
Coupled Fluid Flow and Heat Transfer

5.1 Introduction

The general problem area of convective heat transfer is a particularly challenging one since it represents a situation involving inherently coupled problems, i.e., problems involving multiple physical phenomena. Indeed, the subject itself represents a joining of two classical areas of applied mechanics, namely, fluid mechanics and heat transfer.

Convection problems are generally divided into two major categories depending primarily on the forces that are responsible for the fluid motion. In *forced convection*, the fluid motion is due to the application of pressure or viscous forces on the fluid boundary. *Free* or *natural convection* problems are characterized by the fluid motion which is produced by temperature-induced buoyancy forces. This distinction between the types of convection is not always possible because the two types of driving forces (surface and volume) may appear in varying degrees and combinations.

Another categorization of convection problems may be defined based on the compressibility of the fluid. The most widely studied model for convection utilizes the Boussinesq approximation which assumes an incompressible flow and limits density variations to be small and isolated to the body force term. In a strict Boussinesq model, thermophysical properties are treated as constant, work done by pressure and viscous dissipation are neglected, and the density in the body force term is a linear function of the temperature. The Boussinesq model is computationally convenient because the incompressible viscous flow equations only require minimal alteration for the inclusion of nonisothermal effects. However, the Boussinesq approximation is limited to flows with relatively small temperature differences as outlined in Section 1.7, and this limitation has forced consideration of other low-speed convection models.

Low-speed compressible flows or non-Boussinesq models form another category of convection problems. The standard model in this regime involves the so-called acoustically filtered equations which admit large density (temperature) variations but suppress the propagation of acoustic waves. The resolution of small amplitude (acoustic) pressure disturbances is computationally troublesome and one of the major difficulties in using fully compressible flow formulations at the low-speed or low Mach number limit. The acoustically filtered model may be derived by any of several formal methods [1–3] and these were outlined in Section 1.7. A comparison and comments on the various non-Boussinesq models can also be found in [3,4]. Unfortunately, the finite element methods for this model are not quite the immediate extension of the incompressible methods from the previous chapter.

Regardless of the categorization, the convective heat transfer process depends directly on the motion of a fluid medium. The Navier–Stokes equations from Chapter 4 (or their extension to the compressible case) as well as the advection-diffusion equation that describes thermal energy transport must be considered in the development of a computational scheme. These equations consist of a set of coupled partial differential equations in terms of the velocity field, pressure, and temperature.

In this chapter, finite element models based on the weak formulation of the governing equations of viscous incompressible fluids in the presence of buoyancy forces and inertial effects are developed. Finite element models for nonisothermal porous flow and low-speed compressible flows are also formulated. The finite element model development for this coupled case is completely analogous to the developments for heat conduction in Chapter 3 and viscous incompressible flows in Chapter 4. Therefore, detailed discussions of the actual model development will be less extensive in the present chapter.

5.2 Nonisothermal Incompressible Flows

5.2.1 Governing Equations

The laws describing the nonisothermal flow of an incompressible Newtonian fluid were presented in Chapter 1 and are summarized here for ready reference. The equations are written for a fluid region Ω_f using a Cartesian coordinate system x_i in a Eulerian reference frame, with the index i taking the values $i = 1, 2, 3$ for the three-dimensional case and $i = 1, 2$ for the two-dimensional case; the usual summation convention on repeated indices is used [see Eqs. (1.5.7)–(1.5.9)]; time is denoted by t . An extended form of the Boussinesq approximation is used, which allows the fluid properties to be functions of the thermodynamic state (e.g., pressure P and temperature T) and the density ρ to vary with temperature T according to the relation

$$\rho = \rho_0[1 - \beta(T - T_0)] \quad (5.2.1)$$

where β is the coefficient of thermal expansion and the subscript zero indicates a reference condition. The variation of density as given in (5.2.1) is permitted only in the description of the body force; the density in all other situations is assumed to be that of the reference state, ρ_0 . The governing equations of convective heat transfer are summarized below.

Conservation of Mass

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (5.2.2)$$

Conservation of Momentum

$$\rho_0 \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left[-P\delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \rho_0 g_i \beta(T - T_0) = 0 \quad (5.2.3)$$

Conservation of Energy

$$\rho_0 C_v \left(\frac{\partial T}{\partial t} + v_j \frac{\partial T}{\partial x_j} \right) - \frac{\partial}{\partial x_i} \left(k_{ij} \frac{\partial T}{\partial x_j} \right) - Q - \Phi = 0 \quad (5.2.4)$$

where v_i denote the velocity components, P the pressure, ρ_0 the density, g_i the gravitational force components, T the temperature, C_v the specific heat of the fluid at constant volume, Q the rate of internal heat generation, μ the shear viscosity of the fluid, k_{ij} the components of the thermal conductivity tensor, and Φ is the viscous dissipation in the fluid

$$\Phi = 2\mu D_{ij}D_{ij}; \quad D_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (5.2.5)$$

Recall that Eq. (5.2.2) is also known as the continuity equation or divergence-free condition on the velocity field and relations in Eq. (5.2.3) are known as the Navier–Stokes equations. The equations in (5.2.3) and (5.2.4) are written in advective form.

The advective form of the Boussinesq equations is the standard form for finite element computation. Other discretization methods (particularly finite difference and finite volume methods) have advocated the use of the conservation forms of the equations that conserve various quantities such as kinetic energy, enstrophy, temperature, and temperature squared. It has been observed that the conservation form minimizes aliasing errors and improves temporal stability for these computational schemes. Global conservation of these types of quantities is often useful in proving boundedness and stability of the continuum equations when such proofs are available. Though the theory is incomplete for the continuum, it is often assumed that if the discrete method shares the same conservation properties as the continuum, the discrete solution will behave properly and remain a good approximation to the partial differential equations. The need for the conservation form in a finite element method is uncertain. Gresho, et al. [5] and Cliffe [6] conducted finite element studies of the inviscid Boussinesq equations and determined what forms conserved which quantities and produced the more stable forms. Note, however, that there are limits on what combinations of linear and quadratic quantities may be conserved within any given scheme; discrete systems are limited to conservation of quadratic quantities. Also, because physical dissipation is normally included in convection problems, the need for the conservation form remains unclear. Here a general form of the advection operator (introduced in Section 1.4.7) is used to allow an easy transition from advection to conservation to an “absolute” conservation form. Using a form of (1.4.31) for the incompressible, Boussinesq equations, the balance equations can be written as

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (5.2.6)$$

$$\rho_0 \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} + \alpha_v v_i \frac{\partial v_j}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left[-P \delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \rho_0 g_i \beta (T - T_0) = 0 \quad (5.2.7)$$

$$\rho_0 C_v \left(\frac{\partial T}{\partial t} + v_j \frac{\partial T}{\partial x_j} + \alpha_T T \frac{\partial v_j}{\partial x_j} \right) - \frac{\partial}{\partial x_i} \left(k_{ij} \frac{\partial T}{\partial x_j} \right) - Q - \Phi = 0 \quad (5.2.8)$$

The parameters α_v and α_T may have values of 0, 1, and 1/2 which produce the advective, conservation, and absolute conservation forms of the momentum and energy equations, respectively. Note that the difference between the advective and

conservative equations differ by the occurrence of a moment of the incompressibility constraint. In the continuum, the continuity equation is satisfied exactly and the advective and conservation forms are identical. Because the continuity equation is only satisfied approximately in a computational scheme, the two forms differ in any numerical method. Also, because the α parameters may be selected independently, a number of possible combinations of advective and conservation forms are possible. This also implies that the quantities conserved by the discrete method will differ with choices of α .

Though the demonstration of what quantities are conserved with particular selections of α is not difficult, it need not be repeated here; both references [5] and [6] have extensive discussions of the process, assumptions and results. As we are interested in Boussinesq systems, only a few results will be quoted here for standard mixed finite element methods using a coupled method. For $\alpha_v = \alpha_T = 0$, advective forms are used in both equations and nothing is conserved. If $\alpha_v = \alpha_T = 1$, then the conservation (divergence) form is employed and the global momentum and temperature are conserved. The combination of $\alpha_T = 1/2$ with any allowed value of α_v gives a mixed equation form and ensures conservation of the temperature squared; kinetic energy is conserved for an isothermal flow if $\alpha_v = 1/2$. Many other combinations are possible and additional considerations may be introduced if special pressure interpolation functions are introduced as in [6]. It is the general consensus that conservation of temperature squared is the most important quantity with regard to stability of the inviscid Boussinesq system and therefore $\alpha_v = \alpha_T = 1/2$ are the preferred parameters. However, the benefit of using the absolute conservation form when dissipation is incorporated appears to be minimal.

5.2.2 Boundary Conditions

The boundary conditions are given by

$$v_i = f_i^v(s_k, t) \quad \text{on } \Gamma_v \quad (5.2.9a)$$

$$T_i \equiv \sigma_{ij}(s_k, t) n_j(s_k) = f_i^T(s_k, t) \quad \text{on } \Gamma_T \quad (5.2.9b)$$

for the fluid mechanics part of the problem, and

$$T = f^T(s_k, t) \quad \text{on } \Gamma_T \quad (5.2.10a)$$

$$-\left(k_{ij}\frac{\partial T}{\partial x_j}\right)n_i \equiv q_i n_i = q_c + q_r = q_a = f^q(s_k, t) \quad \text{on } \Gamma_q \quad (5.2.10b)$$

for the heat transfer part of the problem. In Eq. (5.2.9b), σ_{ij} denote the components of the total stress tensor (i.e., viscous and hydrostatic)

$$\sigma_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - P\delta_{ij} \quad (5.2.11)$$

We will not explicitly consider the energy equation for solid regions in this chapter because it was considered in detail in Chapter 3. However, it should be recognized that the inclusion of solid body conduction regions in any convection problem (i.e., the so-called conjugate problem) is a relatively straightforward procedure. Also, as

mentioned in Chapter 1, convection problems may require the addition of auxiliary variables to describe all of the physical phenomena of interest. We will not consider this case explicitly since the formulation and solution methods for transport equations follow exactly the treatment of the energy equation. Note, however, that inclusion of auxiliary transport equations may significantly increase the magnitude of the computational effort.

There are five partial differential equations expressed in terms of five unknowns (v_1, v_2, v_3, P, T) for a three-dimensional problem. Following the developments of Chapters 3 and 4, we shall consider two finite element models of Eqs. (5.2.1)–(5.2.4). The first one is the *velocity-pressure-temperature model* or *mixed model*. The second model is the *penalty-finite element model*. These models were discussed in Chapter 4 for isothermal flows. Extension of these models to convective heat transfer problems is straightforward, as will be shown in this chapter. Note also that the general conservation form of the Boussinesq system, as given in (5.2.6)–(5.2.8), may also be used in the development of a finite element model with the addition of two terms.

5.3 Mixed Finite Element Model

Let us denote expressions on the left side of the equalities in (5.2.2)–(5.2.4) by f_1, \mathbf{f}_2 , and f_3 , respectively. As discussed in Section 4.2, the weighted-integral statements of the three equations over a typical element Ω^e are given by

$$\int_{\Omega^e} w_1 f_1 d\mathbf{x} = 0, \quad \int_{\Omega^e} \mathbf{w}_2 \cdot \mathbf{f}_2 d\mathbf{x} = 0, \quad \int_{\Omega^e} w_3 f_3 d\mathbf{x} = 0 \quad (5.3.1)$$

where (w_1, \mathbf{w}_2, w_3) are the weight functions, which will be equated to the interpolation functions used for (P, \mathbf{v}, T) , respectively. It should be recalled from Chapters 3 and 4 that the weighted-integral statements in Eq. (5.3.1) are reduced to weak statements by integrating the viscous and diffusion parts. All other terms are kept as they are.

Suppose that the dependent variables (T, v_i, P) are approximated by expansions of the form

$$T(\mathbf{x}, t) = \sum_{m=1}^M \theta_m(\mathbf{x}) T_m(t) = \boldsymbol{\Theta}^T \mathbf{T} \quad (5.3.2a)$$

$$v_i(\mathbf{x}, t) = \sum_{n=1}^N \psi_n(\mathbf{x}) v_i^n(t) = \boldsymbol{\Psi}^T \mathbf{v}_i \quad (5.3.2b)$$

$$P(\mathbf{x}, t) = \sum_{l=1}^L \phi_l(\mathbf{x}) P_l(t) = \boldsymbol{\Phi}^T \mathbf{P} \quad (5.3.2c)$$

where $\boldsymbol{\Theta}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\Phi}$ are vectors of interpolation (or shape) functions, \mathbf{T} , \mathbf{v}_i , and \mathbf{P} are vectors of nodal values of temperature, velocity components, and pressure, respectively, and superscript $(\cdot)^T$ denotes a transpose of the enclosed vector or matrix. The weight functions (w_1, \mathbf{w}_2, w_3) have the following correspondence (see Reddy [7] for further details)

$$w_1 \approx \phi_l, \quad \mathbf{w}_2 \approx \boldsymbol{\psi}_n, \quad w_3 \approx \theta_m \quad (5.3.3)$$

Substitution of Eqs. (5.3.2) and (5.3.3) into the weak forms associated with Eq. (5.3.1) results in the following finite element equations:

Continuity

$$-\left[\int_{\Omega^e} \Phi \frac{\partial \Psi^T}{\partial x_i} dx \right] \mathbf{v}_i = \mathbf{0} \quad (5.3.4)$$

Momentum

$$\begin{aligned} & \left[\int_{\Omega^e} \rho_0 \Psi \Psi^T dx \right] \dot{\mathbf{v}}_i + \left[\int_{\Omega^e} \rho_0 \Psi (\Psi^T \mathbf{v}_j) \frac{\partial \Psi^T}{\partial x_j} dx \right] \mathbf{v}_i + \left[\int_{\Omega^e} \mu \frac{\partial \Psi}{\partial x_j} \frac{\partial \Psi^T}{\partial x_j} dx \right] \mathbf{v}_i \\ & + \left[\int_{\Omega^e} \mu \frac{\partial \Psi}{\partial x_j} \frac{\partial \Psi^T}{\partial x_i} dx \right] \mathbf{v}_j - \left[\int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T dx \right] \mathbf{P} \\ & = - \left[\int_{\Omega^e} \rho_0 g_i \beta \Psi \Theta^T dx \right] \mathbf{T} + \left\{ \int_{\Omega^e} \rho_0 g_i \beta \Psi T_0 dx \right\} + \left\{ \oint_{\Gamma^e} \Psi \mathcal{T}_i ds \right\} \end{aligned} \quad (5.3.5)$$

Energy

$$\begin{aligned} & \left[\int_{\Omega^e} \rho_0 C_v \Theta \Theta^T dx \right] \dot{\mathbf{T}} + \left[\int_{\Omega^e} \rho_0 C_v \Theta (\Psi^T \mathbf{v}_j) \frac{\partial \Theta^T}{\partial x_j} dx \right] \mathbf{T} + \left[\int_{\Omega^e} k_{ij} \frac{\partial \Theta}{\partial x_i} \frac{\partial \Theta^T}{\partial x_j} dx \right] \mathbf{T} \\ & = \left\{ \int_{\Omega^e} \Theta Q dx \right\} + \left\{ \int_{\Omega^e} \Theta \Phi dx \right\} + \left\{ \oint_{\Gamma^e} \Theta q ds \right\} \end{aligned} \quad (5.3.6)$$

or

$$\begin{aligned} & \begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}}_1 \\ \dot{\mathbf{v}}_2 \\ \dot{\mathbf{v}}_3 \\ \dot{\mathbf{P}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{v}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\mathbf{v}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}(\mathbf{v}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{P} \end{Bmatrix} + \\ & \begin{bmatrix} \hat{\mathbf{K}}_{11} & \mathbf{K}_{21} & \mathbf{K}_{31} & -\mathbf{Q}_1 \\ \mathbf{K}_{12} & \hat{\mathbf{K}}_{22} & \mathbf{K}_{32} & -\mathbf{Q}_2 \\ \mathbf{K}_{13} & \mathbf{K}_{23} & \hat{\mathbf{K}}_{33} & -\mathbf{Q}_3 \\ -\mathbf{Q}_1^T & -\mathbf{Q}_2^T & -\mathbf{Q}_3^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{P} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1(\mathbf{T}) \\ \mathbf{F}_2(\mathbf{T}) \\ \mathbf{F}_3(\mathbf{T}) \\ \mathbf{0} \end{Bmatrix} \end{aligned} \quad (5.3.7)$$

$$\mathbf{N} \dot{\mathbf{T}} + \mathbf{D}(\mathbf{v}) \mathbf{T} + \mathbf{L} \mathbf{T} = \mathbf{G}(\mathbf{T}) \quad (5.3.8)$$

The coefficient matrices shown in Eqs. (5.3.7) and (5.3.8) are defined by

$$\begin{aligned} \hat{\mathbf{K}}_{11} &= 2\mathbf{K}_{11} + \mathbf{K}_{22} + \mathbf{K}_{33} \\ \hat{\mathbf{K}}_{22} &= \mathbf{K}_{11} + 2\mathbf{K}_{22} + \mathbf{K}_{33} \\ \hat{\mathbf{K}}_{33} &= \mathbf{K}_{11} + \mathbf{K}_{22} + 2\mathbf{K}_{33} \end{aligned} \quad (5.3.9a)$$

$$\mathbf{M} = \int_{\Omega^e} \rho_0 \Psi \Psi^T dx ; \quad \mathbf{C}(\mathbf{v}) = \int_{\Omega^e} \rho_0 \Psi (\Psi^T \mathbf{v}_j) \frac{\partial \Psi^T}{\partial x_j} dx$$

$$\mathbf{K}_{ij} = \int_{\Omega^e} \mu \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_j} dx ; \quad \mathbf{Q}_i = \int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T dx$$

$$\mathbf{F}_i(\mathbf{T}) = - \int_{\Omega^e} \rho_0 g_i \beta \Psi \Psi^T \mathbf{T} dx + \int_{\Omega^e} \rho_0 g_i \beta \Psi T_0 dx + \oint_{\Gamma^e} \Psi \mathcal{T}_i ds$$

$$\begin{aligned}\mathbf{D}(\mathbf{v}) &= \int_{\Omega^e} \rho_0 C \Theta (\Psi^T \mathbf{v}_j) \frac{\partial \Theta^T}{\partial x_j} d\mathbf{x} \\ \mathbf{N} &= \int_{\Omega^e} \rho_0 C_v \Theta \Theta^T d\mathbf{x}; \quad \mathbf{L} = \int_{\Omega^e} k \frac{\partial \Theta}{\partial x_i} \frac{\partial \Theta^T}{\partial x_i} d\mathbf{x} \\ \mathbf{G} &= \int_{\Omega^e} \Theta Q d\mathbf{x} + \int_{\Omega^e} \Theta \Phi d\mathbf{x} + \oint_{\Gamma^e} \Theta q ds\end{aligned}\quad (5.3.9b)$$

where summation on repeated indices is implied. The above equations can be written symbolically as follows.

Continuity

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (5.3.10)$$

Momentum

$$\mathbf{M} \dot{\mathbf{v}} + \mathbf{Cv} + \mathbf{Kv} - \mathbf{QP} + \mathbf{BT} = \mathbf{F} \quad (5.3.11)$$

Energy

$$\mathbf{NT} + \mathbf{DT} + \mathbf{LT} = \mathbf{G} \quad (5.3.12)$$

where the superposed dot represents a time derivative and $\mathbf{v}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$. In writing Eq. (5.3.11) the buoyancy term \mathbf{B} has been separated from the general force vector \mathbf{F} . The expression \mathbf{BT} has the meaning

$$\mathbf{BT} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_3 \end{bmatrix} \begin{Bmatrix} \mathbf{T} \\ \mathbf{T} \\ \mathbf{T} \end{Bmatrix} \quad (5.3.13a)$$

where

$$\mathbf{B}_i = \int_{\Omega^e} \rho_0 \beta g_i \Psi^T \Theta d\mathbf{x} \quad (5.3.13b)$$

Note that the solid body conduction equation of Chapter 3 is obtained from (5.3.12) by neglecting the transport term \mathbf{D} .

Equations (5.3.10)–(5.3.12) can be combined into a single matrix equation

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{v}) + \mathbf{K}(\mathbf{T}) & -\mathbf{Q} & \mathbf{B}(\mathbf{T}) \\ -\mathbf{Q}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}(\mathbf{v}) + \mathbf{L}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(\mathbf{T}) \\ \mathbf{0} \\ \mathbf{G}(\mathbf{T}, \mathbf{u}) \end{Bmatrix} \quad (5.3.14)$$

The viscous and diffusion terms are indicated as being functions of temperature to accommodate the case of temperature dependent viscosity and conductivity; other functional dependencies are also possible. In a more symbolic format, Eq. (5.3.14) can be expressed as

$$\bar{\mathbf{M}} \dot{\mathbf{U}} + \bar{\mathbf{K}} \mathbf{U} = \bar{\mathbf{F}} \quad (5.3.15a)$$

where

$$\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{P}^T, \mathbf{T}^T\} \quad (5.3.15b)$$

Note that the general form of Eq. (5.2.15a) is the same as the nonlinear diffusion Eq. (3.6.3) or the incompressible flow system in Eq. (4.2.15a). Therefore, the time-approximations discussed in Chapters 3 and 4 are readily applicable to the ordinary differential equations in (5.2.15a).

The derivation to this point has considered the advective form of the equations. If conservation forms for the momentum and energy equations are desired, Eqs. (5.2.6)–(5.2.8) may be used for the GFEM. The result is the same as the above equations with additional convection terms $\mathbf{C}^\alpha \mathbf{v}$ and $\mathbf{D}^\alpha \mathbf{v}$ occurring in the momentum and energy equations. Using the above notation, these additional matrices are defined by

$$\mathbf{C}^\alpha \mathbf{v} = \begin{bmatrix} \mathbf{C}_1^\alpha(\mathbf{v}_1) & \mathbf{C}_2^\alpha(\mathbf{v}_1) & \mathbf{C}_3^\alpha(\mathbf{v}_1) \\ \mathbf{C}_1^\alpha(\mathbf{v}_2) & \mathbf{C}_2^\alpha(\mathbf{v}_2) & \mathbf{C}_3^\alpha(\mathbf{v}_2) \\ \mathbf{C}_1^\alpha(\mathbf{v}_3) & \mathbf{C}_2^\alpha(\mathbf{v}_3) & \mathbf{C}_3^\alpha(\mathbf{v}_3) \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{Bmatrix} \quad (5.3.16a)$$

where

$$\mathbf{C}_j^\alpha(\mathbf{v}_i) = \alpha_v \int_{\Omega^e} \rho_0 \boldsymbol{\Psi} (\boldsymbol{\Psi}^T \mathbf{v}_i) \frac{\partial \boldsymbol{\Psi}^T}{\partial x_j} d\mathbf{x} \quad (5.3.16b)$$

and

$$\mathbf{D}^\alpha \mathbf{v} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 \\ \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 \\ \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{Bmatrix} \quad (5.3.17a)$$

where

$$\mathbf{D}_j^\alpha(\mathbf{T}) = \alpha_T \int_{\Omega^e} \rho_0 C \boldsymbol{\Theta} (\boldsymbol{\Theta}^T \mathbf{T}) \frac{\partial \boldsymbol{\Theta}^T}{\partial x_j} d\mathbf{x} \quad (5.3.17b)$$

These forms provide entries for each velocity component \mathbf{v}_j in the momentum and energy equations, respectively. These terms are easily constructed and add little cost to the overall formulation. The α terms may complicate slightly the solution procedure as they add some cross-coupling matrices in the Jacobian of a Newton procedure as outlined in Section 5.7.

The three sets of interpolation functions used in (5.3.2) should be of the Lagrange type, i.e., derived by interpolating only the values of the functions, and not their derivatives. There are three different finite elements associated with the three field variables (v_i , P , T). Although, in principle, one can use different interpolation functions for the velocity field and temperature, it is practical to use the same type of interpolation for the two field variables (T, v_i). The choice of interpolation functions used for the pressure variable (see Section 4.5.2) in the mixed finite element model is constrained by the special role the pressure plays in incompressible flows. In order to prevent an over-constrained system of discrete equations, the interpolation used for pressure must be at least one order lower than that used for the velocity field (i.e., unequal interpolation). Further, pressure need not be made continuous across elements because the pressure variable does not constitute a primary variable of the mixed model.

For two-dimensional flows of viscous incompressible fluids, often the Lagrange quadratic interpolation is used for the velocity components. Two different pressure approximations are available when the velocities are approximated by quadratic Lagrange functions. The first is a continuous-bilinear approximation, in which the pressure is defined at the corner nodes of the element and is made continuous across

element boundaries. The second pressure approximation involves a discontinuous, linear variation defined on the element by

$$\Phi = \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \quad (5.3.18)$$

Here the unknowns are not nodal point values of the pressure but correspond to the coefficients in $P = a \cdot 1 + b \cdot x + c \cdot y$. A standard element used in the analysis of three-dimensional viscous flow problems is the eight-node, hexahedron (brick) element, where the velocity components are approximated using the trilinear Lagrange functions and pressure is a constant (and obviously discontinuous between elements).

5.4 Penalty Finite Element Model

5.4.1 Preliminary Comments

Recall that the velocity field of viscous incompressible flows must satisfy the momentum Eqs. (5.2.3) and in addition the continuity Eq. (5.2.2). Equation (5.2.2) is treated as a constraint on the velocity field. In the penalty function method, the constrained problem is reformulated as an unconstrained problem. The discussion presented in Section 4.3 is applicable here with the body force components f_i in equations (4.3.16a-c) replaced with the buoyancy forces, $g_i\beta(T - T_0)$. The weak statements associated with the penalty formulation of convective heat transfer are as follows:

Continuity and Momentum

$$\begin{aligned} 0 &= \int_{\Omega^e} \rho_0 \delta v_1 \left[\frac{\partial v_1}{\partial t} + \left(v_1 \frac{\partial v_1}{\partial x_1} + v_2 \frac{\partial v_1}{\partial x_2} + v_3 \frac{\partial v_1}{\partial x_3} \right) - g_1 \beta(T - T_0) \right] d\mathbf{x} \\ &\quad + \int_{\Omega^e} \left[2\mu \frac{\partial \delta v_1}{\partial x_1} \frac{\partial v_1}{\partial x_1} + \mu \frac{\partial \delta v_1}{\partial x_2} \left(\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} \right) + \mu \frac{\partial \delta v_1}{\partial x_3} \left(\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1} \right) \right] d\mathbf{x} \\ &\quad - \oint_{\Gamma^e} \delta v_1 T_1 ds + \int_{\Omega^e} \gamma_e \frac{\partial \delta v_1}{\partial x_1} \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \right) d\mathbf{x} \end{aligned} \quad (5.4.1a)$$

$$\begin{aligned} 0 &= \int_{\Omega^e} \rho_0 \delta v_2 \left[\frac{\partial v_2}{\partial t} + \left(v_1 \frac{\partial v_2}{\partial x_1} + v_2 \frac{\partial v_2}{\partial x_2} + v_3 \frac{\partial v_2}{\partial x_3} \right) - g_2 \beta(T - T_0) \right] d\mathbf{x} \\ &\quad + \int_{\Omega^e} \left[2\mu \frac{\partial \delta v_2}{\partial x_2} \frac{\partial v_2}{\partial x_2} + \mu \frac{\partial \delta v_2}{\partial x_1} \left(\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1} \right) + \mu \frac{\partial \delta v_2}{\partial x_3} \left(\frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x_2} \right) \right] d\mathbf{x} \\ &\quad - \oint_{\Gamma^e} \delta v_2 T_2 ds + \int_{\Omega^e} \gamma_e \frac{\partial \delta v_2}{\partial x_2} \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \right) d\mathbf{x} \end{aligned} \quad (5.4.1b)$$

$$\begin{aligned} 0 &= \int_{\Omega^e} \rho_0 \delta v_3 \left[\frac{\partial v_3}{\partial t} + \left(v_1 \frac{\partial v_3}{\partial x_1} + v_2 \frac{\partial v_3}{\partial x_2} + v_3 \frac{\partial v_3}{\partial x_3} \right) - g_3 \beta(T - T_0) \right] d\mathbf{x} \\ &\quad + \int_{\Omega^e} \left[2\mu \frac{\partial \delta v_3}{\partial x_3} \frac{\partial v_3}{\partial x_3} + \mu \frac{\partial \delta v_3}{\partial x_1} \left(\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1} \right) + \mu \frac{\partial \delta v_3}{\partial x_2} \left(\frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x_2} \right) \right] d\mathbf{x} \\ &\quad - \oint_{\Gamma^e} \delta v_3 T_3 ds + \int_{\Omega^e} \gamma_e \frac{\partial \delta v_3}{\partial x_3} \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \right) d\mathbf{x} \end{aligned} \quad (5.4.1c)$$

Energy

$$0 = \int_{\Omega^e} \rho_0 C_v \delta T \left[\frac{\partial T}{\partial t} + \left(v_1 \frac{\partial T}{\partial x_1} + v_2 \frac{\partial T}{\partial x_2} + v_3 \frac{\partial T}{\partial x_3} \right) - Q \right] d\mathbf{x} \\ + \int_{\Omega^e} k_{ij} \frac{\partial \delta T}{\partial x_i} \frac{\partial T}{\partial x_j} d\mathbf{x} - \oint_{\Gamma^e} \delta T q ds \quad (5.4.2)$$

where γ_e is the penalty parameter. We note that the pressure does not appear explicitly in the weak forms (5.4.1) and (5.4.2), although it is a part of the boundary stresses, \mathcal{T}_i [see Eqs. (5.2.9b) and (5.2.11)]. An approximation for the pressure can be post-computed from the relation [see Eq. (4.3.20)]

$$P = -\gamma_e \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \right) = -\gamma_e \frac{\partial v_i}{\partial x_i} \quad (5.4.3)$$

5.4.2 Reduced Integration Penalty Model

The penalty finite element model is obtained from Eq. (5.4.1) by substituting finite element interpolation (5.3.2b) for the velocity field and $\delta v_i = \Psi^T$:

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}}_1 \\ \dot{\mathbf{v}}_2 \\ \dot{\mathbf{v}}_3 \end{Bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{v}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\mathbf{v}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}(\mathbf{v}) \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{Bmatrix} + \\ \begin{bmatrix} 2\mathbf{K}_{11} + \mathbf{K}_{22} + \mathbf{K}_{33} & \mathbf{K}_{21} & \mathbf{K}_{31} \\ \mathbf{K}_{12} & \mathbf{K}_{11} + 2\mathbf{K}_{22} + \mathbf{K}_{33} & \mathbf{K}_{32} \\ \mathbf{K}_{13} & \mathbf{K}_{23} & \mathbf{K}_{11} + \mathbf{K}_{22} + 2\mathbf{K}_{33} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{Bmatrix} \\ + \begin{bmatrix} \hat{\mathbf{K}}_{11} & \hat{\mathbf{K}}_{12} & \hat{\mathbf{K}}_{13} \\ \hat{\mathbf{K}}_{21} & \hat{\mathbf{K}}_{22} & \hat{\mathbf{K}}_{23} \\ \hat{\mathbf{K}}_{31} & \hat{\mathbf{K}}_{32} & \hat{\mathbf{K}}_{33} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \end{Bmatrix} \quad (5.4.4)$$

where \mathbf{M} , $\mathbf{C}(\mathbf{v})$, \mathbf{K}_{ij} , and \mathbf{F}_i are the same as those defined in Eq. (5.3.9), and

$$\hat{\mathbf{K}}_{ij} = \int_{\Omega^e} \gamma_e \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_j} d\mathbf{x} \quad (5.4.5)$$

The finite element model of the energy Eq. (5.3.8) or (5.3.12) remains unchanged.

The comments made in Section 4.5.3 concerning the evaluation of the integrals in $\hat{\mathbf{K}}$ of Eq. (5.4.4) should be recalled here. These coefficients (i.e., penalty terms) should be evaluated using a numerical integration rule of an order less than that required to integrate them exactly. For example, if a linear rectangular element is used to approximate the velocity field for two-dimensional problems, all matrix coefficients except the penalty terms should be evaluated using the 2×2 Gauss quadrature, and the penalty terms should be evaluated using the one-point (1×1) Gauss quadrature. When a quadratic rectangular element is used, the 3×3 Gauss quadrature is used to evaluate the non-penalty terms, and the 2×2 Gauss quadrature is used to evaluate the penalty terms. Similar comments apply to three-dimensional elements.

Concerning the post-computation of pressure in the penalty model, in general, the pressure computed from Eq. (5.4.3) at the integration points is not always reliable and accurate, and one is advised to use either the Poisson equation for pressure [see Eq. (4.5.11)] or the scheme suggested in Section 4.5.4 [see Eq. (4.5.10)].

