{array}{ll} \boldsymbol{\phi} \to \phi & \mathbf{Q} = Q(x_i, \phi) \\ \mathbf{F}_i \to F_i = U_i \phi & \mathbf{G}_i \to G_i = -k \dfrac{\partial \phi}{\partial x_i} \end{array} \tag{1.39}$$

which yields the scalar equation

$$\frac{\partial \phi}{\partial t} + \frac{\partial (U_i \phi)}{\partial x_i} - \frac{\partial}{\partial x_i} \left(k \frac{\partial \phi}{\partial x_i} \right) + Q = 0 \tag{1.40}$$

A time discretization will be assumed as

$$\frac{\partial \phi}{\partial t} \approx \frac{1}{\Delta t} \left(\phi^{n+1} - \phi^n \right) \tag{1.41}$$

Taylor-Galerkin: Scalar equation

A scalar form of the Taylor-Galerkin approach assumes

$$\phi^{n+1} = \phi^n + \Delta t \, \frac{\partial \phi^n}{\partial t} + \tfrac{1}{2} \, \Delta t^2 \, \frac{\partial^2 \phi^n}{\partial t^2} + O(\Delta t^3) \tag{1.42}$$

The first derivative is just the governing equation evaluated at t_n

$$\frac{\partial \phi^n}{\partial t} = \left[-\frac{\partial (U\phi)}{\partial x} + \frac{\partial}{\partial x}\left(k \frac{\partial \phi}{\partial x} \right) - Q \right]^n \tag{1.43}$$

The second derivative gives

$$\frac{\partial^2 \phi^n}{\partial t^2} = \frac{\partial}{\partial t} \left[-\frac{\partial (U\phi)}{\partial x} + \frac{\partial}{\partial x}\left(k \frac{\partial \phi}{\partial x} \right) - Q \right]^n \tag{1.44}$$

If we approximate the solution by C^0 approximations all third derivatives will be discarded and, thus,

$$\frac{\partial^2 \phi^n}{\partial t^2} \approx \frac{\partial}{\partial t} \left[-\frac{\partial (U\phi)}{\partial x} - Q \right]^n \tag{1.45}$$

Assuming now that between t_n and t_{n+1}, U has the constant value U^n, dropping the time derivative of loading during the time increment and interchanging the order of differentiation an approximation to the second derivative becomes

$$\frac{\partial}{\partial t}\left(\frac{\partial (U\phi)}{\partial x} \right)^n = \frac{\partial}{\partial x}\left(\frac{\partial (U\phi)}{\partial t} \right)^n \approx U^n \frac{\partial}{\partial x}\left(\frac{\partial \phi}{\partial t} \right)^n \tag{1.46}$$

and, thus,

$$\frac{\partial^2 \phi^n}{\partial t^t} \approx -U^n \frac{\partial}{\partial x}\left(\frac{\partial \phi^n}{\partial t} \right) \tag{1.47}$$

which again allows direct substitution from the governing differential equation. Collecting all the terms then gives

$$\begin{aligned} \phi^{n+1} - \phi^n = & -\Delta t \left[\frac{\partial (U\phi)}{\partial x} - \frac{\partial}{\partial x}\left(k \frac{\partial \phi}{\partial x} \right) + Q \right]^n \\ & + \tfrac{1}{2}\, \Delta t^2\, U^n \frac{\partial}{\partial x} \left[\frac{\partial (U\phi)}{\partial x} - \frac{\partial}{\partial x}\left(k \frac{\partial \phi}{\partial x} \right) + Q \right]^n \end{aligned} \tag{1.48}$$

Again, with C^0 approximations the third derivative on the diffusion term will be dropped leading to a stabilized form on the convection and loading terms only. A multi-dimensional form of (1.48) is given by

$$\begin{aligned} \phi^{n+1} - \phi^n = & -\Delta t \left[\frac{\partial (U_i\phi)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(k \frac{\partial \phi}{\partial x_i} \right) + Q \right]^n \\ & + \tfrac{1}{2}\, \Delta t^2\, U_j^n \frac{\partial}{\partial x_j} \left[\frac{\partial (U_i\phi)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(k \frac{\partial \phi}{\partial x_i} \right) + Q \right]^n \end{aligned} \tag{1.49}$$

Comparing to eqution (2.105) in Zienkiewicz *et al.* [9] we observe that this is identical to the result for the CBS algorithm. Thus, starting from a conservation form the two are identical.