5.4.3 Consistent Penalty Model

In this model, the finite element model of Eq. (5.4.3) is constructed first. We have

$$\left[\int_{\Omega^e} \Phi \Phi^T d\mathbf{x} \right] \mathbf{P} = - \left[\int_{\Omega^e} \gamma_e \Phi \frac{\partial \Psi^T}{\partial x_i} d\mathbf{x} \right] \mathbf{v}_i \quad (5.4.6a)$$

or in matrix notation

$$\mathbf{M}_p \mathbf{P} = -\gamma_e \mathbf{Q}^T \mathbf{v} \quad (5.4.6b)$$

Inverting Eq. (5.4.6b) for \mathbf{P} , we obtain

$$\mathbf{P} = -\gamma_e \mathbf{M}_p^{-1} \mathbf{Q}^T \mathbf{v} \quad (5.4.7)$$

where $\mathbf{v}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$, $\mathbf{Q}^T = \{\mathbf{Q}_1^T, \mathbf{Q}_2^T, \mathbf{Q}_3^T\}$, and \mathbf{Q}_i and \mathbf{M}_p are given by

$$\mathbf{Q}_i = \int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T d\mathbf{x}; \quad \mathbf{M}_p = \int_{\Omega^e} \Phi \Phi^T d\mathbf{x} \quad (5.4.8)$$

Next, Eq. (5.4.7) is substituted for the pressure into the finite element model (5.3.11) to obtain

$$\mathbf{M} \dot{\mathbf{v}} + (\mathbf{C}(\mathbf{v}) + \mathbf{K} + \mathbf{K}_p) \mathbf{v} = \mathbf{F} \quad (5.4.9a)$$

with

$$\mathbf{K}_p = \gamma_e \mathbf{Q} \mathbf{M}_p^{-1} \mathbf{Q}^T \quad (5.4.9b)$$

It should be noted that reduced integration is not used to evaluate \mathbf{K}_p . For straight-sided elements, one can show the equivalence of matrix $\hat{\mathbf{K}}$ in Eq. (5.4.4) and the matrix \mathbf{K}_p . The post-computation of the pressure P using Eq. (5.4.7) in the consistent penalty method yields very good results.

5.5 Finite Element Models of Porous Flow

The derivation of the finite element model for the nonisothermal, incompressible porous flow problem follows the developments of Section 4.4 in a completely analogous manner. Note that the Boussinesq approximation is utilized in this formulation. The density in the body force term is given by (5.2.1) and is considered a constant at ρ_0 in all other terms. For completeness, we record the mixed finite element model including the buoyancy term:

Continuity

$$-\left[\int_{\Omega^e} \Phi \frac{\partial \Psi^T}{\partial x_i} d\mathbf{x} \right] \mathbf{v}_i = \mathbf{0} \quad (5.5.1)$$

Momentum

$$\left[\int_{\Omega^e} \frac{\rho_0}{\phi} \Psi \Psi^T d\mathbf{x} \right] \dot{\mathbf{v}}_i + \left[\int_{\Omega^e} \frac{\rho_0 \hat{c}}{\sqrt{\kappa}} \Psi (\Psi^T \|\mathbf{v}\|) \Psi^T d\mathbf{x} \right] \mathbf{v}_i + \left[\int_{\Omega^e} \frac{\mu}{\kappa} \Psi \Psi^T d\mathbf{x} \right] \dot{\mathbf{v}}_i$$

$$+ \left[\int_{\Omega^e} \mu_e \frac{\partial \Psi}{\partial x_j} \frac{\partial \Psi^T}{\partial x_j} dx \right] \mathbf{v}_i + \left[\int_{\Omega^e} \mu_e \frac{\partial \Psi}{\partial x_j} \frac{\partial \Psi^T}{\partial x_i} dx \right] \mathbf{v}_j - \left[\int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T dx \right] \mathbf{P} = \\ - \left[\int_{\Omega^e} \rho_0 g_i \beta \Psi \Theta dx \right] \mathbf{T} + \left\{ \int_{\Omega^e} \rho_0 g_i \beta \Psi T_0 dx \right\} + \left\{ \oint_{\Gamma^e} \Psi T_i ds \right\} \quad (5.5.2)$$

Energy

$$\left[\int_{\Omega^e} (\rho C)_e \Theta \Theta^T dx \right] \dot{\mathbf{T}} + \left[\int_{\Omega^e} \rho_0 C \Theta (\Psi^T \mathbf{u}_j) \frac{\partial \Theta^T}{\partial x_j} dx \right] \mathbf{T} + \\ \left[\int_{\Omega^e} k_e \frac{\partial \Theta}{\partial x_i} \frac{\partial \Theta^T}{\partial x_i} dx \right] \mathbf{T} = \left\{ \int_{\Omega^e} \Theta Q dx \right\} + \left\{ \oint_{\Gamma^e} \Theta q_n ds \right\} \quad (5.5.3)$$

Note that the dissipation term is omitted from the energy equation, and the acceleration tensor is taken to be $1/\phi$. The above equations can be written in matrix form as given below.

Continuity

$$-\tilde{\mathbf{Q}}^T \mathbf{v} = \mathbf{0} \quad (5.5.4)$$

Momentum

$$\tilde{\mathbf{M}} \dot{\mathbf{v}} + \tilde{\mathbf{C}} \mathbf{v} + \tilde{\mathbf{K}} \mathbf{v} - \tilde{\mathbf{Q}} \mathbf{P} + \tilde{\mathbf{B}} \mathbf{T} = \tilde{\mathbf{F}} \quad (5.5.5)$$

Energy

$$\tilde{\mathbf{N}} \dot{\mathbf{T}} + \tilde{\mathbf{D}} \mathbf{T} + \tilde{\mathbf{L}} \mathbf{T} = \tilde{\mathbf{G}} \quad (5.5.6)$$

where $\mathbf{v}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$ and

$$\begin{aligned} \tilde{\mathbf{M}} &= \int_{\Omega^e} \frac{\rho_0}{\phi} \Psi \Psi^T dx, \quad \tilde{\mathbf{C}}(\mathbf{v}) = \int_{\Omega^e} \frac{\rho_0 \hat{c}}{\sqrt{\kappa}} \Psi \Psi^T \|\mathbf{v}\| \Psi^T dx \\ \tilde{\mathbf{A}} &= \int_{\Omega^e} \frac{\mu}{\kappa} \Psi \Psi^T dx, \quad \tilde{\mathbf{K}}_{ij} = \int_{\Omega^e} \mu_e \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_j} dx, \quad \tilde{\mathbf{Q}}_i = \int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T dx \\ \tilde{\mathbf{F}}_i(\mathbf{T}) &= - \int_{\Omega^e} \rho_0 g_i \beta \Psi \Theta dx + \int_{\Omega^e} \rho_0 g_i \beta \Psi T_0 dx + \oint_{\Gamma^e} \Psi T_i ds \quad (5.5.7) \\ \tilde{\mathbf{D}}(\mathbf{v}) &= \int_{\Omega^e} \rho_0 C \Theta (\Psi^T \mathbf{v}_j) \frac{\partial \Theta^T}{\partial x_j} dx, \quad \tilde{\mathbf{N}} = \int_{\Omega^e} (\rho C)_e \Theta \Theta^T dx \\ \tilde{\mathbf{L}} &= \int_{\Omega^e} k_e \frac{\partial \Theta}{\partial x_i} \frac{\partial \Theta^T}{\partial x_i} dx, \quad \tilde{\mathbf{G}} = \int_{\Omega^e} \Theta Q dx + \oint_{\Gamma^e} \Theta q_n ds \end{aligned}$$

where summation on repeated indices is implied. The porous flow Eqs. (5.5.4)–(5.5.6) can be arranged into a single matrix equation of the same form as given previously for the flow equations:

$$\begin{bmatrix} \tilde{\mathbf{M}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{N}} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{N}} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \tilde{\mathbf{C}}(\mathbf{v}) + \tilde{\mathbf{A}} + \tilde{\mathbf{K}} & -\tilde{\mathbf{Q}} & \tilde{\mathbf{B}}(\mathbf{T}) \\ -\tilde{\mathbf{Q}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{D}}(\mathbf{v}) + \tilde{\mathbf{L}}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \tilde{\mathbf{F}}(\mathbf{T}) \\ \mathbf{0} \\ \tilde{\mathbf{G}}(\mathbf{T}, \mathbf{v}) \end{Bmatrix} \quad (5.5.8)$$

5.6 Nonisothermal, Low-Speed, Compressible Flows

5.6.1 Governing Equations

The equations describing nonisothermal, low-speed compressible flow of a Newtonian fluid were outlined in Chapter 1 and are summarized here. The equations are written in advective form using the standard acoustically filtered approximation. The equation of state and conservation equations are presented below.

Equation of State

$$\rho = \rho(T, P_0) \quad (5.6.1)$$

Conservation of Mass

$$\frac{\partial \rho v_i}{\partial x_i} = -\frac{\partial \rho}{\partial t} \quad (5.6.2)$$

Conservation of Momentum

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left[-P' \delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \frac{2}{3} \frac{\partial}{\partial x_i} \left[\mu \left(\frac{\partial v_k}{\partial x_k} \right) \right] - \rho g_i = 0 \quad (5.6.3)$$

Conservation of Energy

$$\rho C_v \left(\frac{\partial T}{\partial t} + v_i \frac{\partial T}{\partial x_i} \right) - \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right) - Q - \Phi - \beta T \frac{dP_0}{dt} = 0 \quad (5.6.4)$$

The variables and parameters used in (5.6.1)–(5.6.4) have the standard meaning; the viscous dissipation definition is altered because the stress tensor is expanded to include the velocity divergence term. In the above system of equations, the total pressure P is composed of two parts

$$P(x_i, t) = P_0(t) + P'(x_i, t) \quad (5.6.5)$$

where P_0 is a spatially uniform, time-dependent background (thermodynamic) pressure and P' is a dynamical (mechanical) pressure. The absence of the dynamical pressure from the equation of state precludes acoustic waves and gives the fluid model its name. Formal derivation and theoretical discussion of this system is available in [1-3].

The system in (5.6.2)–(5.6.5) is observed to be quite similar to the Boussinesq system (5.2.2)–(5.2.5) if the density is considered a variable parameter that can be evaluated as needed using (5.6.1). The additional “source” terms in (5.6.2) and (5.6.4) pose little difficulty in evaluation and the inclusion of the added stress term is straightforward. This similarity with the Boussinesq system provides guidance for the development of a finite element method for low-speed compressible flows as outlined in a following section.

For completeness, the conservation form of the non-Boussinesq equations will also be summarized. The conservation form is significantly more complex than the advective form, with a number of additional terms that complicate both the finite element model and associated solution procedures. Though the differences between numerical models for the advection and conservation forms of the Boussinesq equations have been judged to be generally insignificant, the same conclusion is not available

for the non-Boussinesq (acoustically filtered) equations; comparison of the models requires additional evaluation. The generalized advection operator, used previously with the Boussinesq equations, could also be employed here to arrive at the proper conservation form. However, an alternate approach will be used to demonstrate a limiting form of the fully compressible equations. In this approach, the all-Mach number formulation, as defined by Hauke and Hughes [8,9], is the starting point. As noted in [8,9], the fully compressible equations will not have a well defined incompressible limit when density is used as a variable. The most convenient variables are therefore the pressure, primitive variables (P, v_i, T) . Using these variables, the conservation form of the fully compressible flow equations includes an equation of state. The equation of state and conservation equations are given below.

Equation of State

$$\rho = \rho(T, P) \quad (5.6.6)$$

Conservation of Mass

$$\rho\kappa\frac{DP}{Dt} - \rho\beta\frac{DT}{Dt} + \rho\frac{\partial v_i}{\partial x_i} = 0 \quad (5.6.7)$$

Conservation of Momentum

$$\rho\frac{Dv_i}{Dt} + \rho\kappa v_i\frac{DP}{Dt} - \rho\beta v_i\frac{DT}{Dt} + \rho v_i\frac{\partial v_k}{\partial x_k} + \delta_{ij}\frac{\partial P}{\partial x_j} - \frac{\partial\sigma_{ij}}{\partial x_j} - \rho g_i = 0 \quad (5.6.8)$$

Conservation of Energy

$$\rho\frac{De^t}{Dt} + \rho\kappa e^t\frac{DP}{Dt} - \rho\beta e^t\frac{DT}{Dt} + \rho e^t\frac{\partial v_k}{\partial x_k} + \frac{\partial Pv_i}{\partial x_i} - \frac{\partial\sigma_{ij}v_j}{\partial x_i} + \frac{\partial q_i}{\partial x_i} - \rho g_i v_i - Q = 0 \quad (5.6.9)$$

where the additional parameters and variables are the total energy $e^t = e + 1/2v_k v_k$, defined as the sum of the internal and kinetic energies, the isothermal compressibility coefficient, κ , and the coefficient of thermal expansion (at constant pressure) β , given by

$$\kappa = \frac{1}{\rho}\frac{\partial\rho}{\partial P}\Big|_T, \quad \beta = -\frac{1}{\rho}\frac{\partial\rho}{\partial T}\Big|_P \quad (5.6.10)$$

The viscous stress tensor σ_{ij} again has the compressible form which includes the divergence of the velocity. Note also that the substantial derivative of the density is

$$\frac{D\rho}{Dt} = \rho\kappa\frac{DP}{Dt} - \rho\beta\frac{DT}{Dt} \quad (5.6.11)$$

A standard scaling of Eqs. (5.6.7)–(5.6.9) allows the low Mach number form of the equations to be derived. In this limit, the kinetic energy terms and dissipation terms in the energy equation are neglected (small Eckert number). The resulting system of equations is valid for low speed flows without restrictions on the equation of state. If a perfect gas is assumed for convenience, the resulting equations are (5.6.7), (5.6.8), and a simplified energy equation of the form

$$\rho\frac{DC_vT}{Dt} + \rho\kappa C_v T\frac{DP}{Dt} - \rho\beta C_v T\frac{DT}{Dt} + \rho C_v T\frac{\partial v_k}{\partial x_k} + \frac{\partial q_i}{\partial x_i} - Q = 0 \quad (5.6.12)$$

where the relations $P = \rho RT$ and $e = C_v T$ have been defined. This system is not acoustically filtered because the density is still a function of the total (thermodynamic) pressure. However, the system is a valid form for nonisothermal, low-speed flows and could possibly be used for developing finite element models for this class of heat transfer problem. The problem of acoustic waves may cause problems in this system for certain flows and the further reduction to an acoustically filtered form is desirable.

When the isothermal compressibility parameter κ is set to zero, the density relation in Eq. (5.6.11) is simplified and the conservation equations reduce to a conservation form analogous to the non-Boussinesq equations in (5.6.2)–(5.6.4). This system can be expressed as follows.

Conservation of Mass

$$\rho \frac{\partial v_i}{\partial x_i} - \rho \beta \frac{DT}{Dt} = 0 \quad (5.6.13)$$

Conservation of Momentum

$$\rho \frac{Dv_i}{Dt} + v_i \left(\rho \frac{\partial v_k}{\partial x_k} - \rho \beta \frac{DT}{Dt} \right) + \delta_{ij} \frac{\partial P}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i = 0 \quad (5.6.14)$$

Conservation of Energy

$$\rho C_v \frac{DT}{Dt} + C_v T \left(\rho \frac{\partial v_k}{\partial x_k} - \rho \beta \frac{DT}{Dt} \right) + \frac{\partial q_i}{\partial x_i} - Q = 0 \quad (5.6.15)$$

If the mass conservation equation is used to simplify (5.6.14) and (5.6.15), the resulting system is the advective form of the acoustically filtered equations. Though not shown here, it is straightforward to incorporate a spatially uniform, background pressure in the development. Also, if the thermal expansion coefficient is set to zero, the conservation form of the Boussinesq equations are produced though some assumption regarding the body force term is still required.

Both the advective and conservation forms of the acoustically filtered, non-Boussinesq equations may be used for developing finite element models for heat transfer applications. A comparison of the Boussinesq and non-Boussinesq formulations is illustrated in the examples of Section 5.13.

5.6.2 Boundary Conditions

The boundary conditions for the (acoustically filtered) non-Boussinesq equations are formally the same as given previously in Section 5.2.2 for the Boussinesq system. Typically, the velocity and/or tractions are specified on various parts of the fluid mechanics boundary. Also, the temperature and/or heat flux is specified for the thermal boundary of the problem.

5.6.3 Mixed Finite Element Model

The derivation of a finite element model for the acoustically filtered equations follows closely the development of the Boussinesq system. The advective form of the non-Boussinesq system will be considered in detail; the derivation for the conservation form need only be outlined as the models differ by only a few terms. Because

the dynamical part of the pressure P' does not appear in the equation of state, it plays the same role as the pressure in an incompressible flow. That is, the dynamical pressure is a Lagrange multiplier that enforces satisfaction of the continuity constraint. Note that in the time dependent case, the continuity equation has a source term that is not present in the Boussinesq system and will require evaluation based on the equation of state. Overall, the density is generally treated as a variable parameter in the problem that is evaluated from the temperature field through the equation of state.

Spatial discretization of the acoustically filtered Eqs. (5.6.2)–(5.6.4) may be accomplished using the Galerkin form of the method of weighted residuals. The specific approach considered here was designed by Martinez [10] to be a simple extension of the Boussinesq equations. The formulation differs only slightly from other published procedures (see, for example, [11,12]). The weighted integral statements for the conservation of mass, momentum and energy are the same as for the Boussinesq equations and are given in (5.3.1). Let the dependent variables be approximated by expansions of the form

$$T(\mathbf{x}, t) = \sum_{m=1}^M \theta_m(\mathbf{x}) T_m(t) = \boldsymbol{\Theta}^T \mathbf{T} \quad (5.6.16a)$$

$$v_i(\mathbf{x}, t) = \sum_{n=1}^N \psi_n(\mathbf{x}) v_i^n(t) = \boldsymbol{\Psi}^T \mathbf{v}_i \quad (5.6.16b)$$

$$P(\mathbf{x}, t) = \sum_{l=1}^L \phi_l(\mathbf{x}) P_l(t) = \boldsymbol{\Phi}^T \mathbf{P} \quad (5.6.16c)$$

$$\rho(\mathbf{x}, t) = \sum_{m=1}^M \theta_m(\mathbf{x}) \rho_m(t) = \boldsymbol{\Theta}^T \rho \quad (5.6.16d)$$

where the temperature and density variables are assumed to have the same interpolation functions. In general, the interpolation functions for the velocity will also be the same as the temperature functions with the pressure being one order lower than the velocity.

Substitution of (5.6.16) into the weak form associated with (5.3.1) and identification of the appropriate weighting functions as in (5.3.3), produces the following matrix equations.

Continuity

$$\mathbf{H}\dot{\rho} + \mathbf{Q}^T \mathbf{R}\mathbf{v} = \mathbf{0} \quad (5.6.17)$$

Momentum

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{R}\mathbf{v})\mathbf{v} + \tilde{\mathbf{K}}\mathbf{v} - \mathbf{Q}\mathbf{P} + \mathbf{B} = \mathbf{F} \quad (5.6.18)$$

Energy

$$\mathbf{N}\dot{\mathbf{T}} + \mathbf{D}(\mathbf{R}\mathbf{v})\mathbf{T} + \mathbf{L}\mathbf{T} = \mathbf{G} \quad (5.6.19)$$

State

$$\rho = \rho(\mathbf{T}, P_0) \quad (5.6.20)$$

where the superposed dot represents a time derivative and $\mathbf{v}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$. This system is analogous to the Boussinesq equations displayed in (5.3.10)–(5.3.12). The individual matrix terms are defined by

$$\begin{aligned}
 \mathbf{H} &= \int_{\Omega^e} \Phi \Theta^T d\mathbf{x} \\
 \mathbf{M} &= \int_{\Omega^e} \rho \Psi \Psi^T d\mathbf{x}; \quad \mathbf{C}(\mathbf{R}\mathbf{v}) = \int_{\Omega^e} \Psi (\Psi^T \mathbf{R} \mathbf{v}_j) \frac{\partial \Psi^T}{\partial x_j} d\mathbf{x} \\
 \mathbf{K}_{ij} &= \int_{\Omega^e} \mu \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_j} d\mathbf{x}; \quad \mathbf{Q}_i = \int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T d\mathbf{x} \\
 \mathbf{B}_i &= \int_{\Omega^e} \Psi \Theta^T \rho g_i d\mathbf{x}; \quad \mathbf{F}_i = \oint_{\Gamma^e} \Psi \mathcal{T}_i ds \quad (5.6.21a) \\
 \mathbf{D}(\mathbf{R}\mathbf{v}) &= \int_{\Omega^e} C \Theta (\Psi^T \mathbf{R} \mathbf{v}_j) \frac{\partial \Theta^T}{\partial x_j} d\mathbf{x} \\
 \mathbf{N} &= \int_{\Omega^e} \rho C_v \Theta \Theta^T d\mathbf{x}; \quad \mathbf{L} = \int_{\Omega^e} k \frac{\partial \Theta}{\partial x_i} \frac{\partial \Theta^T}{\partial x_i} d\mathbf{x} \\
 \mathbf{G} &= \int_{\Omega^e} \Theta Q d\mathbf{x} + \oint_{\Gamma^e} \Theta q ds \\
 \tilde{\mathbf{K}}_{11} &= \frac{4}{3} \mathbf{K}_{11} + \mathbf{K}_{22} + \mathbf{K}_{33} \\
 \tilde{\mathbf{K}}_{12} &= \mathbf{K}_{21} - \frac{2}{3} \mathbf{K}_{12} \\
 \tilde{\mathbf{K}}_{13} &= \mathbf{K}_{31} - \frac{2}{3} \mathbf{K}_{13} \\
 \tilde{\mathbf{K}}_{21} &= \mathbf{K}_{12} - \frac{2}{3} \mathbf{K}_{21} \\
 \tilde{\mathbf{K}}_{22} &= \mathbf{K}_{11} + \frac{4}{3} \mathbf{K}_{22} + \mathbf{K}_{33} \quad (5.6.21b) \\
 \tilde{\mathbf{K}}_{23} &= \mathbf{K}_{32} - \frac{2}{3} \mathbf{K}_{23} \\
 \tilde{\mathbf{K}}_{13} &= \mathbf{K}_{31} - \frac{2}{3} \mathbf{K}_{13} \\
 \tilde{\mathbf{K}}_{23} &= \mathbf{K}_{32} - \frac{2}{3} \mathbf{K}_{23} \\
 \tilde{\mathbf{K}}_{33} &= \mathbf{K}_{11} + \mathbf{K}_{22} + \frac{4}{3} \mathbf{K}_{33}
 \end{aligned}$$

Equations (5.6.17)–(5.6.20) represent coupled, nonlinear ordinary differential equations for the nodal point unknowns of velocity \mathbf{v}_i , mechanical pressure \mathbf{P} , temperature \mathbf{T} and density ρ . In this formulation the matrix $\mathbf{R} = \text{dia}[\rho]$ is used to include the variable density in the convective terms. In essence, the convective terms have the term ρv_i interpolated as a product rather than a product of interpolants. This manipulation is not mandatory and has not been used in other implementations. Compared to the standard Boussinesq formulation, only the \mathbf{H} matrix is new, though the diffusion matrix $\tilde{\mathbf{K}}$ is altered by the compressible form of the stress tensor and the body force term \mathbf{B} is not explicit in the temperature.

As noted in previous sections, the separate Eqs. (5.6.17)–(5.6.19) can be combined into a single matrix equation as

$$\begin{aligned} \begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{R}\mathbf{v}) + \tilde{\mathbf{K}}(\mathbf{T}) & -\mathbf{Q} & \mathbf{B} \\ -\mathbf{Q}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}(\mathbf{R}\mathbf{v}) + \mathbf{L}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} \\ = \begin{Bmatrix} \mathbf{F} \\ \mathbf{H}\dot{\rho} \\ \mathbf{G}(\mathbf{T}, \mathbf{v}) \end{Bmatrix} \end{aligned} \quad (5.6.22)$$

In a more symbolic format, Eq. (5.6.22) can be expressed as

$$\bar{\mathbf{M}}\dot{\mathbf{U}} + \bar{\mathbf{K}}\mathbf{U} = \bar{\mathbf{F}} \quad (5.6.23a)$$

where again

$$\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{P}^T, \mathbf{T}^T\} \quad (5.6.23b)$$

In the non-Boussinesq system, the density is treated as a temperature dependent property based on (5.6.20); the background pressure is computed as needed from an initial condition or an energy balance on a closed system. The density rate term in (5.6.17) may be evaluated by either of two methods. In the simplest case, the rate term $\mathbf{H}\dot{\rho}$ is moved to the right-hand side of the equation (as shown above) and time-differenced appropriately. Previous values of the density and density rate may be stored or recomputed from the equation of state as needed. In a more implicit method, the density rate is replaced by a temperature rate using

$$\dot{\rho} = \left(\frac{\partial \rho}{\partial T} \right) \dot{T} \quad (5.6.24)$$

An appropriate recursion relation is employed for \dot{T} and the term remains on the left-hand side of the equation. Both methods have been used in practice with the more implicit scheme being somewhat more robust. General solution methods for the non-Boussinesq equations are similar to the Boussinesq system and are outlined in the following section. Further discussion of the non-Boussinesq formulation and application can be found in [4,11,12] and in the examples in a later section.

5.7 Solution Methods

5.7.1 General Discussion

The finite element models developed in the previous sections describe a very large group of heat transfer and fluid mechanics problems ranging in complexity from simple solid body conduction through forced and free convection, to mixed mode convection/conduction. The solution methods used depend on the model, computational resources, the nonlinearity of the system, and the strength of the coupling between equations (see Section 4.6).

The solution procedures described in Section 4.6 were all variants of the basic fixed point method (see Appendix C) and were generally designed to operate concurrently on all of the equations present in a finite element model. The strong coupling

between equations (e.g., momentum and energy) that is characteristic of convective heat transfer problems makes these combined equation methods optimal from the standpoint of convergence rate. The disadvantage, of course, is that a very large and computationally expensive matrix problem must be treated at each iteration. The requirement to perform larger (more elements and higher dimensionality) and more complex (physical phenomena) simulations has reached the point where the usual direct matrix methods for combined equations are prohibitively expensive. A natural choice to make the matrix problem more affordable (while retaining the standard fixed point schemes) is to switch from the Gauss elimination type methods to the iterative methods, such as the preconditioned conjugate gradient (PCG) method. Previously, the development of iterative methods for combined equation sets was slow and has been handicapped by the lack of good preconditioner techniques that can adequately treat the dominating effect of the incompressibility constraint. Various methods now exist that can solve an increasing range of combined equation problems. Though still not as robust or generally applicable as required by the spectrum of applications, there is an increasing use of iterative methods. However, for some demanding applications, the combined equation approach may be sacrificed for alternative formulations of the discrete equations.

In the following sections we first review Newton's method for convective heat transfer problems, a combined equation technique that is heavily used for two-dimensional problems and modest three-dimensional applications. Various segregated solution methods are then reviewed. Throughout this section we will only consider the time-independent equations. From the discussion in Section 4.6 it should be recognized that any of the standard time integration procedures can be applied to the convective heat transfer equations. The resulting nonlinear algebraic equations for each time step are then analogous to the steady state problem, and they can be treated with the algorithms described in the following sections.

5.7.2 Newton's Method

The application of Newton's method to the combined equations for convective heat transfer (5.3.14), (5.5.8) or (5.6.22) follows directly from the isothermal case described in Section 4.6.2. The iterative procedure is given by

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \mathcal{J}^{-1}(\mathbf{U}^n)\mathbf{R}(\mathbf{U}^n) \quad (5.7.1)$$

where the vector of unknowns is $\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{P}^T, \mathbf{T}^T\}$ for the mixed finite element model and $\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{T}^T\}$ for the penalty finite element model. The definition of the Jacobian matrix \mathcal{J} is expanded to include the variations in the equations with respect to the vector of nodal temperatures, \mathbf{T} . Thus for the two-dimensional mixed finite element model of a Boussinesq system we have

$$\mathcal{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{U}} = \begin{bmatrix} \frac{\partial \mathbf{R}_1}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{R}_1}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{R}_1}{\partial \mathbf{P}} & \frac{\partial \mathbf{R}_1}{\partial \mathbf{T}} \\ \frac{\partial \mathbf{R}_2}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{R}_2}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{R}_2}{\partial \mathbf{P}} & \frac{\partial \mathbf{R}_2}{\partial \mathbf{T}} \\ \frac{\partial \mathbf{R}_3}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{R}_3}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{R}_3}{\partial \mathbf{P}} & \frac{\partial \mathbf{R}_3}{\partial \mathbf{T}} \\ \frac{\partial \mathbf{R}_4}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{R}_4}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{R}_4}{\partial \mathbf{P}} & \frac{\partial \mathbf{R}_4}{\partial \mathbf{T}} \end{bmatrix} \quad (5.7.2)$$

where

$$\begin{aligned}\mathbf{R}_1 &= \mathbf{C}_1(\mathbf{v}_1)\mathbf{v}_1 + \mathbf{C}_2(\mathbf{v}_2)\mathbf{v}_1 + (2\mathbf{K}_{11} + \mathbf{K}_{22})\mathbf{v}_1 + \mathbf{K}_{12}\mathbf{v}_2 - \mathbf{Q}_1\mathbf{P} + \mathbf{B}_1\mathbf{T} - \mathbf{F}_1 \\ \mathbf{R}_2 &= \mathbf{C}_1(\mathbf{v}_1)\mathbf{v}_2 + \mathbf{C}_2(\mathbf{v}_2)\mathbf{v}_2 + \mathbf{K}_{21}\mathbf{v}_1 + (\mathbf{K}_{11} + 2\mathbf{K}_{22})\mathbf{v}_2 - \mathbf{Q}_2\mathbf{P} + \mathbf{B}_2\mathbf{T} - \mathbf{F}_2 \\ \mathbf{R}_3 &= -\mathbf{Q}_1^T\mathbf{v}_1 - \mathbf{Q}_2^T\mathbf{v}_2 \\ \mathbf{R}_4 &= \mathbf{D}_1(\mathbf{v}_1)\mathbf{T} + \mathbf{D}_2(\mathbf{v}_2)\mathbf{T} + (\mathbf{L}_{11} + \mathbf{L}_{22})\mathbf{T} - \mathbf{G}\end{aligned}\quad (5.7.3)$$

which correspond to the two momentum equations, the incompressibility constraint, and the energy equation. The matrices appearing in Eq. (5.7.3) are defined in Eqs. (5.3.9a,b). The components of \mathcal{J} are computed directly from the definitions in (5.7.3) and yield the following values:

$$\begin{aligned}\frac{\partial \mathbf{R}_1}{\partial \mathbf{v}_1} &= \mathbf{C}_1(\mathbf{v}_1) + \mathbf{C}_2(\mathbf{v}_2) + \mathbf{C}_1(1)\mathbf{v}_1 + 2\mathbf{K}_{11} + \mathbf{K}_{22} \\ \frac{\partial \mathbf{R}_1}{\partial \mathbf{v}_2} &= \mathbf{C}_2(1)\mathbf{v}_1 + \mathbf{K}_{21}; \quad \frac{\partial \mathbf{R}_1}{\partial \mathbf{P}} = -\mathbf{Q}_1; \quad \frac{\partial \mathbf{R}_1}{\partial \mathbf{T}} = \mathbf{B}_1 \\ \frac{\partial \mathbf{R}_2}{\partial \mathbf{v}_1} &= \mathbf{C}_1(1)\mathbf{v}_2 + \mathbf{K}_{12}; \quad \frac{\partial \mathbf{R}_2}{\partial \mathbf{P}} = -\mathbf{Q}_2 \\ \frac{\partial \mathbf{R}_2}{\partial \mathbf{v}_2} &= \mathbf{C}_1(\mathbf{v}_1) + \mathbf{C}_2(\mathbf{v}_2) + \mathbf{C}_2(1)\mathbf{v}_2 + \mathbf{K}_{11} + 2\mathbf{K}_{22} \\ \frac{\partial \mathbf{R}_2}{\partial \mathbf{T}} &= \mathbf{B}_2; \quad \frac{\partial \mathbf{R}_3}{\partial \mathbf{v}_1} = -\mathbf{Q}_1^T; \quad \frac{\partial \mathbf{R}_3}{\partial \mathbf{v}_2} = -\mathbf{Q}_2^T; \quad \frac{\partial \mathbf{R}_3}{\partial \mathbf{P}} = \mathbf{0} \\ \frac{\partial \mathbf{R}_3}{\partial \mathbf{T}} &= \mathbf{0}; \quad \frac{\partial \mathbf{R}_4}{\partial \mathbf{v}_1} = \mathbf{D}_1(1)\mathbf{T}; \quad \frac{\partial \mathbf{R}_4}{\partial \mathbf{v}_2} = \mathbf{D}_2(1)\mathbf{T}; \quad \frac{\partial \mathbf{R}_4}{\partial \mathbf{P}} = \mathbf{0} \\ \frac{\partial \mathbf{R}_4}{\partial \mathbf{T}} &= \mathbf{D}_1(\mathbf{v}_1) + \mathbf{D}_2(\mathbf{v}_2) + \mathbf{L}_{11} + \mathbf{L}_{22}\end{aligned}\quad (5.7.4)$$

The three-dimensional formulation follows directly from the above definitions.

The advantages and disadvantages of Newton's method as outlined in Section 4.6.2 for the isothermal case hold equally for the convective heat transfer system. Also, the other variations of the fixed point schemes, such as Picard iteration, modified and quasi-Newton methods, and the various continuation techniques can be directly extended to the nonisothermal situation with comparable effectiveness.

5.7.3 Segregated Equation Methods

Segregated equation methods are not specifically solution algorithms but rather an approach to decomposing Eq. (5.3.14) (or similar systems) into a series of smaller problems that can be solved efficiently using direct matrix methods or are more amenable to PCG type methods. For convective heat transfer problems (or other transport problems) a particularly simple segregated or cyclic procedure is given by the following for the $(n+1)$ iteration step:

Step 1 (Energy Equation)

$$\mathbf{D}(\mathbf{v}^n, \mathbf{T}^n)\mathbf{T}^{n+1} + \mathbf{L}(\mathbf{T}^n)\mathbf{T}^{n+1} = \mathbf{G}(\mathbf{v}^n, \mathbf{T}^n) \quad (5.7.5)$$

Step 2 (Continuity and Momentum Equations)

$$-\mathbf{Q}^T\mathbf{v}^{n+1} = \mathbf{0} \quad (5.7.6)$$

$$\mathbf{C}(\mathbf{v}^n)\mathbf{v}^{n+1} + \mathbf{K}(\mathbf{T}^{n+1})\mathbf{v}^{n+1} - \mathbf{Q}\mathbf{P}^{n+1} = \mathbf{F}(\mathbf{T}^{n+1}) \quad (5.7.7)$$

The energy equation is typically solved first in each cycle, especially when considering free convection-dominated flows since the buoyancy term drives the fluid motion; the buoyancy term is included in \mathbf{F} in Eq. (5.7.7).

The procedure given in Eqs. (5.7.5)–(5.7.7), termed a cyclic Picard method, has some advantages over a combined equation method in terms of computer storage and execution time per iteration. Unfortunately, as with most variants of successive substitution, the procedure has a relatively slow rate of convergence; the convergence rate is further reduced due to the lagging of the coefficients induced by the splitting of the equations. This method has been used successfully for convection problems of moderate nonlinearity by several investigators [13–16].

Though the cyclic procedure described above offers some reduction in computational requirements, the combined momentum and continuity equations, especially in three dimensions, can still represent a very large matrix problem. Following the idea of segregating equations to its logical extreme [17], the system in (5.3.14) can be decomposed into the following series of equations:

$$(\mathbf{C}_1(\mathbf{v}) + \mathbf{K}_1)\mathbf{v}_1 - \mathbf{Q}_1\mathbf{P} = \mathbf{F}_1 \quad (5.7.8a)$$

$$(\mathbf{C}_2(\mathbf{v}) + \mathbf{K}_2)\mathbf{v}_2 - \mathbf{Q}_2\mathbf{P} = \mathbf{F}_2 \quad (5.7.8b)$$

$$\mathbf{Q}_1^T\mathbf{v}_1 + \mathbf{Q}_2^T\mathbf{v}_2 = 0 \quad (5.7.8c)$$

$$(\mathbf{D}(\mathbf{v}) + \mathbf{L})\mathbf{T} = \mathbf{G} \quad (5.7.8d)$$

for the case of two-dimensional, nonisothermal flow; the three-dimensional case follows the same procedure as does the inclusion of additional transport equations. The (\mathbf{C}, \mathbf{D}) , and (\mathbf{K}, \mathbf{L}) matrices, represent the advection and diffusion operators, respectively. A segregated solution approach to Eq. (5.7.8) would solve each individual momentum equation, the continuity equation, and the energy equation for the appropriate variable in a cyclic procedure similar to the one shown in Eqs. (5.7.5)–(5.7.7). As each variable is updated from a new solution, it is used where appropriate in the coefficients of the remaining equations, resulting in another type of cyclic Picard scheme. By reducing (5.3.14) to the series of equations in (5.7.8), a significant reduction in size and cost of the matrix problem is realized. The penalty to be paid is the slower convergence rate of the cyclic procedure. However, for very large problems there is a cross-over point for these competing effects and the segregated approach is clearly superior to standard combined equation methods.

A significant complication in the use of Eq. (5.7.8) arises due to the fact that there is no equation in the set from which the pressure may be directly obtained. The solution to this dilemma involves the construction of a pressure Poisson equation. Solving Eqs. (5.7.8a,b) symbolically for the velocities \mathbf{v}_1 and \mathbf{v}_2 and substituting this result into the incompressibility condition (5.7.8c) produces a consistent Poisson equation for the pressure

$$[\mathbf{Q}_1^T(\mathbf{C}_1(\mathbf{v}) + \mathbf{K}_1)^{-1}\mathbf{Q}_1 + \mathbf{Q}_2^T(\mathbf{C}_2(\mathbf{v}) + \mathbf{K}_2)^{-1}\mathbf{Q}_2]\mathbf{P} = \mathbf{F} \quad (5.7.9)$$

where the explicit form of \mathbf{F} is not important in this discussion. Equation (5.7.9), when used in place of the continuity equation, provides a complete set of equations

for all of the unknowns. A number of possible variations exist for the sequential solution of Eqs. (5.7.8) and (5.7.9), each of which produces a slightly different algorithm. In general, the Poisson equation given by (5.7.9) provides an initial estimate of the pressure field which is used in the pressure gradients of the momentum equations; velocity estimates are then obtained from each momentum equation in turn. Since the velocity field is not divergence free at this point, a second application of the Poisson equation is required; this solution supplies a pressure correction that allows the velocity fields to be adjusted to satisfy incompressibility. This general algorithm is analogous to the family of SIMPLE finite difference methods developed by Patankar [18]. Details regarding several variants of the method can be found in the work of Haroutunian et al. [17], who were the original developers of this technique for finite element applications. Note that another significant benefit in the usage of the segregated approach with the pressure Poisson equation is the applicability of iterative matrix solution methods for the individual equations in the sequence. The removal of the incompressibility constraint allows useful preconditioners to be found for each equation in the system. The combination of preconditioned, conjugate gradient (PCG) matrix methods with a segregated equation approach provides the most effective solution method presently available for large scale applications.

5.8 Convection with Change of Phase

There are two major difficulties in simulating melting and solidification change of phase for convection heat transfer problems: (1) accounting for the latent heat effect and (2) the application of fluid boundary conditions on the phase boundary. The latent heat problem has been detailed in the heat conduction section and its solution via the enthalpy method or any other method carries over directly to the convection problem. The boundary condition difficulty, however, is unique to the convection problem though it has some similarities with the free surface problems of Chapter 4.

As part of the formulation of the phase change problem in Section 1.11, it was noted that on the phase boundary the fluid velocity was assumed to be zero [see Eq. (1.11.1)]. This is a particularly difficult condition to apply in standard finite element formulations for two reasons. The location of the phase boundary is not known a priori, and even if it were known, the boundary does not generally coincide with a set of element edges (nodes) to which an essential boundary condition can be applied. One method to circumvent these difficulties is to explicitly track the motion of the phase boundary and allow the finite element mesh to adjust to this motion. Such mesh movement or remeshing schemes allow the phase boundary to always remain aligned with element edges and thus simplify the application of the no-slip boundary condition. However, these procedures are not without their shortcomings. They are expensive, complex to implement for general applications, and are generally limited in application to simple phase boundary motions and shapes. The moving mesh methods described in Section 4.11.3 are candidates for application in this type of phase boundary simulation.

A second, more approximate, method for imposing the no-slip boundary condition relies on the use of a temperature-dependent fluid viscosity and the allowed variation of material properties within each finite element. Depending on the type

of variable coefficient implementation that is used in the construction of element matrices, a variable viscosity is evaluated at the numerical quadrature points or nodal points within an element (see Section 3.9 for details). During each cycle of the solution process the temperature at the viscosity evaluation point is compared with the solidus (liquidus) temperature of the material. If the temperature is above the liquidus, the appropriate fluid viscosity is assigned at the evaluation point; for temperatures below the solidus temperature, the evaluation point viscosity is assigned a “large” value to “solidify” the fluid. Typically, the “solid viscosity” is set at four to five orders of magnitude larger than the normal fluid viscosity; the viscosity function is usually designed to take on a realistic, or at least convenient, variation between the liquidus and solidus temperatures (i.e., through the “mushy” zone). Thermal properties, such as the conductivity, can also be allowed to vary in some physically realistic manner across the phase change zone.

The viscosity evaluation method proposed here allows the phase boundary computation to be maintained on a subelement basis since a given element may contain both fluid and solid phases. However, this procedure is also similar to a smearing technique since a distinct phase boundary is never identified. This is in keeping with the spirit and implementation of the latent heat procedures described previously. This approach is also somewhat like the front tracking, free surface methods of Section 4.11.3. In the present case the front tracking variable is a physical variable, the temperature, rather than an artificial concentration variable [see Eq. (4.11.15)] or a volume fraction [see Eq. (4.11.17)]. The utility of the proposed approach is demonstrated in a subsequent section and in [19].

For materials that change phase over a temperature range, the simple viscosity variation method provides a first-order evaluation of the fluid mechanics and heat transfer processes through the phase change region. In some cases, a more accurate evaluation of the fluid and thermal fields can be achieved. For metal alloys, in which an interdendritic region forms between the liquidus and solidus temperatures, the so-called “mushy” zone is often characterized as a porous media. Careful study of the Darcy-Brinkman porous flow model in (1.6.2) shows that the Navier-Stokes equations can be made a subset of this model if an advective term is added to the porous flow equations. This has been theoretically justified by a number of investigators [20–22] and leads to a momentum equation of the form

$$\frac{\rho_0}{\phi} \frac{\partial v_j}{\partial t} + \frac{\rho_0}{\phi^2} v_j \frac{\partial v_i}{\partial x_j} + \left(\frac{\mu}{\kappa} \right) v_i = \frac{\partial}{\partial x_j} \left[-P \delta_{ij} + \mu_e \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \rho_0 f_i - \rho_0 g_i \beta (T - T_0) \quad (5.8.1)$$

where the Forchheimer term has been replaced by the advective term and the Brinkman viscosity has been equivalenced to the fluid viscosity. Recall that ϕ is the porosity or liquid volume fraction. This equation can be used throughout the fluid domain and within the mushy zone by proper adjustment of the material properties and coefficients as a function of temperature. In the liquid region ($T > T_{liq}$), the porosity is unity and the permeability κ is infinite; Eq. (5.8.1) is the standard Navier-Stokes equation. When the temperature drops below the solidus temperature ($T < T_{sol}$), the porosity and permeability are zero and the fluid viscosity is again very large, resulting in a “solidified” material. Between the liquidus and

solidus temperatures, the porosity goes from unity to zero as a function of temperature according to the particular model being employed [22]. A model for the permeability as a function of porosity transitions the permeability between a very large value and zero. The consequence of all these transitions is that the flow and heat transfer problem are smoothly moved from a viscous flow description to a porous media flow description to a solidified material description as a function of temperature. The model is substantially more detailed than the simple viscosity variation and is grounded in theory. Further details of the this approach and a demonstration of its capabilities are available in [20,22–24].

5.9 Convection with Enclosure Radiation

The inclusion of enclosure radiation effects in convective heat transfer problems involves the direct application of the methods from Section 3.8. By assumption, the fluid in the enclosure is nonparticipating and the radiative exchange therefore only influences the flux and temperature on the fluid boundaries. For purposes of computation the radiation problem is decoupled from the convection/conduction part of the problem and simply provides flux boundary conditions at each iteration or time step of the simulation.

5.10 Post-Computation of Heat Flux

The results of interest from a convective heat transfer analysis generally include fluid velocities, flow patterns, temperature distributions, and heat fluxes. Most of these items are directly available from the finite element results in terms of nodal point quantities. The heat flux (or flux of an auxiliary variable), however, being a derivative of the temperature, requires an auxiliary computation. Details of a method for performing this operation were given in Section 3.10. The simplest procedure for computing the components of the heat flux vector is based on the element level shape function for the temperature. That is, since

$$T(x_i, t) = \boldsymbol{\Theta}^T(x_i)\mathbf{T}(t) \quad (5.10.1)$$

then the flux components are

$$q_i(x_i, t) = -k \frac{\partial \boldsymbol{\Theta}^T}{\partial x_i} \mathbf{T}(t) \quad (5.10.2)$$

The relation in (5.10.2) is valid for any element and allows the flux to be evaluated at any time and at any point within the element. Note that in evaluating the derivative of $\boldsymbol{\Theta}$, the element geometry must be considered for mapped or parametric elements [see Eq. (3.3.3)]. Maximum accuracy in the shape function derivative is obtained by evaluating the derivative at the element numerical integration points (typically, the 2×2 Gauss points in a quadrilateral element) [25]. Since fluxes are normally required on element or domain boundaries, the integration point values are extrapolated to the nodes [26] and the discontinuous element level fluxes averaged between adjacent elements. Other evaluation points within an element have been employed, such as the points midway between nodes on the element boundary, or the Gauss points on

the element boundary. All of these methods provide results of acceptable accuracy provided the computational mesh is sufficiently refined. Note that a common feature of these procedures is the need to compute a unit normal on an element boundary if a total or integrated flux is to be derived. Methods for determining unique and consistent normals to element boundaries are discussed by Engleman, et al. [27].

A second, less familiar, approach for computing flux quantities relies on the consistent application of the weighted residual method. This idea was proposed by Larock and Herrmann [28] and Marshall, et al. [29], and refined by Gresho, et al. [30]. In essence, a weighted residual equation is derived that relates the boundary flux to local heat sources and computed temperature gradients. The consistent flux equation requires a matrix solution, which for many cases can be simplified by the use of a “lumping” procedure. Though somewhat more accurate than the simpler basis function derivative method, this procedure requires more computation, especially for the advection-diffusion equation. For details of this procedure, see [30].

It is important to recall that the above methods were described for the case of a conductive heat flux as would be computed for a solid boundary or a no flow-through surface. The heat flux computation for an open flow boundary would have to include terms describing the advective transport of energy. The appropriate flux description for this case would be

$$q_i(x_i, t) = \rho C_v v_i T - k \frac{\partial T}{\partial x_i} \quad (5.10.3)$$

and using the usual interpolations for velocity and temperature produces

$$q_i(x_i, t) = \rho C_v \boldsymbol{\Psi}^T \mathbf{v}_i(t) - k \frac{\partial \boldsymbol{\Theta}^T}{\partial x_i} \mathbf{T}(t) \quad (5.10.4)$$

the computational form for the total flux. Equation (5.10.4) would be evaluated in the same manner as the conductive flux in (5.10.2) with the same extrapolation and averaging techniques.

In Section 3.10.2, the heat flow function was defined for a conduction problem in two-dimensional geometries. The definition can be extended to convective transport though the restriction to two dimensions remains. Using the fluxes defined in (5.10.3), the heat function is defined by

$$q_x = \rho C_v v_x T - k \frac{\partial T}{\partial x} = \frac{\partial \mathcal{H}}{\partial y} \quad (5.10.5)$$

$$q_y = \rho C_v v_y T - k \frac{\partial T}{\partial y} = -\frac{\partial \mathcal{H}}{\partial x} \quad (5.10.6)$$

As before, the change in the heat function is an exact differential and

$$\delta \mathcal{H} = \int_A^B \mathbf{q} \cdot \mathbf{n} d\Gamma \quad (5.10.7)$$

where \mathbf{n} is the normal to the integration path. The computation of \mathcal{H} follows from the flux definition in terms of the interpolation functions and the definition of the

normal for the element. The details were given in Section 3.10.2 for both the planar and axisymmetric cases and will not be repeated here. It is worth noting again that the integration of (5.10.7) around the boundary of an element should be zero; the departure from zero indicates the error in the element energy balance.

5.11 Turbulent Heat Transfer

The previous sections have been limited to consideration of laminar flows though the extension to turbulent flows is of considerable engineering importance. This section continues and extends the discussion of Section 4.12 on isothermal turbulence modeling to the nonisothermal case. Most of the ideas and turbulence models for isothermal flows have a direct counterpart when the energy equation is included in the formulation. Here we will outline some of the most important concepts for the case of forced convection heat transfer; the theory for strongly coupled, buoyancy driven flows is somewhat less developed.

Extending the assumption given in Eqs. (4.12.1) and (4.12.2), the instantaneous temperature and velocities can be written as

$$\theta = \Theta + \theta', \quad v_i = V_i + v'_i \quad (5.11.1)$$

where Θ is the mean temperature, V_i is the mean velocity, and the primed quantities denote the fluctuations. When this definition and Eq. (4.12.1) are substituted into the instantaneous energy equation and ensemble averaged, the result is an equation of the form

$$\rho C \left(\frac{\partial \Theta}{\partial t} + V_j \frac{\partial \Theta}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left(k \frac{\partial \Theta}{\partial x_j} \right) - \rho C \overline{v'_j \theta'} \quad (5.11.2)$$

The averaging process has introduced the three new variables $\overline{v'_j \theta'}$ which are the components of the Reynolds heat flux vector. These quantities can be modeled in a variety of ways, all of which are analogous to the methods used for the Reynolds stresses. The current status of turbulent heat transfer modeling is reviewed by Launder [31]. Considering only one point closure models (see Section 4.12.4), the Reynolds heat flux can be related to the mean temperature field through a Boussinesq type hypothesis. That is, let

$$-\rho C \overline{v'_j \theta'} = \frac{k_T}{C} \frac{\partial \Theta}{\partial x_j} = \frac{\mu_T}{Pr_T} \frac{\partial \Theta}{\partial x_j} \quad (5.11.3)$$

which can be used in (5.11.2) to produce

$$\rho C \left(\frac{\partial \Theta}{\partial t} + V_j \frac{\partial \Theta}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left[\left(k + \frac{\mu_T}{Pr_T} \right) \frac{\partial \Theta}{\partial x_j} \right] \quad (5.11.4)$$

where k_T is the eddy conductivity and Pr_T is the turbulent Prandtl number. Once the form of the eddy conductivity or the turbulent Prandtl number is specified, then the energy equation in (5.11.4) can be solved in conjunction with the flow equations.

Methods for specifying the eddy conductivity or turbulent Prandtl number parallel the ideas for the eddy viscosity specification. Of all the possible approaches, one of the most popular is the specification of Pr_T as an empirical relation. Reynolds

[32] describes approximately thirty different formulas for Pr_T , all of which have a limited applicability. Clearly, the modeling of turbulent heat transfer processes remains an active area of research, though its progress is directly related to progress made in turbulent flow modeling. It should also be noted that Variational Multi-scale (VMS) models for nonisothermal turbulence could be generated following the ideas of Section 4.12.6.

5.12 Chemically Reacting Systems

5.12.1 Preliminary Comments

The equations describing a reactive material were presented in Section 1.9. The incorporation of chemical kinetics into solid body heat conduction problems was outlined in Section 3.11.5. The complexity of the problem, where general chemical reactions are included with a nonisothermal flow problem, places this topic outside the primary scope of this text. In the following section a brief discussion of reactive flow modeling is provided to simply indicate the scope of the modeling problem. Readers interested in this topic should consult the specialized texts by Oran and Boris [33] or Kee, et al. [34].

5.12.2 Finite Element Modeling of Chemical Reactions

The development of a finite element model for chemically reactive flows follows the usual prescription. The continuity, momentum, and energy equations for the mixture are standard and were considered in Chapter 4 and previous sections of the present chapter. The conservation equations for the species are of the advection-diffusion type and can therefore be treated by the same methods as used for the energy equation. Recall that the auxiliary equations defined in Chapter 1 could be used directly as species equations. The discretized form of the species equations provides an additional set of I matrix equations that can, in principle, be added to the matrix system for the basic nonisothermal flow problem.

Though the development of the reactive flow model appears to be straightforward, the solution of the resulting equations is extremely difficult. First, from the standpoint of problem size, the addition of multistep chemistry leads to a very rapid growth in the number of degrees of freedom considered in any given problem. However, problem size alone is not the most significant difficulty. The equation set is highly coupled and very nonlinear, making the convergence by standard solution methods unpredictable. More importantly, the length and time scales for the chemical processes tend to be significantly different (smaller) than the scales for the fluid and thermal response. This disparity in scales makes the use of fully coupled solution procedures impractical. In mathematical terms, the equation system becomes very stiff. The most effective solution algorithms allow the flow problem to run on its natural time scale, while the chemistry solution is decoupled and run as a subcycled process on a shorter time scale. Obviously, some of the rigorous coupling between variables is destroyed by this technique and variable coefficients must be lagged or predicted and corrected. Despite the drawbacks, this type of approach is the most widely used for problems with “difficult” chemistry.

Chemical reactions also tend to produce large gradients in the dependent variables over small distances. Since the location, extent, and movement of such reaction

zones is not known a priori, excessively refined meshes are often employed to ensure spatial accuracy. Again, this leads to a large problem size and excessive computer resources. A more efficient solution to the length scale problem is the use of adaptive meshes. An example of this technique for a reactive problem can be found in reference [35].

5.13 Numerical Examples

5.13.1 Preliminary Comments

The numerical examples selected for this section are intended to illustrate the performance of the finite element models presented for convective heat transfer problems. Again, the problems are not examined in full engineering detail, but are sufficiently complete to demonstrate the flexibility and capability of the recommended algorithms. All of the examples were solved using the codes NACHOS II [36], FIDAP [37] or KACHINA [38] with the mixed finite element model, unless otherwise stated. Numerical examples of natural convection problems were solved by the penalty finite element model (see Pelletier, Reddy, and Schetz [39]).

5.13.2 Concentric Tube Flow

The first example we consider is the forced convection/conduction problem of flow in a concentric tube. The problem schematic is shown in Figure 5.13.1 along with the finite element mesh of eight-noded elements and boundary conditions. The fluid in the inner tube is a medium-weight oil and the counter-flowing cooling fluid is water. The central tube is made of copper and the outside concentric tube is made of steel. All material properties are assumed to be constant.

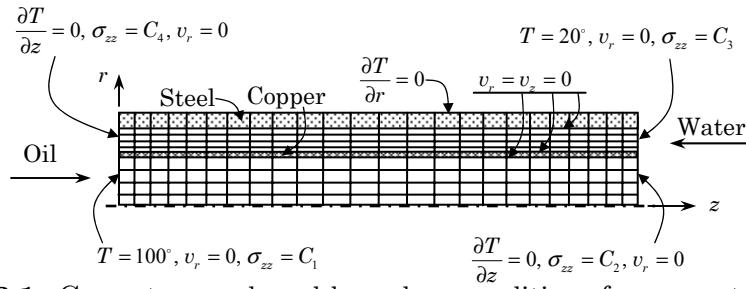


Figure 5.13.1: Geometry, mesh and boundary conditions for concentric tube flow.

As is typical for internal flow problems, flux type boundary conditions are used at the outflow boundary for both fluid regions to allow a smooth solution field without an excessive tube length. Also, in this case a specified traction boundary condition was specified at the inflow boundary; the difference in the normal tractions across the tube length produce the driving force for the fluid. Note that the nonlinear terms are identically zero in the fluid and thermal equations in this case, and therefore the problem can be solved without iteration.

The solution to the hydrodynamic part of the problem consists of a fully developed Poiseuille flow for the oil and flow in a concentric annulus for the water. Both computed profiles are essentially exact for this model. Three cases were computed in which the oil Peclet number ($Pe = \mu C_p/k$) was varied by changing the pressure drop imposed on the tube. A typical isotherm plot is shown in Figure 5.13.2. The

computed temperature profiles along the inside of the copper tube are shown in Figure 5.13.3. Further details on this analysis and other conjugate tube flow problems are available in [40].

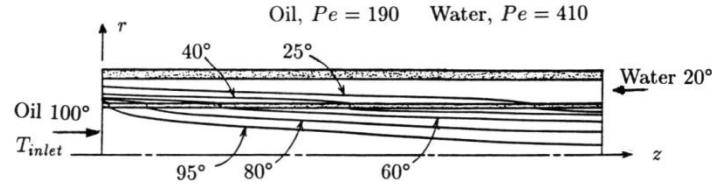


Figure 5.13.2: Isotherms for the concentric tube flow problem.

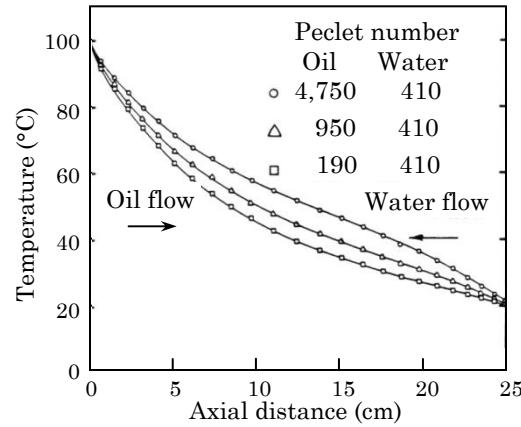


Figure 5.13.3: Axial temperature profiles for the concentric tube flow problem.

5.13.3 Tube Flow with Change of Phase

Consider the problem of a heated fluid flowing through a cold tube, the temperature of which is below the fluid solidification temperature. The problem geometry is shown in Figure 5.13.4. The fluid within the tube is initially quiescent with a uniform temperature equal to the inlet temperature. Fluid motion is initiated by imposing a specified pressure drop on the tube. The methods described in Sections 3.3 and 4.7 were used to model the latent heat release and the immobilization of the fluid at the phase boundary [19].

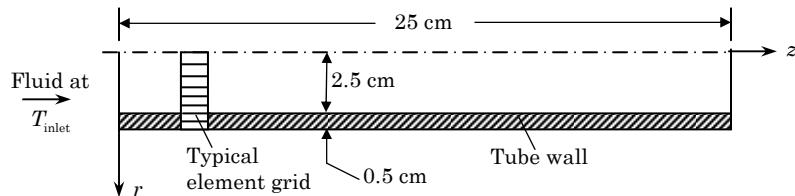


Figure 5.13.4: Schematic for tube flow with change of phase.

The initial temperature response of the fluid is due solely to conduction at the tube wall. The large initial heat loss to the tube cools the fluid below the solidus temperature, producing a solid crust on the tube wall. As the fluid velocity increases, convection along the length of the tube becomes important.

Figure 5.13.5 shows the result of a computation in which the tube eventually becomes obstructed due to the crust formation and growth. Each frame shows the phase change isotherm which corresponds to the phase boundary.

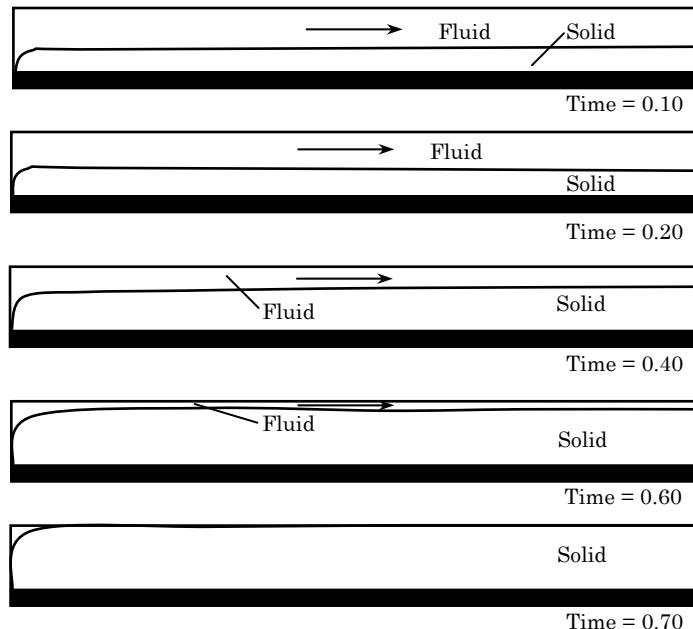


Figure 5.13.5: Motion of phase boundary for impulsively started tube flow.

Figure 5.13.6 shows velocity and temperature profiles at a station four tube diameters from the inlet. The velocity profiles show the effects of initial fluid acceleration, reduction in tube diameter due to crust growth, and finally the reduction in flow rate due to increasing viscosity.

5.13.4 Heated Cavity – Boussinesq Model

A standard benchmark problem for natural convection algorithms consists of a square, closed cavity with insulated top and bottom faces and differentially heated vertical boundaries. For the present case, the contained fluid is a gas with constant material properties. A nondimensional temperature difference of $\epsilon = (T_H - T_C)/(T_H + T_C)$ is taken. For this case, it is set to $\epsilon = 0.005$. For this small value of the temperature difference, the flow is well within the Boussinesq limit. The boundary conditions and nondimensional parameters are shown in Figure 5.13.7a. Since the velocity field is specified on all boundaries of the domain, the pressure field is determined only to within a constant value; the pressure level within the cavity is set by specifying a single pressure value at one node on the cavity boundary. A typical mesh consisting of eight-node elements is also shown in Figure 5.13.7b.

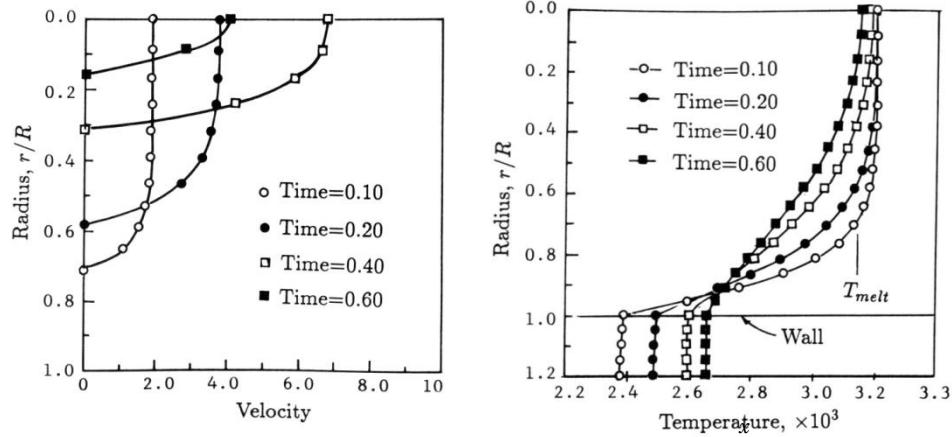


Figure 5.13.6: Velocity and temperature profiles for tube flow, $z/R = 8$.

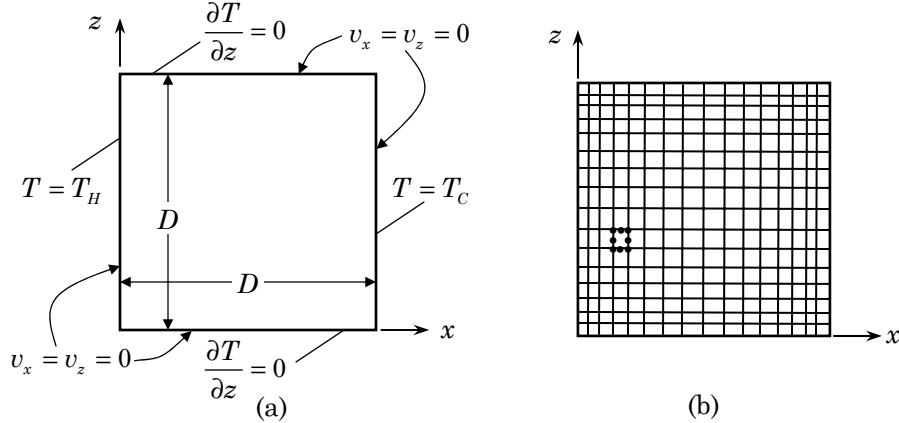


Figure 5.13.7: Schematic for natural convection in a cavity.

The cavity problem was initially solved using the cyclic Picard scheme employed in the earliest versions of the NACHOS code. More recent solutions have been obtained using Newton's method for the combined equation set and zeroth order continuation through the Rayleigh number [$Ra = \beta g(T_H - T_C)D^3/\nu\kappa$] range of interest. Results for this problem in the form of streamline and isotherm plots are shown in Figures 5.13.8 and 5.13.9 for Rayleigh numbers of $Ra = 10^4$ and 10^6 , respectively ($Pr = \nu/\kappa = 0.71$). For the lower Rayleigh number, the flow is relatively weak and the thermal field is only slightly perturbed from a conduction solution. At the higher Rayleigh number, the flow field develops a considerable structure while the thermal field becomes vertically stratified in the core of the cavity with high heat flux regions along the vertical boundaries. As required by the form of the boundary conditions and boundary value problem, the computed solutions are exactly centro-symmetric as seen in the contour plots. A detailed comparison of the present solutions with a large number of other numerical simulations is contained in [41].

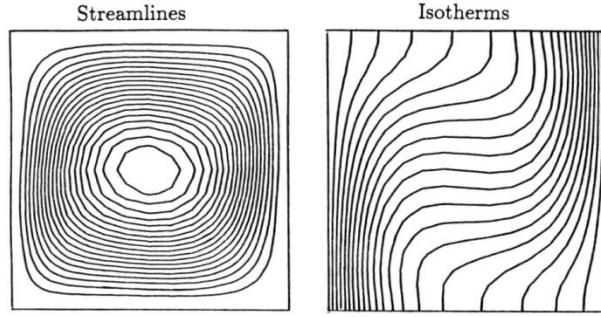


Figure 5.13.8: Streamlines and isotherms for natural convection in a square cavity, Boussinesq model ($Ra = 10^4, \epsilon = 0.005$).