Taylor-Galerkin: Navier-Stokes equation

The Taylor-Galerkin (CBS) form of the Navier-Stokes equations for incompressible flow may be recovered by the followin changes:

$$\begin{aligned} \phi &\to \rho u_a \\ U_i &\to u_i \\ k\phi_{,i} &\to \tau_{ia} \\ Q &\to g_a \end{aligned} \tag{1.50}$$

and adding an approximation for the pressure gradient at time t_{n+1}, (1.49) becomes

$$\begin{aligned} \rho u_a^{n+1} - \rho u_a^n = -\Delta t \left[\frac{\partial(\rho\, u_a\, u_i)}{\partial x_i} - \frac{\partial \tau_{ia}}{\partial x_i} + g_a \right]^n - \Delta t\, \frac{\partial p^{n+1}}{\partial x_a} \\ + \tfrac{1}{2}\, \Delta t^2\, u_j^n \frac{\partial}{\partial x_j} \left[\frac{\partial(\rho\, u_a\, u_i)}{\partial x_i} - \frac{\partial \tau_{ia}}{\partial x_i} + g_a \right]^n + \tfrac{1}{2} \Delta t^2\, u_j\, \frac{\partial^2 p^{n+1}}{\partial x_j\, \partial x_a} \end{aligned} \tag{1.51}$$

Applying the Chorin split to (1.51) yields the pair of equations

$$\begin{aligned} \rho u_a^\star - \rho u_a^n &= \Delta t \left[\frac{\partial(\rho\, u_a\, u_i)}{\partial x_i} - \frac{\partial \tau_{ia}}{\partial x_i} + g_a \right]^n \\ &\quad + \tfrac{1}{2}\, \Delta t^2\, u_j^n \frac{\partial}{\partial x_j} \left[\frac{\partial(\rho\, u_a\, u_i)}{\partial x_i} + g_a \right]^n \\ \rho u_a^{n+1} - \rho u_a^\star &= -\Delta t\, \frac{\partial p^{n+1}}{\partial x_a} + \tfrac{1}{2} \Delta t^2\, u_j\, \frac{\partial^2 p^{n+1}}{\partial x_j\, \partial x_a} \end{aligned} \tag{1.52}$$

The remaining steps, using the continuity equation, are identical to the Chorin implementation. Thus, discretizing the Taylor-Galerkin requires adding the following weak terms. For the first step:

$$\begin{aligned} \delta \Pi_1 &= \tfrac{1}{2} \Delta t^2\, \rho \int_V \delta u_a\, u_j^n \left\{ \frac{\partial}{\partial x_j} \left[\frac{\partial(u_a\, u_i)}{\partial x_i} + g_a \right]^n \right\} \mathrm{d}V \\ &= -\tfrac{1}{2} \Delta t^2\, \rho \int_V \frac{\partial(\delta u_a\, u_j^n)}{\partial x_j} \left[\frac{\partial(u_a\, u_i)}{\partial x_i} + g_a \right]^n \mathrm{d}V \end{aligned} \tag{1.53a}$$

and for the third step

$$\begin{aligned} \delta \Pi_2 &= \tfrac{1}{2} \Delta t^2 \int_V \delta u_a\, u_j^n \left\{ \frac{\partial}{\partial x_j} \left[\frac{\partial p}{\partial x_a} \right]^{n+1} \right\} \mathrm{d}V \\ &= -\tfrac{1}{2} \Delta t^2 \int_V \frac{\partial(\delta u_a\, u_j^n)}{\partial x_j} \left[\frac{\partial p}{\partial x_a} \right]^{n+1} \mathrm{d}V \end{aligned} \tag{1.53b}$$

where constant ρ is assumed due to incompressibility. Descretizing the various terms proceeds as:

$$\begin{aligned} u_j^n &= N_\alpha \, \tilde{u}_j^{\alpha,n} \\ \frac{\partial p}{\partial x_a} &= \frac{\partial N_\alpha}{\partial x_a} \, \delta\tilde{p}^\alpha = r_a^{(p)} \\ \frac{\partial \delta u_a}{\partial x_j} &= \frac{\partial N_\alpha}{\partial x_j} \, \delta\tilde{u}_a^\alpha \\ \frac{\partial (u_a u_i)}{\partial x_i} &= \left(u_a \, \frac{\partial u_i}{\partial x_i} + u_i \, \frac{\partial u_a}{\partial x_i} \right) = r_a^{(u)} \end{aligned} \tag{1.54}$$

Inserting the approximations into (1.53a) yields

$$\delta\Pi_1 = -\tfrac{1}{2} \, \Delta t^2 \, \rho \, \delta\tilde{u}_a^\alpha \int_V \left[N_\alpha \, \frac{\partial u_j^n}{\partial x_j} + \frac{\partial N_\alpha}{\partial x_j} \, u_j^n \right] \, r_a^{(u)} \, \mathrm{d}V \tag{1.55a}$$

and for (1.53b)

$$\delta\Pi_2 = -\tfrac{1}{2} \, \Delta t^2 \, \rho \, \delta\tilde{u}_a^\alpha \int_V \left[N_\alpha \, \frac{\partial u_j^n}{\partial x_j} + \frac{\partial N_\alpha}{\partial x_j} \, u_j^n \right] \, r_a^{(p)} \, \mathrm{d}V \tag{1.55b}$$

Chapter 2

Material set data input

The material set data for the solution of CFD problems for the Navier-Stokes theory requires specification of the element formulation, the material density and the fluid constitution. Currently the constitution is restricted to constant viscosity. The library of fluid elements includes several formulations and, in two-dimensions, a computation of stream lines.