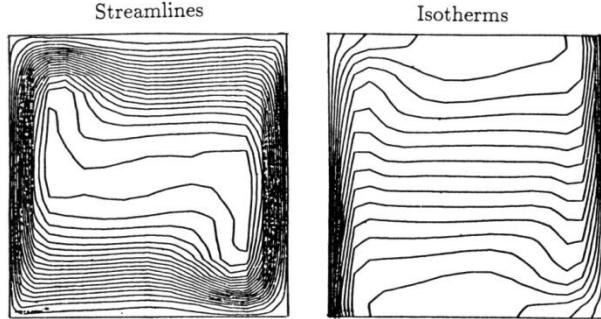


Figure 5.13.9: Streamlines and isotherms for natural convection in a square cavity, Boussinesq model ($Ra = 10^6, \epsilon = 0.005$).

5.13.5 Heated Cavity – Acoustically Filtered Model

The heated cavity flow may also be used for evaluating methods for non-Boussinesq formulations. In the present case the nondimensional temperature difference is increased to $\epsilon = 0.6$ which is outside the Boussinesq range. The flow problem in the previous section is again considered with the same Rayleigh numbers of $Ra = 10^4$ and 10^6 . For this simulation, nine-node elements with a discontinuous pressure approximation were used on a series of regular mesh refinements. Shown in Figures 5.13.10 and 5.13.11 are the acoustically filtered results in terms of the stream function and temperature fields. When compared to the previous Boussinesq results, it is immediately apparent that no symmetries are present. Quantitative comparisons of the heat transfer rates (Nusselt number) have been made with various other simulations in [4] and found to be in excellent agreement. Full Newton iteration was used to compute the acoustically filtered results and the number of iterations for convergence was higher than the comparable Boussinesq case. Although not shown here, a fully compressible fluid model has also been computed and compared to the acoustically filtered result. There is no significant difference between the results which indicates the validity of the acoustically filtered formulation for low-speed, compressible flows.

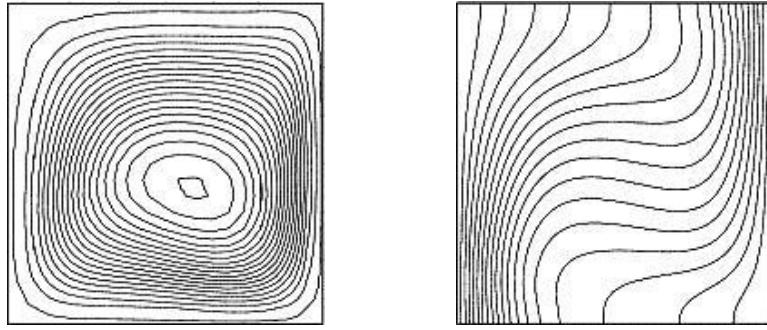


Figure 5.13.10: Streamlines and isotherms for natural convection in a square cavity, Acoustically filtered model ($Ra = 10^4, \epsilon = 0.6$).

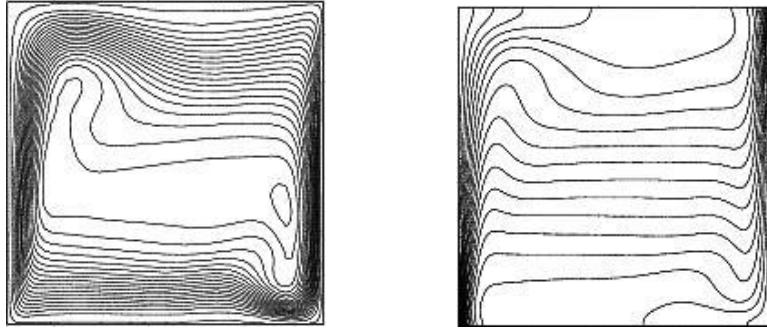


Figure 5.13.11: Streamlines and isotherms for natural convection in a square cavity, Acoustically filtered model ($Ra = 10^6, \epsilon = 0.6$).

5.13.6 Solar Receiver

The schematic shown in Figure 5.13.12 represents a cross section of an annular solar receiver tube surrounded by an eccentrically located glass envelope. The inner tube carries a heat transfer fluid that is heated by a flux that varies with position around the tube; the incident flux is due to solar energy being concentrated on the tube by a parabolic trough collector. The glass envelope provides a shield to reduce the forced convection (wind) heat loss from the collector tube. Of interest in this problem is the prediction of heat loss from the inner tube due to natural convection in the annular space. Parameters such as inner tube temperature distribution, working fluid in the annulus, annular gap, and eccentricity of the envelope were varied to assess their importance on overall heat loss.

Figures 5.13.13 through 5.13.16 contain streamline and isotherm plots for an air-filled annulus for various temperature and geometric configurations. The flow pattern and heat flux distribution is quite sensitive to variations in these parameters even though the Rayleigh number is the same for all cases. The computed heat transfer results for these cases are compared with experimental data in Figure 5.13.17.

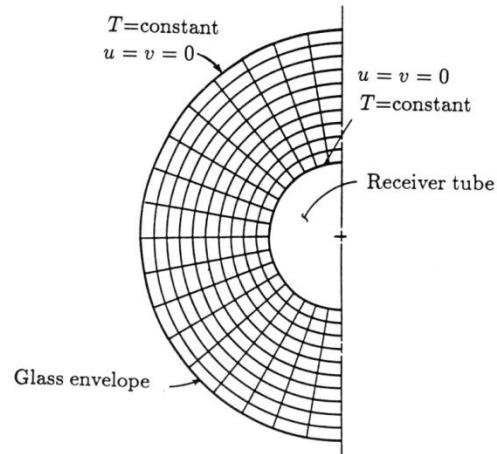


Figure 5.13.12: Mesh and boundary conditions for the annular solar receiver.

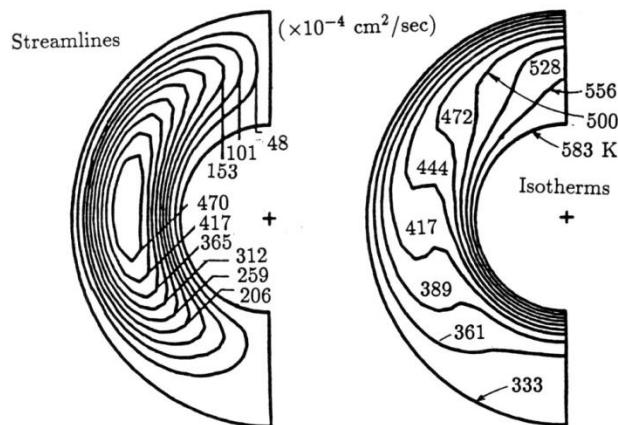


Figure 5.13.13: Streamline and isotherm plots for the solar receiver; *uniform wall temperature*, $Ra = 1.2 \times 10^4$.

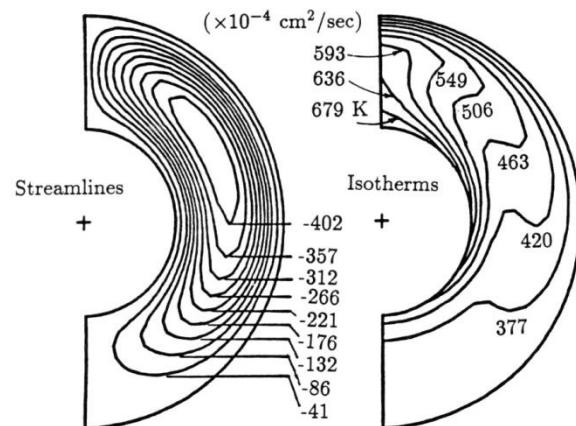


Figure 5.13.14: Streamline and isotherm plots for the solar receiver; *asymmetric wall temperature, hot on top* ($Ra = 1.2 \times 10^4$).

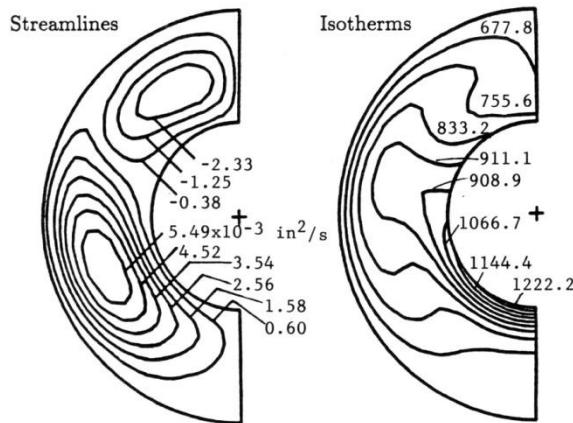


Figure 5.13.15: Streamline and isotherm plots for the solar receiver; uniform wall temperature, *hot on bottom* ($Ra = 1.2 \times 10^4$).

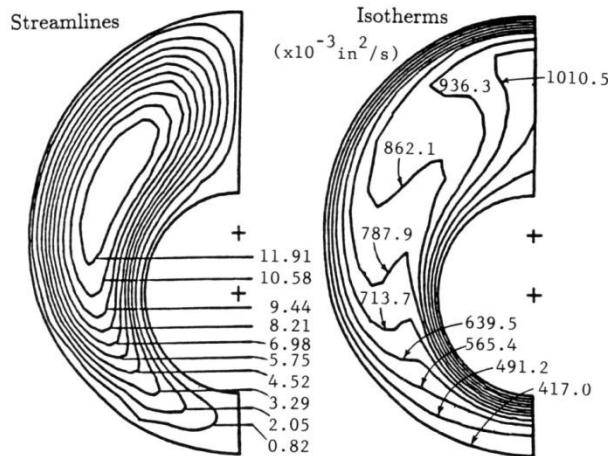


Figure 5.13.16: Streamline and isotherm plots for the solar receiver; uniform wall temperature, *eccentric geometry* ($Ra = 1.2 \times 10^4$).

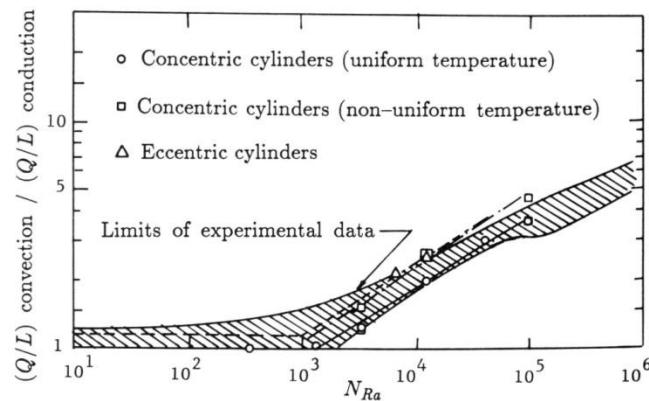


Figure 5.13.17: Heat loss by natural convection in annular spaces of a solar receiver. Computed points are shown as symbols.

5.13.7 Tube Bundle

The natural convection heat transfer from three tubes contained within a larger diameter conduit is modeled with the finite element grids shown in Figure 5.13.18. The geometry is symmetric, allowing the simulation to be performed on one-half of the conduit domain. The interior tubes have constant, but different wall temperatures; the exterior of the conduit is held at a temperature that is lower than the interior tubes. The two meshes shown in Figure 5.13.18 represent two very different mesh generation techniques. The more structured mesh was developed in three sections using a standard mapping technique. The completely unstructured quadrilateral mesh was produced using a method called “paving” [42] which works with the entire region at one time. The only input to the paving algorithm, other than a boundary description, is an element size along each boundary.

Isotherm and streamline plots for two cases are shown in Figures 5.13.19 and 5.13.20. Figure 5.13.19 illustrates the flow produced when a small temperature difference ($\Delta T_{max} = 10$) is maintained between the hottest interior tube and the exterior boundary. The second case shows the result when the temperature difference from the first case is doubled (i.e., $\Delta T_{max} = 20$). Solutions for both cases were obtained using a solution strategy with several Picard iterations followed by a Newton procedure. The computed flow fields show a characteristic plume rising from the lower cylinder with an overall circulation within the larger conduit.

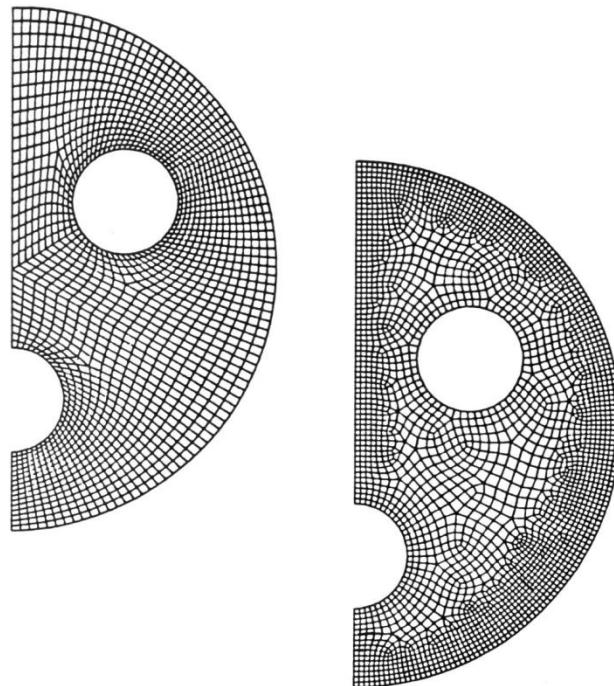


Figure 5.13.18: Finite element meshes for tube bundle geometry.

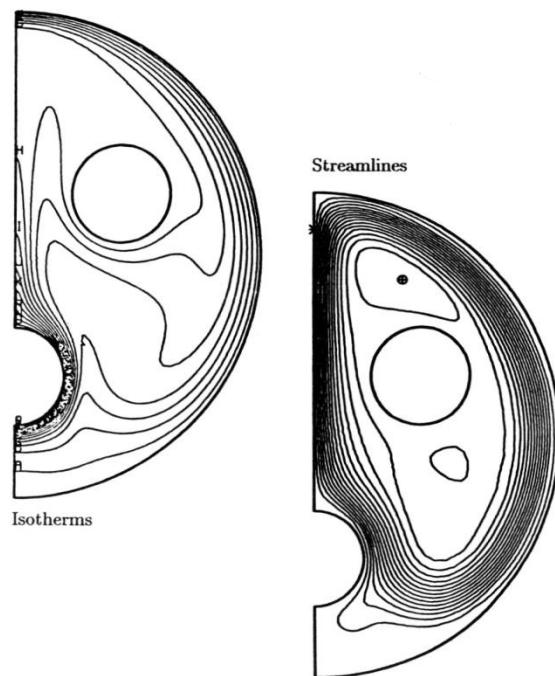


Figure 5.13.19: Contour plots for natural convection in a tube bundle, $\Delta T_{max} = 10$.

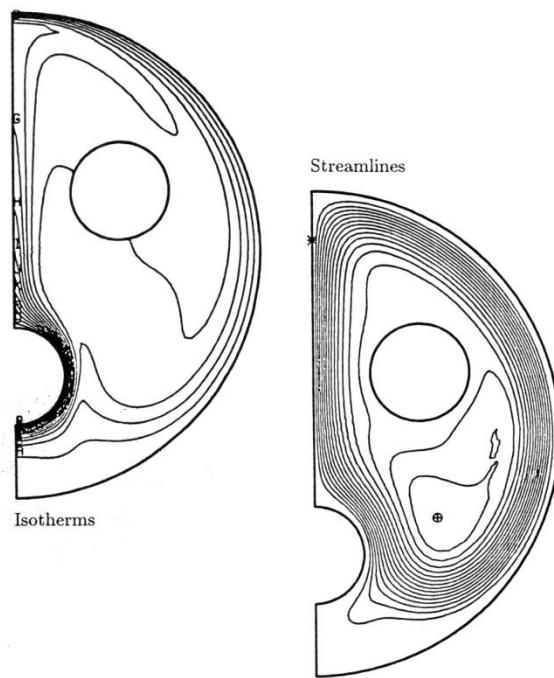


Figure 5.13.20: Contour plots for natural convection in a tube bundle, $\Delta T_{max} = 20$.

5.13.8 Volumetrically Heated Fluid

This example consists of a volumetrically heated fluid contained in an isothermal circular cylindrical container. Many of these types of flows are associated with heat transfer from radioactive gases or liquids. A quadrant of the domain, boundary conditions, and finite element mesh are shown in Figure 5.13.21. At time zero, the initially quiescent, isothermal fluid is heated volumetrically at a uniform rate. The transient solution is obtained by an implicit integration scheme using the trapezoid rule with quasi-linearization.

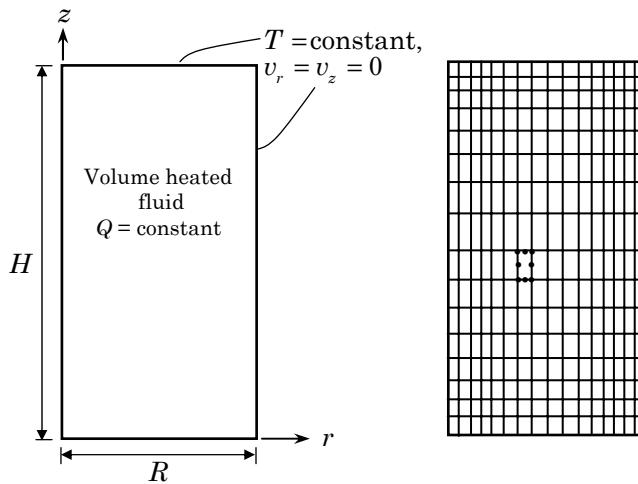


Figure 5.13.21: Computational domain, boundary conditions, and mesh used for volumetrically heated fluid in a circular cylinder.

The series of streamline and isotherm plots in Figure 5.13.22 shows the development of the flow at a moderate modified Grashof number, $Gr = g\beta l^5 Q/\nu^2 k = 2 \times 10^5$. A time history of the maximum stream function shown in Figure 5.13.23 shows the overshoot that occurs during the initial transient. The stream function vs. time plot of Figure 5.13.24 shows the interaction between the two cells at a slightly higher Grashof number, $Gr = 4 \times 10^5$. A further series of plots in Figures 5.13.25 and 5.13.26 illustrate the additional dynamics present in the flow at Grashof number, $Gr = 4 \times 10^5$. The development of a secondary cell that undergoes a damped oscillation is quite evident. The development of secondary cells is common in volumetrically heated fluids. Further details of this analysis are available in [43].

5.13.9 Porous/Fluid Layer

A simple problem that illustrates the use of the saturated porous media formulation in conjunction with an incompressible, viscous flow is shown in Figure 5.13.27. A rectangular geometry is divided vertically with a porous material occupying the right half of the cavity. The top and bottom boundaries of the cavity are insulated while the vertical boundaries are held at uniform but different temperatures. The steady natural convection problem in the cavity was modeled using a Brinkman model for flow in the porous medium. Newton's method was used to solve the flow problem.

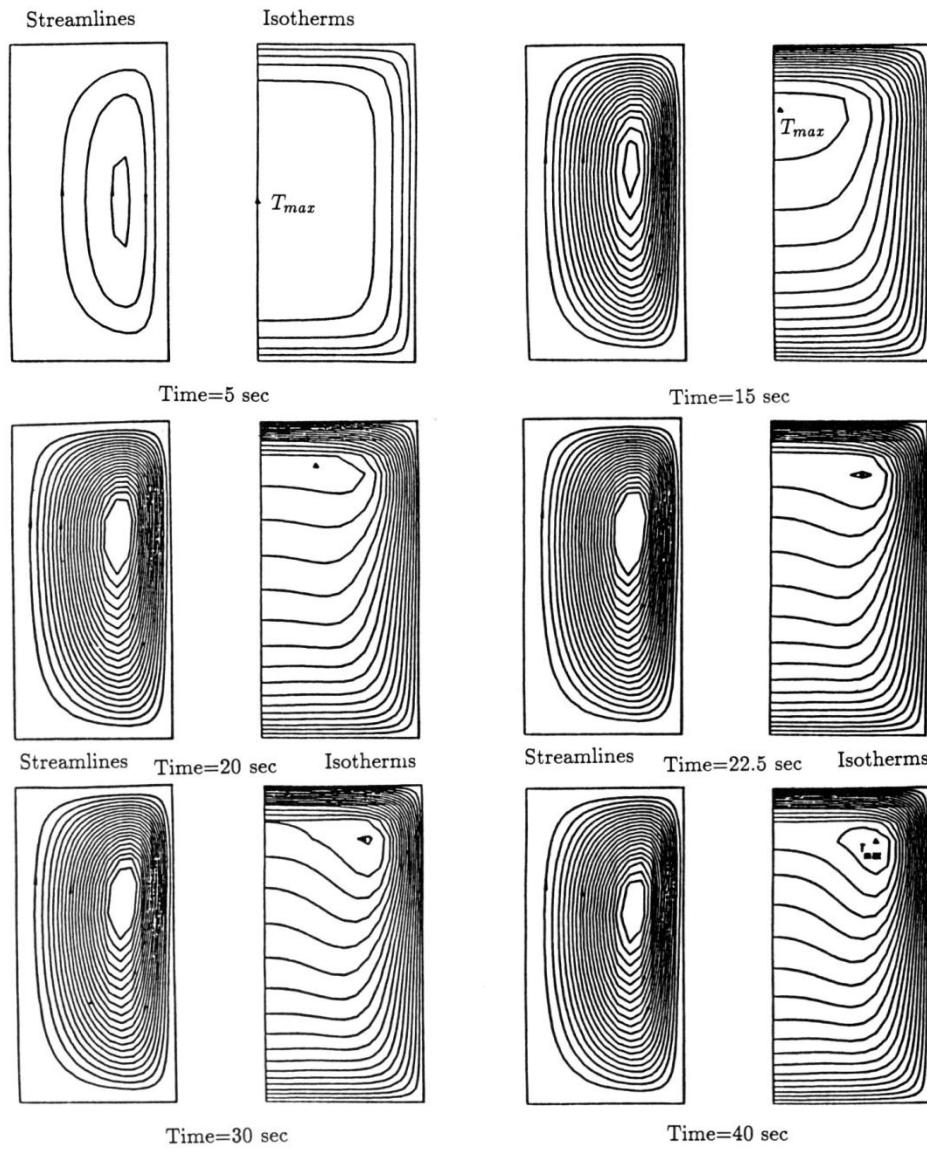


Figure 5.13.22: Streamlines and isotherms for volumetrically heated fluid ($Gr = 2 \times 10^5$).

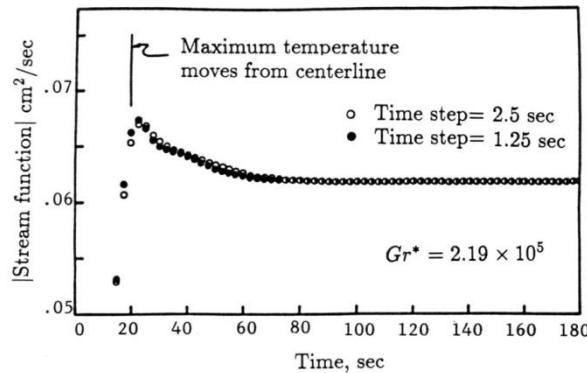


Figure 5.13.23: Stream function vs. time for volumetrically heated fluid ($Gr = 2 \times 10^5$).

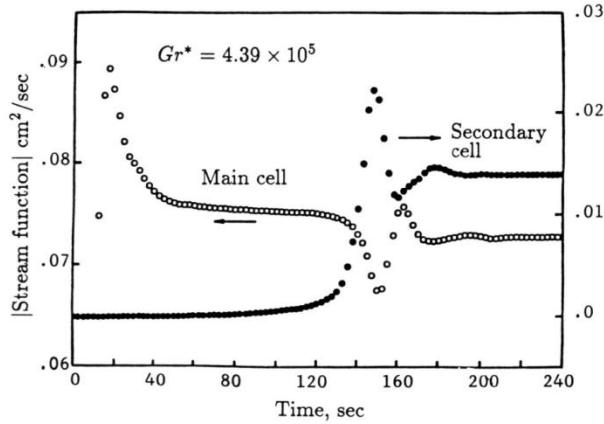


Figure 5.13.24: Stream function vs. time for volumetrically heated fluid ($Gr = 4 \times 10^5$).

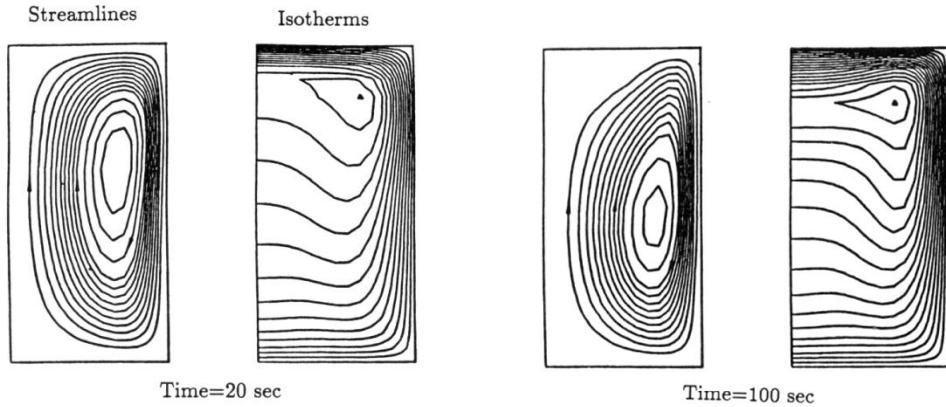


Figure 5.13.25: Streamlines and isotherms for volumetrically heated fluid ($Gr = 4 \times 10^5$).

For small values of the Darcy number (or permeability), $Da = \sqrt{k}/L$, the dominant fluid motion is confined to the fluid layer with little circulation occurring in the porous layer. At higher Darcy numbers, the fluid motion spans the cavity, although the velocities in the porous layer are still much smaller than in the bulk fluid. Representative stream function and isotherm plots are shown in Figures 5.8.28 and 5.8.29 for extremes in the Darcy number. Further discussion on the modeling of conjugate problems and the use of various porous/layer/bulk fluid interface conditions can be found in [22] and [44].

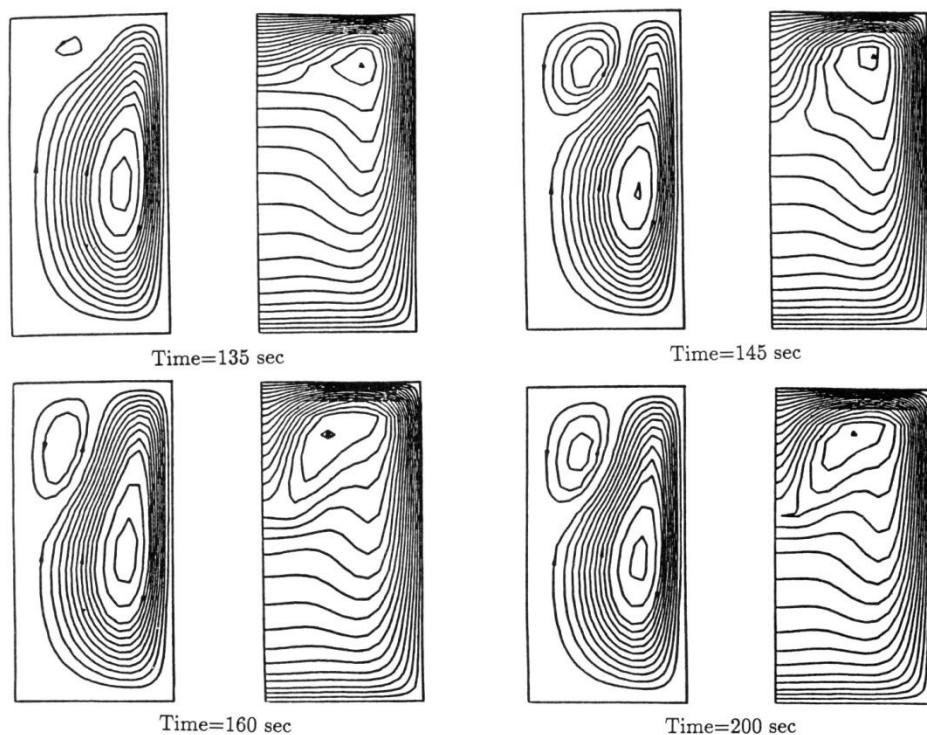


Figure 5.13.26: Streamlines and isotherms for volumetrically heated fluid ($Gr = 4 \times 10^5$).

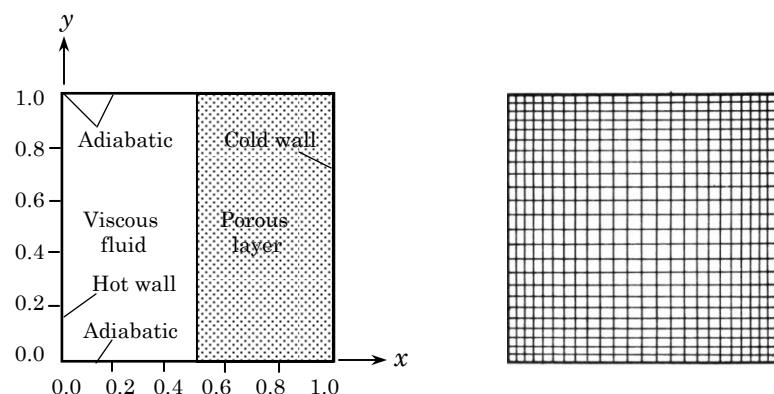


Figure 5.13.27: Schematic and mesh for a conjugate, porous/fluid layer problem.

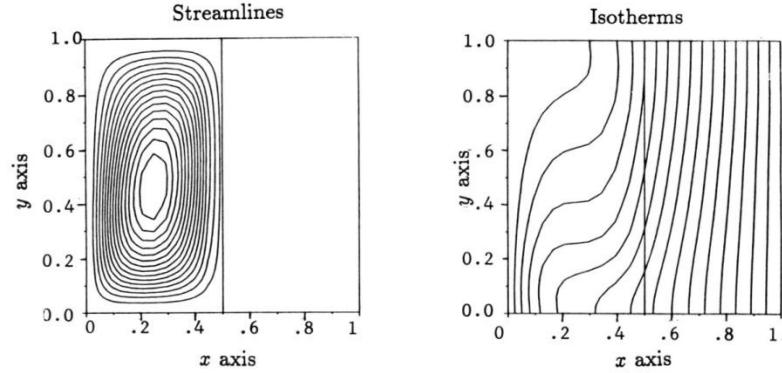


Figure 5.13.28: Streamlines and isotherms for a conjugate, porous/fluid layer ($Ra = 10^5$, $Da = 10^{-5}$).

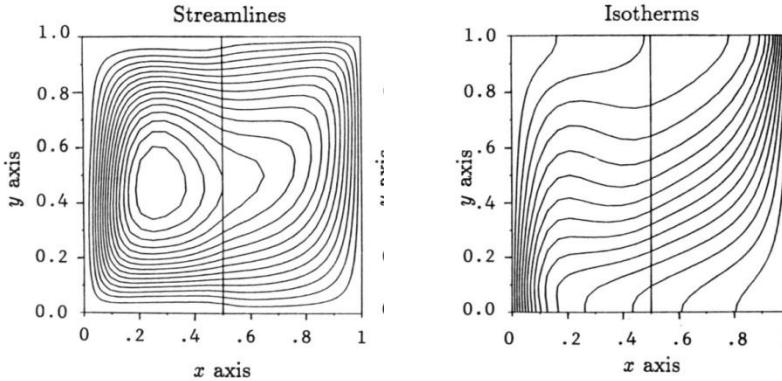


Figure 5.13.29: Streamlines and isotherms for a conjugate, porous/fluid layer ($Ra = 10^5$, $Da = 10^{-3}$).

5.13.10 Curing of an Epoxy

This example demonstrates the use of both the phase change procedures and the auxiliary equation formulation to predict the gelation behavior of an epoxy. The geometry for the simulation is a simple rectangular crucible that initially contains a well-stirred mixture of epoxy resin and curing agent (see Figure 5.13.30). The crucible is held in a constant-temperature oven during the course of the curing process. Since the curing reaction is exothermic, the volumetric heat addition to the fluid is given by

$$Q = \rho \Delta H \frac{D\alpha}{Dt} = \rho (C_p^p - C_p^r) T \frac{D\alpha}{Dt} \quad (5.13.1)$$

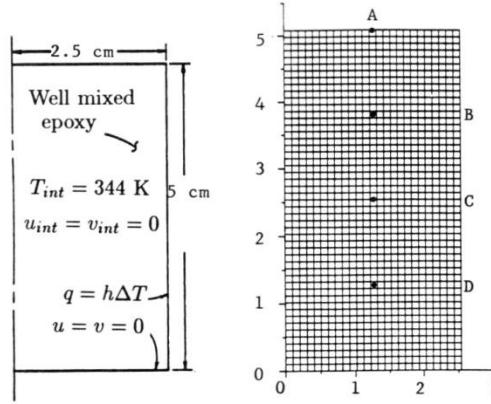


Figure 5.13.30: Schematic and mesh for epoxy curing simulation.

where ΔH is the change in enthalpy of the reacting material due to its change in state, and the superscripts p and r denote the products and reactants, respectively. The variable α is an internal state variable that describes the extent of the gelation reaction; α can be interpreted as the ratio of the mass of reacted material (per unit volume) to the total mass of material (per unit volume). Therefore, α varies between 0 (unreacted) to 1 (fully reacted) with $\alpha = 0.6$ indicating the point of gelation (solidification). The time evolution of the extent of reaction variable, α , is provided by an advection-diffusion equation

$$\frac{\partial \alpha}{\partial t} + v_j \frac{\partial \alpha}{\partial x_j} = \frac{\partial}{\partial x_j} \left(D \frac{\partial \alpha}{\partial x_j} \right) + R \quad (5.13.2)$$

where R is the reaction rate. For this particular material the reaction rate is often assumed to be of second-order with Arrhenius rate constants

$$R = [A_1 \exp(-E_1/\mathcal{R}T) + \alpha A_2 \exp(-E_2/\mathcal{R}T)] (1 - \alpha) \quad (5.13.3)$$

where A_1, A_2 are pre-exponential factors, E_1, E_2 are activation energies, and \mathcal{R} is the gas constant.

Equations (5.13.1) through (5.13.3) must be coupled to the standard nonisothermal, Navier-Stokes equations to describe the entire curing process. In addition, material properties, which vary with the extent of the reaction and the temperature, must be specified. The present problem was solved using the adaptive time step version of the trapezoid rule. A finite element model with quadratic interpolation of the velocity, temperature, and extent of the reaction, and a discontinuous linear approximation of pressure is used [45]. A time history plot of the adaptive time step employed for the analysis is shown in Figure 5.13.31. It is quite evident that the size of the time step reflects the changing complexity of the flow problem. Figures 5.13.32 through 5.13.35 show contour plots of the field variables at four different times during the curing process. As with most volumetrically heated fluids, multiple convection cells are predicted during the low heat release part of the process. When the exothermic reaction begins to accelerate, gelation occurs first at the top of the crucible and proceeds downward as a planar front. A comparison of the velocity of the gel front with the limited experimental data available showed good agreement, considering the uncertainty in material properties.

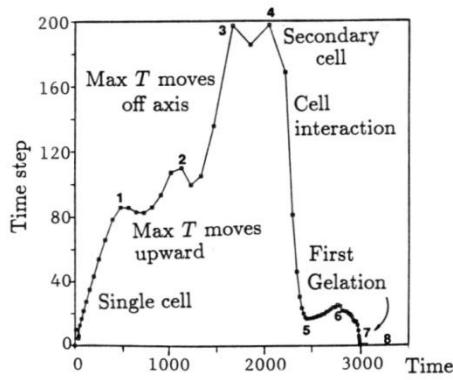


Figure 5.13.31: Time step history for epoxy curing simulation.

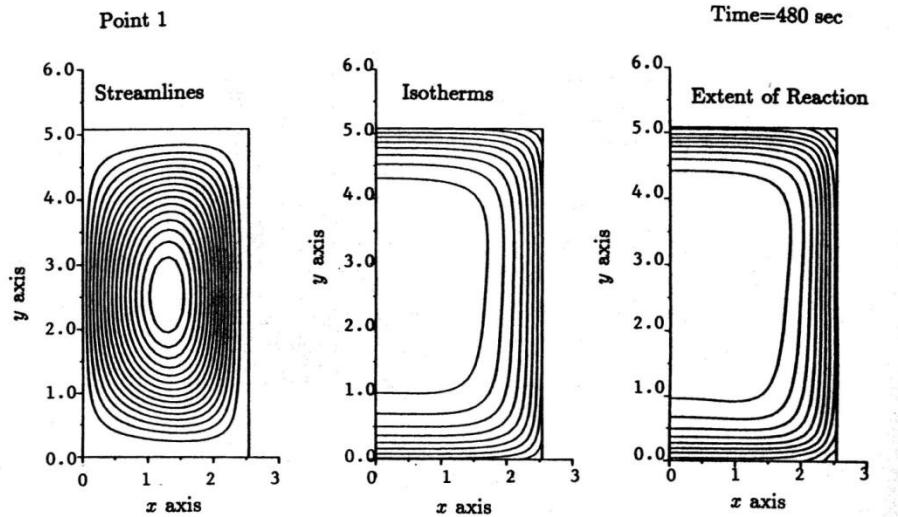


Figure 5.13.32: Contour plots for epoxy curing simulation, time = 480 sec.

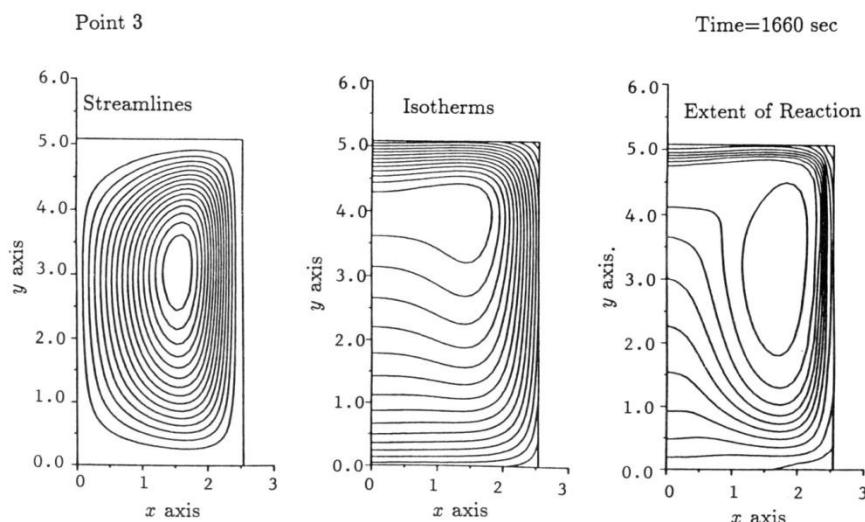


Figure 5.13.33: Contour plots for epoxy curing simulation, time = 1660 sec.

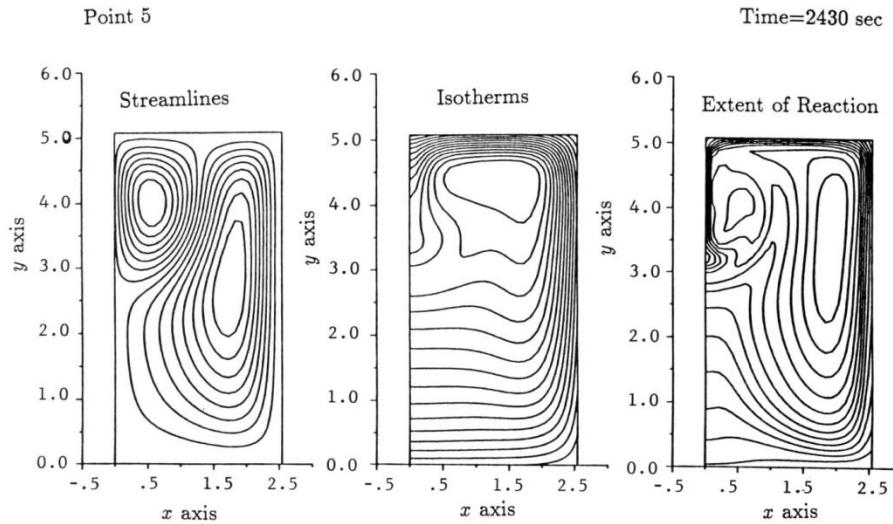


Figure 5.13.34: Contour plots for epoxy curing simulation, time = 2,340 sec.

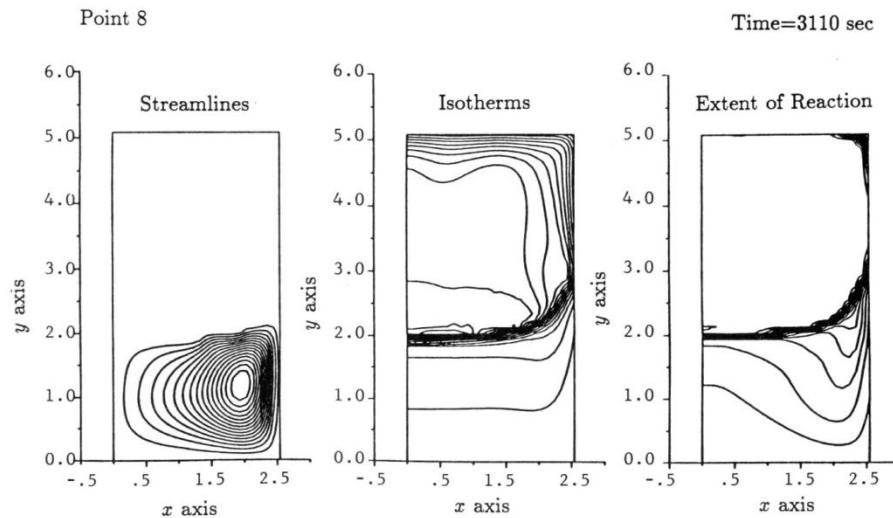


Figure 5.13.35: Contour plots for epoxy curing simulation, time = 3,110 sec.

5.13.11 Heated Channel

The final problem provides an example of the differences between the Boussinesq and acoustically filtered formulations for a time dependent flow. The problem consists of a two-dimensional channel with a heated section in the floor of the channel, as shown in Figure 5.13.36. A fully developed isothermal flow is specified as an initial condition and the transient behavior is produced by the time dependent heating of the floor section. The temperature of the heated section is increased with a linear ramp up to the final steady temperature that is higher than the temperature at the channel inlet. The problem is cast in a nondimensional form with a Reynolds number of $Re = 10$ and a Richardson number of $Ri = Gr/Re^2 = 500$, which indicates the relative importance of buoyancy and inertia forces. Using a perfect gas model, the problem was solved with both the Boussinesq and acoustically filtered equations.

Higher-order elements in a refined mesh were used for the model and an adaptive time stepping, predictor-corrector integration method was employed.

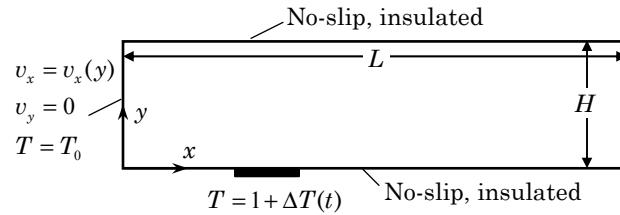


Figure 5.13.36: Schematic of heated channel flow.

Shown in Figure 5.13.37 are a series of isotherm plots for the two formulations. The differences in the fields are clear. A more significant difference in the behavior of the two models can be seen in Figure 5.13.38 where the normalized mass flux at the channel exit is plotted as a function of nondimensional time. The Boussinesq model shows a constant unit efflux while the acoustically filtered model shows an increase in the mass flux while the fluid is heated, followed by a decrease to a unit value at steady state. The differences in the two acoustically filtered curves are due to the use of constant or variable thermophysical properties. Full details of this simulation are available in [4].

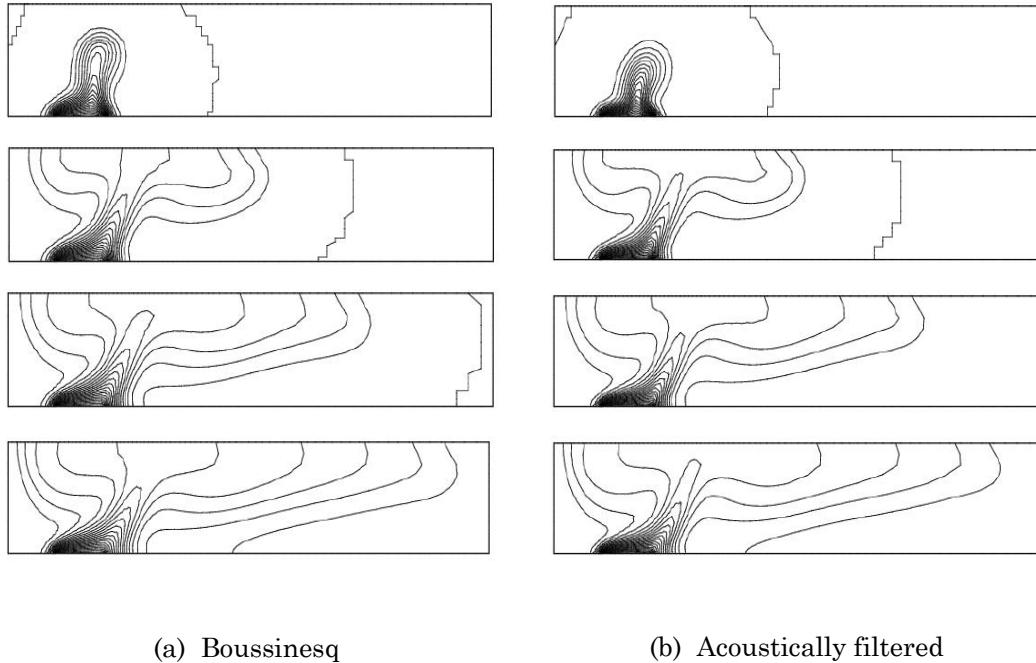


Figure 5.13.37 Isotherms for Boussinesq (left) and acoustically filtered (right) models in heated channel flow at time $t = 0.2, 0.6, 1.0$, and 1.4 .

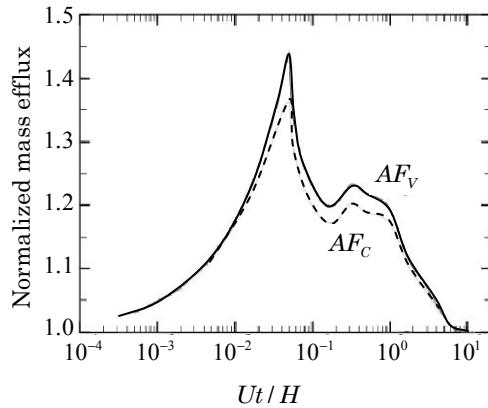


Figure 5.13.38: Normalized mass flux history for channel flow. Boussinesq formulation produces a unit mass flux for all time.

5.13.12 Closure

The numerical examples presented in this section were all based on weak-form Galerkin (mixed or penalty) finite elements models of the Navier-Stokes equations coupled with the energy equation. Coupled fluid flow and heat transfer (i.e., convective heat transfer) problems were also studied using the least-squares finite element models by Bell and Surana [46], Tang and Tsang [47], Surana, et al. [48], Pontaza and Reddy [49], and Prabhakar and Reddy [50].

References for Additional Reading

1. S. Paolucci, “On the Filtering of Sound from the Navier-Stokes Equations,” Sandia National Laboratories Technical Report, SAND82-8253, Albuquerque, NM (1982).
2. A. Majda and J. Sethian, “The Derivation and Numerical Solution of the Equations for Zero Mach Number Combustion,” *Combustion Science and Technology*, **42**, 185–205 (1985).
3. J. Principe and R. Codina, “Mathematical Models for Thermally Coupled Low Speed Flows,” *Advances in Theoretical and Applied Mechanics*, **2**, 93–112 (2009).
4. M. J. Martinez and D. K. Gartling, “A Finite Element Method for Low-Speed Compressible Flows,” *Computer Methods in Applied Mechanics and Engineering*, **193**, 1959–1979 (2004).
5. R. L. Lee, P. M. Gresho, S. T. Chan and M. J. P. Cullen, “Conservation Laws for Primitive Variable Formulations of the Incompressible Flow Equations Using the Galerkin Finite Element Method,” in *Finite Elements in Fluids, Volume 4*, R. H. Gallagher, A. H. Norrie, J. T. Oden and O. C. Zienkiewicz (eds.), John Wiley & Sons, Chichester, U. K. (1982).
6. K. A. Cliffe, “On the Conservative Finite Element Formulations of the Inviscid Boussinesq Equations,” *International Journal for Numerical Methods in Fluids*, **1**, 117–127 (1981).
7. J. N. Reddy, *An Introduction to the Finite Element Method*, 3rd ed., McGraw-Hill, New York (2006).
8. G. Hauke and T. J. R. Hughes, “A Unified Approach to Compressible and Incompressible Flows,” *Computer Methods in Applied Mechanics and Engineering*, **113**, 389–395 (1994).
9. G. Hauke and T. J. R. Hughes, “A Comparative Study of Different Sets of Variables for Solving Compressible and Incompressible Flows,” *Computer Methods in Applied Mechanics and Engineering*, **153**, 1–44 (1998).

10. M. J. Martinez, "Analysis of Anelastic Flow and Numerical Treatment via Finite Elements," Sandia National Laboratories Technical Report, SAND94-0320, Albuquerque, NM (1994).
11. C. Li and R. Glowinski, "Modeling and Numerical Simulation of Low Mach Number Compressible Flows," *International Journal for Numerical Methods in Fluids*, **23**, 77–103 (1996).
12. V. Heuveline, "On the Higher-Order Mixed FEM for Low Mach Number Flows: Applications to a Natural Convection Benchmark Problem," *International Journal for Numerical Methods in Fluids*, **41**, 1339–1356 (2003).
13. D. K. Gartling, "Convective Heat Transfer Analysis by the Finite Element Method," *Computer Methods in Applied Mechanics and Engineering*, **12**, 365–382 (1977).
14. B. Tabarrok and R. C. Lin, "Finite Element Analysis of Free Convection Flows," *International Journal of Heat and Mass Transfer*, **20**, 953–960 (1977).
15. J. N. Reddy and A. Satake, "A Comparison of Various Finite Element Models of Natural Convection in Enclosures," *Journal of Heat Transfer*, **102**, 659–666 (1980).
16. W. N. R. Stevens, "Finite Element, Stream Function-Vorticity Solution of Steady Laminar Natural Convection," *Int. Journal for Numerical Methods in Fluids*, **2**, 349–366 (1982).
17. V. Haroutunian, M. S. Engelmann, and I. Hasbani, "Segregated Finite Element Algorithms for the Numerical Solution of Large-Scale Incompressible Flow Problems," *International Journal for Numerical Methods in Fluids*, **17**, 323–348 (1993).
18. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York (1980).
19. D. K. Gartling, "Finite Element Analysis of Convective Heat Transfer Problems with Phase Change," *Computer Methods in Fluids*, K. Morgan, C. Taylor, and C. A. Brebbia (eds.), Pentech Press, London, U. K. (1974).
20. C. Beckermann and R. Viskanta, "Double-Diffusive Convection During Dendritic Solidification of a Binary Mixture," *Physico Chemical Hydro-dynamics*, **10**, 195–213 (1988).
21. P. J. Prescott, F. P. Incropera and W. D. Bennion, "Modeling of Dendritic Solidification Systems: Reassessment of the Continuum Momentum Equation," *International Journal of Heat and Mass Transfer*, **34**, 2351–2359 (1991).
22. D. K. Gartling, C. E. Hickox and R. C. Givler, "Simulations of Coupled Viscous and Porous Flow Problems," *International Journal of Computational Fluid Dynamics*, **7**, 23–48 (1996).
23. V. R. Voller, A. D. Brent and C. Prakash, "The Modeling of Heat, Mass and Solute Transport in Solidification Systems," *International Journal of Heat and Mass Transfer*, **32**, 1719–1731 (1989).
24. D. K. Gartling and P. A. Sackinger, "Finite Element Simulation of Vacuum Arc Remelting," *International Journal for Numerical Methods in Fluids*, **24**, 1271–1289 (1997).
25. D. J. Naylor, "Stresses in Nearly Incompressible Materials by Finite Elements with Application to the Calculation of Excess Pore Pressures," *International Journal for Numerical Methods in Engineering*, **8**, 443–460 (1974).
26. E. Hinton, F. C. Scott, and R. E. Ricketts, "Local Least Squares Stress Smoothing for Parabolic Isoparametric Elements," *International Journal for Numerical Methods in Engineering*, **9**, 235–256 (1975).
27. M. Engelmann, R. L. Sani, and P. M. Gresho, "The Implementation of Normal and/or Tangential Boundary Conditions in Finite Element Codes for Incompressible Fluid Flow," *International Journal for Numerical Methods in Fluids*, **2**, 225–238 (1982).
28. B. E. Larock and L. R. Herrmann, "Improved Flux Prediction Using Low Order Finite Elements," in *Proc. First International Conference on Finite Elements in Water Resources*, Gray and Pinder (eds.), Pentech Press, London, U. K. (1977).
29. R. S. Marshall, J. C. Heinrich and O. C. Zienkiewicz, "Natural Convection in a Square Enclosure by a Finite Element, Penalty Function Method, Using Primitive Fluid Variables," *Numerical Heat Transfer*, **1**, 315–330 (1978).

30. P. M. Gresho, R. L. Lee, and R. L. Sani, "The Consistent Method for Computing Derived Boundary Quantities When the Galerkin FEM is Used to Solve Thermal and/or Fluids Problems," in *Proc. Second International Conference on Numerical Methods in Thermal Problems*, Venice, Italy (1981).
31. B. Launder, "On the Computation of Convective Heat Transfer in Complex Turbulent Flows," *Journal of Heat Transfer*, **110**, 1112–1128 (1988).
32. A. J. Reynolds, "The Prediction of Turbulent Prandtl and Schmidt Numbers," *International Journal of Heat and Mass Transfer*, **18**, 1055–1069 (1975).
33. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flows*, Elsevier, New York (1987).
34. R. J. Kee, M. E. Coltrin and P. Glarborg, *Chemically Reactive Flow, Theory and Practice*, John Wiley & Sons, New York (2003).
35. M. R. Baer, R. E. Benner, R. J. Gross, and J. W. Nunziato, "Modeling and Computation of Deflagration-to-Detonation Transition (DDT) in Reactive Granular Materials," *Lectures in Applied Mathematics*, **24**, American Mathematical Society, Providence, Rhode Island (1986).
36. D. K. Gartling, "NACHOS II – A Finite Element Computer Program for Incompressible Flow Problems," Sandia National Laboratories Report, SAND86-1816 and SAND86-1817, Albuquerque, New Mexico (1978).
37. FIDAP Manual, Version 5.0, Fluid Dynamics International, Evanston, Illinois (1990).
38. D. K. Gartling, "KACHINA – A Finite Element Computer Program for Incompressible Flow Problems," Sandia National Laboratories Report, SAND2009, Albuquerque, New Mexico (2009).
39. D. H. Pelletier, J. N. Reddy, and J. A. Schetz, "Some Recent Developments and Trends in Finite Element Computational Natural Convection," in *Annual Review of Numerical Fluid Mechanics and Heat Transfer*, Vol. 2, C. L. Tien and T. C. Chawla (eds.), Hemisphere, New York, 39–85 (1989).
40. D. K. Gartling, "Convective Heat Transfer Analysis by the Finite Element Method," *Computer Methods in Applied Mechanics and Engineering*, **12**, 365–382 (1977).
41. G. de Vahl Davis and I. P. Jones, "Natural Convection in a Square Cavity: A Comparison Exercise," *International Journal for Numerical Methods in Fluids*, **3**, 227–248 (1983).
42. T. D. Blacker and M. B. Stephenson, "Paving – A New Approach to Automated Quadrilateral Mesh Generation," Sandia National Laboratories Report, SAND90-0249, Albuquerque, New Mexico (1990).
43. D. K. Gartling, "A Finite Element Analysis of Volumetrically Heated Fluids in an Axisymmetric Enclosure," in *Finite Elements in Fluids*, Vol. 4, R. H. Gallagher, D. Norrie, J. T. Oden, and O. C. Zienkiewicz (eds.), John Wiley & Sons, New York (1982).
44. D.K. Gartling, "A Finite Element Formulation for Incompressible Conjugate Viscous/Porous Flow Problems" in *Proc. International Conference on Computational Methods in Flow Analysis*, Vol. 1, H. Niki and M. Kawahara (eds.), Okayama, Japan, 619–626 (1988).
45. D. K. Gartling, "The Numerical Simulation of Viscous Flow with Change of Phase and Chemical Reaction," in *Proc. International Conference on Computational Engineering Science*, Vol. 2, S. Atluri and G. Yagawa (eds.), Atlanta, Georgia (1988).
46. B. C. Bell and K. S. Surana, " p -Version least squares finite element formulation for two-dimensional non-isothermal incompressible non-Newtonian flow," *International Journal for Numerical Methods in Fluids*, **18**, 127–162 (1994).
47. L. Q. Tang and T. H. Tsang, "Temporal, Spatial and Thermal Features of 3-D Rayleigh-Benard Convection by a Least-Squares Finite Element Method," *Computer Methods in Applied Mechanics and Engineering*, **140** 201 (1997).

48. K. S. Surana, P. Gupta, P. W. TenPas, and J. N. Reddy, “hpk Least Squares Finite Element Processes for 1-D Helmholtz Equation,” *International Journal of Computational Engineering Science and Mechanics*, **7**, 263–291 (2006).
49. J. P. Pontaza and J. N. Reddy, “Spectral/hp Least-Squares Finite Element Formulation for the Navier–Stokes Equations,” *Journal of Computational Physics*, **190**(2), 523–549 (2003).
50. V. Prabhakar and J. N. Reddy, “Spectral/hp Penalty Least-Squares Finite Element Formulation for the Steady Incompressible Navier–Stokes Equations,” *Journal of Computational Physics*, **215**(1), 274–297 (2006).

6

Non-Newtonian Fluids

6.1 Introduction

In Chapters 4 and 5 we studied the finite element models of *Newtonian fluids* (i.e., fluids whose constitutive behavior is linear). Fluids that are not described by the Newtonian constitutive relations are commonly encountered in a wide variety of industrial processes. For example, such materials include motor oils, high molecular weight liquids such as polymers, slurries, pastes, and other complex mixtures. The processing and transport of such fluids are central problems in the chemical, food, plastics, petroleum, and polymer industries.

Non-Newtonian behavior manifests itself in a number of different ways. Most such fluids exhibit a shear rate dependent viscosity, with “shear thinning” (i.e., decreasing viscosity with increasing shear rate) being the most prevalent behavior. Other phenomena associated with the elasticity and memory of the fluid, such as recoil, are also observed in many situations. Differences in the normal stress components occur in many flows and lead to such well-known effects as rod climbing or the Weissenberg effect, and the curvature of the free surface in an open channel flow. A comprehensive list and discussion of these and other non-Newtonian effects is given in the book by Bird et al. [1].

For the present discussion non-Newtonian fluids can conveniently be separated into two distinct categories: (1) inelastic fluids or fluids without memory, and (2) viscoelastic fluids, in which memory effects are significant. The distinction is an important one from both a physical and computational point of view. Basically, inelastic fluids can be viewed as generalizations (in some sense) of the Newtonian fluid. The viscosity function for such materials depends on the rate of deformation of the fluid and thus allows “shear thinning” effects to be modeled. For numerical computations, inelastic fluids can be treated using minor extensions to the standard finite element models developed for Newtonian fluids. Viscoelastic fluids, on the other hand, represent a significant departure from the Newtonian limit in terms of both physical behavior and computational complexity. The primary difficulty here is the “memory” of the fluid; the motion of a material element depends not only on the present stress state, but also on the deformation history of the material element. This history dependence leads to very complex constitutive equations and the need for special computational procedures. It is the purpose of the present chapter to study some aspects of the finite element simulation of both inelastic and viscoelastic fluids.

Following this introduction, a brief outline of the continuum equations, boundary conditions, and pertinent constitutive equations for inelastic fluids will be presented, followed by a description of their finite element models. The emphasis will be on the computational treatment of the material nonlinearities inherent in these types of fluids. The second part of the chapter will be concerned with the so-called simple fluid with fading memory. We will again state the standard balance laws in a convenient form and then describe some of the many different constitutive equations that have been found useful for numerical experimentation and computation. Several numerical examples of both inelastic and viscoelastic flows will also be presented.

The approach taken in the following sections will be very pragmatic with regard to constitutive relations for non-Newtonian fluids. No attempt will be made to theoretically justify the models used or describe in detail their positive or negative aspects, as this topic is well outside the scope of the present text. For readers interested in these questions or other topics in rheology, a number of references can be consulted, including the textbooks by Bird et al. [1], Lodge [2], Walters [3] and Tanner [4]. The book by Crochet et al. [5] covers applications of finite difference and finite element methods to non-Newtonian fluids. The text by Pearson [6] treats a variety of theoretical and practical topics regarding the analysis of industrial flow problems.

6.2 Governing Equations of Inelastic Fluids

For completeness and ready reference, the basic equations of motion are summarized once again. The section concludes with a discussion of representative inelastic, non-Newtonian constitutive relations.

6.2.1 Conservation Equations

The equations resulting from the principles of conservation of mass, momentum, and energy for flows of viscous, incompressible, inelastic fluids consist of the continuity equation, the Navier–Stokes equations, and the energy equation. For most flows of interest, the assumptions of incompressibility and laminar flow are easily justified. Utilizing standard index notation, where repeated subscripts imply summation, the governing equations in a rectangular Cartesian coordinate system are summarized here [7].

Conservation of Mass

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.2.1)$$

Conservation of Momentum

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i \quad (6.2.2)$$

Conservation of Energy

$$\rho C \left(\frac{\partial T}{\partial t} + v_j \frac{\partial T}{\partial x_j} \right) = - \frac{\partial q_i}{\partial x_i} + Q + \Phi \quad (6.2.3)$$

The constitutive equations for the total stress, σ_{ij} , and heat flux, q_i , are given by

$$\sigma_{ij} = -P\delta_{ij} + 2\mu D_{ij}, \quad \mu = \mu(D_{ij}) \quad (6.2.4)$$

and

$$q_i = -k \frac{\partial T}{\partial x_i} \quad (6.2.5)$$

where Eq. (6.2.4) is written for an isotropic, inelastic, non-Newtonian or generalized Newtonian fluid [1]. Equation (6.2.5) is the well-known Fourier's heat conduction law. The flow kinematics are given by

$$D_{ij} = \frac{1}{2}(L_{ij} + L_{ji}); \quad L_{ij} = \frac{\partial v_i}{\partial x_j} \quad (6.2.6)$$

where D_{ij} and L_{ij} are the rate of deformation and velocity gradient tensors, respectively. The coefficient μ in Eq. (6.2.4) is a scalar viscosity function which, in general, depends on the rate of deformation tensor \mathbf{D} in a manner that will be made clear in a subsequent section. The function Φ in Eq. (6.2.3) represents the viscous dissipation in the flow, and it enters the energy equation as a volume source of thermal energy. The dissipation depends on the velocity gradients and is given by

$$\Phi = 2\mu D_{ij} D_{ij} \quad (6.2.7)$$

The remaining terms in Eqs. (6.2.1)–(6.2.5) are the same as defined in earlier chapters for Newtonian flows.

Though the above set of equations is quite general and describes most practical inelastic flow problems, several simplifications to the equation system are often considered. Many processing operations involving non-Newtonian fluids occur at very low Reynolds numbers, where $Re = \rho V_{ref} D / \mu$ and V_{ref} is a reference velocity and D is a characteristic dimension in the flow problem. For such cases the nonlinear terms in the momentum equation may be neglected. Despite this simplification, the momentum equations are still nonlinear due to the variation of the viscosity function with the rate of deformation tensor. There are many instances when viscous dissipation can also be neglected, in which case the energy equation is simplified. Various other simplifications involving the energy equation are possible, depending on whether buoyancy effects or temperature-dependent material properties are considered. Such possibilities are treated in the same manner as for a Newtonian fluid and thus will not be considered further in this chapter.

6.2.2 Boundary Conditions

Equations (6.2.1)–(6.2.7) represent, in general, a coupled problem that requires boundary conditions on both the fluid motion and the energy transport. The necessary boundary conditions are of the standard type and consist of specified velocities or tractions for the momentum equations and specified temperature or heat flux for the energy equation. Symbolically the boundary conditions for the momentum equation are

$$v_i = f_i^v(s_k, t) \quad \text{on } \Gamma_v \quad (6.2.8a)$$

$$T_i = \sigma_{ij}(s_k, t) n_j(s_k) = f_i^T(s_k, t) \quad \text{on } \Gamma_T \quad (6.2.8b)$$

where s_k are the coordinates along the boundary, t is the time, n_i is the outward unit normal to the boundary, and Γ_f is the total boundary enclosing the fluid domain, Ω_f , with $\Gamma_f = \Gamma_v \cup \Gamma_T$. Note that the conditions written in Eq. (6.2.8) are in component form, i.e., there is a condition on each component of the velocity and stress vectors.

The thermal boundary conditions, for the nonisothermal case, are given by

$$T = f^T(s_k, t) \quad \text{on } \Gamma_T \quad (6.2.9a)$$

$$- \left(k \frac{\partial T}{\partial x_i} \right) n_i = q_i n_i = q_c + q_r + q_a = f_q(s_k, t) \quad \text{on } \Gamma_q \quad (6.2.9b)$$

where Γ_{ht} is the total boundary enclosing the heat transfer region and $\Gamma_{ht} = \Gamma_T \cup \Gamma_q$. Also, q_a indicates a specified flux and q_c and q_r refer to the convective and radiative components given by

$$q_c = h_c(s_k, T, t)(T - T_c) \quad (6.2.10a)$$

$$q_r = h_r(s_k, T, t)(T - T_r) \quad (6.2.10b)$$

where h_c and h_r are the convective and radiative heat transfer coefficients, and T_c and T_r are the reference (or sink) temperatures for convective and radiative heat transfer.