2.1 Fluid property data

The basic description for a CFD analysis is controlled by the material set data records. These are given by:

```
MATErial ma
  FLUId
    NEWTonian VISCosity mu
    DENSity   MASS      rho
    TYPE     <VELOcity,HOOD,BOCHev,COURant,DONEa,STREam>
    INCOMpressible ,,   npart
      ! Blank termination record
```

where `mu` is the viscosity, `rho` the mass density, and `npart` is the partition to apply the incompressible continutity constraint. The `TYPE` record controls the specific element formulation that is used in the analysis, as well as, activating the computation of streamlines for 2-d analyses.

2.1.1 Two-dimensional element types

The basic element types for two-dimensional analyses have either quadrilateral or triangular shape and are shown in Figures 2.1 and 2.2.

Type: Velocity analysis

The solution of problems using the `TYPE=VELOcity` formulation uses element types $Q4P1$, $Q9P3$, $T6P1$ or $T7P3$ as shown in Figures 2.1 and 2.2. The element approximation for the u_1, u_2 velocity are nodal iso-parametric interpolations. The element approximation for the pressure are polynomials 1, x_1, x_2 (lowest order elements use only the constant value). Thus, for the incompressible behavior of the continuity equation the pressure degrees of freedom are element Lagrange multipliers. No other type of elements may be used with this formulation.

Type: Taylor-Hood analysis

The solution of problems using the `TYPE=HOOD` formulation uses element types $Q9Q4$, $T6T3$ or $T7T3$ type elements. Thus, the velocity interpolation is a nodal isoparametric quadratic approximation while the pressure is a linear nodal (sub)parametric approximation.

Type: Dohrmann-Bochev stabilized analysis

The solution of problems using the `TYPE=BOCHev` formulation uses isoparametric equal-order nodal interpolations for both the pressure and the velocity. Thus elements of type $T3T3$, $T6T6$, $Q4Q4$ or $Q9Q9$ may be used. The interpolations for both velocity and pressure are isoparametric.

Type: Courant analysis

The solution of problems using the `TYPE=COURant` formulation use the same element types as for the Taylor-Hood formulation. In addition a special solution algorithm which uses the *partition* options of *FEAP* must be employed. The algorithm is described further in 3.

Type: Donea `et al.` analysis

The solution of problems using the `DONEa` *et al.* formulation use only element type $Q4P1$. In addition a special solution algorithm which uses the *partition* options of *FEAP* must be employed. The algorithm is a little different than the Courant type due to the special need to form the pressure matrix and is described further in 3.

As implemented in *FEAP* it is necessary to assign a nodal pressure degree of freedom in order to use partitions. However, all nodal values must have a fixed boundary condition so that the pressure solution is correct.

2.1.2 Streamline data

For two-dimensional problems the results for the streamlines may be computed as an extra degree of freedom to the problem. The analysis is activated by including a material data set

```
MATErial ms
  FLUId
      ....
    TYPE STREam  st       ! Uses nodal degree of freedom 'st'
      ! Termination record
```

The `TYPE` command for streamlines is given in addition to that describing the element formulation to use.

The stream function, ψ, defines the velocities as[9]

$$u_1 = -\frac{\partial \psi}{\partial x_2} \quad \text{and} \quad u_2 = \frac{\partial \psi}{\partial x_1} \tag{2.1}$$

and thus satisfies the continuity condition $u_{i,i}$. A Poisson equation governing the stream function may be deduced as

$$\frac{\partial}{\partial x_1}\left(\frac{\partial \psi}{\partial x_1} - u_2\right) + \frac{\partial}{\partial x_2}\left(\frac{\partial \psi}{\partial x_2} + u_1\right) = 0 \tag{2.2}$$

This has a weak form

$$\int_V \left[\frac{\partial \delta\psi}{\partial x_1}\left(\frac{\partial \psi}{\partial x_1} - u_2\right) + \frac{\partial \delta\psi}{\partial x_2}\left(\frac{\partial \psi}{\partial x_2} + u_1\right)\right] \mathrm{d}V = 0 \tag{2.3}$$