The above boundary conditions apply to most standard situations where the fluid is contained by fixed boundaries or enters/leaves the domain, Ω_f . One other type of boundary (or interface) condition that requires mention concerns conditions along a free surface between two fluids. Many non-Newtonian (and Newtonian) flows involve situations where the fluid (liquid) is extruded, spun, drawn, or flows in a sheet or jet, such that a free surface interface exists in the problem domain. In most cases, one of the fluids is a gas and its motion relative to the other fluid is neglected. The case of an interface between two immiscible liquids can also be included in this formulation (see Section 1.10.1). The free surface problem was considered in detail in Chapter 4 for the Newtonian fluid and does not differ significantly for the non-Newtonian case. When stress or force balances are considered along the interface, the inelastic formulation is complicated by the nonlinear constitutive behavior defined in (6.2.4). Otherwise, the interface formulation and solution methods are the same as described in Section 4.11 and they will not be repeated here.

6.2.3 Constitutive Equations

The form of the constitutive equation for a generalized Newtonian fluid was given above as

$$\sigma_{ij} = -P\delta_{ij} + 2\mu(D_{ij})D_{ij} \quad (6.2.11)$$

where σ_{ij} are the components of the total stress tensor, P is the pressure, δ_{ij} is the Kronecker delta (or components of the unit tensor), and D_{ij} are the components of the rate of deformation tensor. Of interest in the present section are particular forms for the *deviatoric stress* or *extra stress* components defined by

$$\tau_{ij} = 2\mu(D_{ij})D_{ij} \quad (6.2.12)$$

The viscosity for non-Newtonian fluids is found to depend on the rate of deformation tensor (see [1,4,7,8]):

$$\mu = \mu(D_{ij}) = \mu(I_1, I_2, I_3) \quad (6.2.13)$$

where the I_i are the *invariants* of D_{ij} , defined by

$$I_1 = \text{tr}(\mathbf{D}) = \sum_i D_{ii} \quad (6.2.14a)$$

$$I_2 = \frac{1}{2} \text{tr}(\mathbf{D}^2) = \frac{1}{2} \sum_i \sum_j D_{ij} D_{ji} \quad (6.2.14b)$$

$$I_3 = \frac{1}{3} \text{tr}(\mathbf{D}^3) = \frac{1}{3} \sum_i \sum_j \sum_k D_{ij} D_{jk} D_{ki} \quad (6.2.14c)$$

where tr denotes the trace. For an incompressible fluid, $I_1 = \nabla \cdot \mathbf{v} = 0$. Also, there is no theoretical or experimental evidence to suggest that the viscosity depends on I_3 ; thus, the dependence on the third invariant is eliminated. Equation (6.2.13) then reduces to

$$\mu = \mu(D_{ij}) = \mu(I_2) \quad (6.2.15)$$

for a generalized Newtonian fluid. The viscosity can also depend on the thermodynamic state of the fluid, which for incompressible fluids usually implies a dependence only on the temperature.

Though Eq. (6.2.15) gives the general functional form for the viscosity function, experimental observation and a limited theoretical base must be used to provide specific models for non-Newtonian viscosities. A variety of models have been proposed and correlated with experimental data (e.g., [1]). Several of the most useful and popular models are cataloged below.

6.2.3.1 Power-law model

The simplest and most familiar non-Newtonian viscosity model is the power-law model which has the form

$$\mu = K I_2^{(n-1)/2} \quad (6.2.16)$$

where n and K are parameters, which are perhaps functions of temperature, termed the *power law index* and *consistency*, respectively. One of the most common features of many non-Newtonian fluids is the “power law” decrease in the apparent viscosity with increasing shear (deformation) rate as modeled by Eq. (6.2.16). Such fluids, with an index $n < 1$ are termed *shear thinning* or *pseudoplastic*. A few materials are *shear thickening* or *dilatant* and have an index $n > 1$. The Newtonian viscosity function is obtained with $n = 1$. The admissible range of the index is bounded below by zero due to stability considerations.

When considering nonisothermal flows, the following empirical relations for n and K have proved useful

$$n = n_0 + B \left(\frac{T - T_0}{T_0} \right) \quad (6.2.17)$$

$$K = K_0 \exp[-A(T - T_0)/T_0] \quad (6.2.18)$$

where subscript zero indicates a reference condition and A and B are constants for each fluid.

6.2.3.2 Carreau model

A major deficiency in the power-law model is that it fails to predict upper and lower limiting viscosities for extreme values of the deformation rate, I_2 . This problem is alleviated in the multiple parameter Carreau model, which is of the form

$$\mu = \mu_\infty + (\mu_0 - \mu_\infty) \left(1 + [\lambda I_2]^2\right)^{(n-1)/2} \quad (6.2.19)$$

In Eq. (6.2.19), μ_0 and μ_∞ are the initial and infinite shear rate viscosities, respectively, and λ is a time constant. The remaining parameters were defined previously.

To illustrate the differences between the power-law and Carreau models, a plot of $\log \mu$ versus $\log I_2$ is shown in Figure 6.2.1 for several examples of each model. Like the power-law model, the Carreau viscosity is seen to have a “power law” region, which is bounded on either end by regions of constant viscosity.

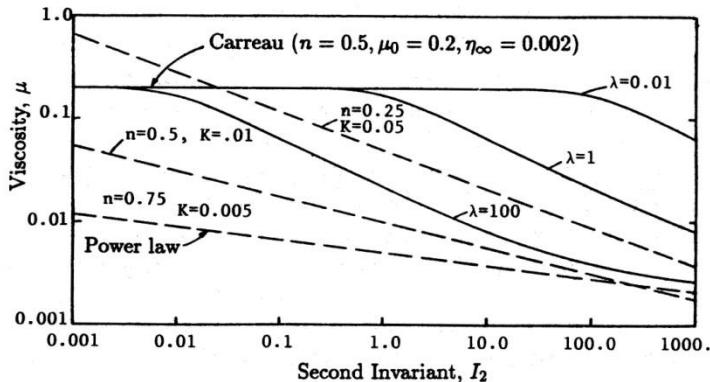


Figure 6.2.1: Viscosity functions for the power-law and Carreau models.

6.2.3.3 Bingham model

The Bingham fluid differs from most other fluids in that it can sustain an applied stress without fluid motion occurring. The fluid possesses a yield stress, τ_0 , such that when the applied stresses are below τ_0 no motion occurs; when the applied stresses exceed τ_0 the material flows, with the viscous stresses being proportional to the excess of the stress over the yield condition. Typically, the constitutive equation after yield is taken to be Newtonian (Bingham model), though other forms such as a power-law equation (Herschel–Buckley model) are possible.

In a general form, the Bingham model is expressed as [9]

$$\tau_{ij} = \left(\frac{\tau_0}{\sqrt{I_2}} + 2\mu \right) D_{ij} \quad \text{when } \frac{1}{2} tr(\tau^2) \geq \tau_0^2 \quad (6.2.20)$$

$$\tau_{ij} = 0 \quad \text{when } \frac{1}{2} tr(\tau^2) < \tau_0^2 \quad (6.2.21)$$

From Eq. (6.2.20) the apparent viscosity of the material beyond the yield point is $(\tau_0/\sqrt{T_2} + 2\mu)$. For a Herschel–Buckley fluid the μ in (6.2.20) is given by Eq. (6.2.16). The inequalities in Eqs. (6.2.20) and (6.2.21) describe a von Mises yield criterion. The implementation of the Bingham model into a computational procedure requires that (6.2.20) and (6.2.21) be modified slightly due to the condition $D_{ij} = 0$. This point will be covered in detail in a later section.

The above models serve to illustrate some of the typical viscosity functions that are available to model inelastic, non-Newtonian fluids. A more extensive list is available in [1].

6.3 Finite Element Models of Inelastic Fluids

6.3.1 Introduction

The finite element formulation of non-Newtonian flows follows very closely the formulations developed for Newtonian flow problems. Therefore, in the present section only a brief overview of the general formulation will be given, with more attention focused on those aspects that are unique to the non-Newtonian problem. For a detailed description of the finite element models of viscous incompressible flow problems, the reader is referred to Chapter 4.

As in the case of Newtonian fluids, there are several different formulations that may be used to construct the finite element models of a non-Newtonian fluid. The equations given in Section 6.2.1 are in terms of the velocity, pressure, and temperature as the dependent variables. However, equations using the stream function and vorticity or the stream function alone could also be employed. The primary argument in favor of the use of primitive variables (i.e., velocity, pressure, and temperature) for non-Newtonian flows comes from the free surface problem. The free surface boundary conditions given previously in Section 4.11 contain the pressure explicitly, and therefore can be more conveniently imposed in a primitive variable formulation. Also, as stated earlier, the use of the stream function-vorticity or stream function formulations presents difficulties in the imposition of boundary conditions, and the stream function formulation requires higher-order interpolation (i.e., C^1 -continuity).

Here, we develop finite element models based on the primitive variables. The mixed and penalty models described in Chapter 4 are presented here for the nonisothermal flow of inelastic fluids.

6.3.2 Mixed Model

Weak forms of Eqs. (6.2.1)–(6.2.3) can be developed from their Galerkin integrals, as explained in Chapters 4 and 5 (see Sections 4.2 and 5.3). The velocity components are approximated by Lagrange interpolation functions, one order higher than those used for the pressure. Suppose that the dependent variables (v_i, P, T) are approximated by expansions of the form

$$T(\mathbf{x}, t) = \sum_{n=1}^N \psi_n(\mathbf{x}) T_n(t) = \boldsymbol{\Psi}^T \mathbf{T} \quad (6.3.1a)$$

$$v_i(\mathbf{x}, t) = \sum_{m=1}^M \psi_m(\mathbf{x}) v_i^m(t) = \boldsymbol{\Psi}^T \mathbf{v}_i \quad (6.3.1b)$$

$$P(\mathbf{x}, t) = \sum_{l=1}^L \phi_l(\mathbf{x}) P_l(t) = \boldsymbol{\Phi}^T \mathbf{P} \quad (6.3.1c)$$

where $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ are vectors of interpolation (or shape) functions, \mathbf{T} , \mathbf{v}_i , and \mathbf{P} are (column) vectors of nodal values of temperature, velocity components, and pressure, respectively, and superscript $(\cdot)^T$ denotes a transpose of the enclosed vector or matrix. Note that the standard practice of interpolating the temperature and velocity variables with the same shape functions has been employed here. Substitution of Eqs. (6.3.1a)–(6.3.1c) into the weak forms of Eqs. (6.2.1)–(6.2.4) [i.e., Eq. (5.3.1)] results in the following set of nonlinear algebraic equations [see Eqs. (5.3.4)–(5.3.6) for details].

Continuity

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (6.3.2)$$

Momentum

$$\mathbf{M} \dot{\mathbf{v}} + \mathbf{C}(\mathbf{v}) \mathbf{v} + \mathbf{K}(\mathbf{v}, \mathbf{T}) \mathbf{v} - \mathbf{Q} \mathbf{P} + \mathbf{B} \mathbf{T} = \mathbf{F} \quad (6.3.3)$$

Energy

$$\mathbf{N} \dot{\mathbf{T}} + \mathbf{D} \mathbf{T} + \mathbf{L} \mathbf{T} = \mathbf{G} \quad (6.3.4)$$

where the superposed dot represents a time derivative and $\mathbf{v}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\}$. This set of equations is virtually identical to those used in Newtonian problems except for the dependence of the viscous diffusion term \mathbf{K} on the velocity (because of viscosity's dependence on the rate of deformation tensor), and possibly temperature.

For the three-dimensional case, Eqs. (6.3.2)–(6.3.4) have the following explicit form [the continuity equation (6.3.2) and momentum equations (6.3.3) are combined into one]:

$$\begin{aligned} & \left[\begin{array}{cccc} \mathbf{M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \dot{\mathbf{v}}_1 \\ \dot{\mathbf{v}}_2 \\ \dot{\mathbf{v}}_3 \\ \dot{\mathbf{P}} \end{array} \right\} + \left[\begin{array}{cccc} \mathbf{C}(\mathbf{v}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\mathbf{v}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}(\mathbf{v}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{P} \end{array} \right\} \\ & + \left[\begin{array}{cccc} \hat{\mathbf{K}}_{11} & \mathbf{K}_{21} & \mathbf{K}_{31} & -\mathbf{Q}_1 \\ \mathbf{K}_{12} & \hat{\mathbf{K}}_{22} & \mathbf{K}_{32} & -\mathbf{Q}_2 \\ \mathbf{K}_{13} & \mathbf{K}_{23} & \hat{\mathbf{K}}_{33} & -\mathbf{Q}_3 \\ -\mathbf{Q}_1^T & -\mathbf{Q}_2^T & -\mathbf{Q}_3^T & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{P} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{F}_1(\mathbf{T}) \\ \mathbf{F}_2(\mathbf{T}) \\ \mathbf{F}_3(\mathbf{T}) \\ \mathbf{0} \end{array} \right\} \end{aligned} \quad (6.3.5)$$

$$[\mathbf{N}] \{\dot{\mathbf{T}}\} + [\mathbf{D}(\mathbf{v})] \{\mathbf{T}\} + [\mathbf{L}] \{\mathbf{T}\} = \{\mathbf{G}(\mathbf{T})\} \quad (6.3.6)$$

The coefficient matrices shown in Eqs. (6.3.5) and (6.3.6) are defined by [from Eq. (6.2.12)]

$$\begin{aligned} \hat{\mathbf{K}}_{11} &= 2\mathbf{K}_{11} + \mathbf{K}_{22} + \mathbf{K}_{33} \\ \hat{\mathbf{K}}_{22} &= \mathbf{K}_{11} + 2\mathbf{K}_{22} + \mathbf{K}_{33} \\ \hat{\mathbf{K}}_{33} &= \mathbf{K}_{11} + \mathbf{K}_{22} + 2\mathbf{K}_{33} \end{aligned} \quad (6.3.7a)$$

$$\begin{aligned}
\mathbf{M} &= \int_{\Omega^e} \rho_0 \Psi \Psi^T d\mathbf{x}; \quad \mathbf{C}(\mathbf{v}) = \int_{\Omega^e} \rho_0 \Psi (\Psi^T \mathbf{v}_j) \frac{\partial \Psi^T}{\partial x_j} d\mathbf{x} \\
\mathbf{K}_{ij} &= \int_{\Omega^e} \mu \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_j} d\mathbf{x}; \quad \mathbf{Q}_i = \int_{\Omega^e} \frac{\partial \Psi}{\partial x_i} \Phi^T d\mathbf{x} \\
\mathbf{F}_i(\mathbf{T}) &= - \int_{\Omega^e} \rho_0 g_i \beta \Psi \Phi^T d\mathbf{x} + \int_{\Omega^e} \rho_0 g_i \beta T_0 \Psi d\mathbf{x} + \oint_{\Gamma^e} \Psi T_i ds \\
\mathbf{D}(\mathbf{v}) &= \int_{\Omega^e} \rho_0 C \Phi (\Psi^T \mathbf{v}_j) \frac{\partial \Phi^T}{\partial x_j} d\mathbf{x} \\
\mathbf{N} &= \int_{\Omega^e} \rho_0 C \Psi \Psi^T d\mathbf{x}; \quad \mathbf{L} = \int_{\Omega^e} k \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_i} d\mathbf{x} \\
\mathbf{G} &= \int_{\Omega^e} \Psi Q d\mathbf{x} + \int_{\Omega^e} 2\mu \Psi \Phi d\mathbf{x} + \oint_{\Gamma^e} \Psi q_n ds
\end{aligned} \tag{6.3.7b}$$

where summation on repeated indices is implied. Note that the finite element model is nonlinear because of the nonlinearity in the convective terms as well as the viscosity. In addition, the conductivity k can be a function of temperature. Equations (6.3.2)–(6.3.4) can be combined into a single matrix equation

$$\begin{aligned}
\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{v}) + \mathbf{K}(\mathbf{v}, \mathbf{T}) & -\mathbf{Q} & \mathbf{B}(\mathbf{T}) \\ -\mathbf{Q}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}(\mathbf{v}) + \mathbf{L}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(\mathbf{T}) \\ \mathbf{0} \\ \mathbf{G}(\mathbf{T}, \mathbf{v}) \end{Bmatrix}
\end{aligned} \tag{6.3.8}$$

or in a more symbolic format as

$$\bar{\mathbf{M}} \dot{\mathbf{U}} + \bar{\mathbf{K}}(\mathbf{v}, \mathbf{T}) \mathbf{U} = \bar{\mathbf{F}} \tag{6.3.9}$$

where

$$\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{P}^T, \mathbf{T}^T\} \tag{6.3.10}$$

This completes the development of the mixed finite element model for the inelastic case.

6.3.3 Penalty Model

In the penalty function method, the continuity equation is treated as a constraint (see Section 5.4) and the problem is reformulated as an unconstrained problem. The pressure, which is the Lagrange multiplier, does not appear explicitly as a dependent variable in the formulation, although it is a part of the boundary stresses [see Eq. (4.3.5)]. In two dimensions, an approximation for the pressure can be post-computed from the relation

$$P = -\gamma_e \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} \right) \tag{6.3.11}$$

where γ_e is the penalty parameter. The penalty finite element model is given by

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{v}}_1 \\ \dot{\mathbf{v}}_2 \end{Bmatrix} + \begin{bmatrix} \mathbf{C}(\mathbf{v}) & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\mathbf{v}) \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{Bmatrix} +$$

$$\left(\begin{bmatrix} 2\mathbf{K}_{11} + \mathbf{K}_{22} & \mathbf{K}_{21} \\ \mathbf{K}_{12} & \mathbf{K}_{11} + 2\mathbf{K}_{22} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{K}}_{11} & \hat{\mathbf{K}}_{21} \\ \hat{\mathbf{K}}_{12} & \hat{\mathbf{K}}_{22} \end{bmatrix} \right) \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{Bmatrix} \quad (6.3.12)$$

where \mathbf{M} , $\mathbf{C}(\mathbf{v})$, \mathbf{K}_{ij} (which depend on the viscosity) and \mathbf{F}_i are the same as those defined in Eq. (6.3.7), and

$$\hat{\mathbf{K}}_{ij} = \int_{\Omega^e} \gamma_e \frac{\partial \Psi}{\partial x_i} \frac{\partial \Psi^T}{\partial x_j} d\mathbf{x} \quad (6.3.13)$$

The energy equation remains unchanged as in Eq. (6.3.6). In matrix form, Eqs. (6.3.13) and (6.3.6) can be expressed as

$$\bar{\mathbf{M}} \dot{\mathbf{U}} + \bar{\mathbf{K}}(\mathbf{v}, \mathbf{T}) \mathbf{U} = \bar{\mathbf{F}} \quad (6.3.14)$$

where

$$\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{T}^T\} \quad (6.3.15)$$

This completes the discussion of the penalty finite element model for the inelastic case.

6.3.4 Matrix Evaluations

The appearance of a nonlinear diffusion term, \mathbf{K} , in Eqs. (6.3.9) and (6.3.14) influences two basic aspects of the finite element analysis, namely, the matrix evaluation and solution algorithms. Matrix evaluation methods will be discussed first.

The techniques used for the construction of element level matrices that contain variable coefficients have generally fallen into one of two categories — the reconstruction methods and the hypermatrix methods. The differences in these methods and their application to non-Newtonian formulations are best explained via a specific example. Consider the construction of a particular component of the \mathbf{K} matrix in Eq. (6.3.9), for example,

$$\mathbf{K}_{11} = \mathbf{K}_{\mathbf{xx}} = \int_{\Omega^e} \frac{\partial \Psi}{\partial x} \mu(I_2, T) \frac{\partial \Psi^T}{\partial x} d\mathbf{x} \quad (6.3.16)$$

where Ψ is the vector of shape functions used for the velocity field. The viscosity function is shown in its most general form with a dependence on I_2 and the temperature; the invariant I_2 depends on velocity gradients [see Eq. (6.2.14)].

For most finite element applications the integration in Eqs. (6.3.7) and (6.3.16) is performed via a numerical quadrature. Such a computation requires that each function in the integrand be evaluated at each integration point within the element. Therefore, the evaluation of \mathbf{K} for a non-Newtonian fluid requires that the velocity gradients and perhaps temperature be evaluated at integration points, using the values from the latest available solution (\bar{u}_m, \bar{T}_n) at time t_s :

$$T(\mathbf{x}, t_s) = \sum_{n=1}^N \theta_n(\mathbf{x}) \bar{T}_n(t_s) \quad (6.3.17a)$$

$$v_i(\mathbf{x}, t_s) = \sum_{m=1}^M \psi_m(\mathbf{x}) \bar{v}_i^m(t_s) \quad (6.3.17b)$$

In many standard finite element programs the above quadrature procedure is carried out each time the element matrix is required. As a result of the iterative methods used to solve the nonlinear equations, the matrix formation operations may be required many times, especially for time-dependent problem, resulting in a large computational cost.

An alternative to the matrix reconstruction method, the hypermatrix approach may be used. This method reduces some of the computational cost but at the cost of some additional storage and I/O in the computer program. Since the viscosity is a function of position in the element (due to its functional dependence on the velocity field and temperature), it is natural to interpolate the viscosity in the same way as a dependent variable. Thus, let the viscosity be represented by

$$\mu = \Psi^T \hat{\mu} \quad (6.3.18)$$

where Ψ is a vector of shape functions and $\hat{\mu}$ is a vector of nodal point viscosity values. Substitution of Eq. (6.3.18) in Eq. (6.3.16) yields

$$\mathbf{K}_{\mathbf{xx}} = \int_{\Omega_e} \frac{\partial \Psi}{\partial x} (\Psi^T \hat{\mu}) \frac{\partial \Psi^T}{\partial x} dx \quad (6.3.19)$$

Since the $\hat{\mu}$ are the nodal values, the integral in Eq. (6.3.19) can be constructed once and stored as a three-dimensional array $\mathbf{K}(NI, NJ, NK)$, where each index corresponds to one of the shape functions in (6.3.19). This matrix is called the *hypermatrix*. A product of the hypermatrix with a known vector of nodal point viscosities produces the required element matrix without repeated numerical quadrature. This technique has been used successfully (see [10,11]) for the nonlinear advection terms in the momentum equations [i.e., the $\mathbf{C}(\mathbf{v})\mathbf{v}$ term in Eq. (6.3.7)] as well as for material property variations (see [12]). It is most effective when the variable coefficient is a nodal quantity or depends on a nodal quantity, since this permits shape function evaluations to be avoided. Unfortunately, this is not the case for viscosity which depends on velocity gradients. It is well-known that the most accurate points within a quadrilateral element at which to evaluate derivatives are the 2×2 Gauss integration points (see [13]). Thus, for maximum accuracy in the viscosity evaluation the invariant I_2 should be evaluated at the Gauss points. However, the extrapolation of Gauss point values of I_2 to the nodes by standard methods (see [14,15]), in order to use Eq. (6.3.19), has not proven to be a viable technique for most non-Newtonian models. In general, predictions of I_2 at the nodes via extrapolation are very inaccurate and lead to poor overall accuracy of the solution method. As an alternative to extrapolation, an averaging method is used in which the Gauss point values of I_2 are averaged over the element and a single value is used to evaluate the viscosity at the nodes.

Figure 6.3.1 shows velocity profiles obtained with both the extrapolation and averaging procedures described above. The results are for a power-law fluid in a cylindrical tube. The inaccuracies generated by extrapolation are on the order of 30% while the averaging procedure yields results within a few percent of the analytical result. The averaging procedure in conjunction with Eq. (6.3.19) has been used successfully in other problems with a variety of viscosity models (see [16,17]).

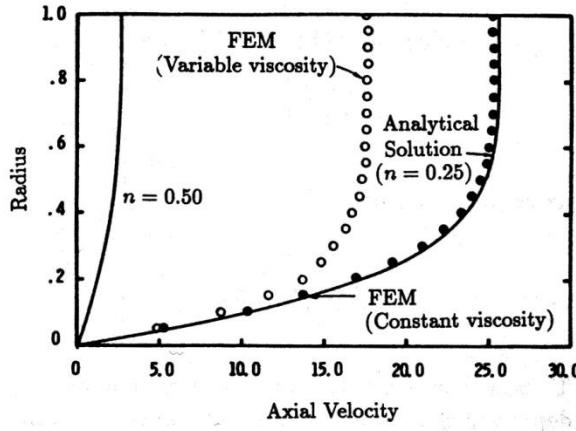


Figure 6.3.1: Velocity profile for a power fluid in a tube; various methods for viscosity computation.

To conclude this section it should be pointed out that the choice of a reconstruction or hypermatrix procedure is dependent on the type of computing resources available and the structure of the finite element software. Both methods are equivalent and effective. The relative costs of CPU and I/O time will heavily influence the selection.

6.4 Solution Methods for Inelastic Fluids

The solution procedure selected for the inelastic, non-Newtonian problem given by the finite element models in Section 6.3 must be capable of treating several different types of nonlinearities. For nonzero Reynolds number flows the nonlinear advection terms are significant and in many cases may dominate the problem. The non-Newtonian fluid introduces a second type of nonlinear behavior while the presence of unknown free surface boundaries can introduce a geometric nonlinearity. In the present section, only the first two types of phenomena will be considered; free surface solutions were covered in a previous chapter. Also, the discussion will focus on steady isothermal problems since many of the procedures and results obtained for this type of problem can be extended in a straightforward manner to more complex situations. The solution methods described in Chapters 4 and 5 are also applicable here with only minor alterations.

For the steady-state case, the mixed method Eqs. (6.3.2) and (6.3.3) take the form

$$\mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{K}(\mathbf{v}) - \mathbf{Q}\mathbf{P} = \mathbf{F} \quad (6.4.1a)$$

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (6.4.1b)$$

or, as a single matrix equation we have

$$\hat{\mathbf{K}}(\mathbf{U})\mathbf{U} = \hat{\mathbf{F}} \quad (6.4.2)$$

where the vector \mathbf{U} now contains the velocity components (v_1, v_2, v_3) and the pressure \mathbf{P} . The dependence of μ on \mathbf{v} has been absorbed into \mathbf{K} and hence into $\hat{\mathbf{K}}$.

There are a wide variety of iterative algorithms that can be applied to Eq. (6.4.2), and we will consider only two such schemes in detail here. Similar comments apply to the penalty method model from Eq. (6.3.14).

The simplest method is Picard's method (also known as successive substitution or functional iteration), which can be written in the following form

$$\hat{\mathbf{K}}(\mathbf{U}^n)\mathbf{U}^{n+1} = \hat{\mathbf{F}}^n \quad (6.4.3)$$

where the superscript n indicates the iteration level. Equation (6.4.3) describes a particularly simple procedure in which the nonlinear coefficients in the problem are evaluated using dependent variable data from the previous iteration. The method has a linear rate of convergence but works for a relatively large range of problems. For shear thinning materials it is observed that as the power law index, n [see Eq. (6.2.16) for example] decreases, the rate of convergence decreases markedly (see [5]).

A more sophisticated algorithm with a higher rate of convergence is Newton's method. For the nonlinear equation in (6.4.2) Newton's method may be written as

$$\mathbf{J}(\mathbf{U}^n)[\mathbf{U}^{n+1} - \mathbf{U}^n] = -\hat{\mathbf{K}}(\mathbf{U}^n)\mathbf{U}^n + \hat{\mathbf{F}}^n \quad (6.4.4)$$

where \mathbf{J} is the Jacobian matrix defined by

$$\mathbf{J}(\mathbf{U}^n) = \frac{\partial}{\partial \mathbf{U}} [\hat{\mathbf{K}}(\mathbf{U})\mathbf{U} - \hat{\mathbf{F}}] \Big|_{\mathbf{U}^n} \quad (6.4.5)$$

Newton's method is the standard solution procedure for Newtonian problems since it is well-suited to the quadratic nonlinearity occurring in the advection terms of the momentum equations. However, experience has shown that this procedure, as written in Eq. (6.4.4), does not perform well for many types of generalized Newtonian fluids (see [18]). In particular, Newton's method does not work well for viscosity models with shear thinning. It is therefore recommended that the non-Newtonian behavior in Eq. (6.4.4) be treated using the Picard method; the advection terms, if present, should be treated using Newton's method.

To illustrate the behavior of the Picard and Newton algorithms, a simple creeping flow problem ($Re = 0$) was solved using a power-law viscosity model. The problem consists of the ubiquitous driven cavity flow in which a fluid is contained in a square cavity three sides of which are stationary; the fourth side of the cavity moves at unit velocity in its own plane. Shown in Figure 6.4.1 are plots of the convergence measure (relative norm on the change in the solution between iterations) versus iteration number for both algorithms and several values of the power-law index. The Picard scheme converges for all values of the index below unity (shear thinning) but diverges for shear thickening fluids. The Newton method performs in the opposite way, with rapid convergence for shear thickening and divergence for smaller values of the index.

Other methods of solution of Eq. (6.4.2) are possible though the Picard and Newton methods represent the behavioral limits for most algorithms. Tanner and Milthorpe [18] have used a combination of the Picard and the Newton method successfully for several nonlinear viscosity models. Also, Engelman [19] has found

the quasi-Newton or variable metric method to be very cost effective for non-Newtonian flows.

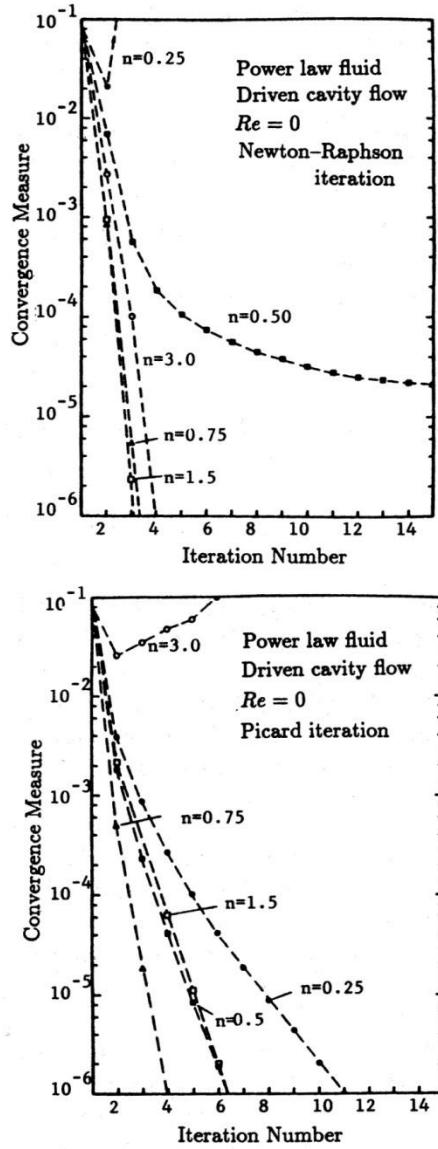


Figure 6.4.1: Iterative convergence for the Picard and Newton methods.

The solution procedures outlined above are applicable to all of the common inelastic, non-Newtonian models. However, the Bingham model [Eqs. (6.2.20) and (6.2.21)] deserves some further comments due to its unique method of implementation. For a typical velocity-based finite element method, the exact Bingham constitutive equation is quite difficult to use, mainly as a result of the requirement that no motion take place below the yield condition. To circumvent this difficulty it has proved useful to employ an approximate Bingham equation

given by

$$\tau_{ij} = \left(\frac{\tau_0(1 - \mu/\mu_r)}{\sqrt{I_2}} + 2\mu \right) D_{ij} \quad \text{when } \frac{1}{2} \operatorname{tr}(\tau^2) \geq \tau_0^2 \quad (6.4.6a)$$

$$\tau_{ij} = 2\mu_r D_{ij} \quad \text{when } \frac{1}{2} \operatorname{tr}(\tau^2) < \tau_0^2 \quad (6.4.6b)$$

where μ_r is a pre-yield viscosity and $\mu/\mu_r \ll 1$. A plot of the constitutive model given by (6.4.6) is shown in Figure 6.4.2. Unlike the true Bingham material, the fluid described by Eq. (6.4.6) can undergo deformation below the yield point though the magnitude of the motion can be made arbitrarily small (i.e., approach the Bingham model) by increasing μ_r relative to μ . Experience has shown that when $\mu/\mu_r \sim .01$ or less then the solution is virtually independent of μ_r and thus approximates the Bingham model with excellent accuracy [17,18].

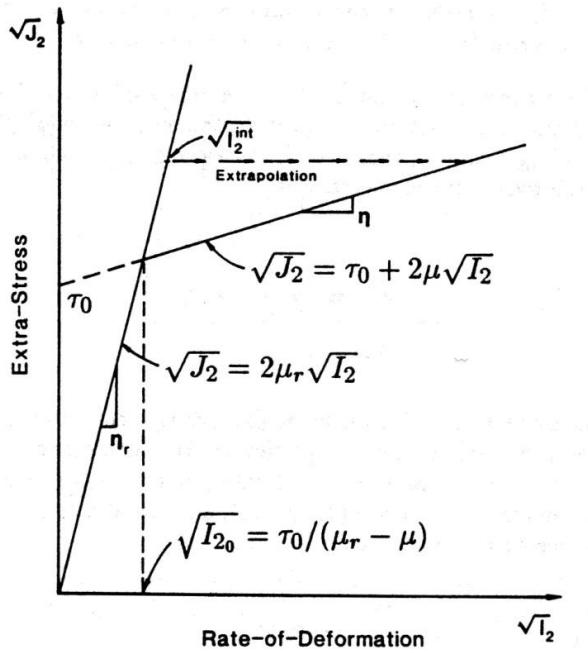


Figure 6.4.2: Bi-viscosity model for computations involving Bingham fluids.