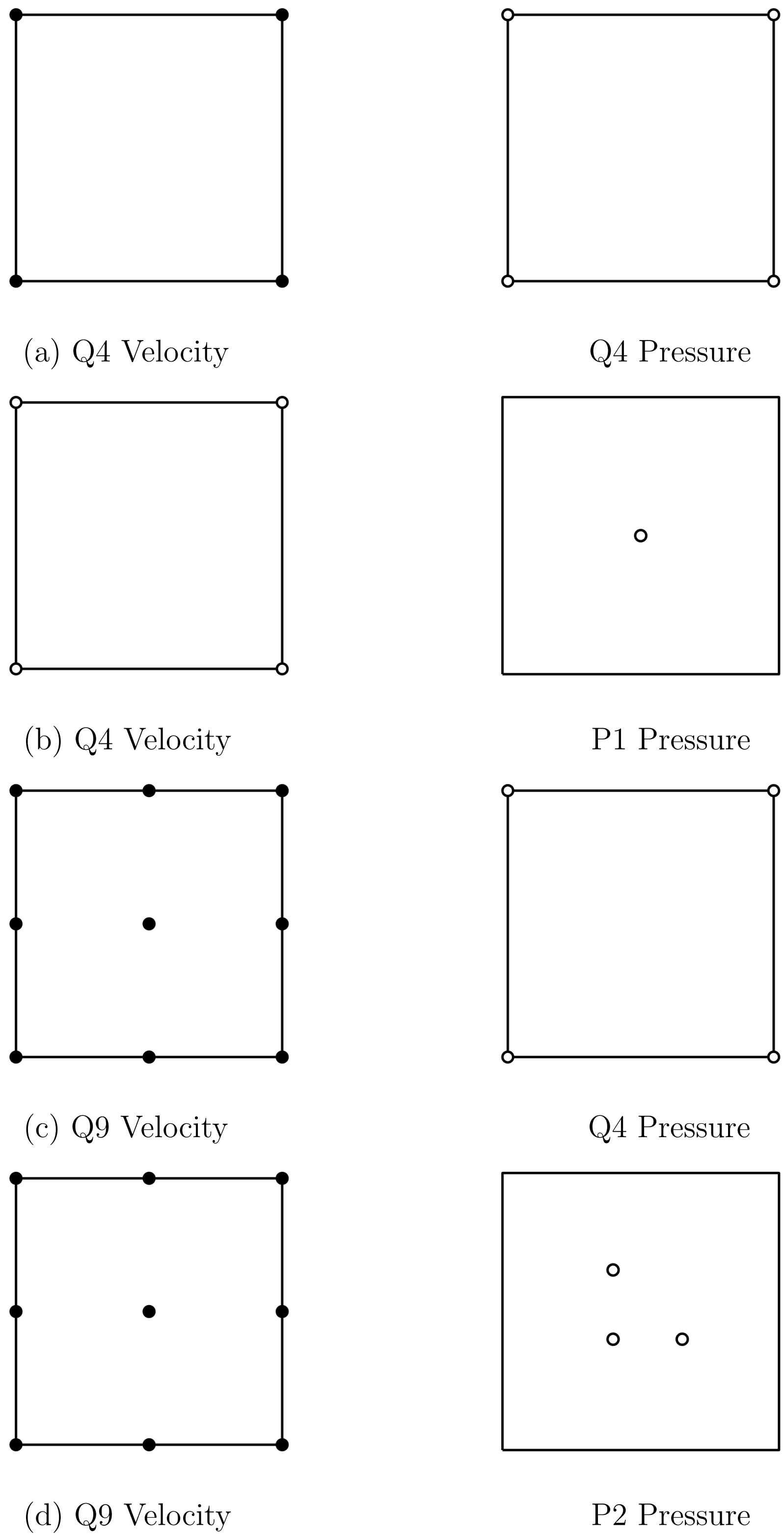
which may be discretized by

$$\psi = N_\alpha \, \tilde{\psi}^\alpha \tag{2.4}$$

and solved as a post-processing step in the analysis once the velocities are known.

2.1.3 Three-dimensional element types

The formulations for the `VELOcity` and `BOCHev` stabilized forms have been implemented for the appropriate `QnPm` and `QnQn` forms, respectively. Element node numbering is given in the *FEAP* User Manual [1].



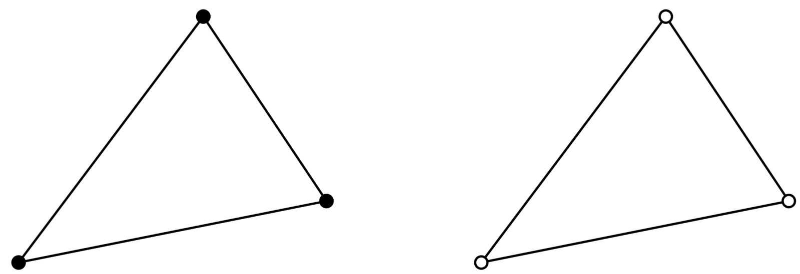
(a) Q4 Velocity — Q4 Pressure

(b) Q4 Velocity — P1 Pressure

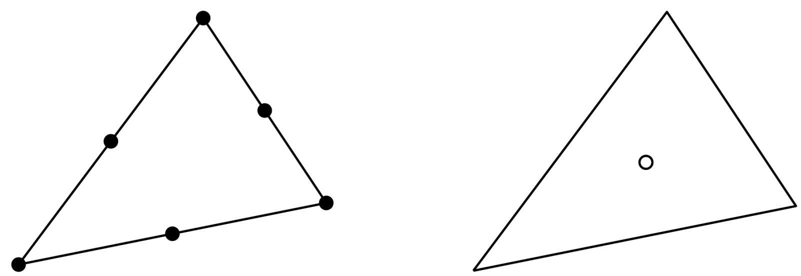
(c) Q9 Velocity — Q4 Pressure

(d) Q9 Velocity — P2 Pressure

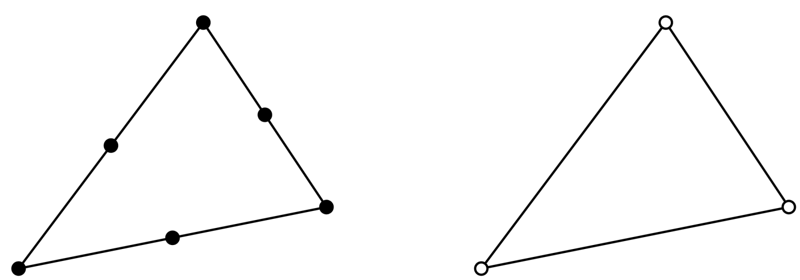
Figure 2.1: Quadrilateral 2-d fluid element velocity and pressure.



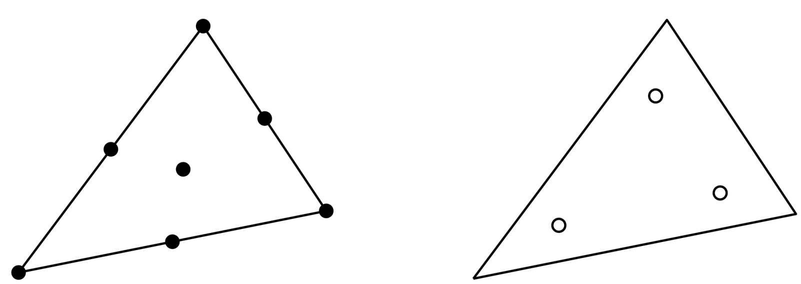
(a) T3 Velocity T3 Pressure



(b) T6 Velocity P1 Pressure



(c) T6 Velocity T3 Pressure



(d) T7 Velocity P2 Pressure

Figure 2.2: Triangular 2-d fluid element velocity and pressure.

Chapter 3

Split algorithm solutions

The Chorin split and Donea *et al.* formulations to solve CFD problems both use a transient solution scheme and a three step solution process based on the original work in Reference [5].

3.1 Chorin split solution statements

For the Chorin split the five degrees of freedom are u_1, u_2, p, $u_1^\star$ and $u_2^\star$; they are assigned in this order in *FEAP*. The split is first defined by the partitioning with the $\mathbf{u}^\star$ variables computed in the first partition, the pressure p in the second and finally the velocities $\mathbf{u}$ in the third. In addition a sixth degree of freedom will be used to compute the stream lines in a fourth partition. Accordingly, the partition data is given in Table 3.1

```
PARTition
  0 0 0 1 1 0  ! u-star
  0 0 1 0 0 0  ! pressure
  1 1 0 0 0 0  ! u
  0 0 0 0 0 1  ! Streamlines
```

Table 3.1: Partition data for 2-d split solutions.

Since this is a transient solution it necessary to compute a mass matrix for the first and third partition, requiring the data which is best given in batch mode as shown in Table 3.2.

```
BATCh
  PARTition,,1
    MASS LUMP    ! Form a diagonal mass
  PARTition,,2
    TANG         ! Form and factor pressure tangent matrix
  PARTition,,3
    MASS LUMP    ! Form a diagonal mass
  PARTition,,4
    TANG         ! Form and factor streamline tangent matrix
END
```

Table 3.2: Matrix forms for 2-d Chorin split solutions.