Iterative convergence of Bingham fluids using the Picard method [see Eq. (6.4.3)] can be very slow since the apparent viscosity after yield is essentially like that of a power-law fluid with index $n = 0$. To accelerate the convergence a particular form of extrapolation can be used. Note first that for the present model the yield stress, τ_0 , corresponds to a value of the rate of deformation (invariant) given by $\sqrt{I_{20}} = \tau_0/(\mu_r - \mu)$ as seen in Figure 6.4.2. At the start of the iterative procedure the fluid is usually assumed to have a viscosity, μ_r , from which an initial velocity field and values of I_2 may be computed. For those elements which have yielded ($\sqrt{I_2} \geq \sqrt{I_{20}}$) an extrapolation at constant stress should be used to estimate a new value of the apparent viscosity. Failure to perform this extrapolation can double or

triple the number of iterations needed to reach convergence. Based on Figure 6.4.2 (dashed arrow) it is possible to provide an analytical expression for the projected value of I_2 needed for the viscosity evaluation. That is,

$$\sqrt{I_2} = \frac{2\mu_r \sqrt{I_2^{int}} - \tau_0}{2\mu} \quad (6.4.7)$$

where the superscript *int* indicates the value of I_2 computed initially using a viscosity of μ_r . The above procedure has been used in a variety of steady-state problems with good success [16,17] and is readily extended to time-dependent flows.

6.5 Governing Equations of Viscoelastic Fluids

As noted in the introduction, non-Newtonian behavior has many facets. In the previous sections we dealt with the modeling of “shear rate” dependent viscosity behavior but ignored the problems associated with fluid elasticity and memory. In the following several sections we will focus on viscoelastic fluids and their numerical simulation by the finite element method.

6.5.1 Conservation Equations

The analytical description of the motion of a continuous medium is based on conservation of mass, momentum, and energy, and the associated equations of state and constitutive relations. The present development will be limited to laminar flows of incompressible, viscous isotropic fluids. The fluid motion is assumed to be isothermal to avoid the discussion of the energy equation, as it was discussed in detail in Chapter 5. We also assume that, for simplicity, the flows are two-dimensional. Of course, the extension of the present discussion to three dimensions is conceptually straightforward, although in practice the subject is sufficiently complex and computationally taxing to have received relatively little attention. The discussion will also be limited to simple fluids with fading memory [4–7,20,21], a material description that has received the most widespread attention and development.

The equations of interest for two-dimensional flows are written in the Cartesian coordinate system (x_1, x_2) using the Eulerian description.

Conservation of Mass

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.5.1)$$

Conservation of Momentum

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i \quad (6.5.2)$$

where v_i denotes the i th component of velocity, σ_{ij} the components of the total (Cauchy) stress tensor, ρ the density, g_i the component of body force, and t denotes time. The standard index notation is used, with summation on repeated indices.

The field equations in (6.5.1) and (6.5.2) are to be solved in conjunction with a constitutive equation for the fluid and boundary and initial conditions of the flow

problem. The various constitutive equations of viscoelastic fluid behavior are a major focus of this chapter, and will be discussed in the next section. Since the boundary conditions depend on the dependent variables employed in the problem, they are discussed subsequent to the discussion of the constitutive relations.

6.5.2 Constitutive Equations

For an incompressible fluid the total stress is given by

$$\sigma_{ij} = -P\delta_{ij} + \tau_{ij} \quad (6.5.3)$$

where P is a pressure and τ_{ij} is the extra-stress. It should be noted that due to the particular forms that will be used to describe the extra-stress, the pressure defined in Eq. (6.5.3) will not, in general, be the same as a Newtonian pressure or hydrostatic pressure. That is, the extra-stress may not be traceless (i.e., $\tau_{ii} \neq 0$), in which case P is not the usual mean normal stress found in Newtonian fluid mechanics [22,23].

The central issue for the constitutive description of a viscoelastic fluid is the choice of an equation that relates the extra-stress in (6.5.3) to the flow kinematics. For the general class of materials called simple fluids, such a relationship can be established in functional form where the current extra-stress is related to the history of deformation in the fluid. Typically, such a relation is expressed as

$$\tau_{ij} = \mathcal{F}_{ij}[G_{kl}(s), 0 < s < \infty] \quad (6.5.4)$$

where \mathcal{F}_{ij} is a tensor-valued functional, G_{kl} is a finite deformation tensor (related to the Cauchy–Green tensor), and $s = t - t'$ is the time lapse from time t' to the present time, t . Unfortunately, the generality of a functional form, such as (6.5.4), does not produce usable constitutive equations for general flow problems; the solution of practical problems requires that approximate forms of (6.5.4) be developed.

Since there are numerous ways to approximate the general functional given in (6.5.4), a wide variety of constitutive equations have developed out of the simple fluid theory. To date, none of these approximate constitutive equations are universally applicable; that is, no single constitutive equation is capable of predicting all of the observed behavior in elastic fluids. In essence, the present state-of-the-art mandates that a spectrum of constitutive equations be considered for use, with the specific choice being dictated by the ability of a given model to predict the dominant, non-Newtonian effects expected in a particular application.

To further complicate the situation, approximate constitutive equations can be of several different forms. For certain limiting flow conditions, such as “small” strains, “slow” flows, or “slightly” elastic fluids, the functional in (6.5.4) can be approximated by an expansion in terms of a small parameter. Such a formulation leads to a set of “hierarchical” equations for the extra-stress in terms of the rate of deformation tensor and its various derivatives [5,6]. In these models the extra-stress is given explicitly in terms of the flow kinematics, which in principle, allows Eqs. (6.5.1)–(6.5.3) to be written in terms of only the velocity and pressure, as in a Newtonian flow. Though convenient in form, these hierarchical formulas are of very limited value in general flows due to the “smallness” assumptions used in their derivation. Some computational work has been done using these types of models

[24–26]; however, due to their limited applicability they will not be given further consideration here.

The two remaining major categories of approximate constitutive relations include the integral and differential models. The integral model represents the extra-stress in terms of an integral over past time of the fluid deformation history. For a differential model the extra-stress is determined from a differential (evolution) equation that relates the stress and stress rate to the flow kinematics. The choice between a differential and integral formulation is a crucial one with regard to numerical simulation since the computational algorithms for each type are very different. In the present book we will emphasize the differential form since the majority of the current numerical work involves this formulation. References to work with integral models will also be given.

6.5.2.1 Differential models

The differential constitutive equations to be considered here are implicit, rate-type models, generally associated with the names of Oldroyd, Maxwell, and Jeffrey. Before proceeding to the description of specific models it is important to list a number of definitions. For an Eulerian reference frame the material time derivative (or convected derivative) of a symmetric, second-order tensor can be defined in several ways, all of which are frame invariant. Let φ_{ij} denote the Cartesian components of a second-order tensor. Then the *upper-convected* (or co-deformational) derivative is defined by

$$\overset{\nabla}{\varphi}_{ij} = \frac{\partial \varphi_{ij}}{\partial t} + v_k \frac{\partial \varphi_{ij}}{\partial x_k} - L_{ik}\varphi_{kj} - L_{jk}\varphi_{ki} \quad (6.5.5a)$$

and the *lower-convected* derivative is defined as

$$\overset{\Delta}{\varphi}_{ij} = \frac{\partial \varphi_{ij}}{\partial t} + v_k \frac{\partial \varphi_{ij}}{\partial x_k} + L_{ki}\varphi_{kj} + L_{kj}\varphi_{ki} \quad (6.5.5b)$$

where v_k are the components of the velocity vector and L_{ij} are the components of the velocity gradient tensor \mathbf{L} , defined by

$$\mathbf{L} = \nabla \mathbf{v}, \quad \text{or} \quad L_{ij} = \frac{\partial v_j}{\partial x_i} \quad (6.5.6)$$

Since both Eqs. (6.5.5a) and (6.5.5b) are admissible convected derivatives, their linear combination is also admissible:

$$\dot{\varphi}_{ij} = \left(1 - \frac{a}{2}\right) \overset{\nabla}{\varphi}_{ij} + \left(\frac{a}{2}\right) \overset{\Delta}{\varphi}_{ij} \quad (6.5.7)$$

Equation (6.5.7) is a general convected derivative which reduces to (6.5.5a) for $a = 0$ and (6.5.5b) for $a = 2$. When $a = 1$ [average of (6.5.5a) and (6.5.5b)] the convected derivative in (6.5.7) is termed a *corotational* or *Jaumann derivative*.

Using τ as the extra-stress tensor with the Cartesian components denoted by τ_{ij} , the upper-convected derivative of the stress components is given by

$$\overset{\nabla}{\tau}_{ij} = \frac{\partial \tau_{ij}}{\partial t} + v_k \frac{\partial \tau_{ij}}{\partial x_k} - L_{ik}\tau_{kj} - L_{jk}\tau_{ki} \quad (6.5.8a)$$

and the lower-convected derivative is

$$\overset{\Delta}{\tau}_{ij} = \frac{\partial \tau_{ij}}{\partial t} + v_k \frac{\partial \tau_{ij}}{\partial x_k} + L_{ki} \tau_{kj} + L_{kj} \tau_{ki} \quad (6.5.8b)$$

The corotational derivative is

$$\overset{\circ}{\tau}_{ij} = \left(1 - \frac{a}{2}\right) \overset{\nabla}{\tau}_{ij} + \left(\frac{a}{2}\right) \overset{\Delta}{\tau}_{ij} \quad (6.5.9)$$

All of these derivatives have been used in various differential constitutive equations. The selection of one type of derivative over another is usually based on the physical plausibility of the resulting constitutive equation and the matching of experimental data to the model for simple (viscometric) flows.

The simplest differential constitutive models are the upper- and lower-convected Maxwell fluids, which are defined by the following equations.

Upper-Convected Maxwell Fluid

$$\tau_{ij} + \lambda \overset{\nabla}{\tau}_{ij} = 2\mu^p D_{ij} \quad (6.5.10a)$$

Lower-Convected Maxwell Fluid

$$\tau_{ij} + \lambda \overset{\Delta}{\tau}_{ij} = 2\mu^p D_{ij} \quad (6.5.10b)$$

where λ is a relaxation time for the fluid, μ^p is a viscosity, and D_{ij} are the components of the rate of deformation (or strain rate) tensor defined by

$$\mathbf{D} = \frac{1}{2} [\nabla \mathbf{v} + (\nabla \mathbf{v})^T], \text{ or } D_{ij} = \frac{1}{2} (L_{ij} + L_{ji}) \quad (6.5.11)$$

The upper-convected Maxwell model (6.5.10a) has been used extensively in testing numerical algorithms; the lower-convected and corotational forms of the Maxwell fluid predict physically unrealistic behavior and are not generally used.

By employing the general convected derivative (6.5.9) in a Maxwell-like model the following constitutive equation is produced.

Johnson–Segalman Model

$$\tau_{ij} + \lambda \overset{\circ}{\tau}_{ij} = 2\mu^p D_{ij} \quad (6.5.12)$$

which is the Johnson–Segalman fluid [27]. By slightly modifying Eq. (6.5.12) to include a variable coefficient for τ_{ij} , the Phan Thien–Tanner model [28] is derived.

Phan Thien–Tanner Model

$$Y(\tau) \tau_{ij} + \lambda \overset{\circ}{\tau}_{ij} = 2\mu^p D_{ij} \quad (6.5.13a)$$

where

$$Y(\tau) = 1 + \frac{\epsilon \lambda}{\mu^p} \operatorname{tr}(\tau) \quad (6.5.13b)$$

and ϵ is a constant. This equation is somewhat better than (6.5.12) in representing actual material behavior.

The Johnson–Segalman and Phan Thien–Tanner models suffer a common defect in that, for a monotonically increasing shear rate, there is a region where the shear stress decreases — a physically unrealistic behavior. To correct this anomaly the constitutive equations can be altered using the following procedure. Let the extra-stress be decomposed into two partial stresses, τ_{ij}^s and τ_{ij}^p such that

$$\tau_{ij} = \tau_{ij}^s + \tau_{ij}^p \quad (6.5.14)$$

where τ_{ij}^s is a purely viscous and τ_{ij}^p is a viscoelastic stress component. The decomposition used here is sometimes thought of in terms of a combination of a Newtonian solvent (superscript s) and a polymer additive (superscript p). Using the Johnson–Segalman fluid as an example, then

$$\tau_{ij}^s = 2\mu^s D_{ij}$$

$$\tau_{ij}^p + \lambda \overset{\circ}{\tau}_{ij}^p = 2\mu^p D_{ij} \quad (6.5.15)$$

It is possible to eliminate the partial stresses in (6.5.14) and (6.5.15) to produce a new constitutive relation of the following form

$$\tau_{ij} + \lambda \overset{\circ}{\tau}_{ij} = 2\bar{\mu} \left(D_{ij} + \lambda' \overset{\circ}{D}_{ij} \right) \quad (6.5.16)$$

with $\bar{\mu} = (\mu^s + \mu^p)$ and $\lambda' = \lambda\mu^s/\bar{\mu}$, where λ' is a retardation time. The constitutive equation in (6.5.16) is recognized as a type of Oldroyd fluid; it is, in fact, a reduced form of the eight-constant Oldroyd model [1,5]. For particular choices of the convected derivative in Eq. (6.5.16), specific models can be generated. When $a = 0$ ($\overset{\circ}{\tau}_{ij}$ becomes $\overset{\nabla}{\tau}_{ij}$) then (6.5.14) becomes the Oldroyd B fluid; the case $a = 2$ ($\overset{\circ}{\tau}_{ij}$ becomes $\overset{\Delta}{\tau}_{ij}$) produces the Oldroyd A fluid. In order to ensure a monotonically increasing shear stress the inequality $\mu^s \geq \mu^p/8$ must be satisfied. The stress decomposition employed above can also be used with the Phan Thien–Tanner model to produce a correct shear stress behavior.

In all of the above constitutive equations the material parameters, λ and μ^p , were assumed to be constants. For some constitutive equations the constancy of these parameters leads to material (or viscometric) functions that do not accurately represent the behavior of real elastic fluids. For example, the shear viscosity predicted by a Maxwell fluid is a constant, when in fact viscoelastic fluids normally exhibit a shear thinning behavior. This situation can be remedied to some degree by allowing the parameters λ and μ^p to be functions of the invariants of the rate of deformation tensor as was done for the generalized Newtonian fluid (see Section 6.2.3). Using the upper-convected Maxwell fluid as an example, then

$$\tau_{ij} + \lambda(I_2)\overset{\nabla}{\tau}_{ij} = 2\mu^p(I_2)D_{ij} \quad (6.5.17)$$

where I_2 is the second invariant of the strain rate tensor \mathbf{D} [see Eq. (6.5.11)], $I_2 = 1/2(D_{ij}D_{ij})$. The constitutive equation in (6.5.17) is usually termed a White–Metzner model [29]. White–Metzner forms of other differential models, such as the Oldroyd fluids, have also been developed and used in various situations.

The above list of differential models is by no means exhaustive though it does include many of the constitutive equations that have been used in computational work. Though the equations have been stated without derivation or motivation, they all rest on a reasonable theoretical basis. Some of the equations (e.g., Oldroyd and Maxwell) were developed as generalizations of simple, linear viscoelastic models. Linear viscoelasticity can be formally developed from the previously cited “hierarchical” equations or informally from mechanical analogies and heuristic arguments [1]. Other differential models, such as the Johnson–Segalman and Phan Thien–Tanner fluids, were derived using statistical mechanics ideas and a conceptual (network) model for the microstructure of a viscoelastic fluid. Due to their approximate nature, all of the above models are limited in their ability to represent true viscoelastic behavior; the more complex models provide a reasonable qualitative description of many observed phenomena. A catalog of the strengths, weaknesses, and limitations of these and other differential models can be found in [1,4,5,23].

6.5.2.2 Integral models

An approximate integral model for a viscoelastic fluid represents the extra-stress in terms of an integral over the past history of the fluid deformation. A general form for a single integral model can be expressed as (see [30])

$$\tau_{ij} = \int_{-\infty}^t 2m(t-t')H_{ij}(t,t')dt' \quad (6.5.18)$$

where t is the current time, m is a scalar memory function (or relaxation kernel), and H_{ij} is a nonlinear deformation measure (tensor) between the past time, t' , and t .

There are many possible forms for both the memory function and the deformation measure. Normally the memory function is a decreasing function of the time lapse $s = t - t'$. Typical of such a function is the exponential given by

$$m(t-t') = m(s) = \frac{\mu_0}{\lambda^2} e^{-s/\lambda} \quad (6.5.19)$$

where the parameters μ_0 , λ , and s were defined previously. Like the choice of a convected derivative in a differential model, the selection of a deformation measure for use in Eq. (6.5.18) is somewhat arbitrary. One particular form that has received some attention is given by

$$H_{ij} = \phi_1(I, II)C_{ij}^{-1} + \phi_2(I, II)C_{ij} \quad (6.5.20)$$

In Eq. (6.5.20) C_{ij} is the Cauchy–Green deformation tensor, C_{ij}^{-1} is its inverse, called the Finger tensor, and the ϕ_i are scalar functions of the invariants of the deformation tensors, i.e., $I = \text{tr}(C_{ij}^{-1})$ and $II = \text{tr}(C_{ij})$. The indicated deformation tensor is defined by (see Malvern [31])

$$C_{ij} = \frac{\partial \chi_m}{\partial x_i} \frac{\partial \chi_m}{\partial x_j} \quad (6.5.21)$$

where χ_m is the location at time t' of a fluid particle that is now at position x_i and time t . The form of the deformation measure in Eq. (6.5.20) is still quite general, though specific choices for the functions ϕ_i and the memory function m lead to several well-known constitutive models. Among these are the Kaye–BKZ fluid [1,4,23] and the Lodge rubber-like liquid [32].

As a specific example of an integral model of the type given by the previous equations we consider the Maxwell fluid. Setting $\phi_1 = 1$ and $\phi_2 = 0$ in Eq. (6.5.20) and using the memory function shown in (6.5.19) allow a constitutive equation of the following form

$$\tau_{ij} = \frac{\mu_0}{\lambda^2} \int_{-\infty}^t \exp [-(t-t')/\lambda] \left[C_{ij}^{-1}(t') - \delta_{ij} \right] dt' \quad (6.5.22)$$

In writing Eq. (6.5.22), the Finger deformation tensor in (6.5.20) has been replaced with the more usual Finger strain tensor (e.g., [4,30,31]). The above constitutive equation is an integral equivalent to the upper-convected Maxwell model shown in differential form in Eq. (6.5.10a). Note that in this case the extra-stress is given in an explicit form, though its evaluation requires that the strain history be known for each fluid particle. Also, it is important to emphasize that though the Maxwell fluid has both a differential and integral form, this is not generally true for other constitutive equations.

Since the emphasis here is on differential models we will not dwell on other specific forms of integral constitutive equations. A good source for further discussion of such models is the book by Bird et al. [1]. It is appropriate to note in closing this section that some of the integral models are very good at reproducing realistic viscoelastic behavior [4,23]. However, this improvement in modeling accuracy is offset to a large degree by the difficulties in using such models in general computational schemes. Further comments on computational procedures for integral models will be reserved for a later section.

6.5.3 Boundary Conditions

The equations in (6.5.1) and (6.5.2) are to be solved in conjunction with an appropriate constitutive relation as discussed in the previous section. The set of boundary conditions for the problem at hand consists of either specified velocities or tractions on the boundary of the fluid domain, with one condition needed for each component of velocity. However, there is also an additional set of conditions for the viscoelastic problem due to the “memory” of the fluid. The extra condition, in reality, is an initial condition on the components of the fluid extra-stress, though it is often implemented as a boundary condition, especially for flow-through type boundaries. For time-dependent problems the initial fluid stresses must be given at all points in the problem domain. All problems that contain an inflow boundary require the specification of a strain (stress) history of the fluid crossing the boundary. Completely confined flows do not require any additional data beyond the initial stress-state. Additional comments on boundary conditions can be found in [33].

6.6 Finite Element Model of Differential Form

6.6.1 Preliminary Comments

In this section we will outline typical finite element procedures for the solution of viscoelastic flow problems. At the outset it is important to realize that a standard, well-established computational procedure has not yet evolved for this particular class of flow problems. Numerous different formulations have been proposed [21], but none has proved adequate for the solution of flows with highly elastic fluids. Further comments on the unresolved general problems in this area will be given in a later section.

The wide variety of possible constitutive equations and finite element formulations makes detailed explanation of many specific algorithms impossible in this limited text. Therefore, the detailed discussion will be focused on a single, mixed finite element formulation for a particular differential constitutive equation. Other models and formulations follow easily from this basic outline. A brief description of typical procedures for integral models will also be presented.

6.6.2 Summary of Governing Equations

The governing equations of interest are summarized below in rectangular Cartesian form.

Conservation of Mass

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.6.1)$$

Conservation of Momentum

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = \frac{\partial \sigma_{ij}}{\partial x_j} \quad (6.6.2)$$

Constitutive Equations

$$\sigma_{ij} = -P\delta_{ij} + \tau_{ij} \quad (6.6.3)$$

$$\tau_{ij} = \tau_{ij}^s + \tau_{ij}^p \quad (6.6.4)$$

$$\tau_{ij}^s = 2\mu^e D_{ij} \quad (6.6.5)$$

$$\tau_{ij}^p + \lambda \nabla_p^2 v_i = 2\mu^p D_{ij} \quad (6.6.6)$$

where in Eq. (6.6.2) the body force term has been neglected for simplicity. The constitutive equation employed is the Oldroyd B model. Other models could be included by adding a generic function of the stress to the right-hand side of the constitutive equation. However, the basic algorithm remains unchanged. Also, note that the solvent or viscous stress in (6.6.5) has been expressed in terms of an effective viscosity. In many cases, this is taken as the solvent viscosity ($\mu^e = \mu^s$) as illustrated in equation (6.5.15). In other situations, the stress split is made arbitrary and μ^e represents only a part of the total viscosity. The permissible variation in this definition has been used to advantage in developing solution algorithms.

The above set of equations can be combined in various ways, each of which leads to a slightly different finite element procedure. For example, combining

Eqs.(6.6.3)–(6.6.5) and substituting for σ_{ij} in the momentum balance (6.6.2) leads to the following set of equations:

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.6.7)$$

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = - \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (2\mu^e D_{ij}) + \frac{\partial}{\partial x_j} (\tau_{ij}^p) \quad (6.6.8)$$

$$\tau_{ij}^p + \lambda \tau_{ij}^{\nabla p} = 2\mu^p D_{ij} \quad (6.6.9)$$

This equation set is a natural extension of work on a Newtonian fluid, and was the first one considered by workers in viscoelastic flows [34–36]. Much of the early work centered on the Upper Convected Maxwell (UCM) fluid ($\mu^e = 0$) in the creeping or Stokes flow limit. Under these circumstances, as pointed out by Crochet et al. [5], the mixed finite element formulation associated with (6.6.7)–(6.6.9) has some restrictions with regard to the choice of approximating functions for the extra-stress variables.

Another possible formulation was proposed by Chang, et al. [37] and is the one that will be described here, in some detail. The implicit constitutive equation (6.6.6) can be rearranged and substituted into (6.6.8). The result is a set of equations of the following form

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.6.10)$$

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = - \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (2\bar{\mu} D_{ij}) - \frac{\partial}{\partial x_j} (\lambda \tau_{ij}^{\nabla p}) \quad (6.6.11)$$

$$\tau_{ij}^p + \lambda \tau_{ij}^{\nabla p} = 2\mu^p D_{ij} \quad (6.6.12)$$

where $\bar{\mu} = (\mu^e + \mu^p)$, D_{ij} is defined in (6.5.11), and the stress rate, τ_{ij}^{∇} , is given in (6.5.8a). The inclusion of the stress rate in the momentum equation appears to be of some benefit for numerical computations [5]. The finite element forms of the above two formulations are often termed MIX (mixed method) formulations and form the basis for many of the later advanced developments.

For two-dimensional geometries the equations in (6.6.10)–(6.6.12) provide six relations for the six unknowns, v_1 , v_2 , P , τ_{11}^p , τ_{22}^p , and τ_{12}^p . In an axisymmetric geometry the system would increase to seven equations with the circumferential (hoop) stress, τ_{33}^p , as the seventh unknown. The distinguishing feature of this system, as compared to a Newtonian problem, is the implicit nature of the constitutive equation that forces the extra-stresses to remain as dependent variables.

6.6.3 Finite Element Model

Application of the finite element procedure to the partial differential equations in (6.6.10)–(6.6.12) follows the standard format. The velocity components, pressure, and extra-stress components are approximated by expansions of the form

$$v_i = \sum_{m=1}^M \Psi_m v_i^m = \Psi^T \mathbf{v}_i \quad (6.6.13a)$$

$$P = \sum_{n=1}^N \Phi_n P_n = \Phi^T \mathbf{P} \quad (6.6.13b)$$

$$\tau_{ij} = \sum_{k=1}^K \Pi^k \tau_{ij}^k = \Pi^T \tau_{ij} \quad (6.6.13c)$$

where Φ , Ψ , and Π are vectors of basis functions, \mathbf{v}_i , \mathbf{P} , and τ_{ij} are vectors of nodal unknowns, and M , N , and K indicate the number of nodes at which the various unknowns are defined. The superscript p for the extra-stress has been omitted for clarity. Using the finite element approximations (6.6.13) in standard weighted-integral statements (i.e., weak forms) of the system in (6.6.10)–(6.6.12) (see Chapter 5), the following system of finite element equations can be obtained.

Momentum

$$\begin{aligned} & \left[\int_{\Omega^e} \rho_0 \Phi \Phi^T \mathbf{v}_j \frac{\partial \Phi^T}{\partial x_j} d\mathbf{x} \right] \mathbf{v}_i + \left[\int_{\Omega^e} \mu \frac{\partial \Phi}{\partial x_j} \frac{\partial \Phi^T}{\partial x_j} d\mathbf{x} \right] \mathbf{v}_i + \left[\int_{\Omega^e} \mu \frac{\partial \Phi}{\partial x_j} \frac{\partial \Phi^T}{\partial x_i} d\mathbf{x} \right] \mathbf{v}_j \\ & + \left[\int_{\Omega^e} -\frac{\partial \Phi}{\partial x_i} \Psi^T d\mathbf{x} \right] \mathbf{P} - \left[\int_{\Omega^e} \lambda \frac{\partial \Phi}{\partial x_j} \nabla \tau_{ij} d\mathbf{x} \right] = \left[\oint_{\Gamma^e} \Phi \tau_{ij} n_j ds \right] \end{aligned} \quad (6.6.14)$$

Continuity

$$\left[\int_{\Omega^e} -\Psi \frac{\partial \Phi^T}{\partial x_i} d\mathbf{x} \right] \mathbf{v}_i = \mathbf{0} \quad (6.6.15)$$

Constitutive

$$\begin{aligned} & \left[\int_{\Omega^e} \Pi \Pi^T d\mathbf{x} \right] \tau_{ij} + \left[\int_{\Omega^e} \Pi \lambda \nabla \tau_{ij} d\mathbf{x} \right] - \left[\int_{\Omega^e} \mu^p \Pi \frac{\partial \Phi^T}{\partial x_j} d\mathbf{x} \right] \mathbf{v}_i \\ & - \left[\int_{\Omega^e} \mu^p \Pi \frac{\partial \Phi^T}{\partial x_i} d\mathbf{x} \right] \mathbf{v}_j = \mathbf{0} \end{aligned} \quad (6.6.16)$$

where the steady-state form of the equations has been considered. In arriving at the above equations, the Green–Gauss or divergence theorem has been used to reduce the second-order diffusion terms to first-order terms plus a boundary integral. The appearance of the boundary integral corresponds to the “natural” boundary conditions for the problem. Also, the stress rate has not been written out due to its complexity; the two-dimensional components of the stress rate are written below in terms of the finite element functions,

$$\begin{aligned} \nabla \tau_{11} &= \Phi^T \mathbf{v}_1 \frac{\partial \Pi^T}{\partial x_1} \tau_{11} + \Phi^T \mathbf{v}_2 \frac{\partial \Pi^T}{\partial x_2} \tau_{11} - 2 \frac{\partial \Phi^T}{\partial x_1} \mathbf{v}_1 \Pi^T \tau_{11} - 2 \frac{\partial \Phi^T}{\partial x_2} \mathbf{v}_1 \Pi^T \tau_{12} \\ & \quad (6.6.17) \end{aligned}$$

$$\begin{aligned} \nabla \tau_{22} &= \Phi^T \mathbf{v}_1 \frac{\partial \Pi^T}{\partial x_1} \tau_{22} + \Phi^T \mathbf{v}_2 \frac{\partial \Pi^T}{\partial x_2} \tau_{22} - 2 \frac{\partial \Phi^T}{\partial x_1} \mathbf{v}_2 \Pi^T \tau_{12} - 2 \frac{\partial \Phi^T}{\partial x_2} \mathbf{v}_2 \Pi^T \tau_{22} \\ & \quad (6.6.18) \end{aligned}$$

$$\begin{aligned} \nabla \tau_{12} &= \Phi^T \mathbf{v}_1 \frac{\partial \Pi^T}{\partial x_1} \tau_{12} + \Phi^T \mathbf{v}_2 \frac{\partial \Pi^T}{\partial x_2} \tau_{12} - \frac{\partial \Phi^T}{\partial x_1} \mathbf{v}_2 \Pi^T \tau_{11} - \frac{\partial \Phi^T}{\partial x_2} \mathbf{v}_1 \Pi^T \tau_{22} \\ & - \left(\frac{\partial \Phi^T}{\partial x_1} \mathbf{v}_1 + \frac{\partial \Phi^T}{\partial x_2} \mathbf{v}_2 \right) \Pi^T \tau_{12} \end{aligned} \quad (6.6.19)$$

Equations (6.6.17)–(6.6.19) can be substituted directly into (6.6.14) and (6.6.16) to produce the complete form of the finite element equations.