The remainder of the algorithm describes the transient solution process and is shown in Table 3.3. [1] Outputs and plots may be added before the last NEXT statement.

3.1.1 Donea *et al.* solution statements

For the Donea *et al.* split the five degrees of freedom are also u_1, u_2, p, $u_1^\star$ and $u_2^\star$ and are assigned in this order in *FEAP*. The split partitioning is defined in Table 3.1 with the $\mathbf{u}^\star$ variables computed in the first partition, the pressure p in the second and the velocities $\mathbf{u}$ in the third. In addition a sixth degree of freedom is used to compute the stream lines in a fourth partition. The first part of the solution is identical to Table 3.2 except the partition 2 command

```
TANG
```

is replaced by

```
SPLIt INIT  ! Form and factor Donea pressure tangent matrix
```

The remaining solution steps also are identical to the Chorin split in Table 3.3 except for the pressure solution in partition 2 where

```
SOLVE
```

is replaced by

```
SPLIt STEP 2
```

[1]The use of the LOOP-NEXT pairs on solution steps forces *FEAP* to make a solution even if the residual is zero.

```
BATCH
  DT,,dt ! where dt is a time step satisfying CFL condition
  LOOP,INFInite
    LOOP,,20          ! Interval between plots
      TIME,,t           ! "t" specified time to stop solution
      PART,,1
        LOOP,,1
          FORM          ! Form partition 1 residual
          SPLIt STEP 1  ! Solves for u-star
        NEXT
      PART,,2
        LOOP,,1
          FORM          ! Form partition 2 residual
          SOLVe         ! Solve equations for pressure
        NEXT
      PART,,3
        LOOP,,1
          FORM          ! Form partition 3 residual
          SPLIt STEP 3  ! Solves for u
        NEXT
    NEXT              ! End of time step loops in interval
    PART,,4           ! Solve for Streamlines
      LOOP,,1
        FORM          ! Form partition 2 residual
        SOLVe         ! Solve equations for streamlines
      NEXT
       ....           ! Outputs and plots
  NEXT ! time intervalt
END ! Batch
```

Table 3.3: Solution steps for split solutions.

Appendix A

Driven Cavity Input files

As a simple example we consider the driven cavity problem. The domain is a unit square as shown in Figure A.1 and is subjected to a uniform tangential velocity u_1 along the entire top. The properties for the analysis are $\nu = 0.01$, $\rho = 1$ and $u_1 = 1$ which, at steady state yields a Reynolds number of 100. The problem may be analyzed for each of the formulations described above using the input files listed in the following sections. Each problem uses a uniform mesh of square elements.

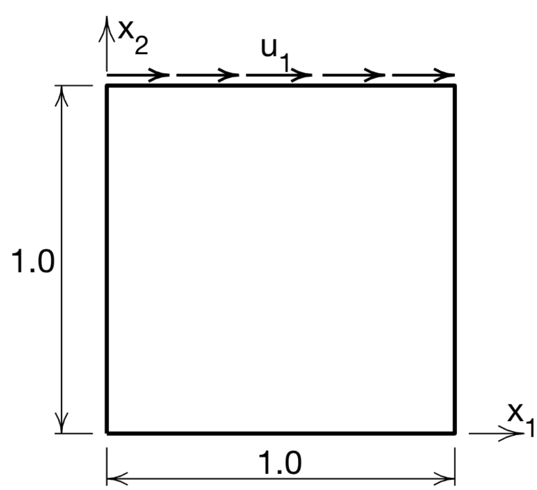


Figure A.1: Driven cavity problem geometry.

A.1 Velocity formulation

```
feap * * Driven cavity 2-d form: Qn/Pm type formulation
  0 0 0 2 3 9

 mate 1
   fluid    ! Uses dofs 1-2 for velocity; 3 for pressure
     newtonian viscosity 0.01
     type velocity 1
     type stream   3
     density mass  1.0

noprint
param
 n = 100

block
  cart n  n
    quad  9
    1  0 0
    2  1 0
    3  1 1
    4  0 1

ebou
  1  0 1 1   1
  2  0 1 1   1
  1  1 1 1   1
  2  1 1 1   1

cbou set
  node 0.5 0 1 0 1

csurf
  displ  1
    line
      1 1 1 1.0
      2 0 1 1.0

end


partition
 1 1 0
 0 0 1

batch
  noprint
  prop
  dt,,1
```

```
  loop,,1
    time
    partition,,1
      loop,,10
        utang,,1
      next
    partition,,2
      loop,,10
        tang,,1
      next
    plot frame 1
    plot cont 1 0 1
    plot frame 2
    plot cont 2 0 1
    plot frame 3
    plot rang -0.5 0.5
    plot stre 7 0 1     ! Pressure plot
    plot rang 0 0
    plot frame 4
    plot cont 3  0 1
  next
  disp coor 1 0.5
  disp coor 2 0.5
end

inter
stop
```

A.2 Taylor-Hood formulation

```
feap * * Driven cavity 2-d form: Taylor-Hood type formulation
  0 0 0 2 4 9

parameter
  st = 4    ! DOF for streamlines

 mate 1
   fluid    ! Uses dofs 1-2 for velocity; 3 for pressure
     newtonian viscosity 0.01
     density mass  1.0
     type    hood
     type    stream st

noprint
param
 n = 100

block
  cart n  n
    quad  9
    1  0 0
    2  1 0
    3  1 1
    4  0 1

ebou
  1  0 1 1   0 1
  2  0 1 1   0 1
  1  1 1 1   0 1
  2  1 1 1   0 1


cbou set
   node 0.5 0 1 0 1 1

csurf
  displ  1 1.0
    line
      1 1 1 1.0
      2 0 1 1.0

end


partition
 1 1 1 0
 0 0 0 1
```

```
batch
  noprint
  prop
  dt,,1
  loop,,1
    time
    partition,,1
      loop,,10
        utang,,1
      next
    partition,,2
      loop,,10
        tang,,1
      next
    plot frame 1
    plot cont 1 0 1
    plot frame 2
    plot cont 2 0 1
    plot frame 3
    plot rang -0.5 0.5
    plot stre 7 0 1     ! Pressure plot
    plot rang 0 0
    plot frame 4
    plot cont st 0 1
  next
  disp coor 1 0.5
  disp coor 2 0.5
end

inter
stop
```

A.3 Dohrmann-Bochev formulation

```
feap * * Driven cavity 2-d form: Dohrmann-Bochev stabilized
  0 0 0 2 4 4

 mate 1
   fluid    ! Uses dofs 1-2 for velocity; 3 for pressure
     newtonian viscosity 0.01
     type bochev 1
     type stream 4
     density mass  1.0
     penalty,,2   ! Stabilizing value

noprint
param
 n = 100

block
  cart n  n
    quad  4
    1  0 0
    2  1 0
    3  1 1
    4  0 1

ebou
  1  0   1 1   0 1
  2  0   1 1   0 1
  1  1   1 1   0 1
  2  1   1 1   0 1

cbou set
   node 0.5 0.0   1 0   1 1

csurf
  displ  1 1.0  ! Top velocity
    line
      1 1 1 1.0
      2 0 1 1.0

end mesh

partition
 1 1 1 0    ! Solve for velocity/pressure
 0 0 0 1    ! Streamline solution

 batch
  noprint
  prop
  dt,,1
```

```
  loop,,1
    time
    partition,,1
      loop,,10
        utang,,1
      next
    partition,,2
      loop,,10
        tang,,1
      next
    plot frame 1
    plot cont 1 0 1
    plot frame 2
    plot cont 2 0 1
    plot frame 3
    plot rang -0.5 0.5
    plot cont 3 0 1     ! Pressure plot
    plot rang 0 0
    plot frame 4
    plot cont 4 0 1
  next
end

inter
stop
```

A.4 Chorin split formulation

```
param
  q = 9

feap * * Driven cavity 2-d Chorin Split form
  0 0 0 2 6 q

noprint
parameter
  st = 6  ! Streamline dof

 mate 1
   fluid
     newtonian viscosity 0.01
     type      chorin    2     !  ! Applied to partition 2 of split
     type      stream    st
     density   mass      1.0

param
 n = 40
 m = n + 1
 m1 = n*m + 1
 m2 = m*m
 nh = n/2


block
  cart n  n
    quad  q
    1  0 0
    2  1 0
    3  1 1
    4  0 1

! Normal velocity boundaries are fixed
ebou
  1  0  1 0   0  1 1   1
  1  1  1 0   0  1 1   1
  2  0  0 1   0  1 1   1
  2  1  0 1   0  1 1   1

cbou  ! To release normal nodal velocity at center bottom
  node 0.5 0.0   1 0 1 1 0 1

! set tangential velocity (leaky)
 disp
  m1 1 1.0 0.0 0.0 1.0 0.0
  m2 0 1.0 0.0 0.0 1.0 0.0
```

```
end mesh

partition     ! Split algorithm for u, u-star & p
 0 0 0 1 1 0  ! u-star
 0 0 1 0 0 0  ! pressure
 1 1 0 0 0 0  ! u
 0 0 0 0 0 1  ! Streamlines

batch
  tplot,,50
end
disp nh+1 2
show

batch
  print off
  noprint log
  dt,,0.005        ! n = 40
! u_star
  part,,1
    mass lump
! Pressure
  part,,2
    tang
! u
  part,,3
    mass lump
! Streamline
  part,,4
    tang
  loop,,500
    loop,,50     ! Plot every 50 time increments
      time,,5   ! Stop at time = 5
      part,,1
        loop,,1
          form
          split step 1
        next
      part,,2
        loop,,1
          form
          solv
        next
      part,,3
        loop,,1
          form
          split step 3
        next
    next
    plot fram 1
    plot cont 1 0 1
```

```
    plot fram 2
    plot cont 2 0 1
    plot fram 3
    plot cont 4 0 1
    plot fram 4
    plot cont 5 0 1
  next

! Output solution at mid levels
! disp coor 1 0.5
! disp coor 2 0.5
end


inter
!  Streamline solution
batch
  part,,4
  loop,,1
    tang,,1
  next
  plot wipe
  plot frame
  plot cont 6 0 1
end

inter
stop
```

A.5 Donea *et al.* split formulation

```
feap * * Driven cavity 2-d Donea Split form
  0 0 0 2 6 4

noprint
parameter
  st = 6  ! Streamline dof

 mate 1
   fluid
     newtonian viscosity 0.01
     type donea  2              ! Applied to partition 2 of split
     type stream st
     density   mass      1.0

param
 n = 40
 m = n + 1
 m1 = n*m + 1
 m2 = m*m
 nh = n/2


block
  cart n  n
    quad  4
    1  0 0
    2  1 0
    3  1 1
    4  0 1

! Element pressure boundary condition
lbou
  m/2 1

boun      ! All nodal pressures are out
  1 1 0 0 -1 0 0
 m2 0 0 0  1 0 0

! Normal velocity boundaries are fixed
ebou
  1  0  1 0   1  1 1   1
  1  1  1 0   1  1 1   1
  2  0  0 1   1  1 1   1
  2  1  0 1   1  1 1   1

cbou ! To release nodal velocity at center bottom
  node 0.5 0.0   1 0 1 1 0 1
```

```
! set tangential velocity (leaky)
disp
  m1 1 1.0 0.0 0.0 1.0 0.0
  m2 0 1.0 0.0 0.0 1.0 0.0

end mesh

partition    ! Split algorithm for u, u-star & p
 0 0 0 1 1 0  ! u-star
 0 0 1 0 0 0  ! pressure
 1 1 0 0 0 0  ! u
 0 0 0 0 0 1  ! Streamlines

batch
  print off
  noprint log
  dt,,0.005       ! n = 40
! u_star
  part,,1
    mass lump
! Pressure matrix
  part,,2
    split init
! u
  part,,3
    mass lump
  loop,,500
    loop,,50     ! Plot every 50 time increments
      time,,5    ! Stop at time = 5
      part,,1
        loop,,1
          form
          split step 1
        next
      part,,2
        loop,,1
          form
          split step 2
        next
      part,,3
        loop,,1
          form
          split step 3
        next
    next
    plot fram 1
    plot cont 1 0 1
    plot fram 2
    plot cont 2 0 1
    plot fram 3
    plot cont 4 0 1
```

```
    plot fram 4
    plot cont 5 0 1
  next

! Output solution at mid levels
! disp coor 1 0.5
! disp coor 2 0.5
end

inter
!  Streamline solution
batch
  part,,4
  loop,,1
    tang,,1
  next
  plot wipe
  plot frame
  plot cont 6 0 1
end

inter
stop
```

Bibliography

[1] R.L. Taylor and S. Govindjee. *FEAP - A Finite Element Analysis Program, User Manual*. University of California, Berkeley. http://projects.ce.berkeley.edu/feap.

[2] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier, Oxford, 7^{th} edition, 2013.

[3] O.C. Zienkiewicz, R.L. Taylor, and D. Fox. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier, Oxford, 7^{th} edition, 2013.

[4] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1:73–100, 1973.

[5] A.J. Chorin. A numerical method for solving incompressible viscous problems. *Journal of Computational Physics*, 2:12–26, 1967.

[6] J. Donea, S. Giuliani, H. Laval, and L. Quartapelle. Finite element solution of unsteady Navier-Stokes equations by a fractional step method. *Computer Methods in Applied Mechanics and Engineering*, 33:53–73, 1982.

[7] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, Chichester, 2003. https://onlinelibrary.wiley.com/doi/book/10.1002/0470013826.

[8] C.R. Dohrmann and P.B. Bochev. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *International Journal for Numerical Methods in Fluids*, 46:183–201, 2004.

[9] O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics*. Elsevier, Oxford, 7^{th} edition, 2014.

Index