Once the form of the interpolation functions Φ , Ψ , and Π is specified (i.e., a particular element is selected), and the geometry of the element is known (i.e., x_i) then the integrals in (6.6.14)–(6.6.16) may be evaluated to produce the required coefficient matrices. The integrals are evaluated via a numerical quadrature procedure, as discussed in Chapters 2 and 3. The discrete system is given by the following matrix equations (it is common to write the momentum and continuity equations as a single system):

$$\begin{aligned} & \left[\begin{array}{ccc} \mathbf{C}_i(\mathbf{v}_i) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_i(\mathbf{v}_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{P} \end{Bmatrix} + \left[\begin{array}{ccc} 2\mathbf{K}_{11} + \mathbf{K}_{22} & \mathbf{K}_{21} & \mathbf{Q}_1 \\ \mathbf{K}_{12} & \mathbf{K}_{11} + 2\mathbf{K}_{22} & \mathbf{Q}_2 \\ \mathbf{Q}_1^T & \mathbf{Q}_2^T & \mathbf{0} \end{array} \right] \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{P} \end{Bmatrix} \\ & + \left[\begin{array}{ccc} \mathbf{R}_1^{11}(\mathbf{v}_1, \mathbf{v}_2) & \mathbf{R}_1^{22}(\mathbf{v}_1, \mathbf{v}_2) & \mathbf{R}_1^{12}(\mathbf{v}_1, \mathbf{v}_2) \\ \mathbf{R}_2^{11}(\mathbf{v}_1, \mathbf{v}_2) & \mathbf{R}_2^{22}(\mathbf{v}_1, \mathbf{v}_2) & \mathbf{R}_2^{12}(\mathbf{v}_1, \mathbf{v}_2) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \begin{Bmatrix} \tau_{11} \\ \tau_{22} \\ \tau_{12} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{0} \end{Bmatrix} \quad (6.6.20) \end{aligned}$$

and for the constitutive equation

$$\begin{aligned} & \left[\begin{array}{ccc} \mathbf{N} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} \end{array} \right] \begin{Bmatrix} \tau_{11} \\ \tau_{22} \\ \tau_{12} \end{Bmatrix} - \left[\begin{array}{ccc} 2\mathbf{D}_1(\mathbf{v}_1) & \mathbf{0} & 2\mathbf{D}_2(\mathbf{v}_1) \\ \mathbf{0} & 2\mathbf{D}_2(\mathbf{v}_2) & 2\mathbf{D}_1(\mathbf{v}_2) \\ \mathbf{D}_1(\mathbf{v}_2) & \mathbf{D}_2(\mathbf{v}_1) & \mathbf{D}_i(\mathbf{v}_i) \end{array} \right] \begin{Bmatrix} \tau_{11} \\ \tau_{22} \\ \tau_{12} \end{Bmatrix} \\ & + \left[\begin{array}{ccc} \mathbf{C}_i^*(\mathbf{v}_i) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_i^*(\mathbf{v}_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_i^*(\mathbf{v}_i) \end{array} \right] \begin{Bmatrix} \tau_{11} \\ \tau_{22} \\ \tau_{12} \end{Bmatrix} - \left[\begin{array}{ccc} 2\mathbf{L}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2\mathbf{L}_2 & \mathbf{0} \\ \mathbf{L}_1 & \mathbf{L}_2 & \mathbf{0} \end{array} \right] \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{P} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \quad (6.6.21) \end{aligned}$$

where sum on repeated indices is implied. The coefficient matrices shown in Eqs. (6.6.20) and (6.6.21) are defined by

$$\begin{aligned} \mathbf{C}_i(\mathbf{v}_j) &= \int_{\Omega^e} \rho_0 \Phi \Phi^T \mathbf{v}_j \frac{\partial \Phi^T}{\partial x_i} d\mathbf{x}, \quad \mathbf{K}_{ij} = \int_{\Omega^e} \bar{\mu} \frac{\partial \Phi}{\partial x_i} \frac{\partial \Phi^T}{\partial x_j} d\mathbf{x} \\ \mathbf{Q}_i &= - \int_{\Omega^e} \frac{\partial \Phi}{\partial x_i} \Psi^T d\mathbf{x}, \quad \mathbf{F}_i = \oint_{\Gamma^e} \Phi \tau_{ij} n_j ds \\ \mathbf{N} &= \int_{\Omega^e} \Pi \Pi^T d\mathbf{x}, \quad \mathbf{C}_i^*(\mathbf{v}_j) = \int_{\Omega^e} \lambda \Pi \Phi^T \mathbf{v}_j \frac{\partial \Pi^T}{\partial x_i} d\mathbf{x} \\ \mathbf{D}_i(\mathbf{v}_j) &= \int_{\Omega^e} \lambda \Pi \frac{\partial \Phi^T}{\partial x_i} \mathbf{v}_j \Pi^T d\mathbf{x}, \quad \mathbf{L}_i = \int_{\Omega^e} \mu^p \Pi \frac{\partial \Phi^T}{\partial x_i} d\mathbf{x} \end{aligned} \quad (6.6.22)$$

and the matrices \mathbf{R}_k^{ij} are defined by

$$\begin{aligned} \mathbf{R}_1^{11} &= -\mathbf{S}_{11}(\mathbf{v}_1) - \mathbf{S}_{12}(\mathbf{v}_2) + 2\mathbf{T}_{11}(\mathbf{v}_2) + \mathbf{T}_{21}(\mathbf{v}_2), \quad \mathbf{R}_1^{22} = \mathbf{T}_{22}(\mathbf{v}_1) \\ \mathbf{R}_1^{12} &= -\mathbf{S}_{21}(\mathbf{v}_1) - \mathbf{S}_{22}(\mathbf{v}_2) + 2\mathbf{T}_{12}(\mathbf{v}_1) + \mathbf{T}_{21}(\mathbf{v}_1) + \mathbf{T}_{22}(\mathbf{v}_2) \\ \mathbf{R}_2^{11} &= \mathbf{T}_{11}(\mathbf{v}_2), \quad \mathbf{R}_2^{22} = -\mathbf{S}_{21}(\mathbf{v}_1) - \mathbf{S}_{22}(\mathbf{v}_2) + \mathbf{T}_{12}(\mathbf{v}_1) + 2\mathbf{T}_{22}(\mathbf{v}_2) \\ \mathbf{R}_2^{12} &= -\mathbf{S}_{12}(\mathbf{v}_2) - \mathbf{S}_{11}(\mathbf{v}_1) + 2\mathbf{T}_{21}(\mathbf{v}_2) + \mathbf{T}_{12}(\mathbf{v}_2) + \mathbf{T}_{11}(\mathbf{v}_1) \end{aligned} \quad (6.6.23)$$

The \mathbf{S} and \mathbf{T} matrices are defined as

$$\mathbf{S}_{ij}(\mathbf{v}_k) = \int_{\Omega^e} \lambda \frac{\partial \Phi}{\partial x_i} \Phi^T \mathbf{v}_k \frac{\partial \Pi^T}{\partial x_j} d\mathbf{x}, \quad \mathbf{T}_{ij}(\mathbf{v}_k) = \int_{\Omega^e} \lambda \frac{\partial \Phi}{\partial x_i} \frac{\partial \Phi^T}{\partial x_j} \mathbf{v}_k \Pi^T d\mathbf{x} \quad (6.6.24)$$

Finally, Eqs. (6.6.20) and (6.6.21) can be expressed symbolically as

$$\mathbf{C}(\mathbf{v})\mathbf{U} + \mathbf{K}\mathbf{U} + \mathbf{R}(\mathbf{v})\tau = \mathbf{F} \quad (6.6.25)$$

$$\mathbf{N}\tau - \mathbf{D}(\mathbf{v})\tau + \mathbf{C}^*(\mathbf{v})\tau - \mathbf{L}\mathbf{U} = \mathbf{0} \quad (6.6.26)$$

where the vector of unknowns is $\mathbf{U}^T = \{\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{P}^T\}$. Equations (6.6.25) and (6.6.26) represent the finite element equations for a viscoelastic (Oldroyd B) fluid for the steady-state case; very little consideration has been given to the time-dependent case. The displayed matrix system is highly nonlinear and fully coupled. Equation (6.6.25) is recognized as the standard form for a Newtonian problem with the addition of the $\mathbf{R}(\mathbf{v})\sigma$ term which comes from the stress rate. Equation (6.6.26) is the finite element analog of the implicit constitutive equation and is seen to be an equation of the advection type for the extra-stress.

Before discussing solution procedures for the matrix system, some comments on the approximations used in (6.6.13) are required. In defining the approximations to the dependent variables all of the basis functions were taken to be different. From experience with mixed methods for Newtonian problems it is well-known that the pressure approximation should be of lower order than the velocity approximation. For most applications in viscoelastic simulations this condition has been met by using quadratic functions for the velocity and linear approximations for the pressure. The choice of an appropriate approximation for the extra-stress is less clearly defined. As mentioned earlier, the problem formulation based on Eqs. (6.6.7)–(6.6.9) has a restriction on the permissible extra-stress basis functions. A solvability condition for the resulting matrix problem requires that the extra-stresses be approximated to the same order as the velocity components [5].

In the present formulation there are no such restrictions and the choice of the interpolation functions used for stress components is quite arbitrary. Heuristic arguments were made for having the extra-stress and pressure be of the same order since they are both “stress-like” variables. However, many studies using both linear and quadratic stress approximations have been produced, and in virtually all cases the mixed method has been limited to flows with small fluid elasticity. In the mid 1980s Marchal and Crochet [38,39] demonstrated that some improvement in behavior of a mixed method could be achieved by using higher-order approximations for the extra-stress. Their general criterion was that the stress approximation should be of sufficient order to represent velocity gradients. They achieved this approximation in practical computations by subdividing each velocity element into 3×3 or 4×4 bilinear subelements on which the extra-stresses were interpolated. This procedure, along with some upwinding techniques, allowed computations to be made at large values of fluid elasticity. The benefits of higher-order stresses were also examined by Rao and Finlayson [40], where a cubic stress approximation was employed. Despite this and more recent work, the problem of stress approximation is not fully understood nor resolved. Further comments will be made in Section 6.9.

6.6.4 Solution Methods

To discuss possible solution procedures for the discretized viscoelastic problem in (6.6.25) and (6.6.26), it is convenient to write the matrix system in a more compact form. Combining the incompressibility constraint with the momentum equation allows the following system to be derived

$$\mathbf{K}_{\mathbf{vv}}(\mathbf{v})\mathbf{U} + \mathbf{K}_{\mathbf{v}\tau}(\mathbf{v})\tau = \mathbf{F} \quad (6.6.27a)$$

$$\mathbf{K}_{\tau\tau}(\mathbf{v})\tau + \mathbf{K}_{\tau\mathbf{v}}\mathbf{U} = \mathbf{0} \quad (6.6.27b)$$

where the vector \mathbf{U} now represents the velocity and pressure variables. The $\mathbf{K}_{\mathbf{vv}}$ term contains the advection, diffusion, pressure gradient, and incompressibility terms from the momentum equation. The second equation in (6.6.27) is a rearranged version of the constitutive equation in (6.6.26).

As with any strongly coupled system of equations, there are two basic approaches to the solution of the system given in (6.6.27): artificially decouple the system and employ a cyclic solution process on the decoupled equations, or directly solve the combined equation set. Though split equation algorithms are generally less effective than combined schemes, they have the potential for some significant computational benefits in the present system.

Consider the simplest cyclic Picard method available for (6.6.27), which is written here for cycle $n + 1$ as

$$\text{Step 1: } \mathbf{K}_{\mathbf{vv}}(\mathbf{v}^n)\mathbf{U}^{n+1} = \mathbf{F} - \mathbf{K}_{\mathbf{v}\tau}(\mathbf{v}^n)\tau^n \quad (6.6.28a)$$

$$\text{Step 2: } \mathbf{K}_{\tau\tau}(\mathbf{v}^{n+1})\tau^{n+1} = -\mathbf{K}_{\tau\mathbf{v}}\mathbf{U}^{n+1} \quad (6.6.28b)$$

At each step of (6.6.28a) a matrix solution is required. However, the matrices in (6.6.28a) are significantly smaller than the matrix associated with the combined system in (6.6.27). Reduction of the size of the matrix problem is especially important for the viscoelastic problem since the number of unknowns per nodal point is very large (e.g., the number of unknowns in a viscoelastic flow can be more than double those of a Newtonian problem for the same finite element mesh). Unfortunately, little work has been done to improve the relatively poor iterative (convergence) performance of methods such as (6.6.28); virtually all viscoelastic problems to date have been solved using a combined equation method.

If the solution to the coupled matrix system is to be obtained by simultaneous solution of the equations in (6.6.26), then it is appropriate to rewrite the system in the operator form

$$\hat{\mathbf{K}}(\hat{\mathbf{U}})\hat{\mathbf{U}} = \hat{\mathbf{F}} \quad (6.6.29)$$

where $\hat{\mathbf{U}}$ now contains all of the problem unknowns. The nonlinear problem shown in (6.6.19) may be solved using any of the fixed point iteration schemes. The Picard method

$$\hat{\mathbf{K}}(\hat{\mathbf{U}}^n)\hat{\mathbf{U}}^{n+1} = \hat{\mathbf{F}} \quad (6.6.30)$$

was used by Coleman [36] for several viscoelastic flow problems. The most commonly used method, however, is Newton's method [26,41–44], which for (6.6.19) can be written as

$$\mathbf{J}(\hat{\mathbf{U}}^n)[\hat{\mathbf{U}}^{n+1} - \hat{\mathbf{U}}^n] = -\hat{\mathbf{K}}(\hat{\mathbf{U}}^n)\hat{\mathbf{U}}^n + \hat{\mathbf{F}} \quad (6.6.31a)$$

where

$$\mathbf{J}(\hat{\mathbf{U}}^n) = \frac{\partial}{\partial \hat{\mathbf{U}}} [\hat{\mathbf{K}}(\hat{\mathbf{U}})\hat{\mathbf{U}} - \hat{\mathbf{F}}] \Big|_{\hat{\mathbf{U}}^n} \quad (6.6.31b)$$

The Jacobian \mathbf{J} for the system given in (6.6.25) and (6.6.26) is very complex but can, in fact, be evaluated analytically. Newton's method converges rapidly and, in general, works very well for flows involving materials with low to moderate elasticity. For highly elastic fluids none of the above algorithms allow convergence, as will be discussed subsequently.

There has been no discussion of solution methods for transient flows, mainly due to the lack of previous work in this area. It is certainly possible to extend the steady-state formulation of Section 6.5 into the time-dependent regime. Following the ideas developed for Newtonian problems [45], the resulting ordinary differential equation system could be integrated in time using an implicit method, such as the trapezoid rule. However, there are two basic difficulties in such an approach. First, since the matrix system for a viscoelastic formulation is so large, the repeated solution of an implicit integration algorithm may not be economically justifiable. Splitting the system during the integration procedure [as in (18)] could alleviate some of the computational burden but would increase concerns regarding stability. The second concern is with the constitutive equation and the fact that the time scale for the material response can be significantly different than the characteristic time for the flow process [46]. Such a situation suggests that a single integration time scale may be inappropriate or not economical; the momentum and constitutive equations may require quite different integration procedures. A method developed by Gartling [47], which is similar to the work of Fortin and Fortin [48], addresses some of the questions regarding transient flows.

6.7 Additional Models of Differential Form

The finite element models developed in Section 6.6 formed the basis for viscoelastic simulations until the later part of the 1980s. Various perturbations on the basic mixed method were attempted to improve computational performance, improve convergence, and most importantly, provide solutions at increasing levels of fluid elasticity. This last problem, the "high Weissenberg number problem" (see Section 6.9), was particularly troublesome as the MIX type algorithms all failed at moderate levels of fluid elasticity. The cause of this failure is complex; some of the issues associated with the phenomena are discussed in a later section.

A number of other formulations have been developed to try to alleviate the high Weissenberg difficulties. In each case, the main objective has been to maintain the momentum and continuity equations as a fully elliptic equation system with well-understood mathematical and numerical properties (e.g., saddle point problem, LBB condition, dominant advection terms, etc.). Also, the constitutive equation treatment has been altered to better account for the hyperbolic nature of the equation set. The following subsections describe the main algorithms developed from these basic ideas. The finite element form of these algorithms will not be detailed as the computational implementation follows directly from the previous section.

6.7.1 Explicitly Elliptic Momentum Equation Method

The first significant reformulation of the viscoelastic flow problem is due to King et al. [49] and is termed the explicitly elliptic momentum equation (EEME) method. It was proven by Joseph et al. [50] that under certain conditions the equation set for the UCM model formulated via a MIX type method could undergo a change of type (change in the characteristics of the equations from imaginary to real) and lose the dominant elliptic character of the system. The essence of the analysis was a criterion on the dependent variables that must be met in order to ensure a consistent and stable numerical solution and prevent any change in the basic characteristics of the equation set. The EEME method was the result of a search for a method that would explicitly satisfy the change of type criterion.

The formulation begins with the basic continuity, momentum, and UCM constitutive models

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.7.1)$$

$$-\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\tau_{ij}^p) = 0 \quad (6.7.2)$$

$$\tau_{ij}^p + \lambda \nabla_{ij}^p = 2\mu^p D_{ij} \quad (6.7.3)$$

where the momentum equation is simplified by considering only steady Stokes flow and no body forces. Taking the divergence of the stress in (6.7.3) and substituting into (6.7.2) produces a momentum equation of the form

$$-\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (2\mu^p D_{ij}) - \frac{\partial}{\partial x_j} (\lambda \nabla_{ij}^p) = 0 \quad (6.7.4)$$

This equation is similar in form to (6.6.11) though for a different constitutive model. Equation (6.7.4) may be rewritten using the definition of the upper convected derivative as

$$-\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\mu^p L_{ij} + \mu^p L_{ji}) - \frac{\partial}{\partial x_j} \left(\lambda v_k \frac{\partial \tau_{ij}^p}{\partial x_k} - L_{ik} \tau_{kj}^p - L_{jk} \tau_{ki}^p \right) = 0 \quad (6.7.5)$$

Defining a new tensor variable, χ_{ij} , based on the change of type criterion, the momentum equation may be reformed as

$$\frac{\partial}{\partial x_k} (\chi_{ij} L_{jk}) + L_{ij} \frac{\partial \chi_{jk}}{\partial x_k} = \frac{\partial q}{\partial x_k} \quad (6.7.6)$$

where

$$\chi_{ij} = \delta_{ij} - \lambda \tau_{ij}^p \quad (6.7.7)$$

$$q = P + \lambda v_j \frac{\partial P}{\partial x_j} \quad (6.7.8)$$

The EEME method consists of Eqs. (6.7.1), (6.7.3), and (6.7.6) with the definitions in (6.7.7) and (6.7.8). This formulation is similar in development to the

MIX method of Section 6.6. The major difference between the methods is that the EEME was conceived for an UCM model (no solvent or viscous stress component) while the MIX methods were usually employed with constitutive models having a viscous component. A MIX formulation for an UCM has been shown by a number of investigators to be very limited in terms of maximum attainable Weissenberg number. The advantage of the EEME method is the enforced elliptic nature of the momentum equation in (6.7.6), which is maintained as the fluid elasticity is increased. The tensor variable χ_{ij} , is analogous to an anisotropic diffusivity; the variable q is a modified pressure. The lower-order derivative terms are similar to advection terms. The momentum equation is thus similar to a Navier Stokes problem. In a finite element procedure the modified pressure q is used (and interpolated) in the same way as in a Newtonian formulation.

The EEME formulation was used with a finite element implementation, primarily by the MIT group [49,51], to solve a number of viscoelastic problems. A variety of stress approximations were tested and found to produce consistent and convergent results at reasonably large values of fluid elasticity. The failure of the EEME method at higher values of elasticity was attributed to the poor resolution of elastic boundary layers. The low-order derivative terms in (6.7.6) become dominant at high elasticity, much like the advection terms in a high Reynolds number viscous flow. Note that the EEME method can be extended to constitutive models with a viscous stress contribution, such as the Oldroyd models. However, its inherent design makes it most relevant to situations where the viscous stress is much less than the viscoelastic stress.

6.7.2 Elastic Viscous Stress Splitting Method

Despite the improvements seen with the EEME algorithm, its focus on UCM type models reduced its universality. However, the philosophy of the EEME formulation, involving a strongly elliptic momentum/continuity system and a properly represented hyperbolic constitutive equation, was accepted as a way to improve viscoelastic simulation methods. A number of investigators had also demonstrated that the addition of a viscous contribution to the stress model was computationally helpful. The viscous contribution regularizes the mathematical problem and eliminates the deleterious change of equation type problem that plagues the UCM model.

The elastic viscous split stress (EVSS) method was developed [51] to generalize the EEME method and focus on constitutive models containing a purely viscous component. The formulation followed from previous work on stress splitting, but changed significantly the method for handling higher-order derivatives in the resulting equations. The development of the method again begins with the steady Stokes form of the continuity and momentum equations. In this illustration the Oldroyd model will be used as the constitutive equation with a viscous component.

The basic equations are

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.7.9)$$

$$-\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} = 0 \quad (6.7.10)$$

$$\tau_{ij}^p + \lambda \nabla_{ij}^p = 2\mu^p D_{ij} \quad (6.7.11)$$

$$\tau_{ij} = 2\mu^e D_{ij} + \tau_{ij}^p \quad (6.7.12)$$

For the EVSS scheme, a change of variable is introduced with

$$S_{ij} = \tau_{ij}^p - 2\mu^p D_{ij} \quad (6.7.13)$$

Substituting (6.7.13) and (6.7.12) into the momentum equation and (6.7.13) into the constitutive relation yields the following momentum and stress rate equations

$$-\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (2\bar{\mu}^e D_{ij}) + \frac{\partial S_{ij}}{\partial x_j} = 0 \quad (6.7.14)$$

$$S_{ij}^p + \lambda S_{ij}^p = 2\lambda\mu^p \nabla_{ij} \quad (6.7.15)$$

where the viscosity $\bar{\mu}^e$ is the sum of μ^e and μ^p .

The standard EVSS formulation thus consists of equations (6.7.9), (6.7.14), and (6.7.15) with the definitions for S_{ij} and D_{ij} . This system maintains the elliptic nature of the continuity and momentum equation set but introduces a complication for the finite element implementation. The upper convected derivative of the rate of deformation tensor D_{ij} in (6.7.15) contains second derivatives of the velocity field. A number of approaches have been demonstrated to circumvent this problem and include an integration by parts [52] leading to an inconvenient boundary term, the retention of D_{ij} as an unknown and use of a projection method [51], and a discrete version of retaining D_{ij} as an unknown variable [52]. This last method is termed a Discrete EVSS (DEVSS) and has found considerable favor within the viscoelastic community. Other perturbations and refinements of the EVSS formulation have recently been demonstrated. An adaptive procedure has been employed in [53] and [54] to split the stress leading to methods termed Adaptive Viscous Split Stress (AVSS) and Discrete AVSS (DAVSS). The effective viscosity μ^e in (6.7.12) is allowed to vary such that a balance between the viscous and viscoelastic stress levels is maintained and thereby keep the solution procedure under control.

It must be emphasized that the stress splitting methods are beneficial in maintaining the mathematical properties of the momentum/continuity problem; the proper treatment of the hyperbolic constitutive equation remains an important part of the overall algorithm. In most of the EEME and EVSS methods, some type of upwind method is employed in the formulation of the constitutive equations. The Streamline Upwind Petrov-Galerkin (SUPG) method mentioned in Chapter 4 is one popular approach to the convection operator. The discontinuous Galerkin (DG) method, first outlined by Lesaint and Raviart [55], is another method for computationally treating first-order hyperbolic equations. This method was first applied to the viscoelastic problem in [48].

The additional formulations outlined above have improved considerably the finite element computational capabilities for viscoelastic simulations. However, substantial work remains to be done, since most of these methods are not affordable in three-dimensional, transient, and/or multiple relaxation time simulations.

6.8 Finite Element Model of Integral Form

Finite element procedures for viscoelastic fluids of the integral type have generally received less attention than those for differential models. This can be attributed primarily to the difficulties in implementation of an integral constitutive equation. In this section we will outline the steps needed to use such a model and provide references to work that describes the details of several algorithms.

For consideration of a specific example the Maxwell fluid described by Eq. (6.5.22) will be used. The basic equations are rewritten here for reference

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (6.8.1)$$

$$\rho \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\tau_{ij}) \quad (6.8.2)$$

$$\tau_{ij} = \frac{\mu_0}{\lambda^2} \int_{-\infty}^t \exp [-(t-t')/\lambda] [C_{ij}^{-1}(t') - \delta_{ij}] dt' \quad (6.8.3)$$

Note that the stress decomposition procedure of Eq. (6.5.13) can also be used with an integral model; its use in the present case would lead to an integral form of the Oldroyd B fluid.

The balance laws, (6.8.1) and (6.8.2), can be discretized in the standard way, using a mixed (velocity, pressure) finite element procedure. The result of such a process is the following familiar set of matrix equations

$$\mathbf{Q}^T \mathbf{v} = 0 \quad (6.8.4)$$

$$\mathbf{C}_i(\mathbf{v})\mathbf{v} + \mathbf{Q}_i\mathbf{P} = \mathbf{F}_i - \int_{\Omega^e} \frac{\partial \Phi}{\partial x_j} \tau_{ij} d\mathbf{x} \quad (6.8.5)$$

which are written here for the steady-state case. Note that in writing Eq. (6.8.5), Φ is assumed to be the velocity basis (weighting) function; an integration by parts is responsible for the form of the stress integral. The arrangement of terms in (6.8.5) anticipates the use of an iterative solution procedure. In the first part of the iteration scheme, Eqs. (6.8.4) and (6.8.5) can be solved for the velocity and pressure given a known (or predicted) stress field. The task then is to develop an algorithm for the prediction of the extra-stress given a velocity field from (6.8.5).

From Eq. (6.8.3), it is apparent that to predict the extra-stress at a particular location, say, a nodal point or quadrature point, the strain history for that point must be evaluated over past time. This computation is usually done in three stages. In the first stage, the particle path (streamline for steady flow) through the point in question is constructed. Procedures for this construction range from the Lagrangian-like computation of displacement fields for each finite element [56] to the direct computation of the streamlines [57]. The construction of the trajectory allows the previous locations of the particle (i.e., node or quadrature point) to be established. The second stage requires the evaluation of the strain measure (e.g., C_{ij}^{-1}) along the particle trajectory. The final stage uses a numerical (Laguerre) quadrature to evaluate the history integral (6.8.3) and produce the extra-stress at a particular location.

The details of the particle tracking and strain evaluation depend directly on the particular type of finite element approximation used for the velocity field. A discussion of several different approaches may be found in the work of Viriyayuthakorn and Caswell [56], Dupont et al. [57] and Bernstein et al. [58]. Once a prediction of the stress field has been made the stress forcing function in (6.8.5) can be evaluated and a new velocity obtained. This cyclic procedure is continued until convergence of the dependent variables is achieved. Details regarding the implementation of the various iteration procedures may also be found in [56,57].

There are several drawbacks to the integral methods as currently implemented. The basic structure of the equations in (6.8.1)–(6.8.3) does not permit the easy use of higher-order iterative methods, such as Newton's method. Convergence is thus limited to a linear rate, which can be a severe computational burden. The particle tracking algorithms appear to be the weak point of the overall procedure. For recirculating flows the methods must be of sufficient accuracy to ensure the closure of streamlines. This is a rather difficult task when dealing with approximate numerical schemes and an incompressible fluid. Also, the extension of the particle tracking ideas to time-dependent flows would appear to be very difficult. Despite these disadvantages the work on integral constitutive equations should continue to increase, since these models are somewhat better at predicting realistic viscoelastic behavior than their differential counterparts.

6.9 Unresolved Problems

6.9.1 General Comments

To conclude this brief treatment of finite element methods for viscoelastic flows, it is appropriate to outline some of the unresolved problems and areas of research in this field. Numerical difficulties in the simulation of Newtonian fluids are generally associated with the occurrence of large values of some relevant nondimensional parameter, e.g., the Reynolds number, Peclet number, or Rayleigh number. The situation is quite similar for the viscoelastic problem though the nondimensional parameters are not as familiar.

As noted in a previous section, there are two time scales involved in a viscoelastic flow: the time constant for the material (relaxation time, λ) and the characteristic time for the flow process, t_f . The ratio of these time parameters is called the Deborah number, $De = \lambda/t_f$. Very often the characteristic time for the flow is taken as the reciprocal of a typical (wall) shear rate, so that $De = \lambda\dot{\gamma}_w$. The nondimensionalization of simple constitutive equations, such as the Maxwell model, can lead to the definition of a second characteristic parameter. The Weissenberg number is typically defined as $W = \lambda U/L$ where U and L are a characteristic velocity and length scale for the flow. In many cases the Weissenberg and Deborah numbers can have the same definition; they also tend to be used interchangeably, which leads to some confusion in the literature.

The Deborah and Weissenberg numbers both indicate the relative importance of fluid elasticity for a given flow problem. High values of W (or De) indicate a material response that is very solid-like, while low values of the parameter represent “small” departures from normal viscous fluid behavior.

The continuing outstanding problem in the numerical simulation of viscoelastic fluids is the “high Weissenberg (or Deborah) number problem.” In essence, this phrase refers to the observed behavior that, as the Weissenberg number is increased for a given problem, a critical value is reached (W_{cr}) beyond which the numerical algorithm fails. Failure in the present context normally consists of an initial degradation in the solution (e.g., spatial oscillations in the dependent variables, especially the stresses) followed by divergence of the solution algorithm. The high Weissenberg limit is a universal phenomenon in that it has been observed in all weak form Galerkin finite element models and finite difference methods and for all types of constitutive equations. The critical value of the Weissenberg number is very sensitive to the particular flow geometry, numerical algorithm, constitutive equation, and computational mesh. Unfortunately, from an engineering analysis viewpoint the limiting value W_{cr} is usually too small to be of practical interest (i.e., they are perturbations to the Newtonian solution).

The failure of the most numerical models to provide solutions for highly elastic fluids has spawned a great deal of research activity as evidenced by the numerous publications and several international workshops on the subject [59]. Though the basic causes of the problem have yet to be fully identified, a number of issues have been explored that contribute to the basic understanding of several parts of this complex problem. A summary of some of these topics is given below. For the reader interested in further details regarding the many facets of the “high Weissenberg number problem” the text by Crochet et al. [5], the reviews by Crochet [60] and Keunings [61], and recent issues of the *Journal of Non-Newtonian Fluid Mechanics* are recommended. An excellent review of the current computational state-of-the-art has been compiled by Baaijens [62] and a review of the more mathematical issues associated with modeling viscoelastic fluids can be found in the work of Renardy [63]. Recent studies using the least-squares finite element models and k -version finite element models [64–73] have the potential to solve high Weissenberg number problems.

6.9.2 Choice of Constitutive Equation

Much of the early numerical work on viscoelastic flows was carried out using the Maxwell constitutive model. This was mainly due to the relative simplicity of the equation and the belief that it was prototypical of the more complex constitutive relations. Work by Phan Thien [74–76] demonstrated that for some flows there are inherent instabilities in the Maxwell constitutive model that occur above a critical Weissenberg number. Such a proof demonstrates one of the subtle dangers of this class of problems — the use of approximate constitutive equations that are difficult or impossible to test analytically for non-physical behavior in complex flows.

Many recent investigations have abandoned the Maxwell/Oldroyd family of fluid models in favor of more realistic (and complex) equations such as those due to Johnson–Segalman and Phan Thien–Tanner. There is, of course, no proof that the more complex models do not also contain built-in instabilities that may or may not be physically realistic. Despite the results shown in [48,56,57] most investigators do not feel that the high Weissenberg limit is due solely to the choice of the constitutive model, though it is certainly regarded as a contributing factor. It should also be noted that virtually all numerical work has considered only single relaxation time

models, while it is clear that most constitutive models require multiple relaxation times to approximate realistic fluid behavior. The large computational burden for single relaxation time models has prevented more realistic relaxation spectra from being investigated.

6.9.3 Uniqueness and Existence of Solutions

The complexity of the equations describing viscoelastic flows precludes any proof of solution uniqueness or existence. Indeed, from the behavior of the simple Newtonian fluid it is to be expected that bifurcations and multiple solutions will exist for flows of these types of fluids. Various investigators [26,41,46] working with several different geometries and constitutive relations have demonstrated the occurrence of limit points (loss of solution with increasing Weissenberg number) and bifurcation points (multiple solutions) in numerical simulations. Typically such features of the flow are computed using a continuation (or imbedding) procedure and Newton's method. Corner singularities routinely cause numerical difficulties and mathematical uncertainty in viscoelastic flows.

The question remains as to whether such behavior is an attribute of the physical model (differential constitutive equation) or is an artifact of the numerical discretization process. Keunings [77] and Debbaut and Crochet [78] have provided arguments for such phenomena being of a numerical nature in at least two particular simulations. Their findings, however, do not eliminate the concern over encountering true bifurcations or limit points under other circumstances.

Various experimental investigations of viscoelastic flows, such as [79], have demonstrated the occurrence of time-dependence and three-dimensional flow structure in relatively simple two-dimensional geometries. With the numerical emphasis on two-dimensional, steady simulations, this type of behavior is clearly not predictable. Issues such as these emphasize the difficulty in discriminating between behavior of the discretized equations, the continuum equations, and the physical system.

6.9.4 Numerical Algorithm Problems

A particularly disturbing behavior that was observed in many numerical simulations is a pathology associated with mesh refinement studies. Typically, a solution on one mesh discretization could be obtained up to a critical Weissenberg number of, say, W_{cr}^{coarse} ; a refined mesh solution would produce a solution up to W_{cr}^{fine} , where $W_{cr}^{fine} < W_{cr}^{coarse}$. Stated differently, the finite element procedure did not appear to converge as the mesh was refined — a fundamental property of a stable algorithm. Such behavior was reported for several two-dimensional simulations [5,44] and studied in detail in a one-dimensional flow [42].

In [42] it was conjectured that this behavior was due to an improper choice of the approximating functions for the extra-stress. Little theoretical guidance is available for the choice of stress approximation relative to the velocity or pressure; the LBB condition must usually still be satisfied for the momentum and continuity equations. As noted previously, the work of Marchal and Crochet [38,39] and others [40] produced some support for the need for higher-order stress approximations. Such anomalous convergence behavior could also stem from the use of an inappropriate solution algorithm or formulation for the constitutive equation.

Again, improvements in the treatment of the hyperbolic nature of the constitutive equation seem to yield improved computations.

Extensive work in this area by Dupret, et al. [80] eventually led to a better understanding of some aspects of the problem. Viscoelastic flows are particularly difficult to model due to: (a) the presence of normal stresses, which lead to very high stress gradients and thin elastic boundary layers, (b) the occurrence of singularities and stress concentrations, and (c) the strong hyperbolic character of the constitutive equation. Large stress gradients and singularities place a high demand on mesh resolution for the stress variables. The hyperbolic nature of the constitutive equation requires that an appropriate numerical method be used for the equation set; upwind techniques or Petrov–Galerkin methods are one possible approach to this problem. When any of the above demands are not met, as was the case in most standard, mixed finite element methods, the discrete equation systems for the viscoelastic problem could produce improper evolutionary behavior. Dupret, et al. [80] tracked this behavior by monitoring a characteristic tensor and correlating its loss of positive definiteness with the occurrence of numerical limit points in the solution. The combination of higher-order stress approximations and upwind techniques for the constitutive equation proved to stabilize the method, and allow solutions at very high Weissenberg numbers with appropriate mesh convergence behavior. However, upwinding and Petrov–Galarkin methods are not without their own problems. The addition of artificial diffusion with these methods must be carefully evaluated to ensure that a true solution is not masked/ altered by the numerical technique. To some extent, the use of streamline upwind methods or discontinuous Galerkin methods has been beneficial in this regard.

A proof by Fortin and Pierre [81] showed under what conditions a higher-order stress approximation was required and confirmed the method developed by Marchal and Crochet [39]. This proof, though not applicable to all formulations, also pointed to the need to properly formulate the momentum/continuity equations as a strongly elliptic system. The EVSS formulation [51] eventually developed out of these considerations and led to substantial improvements in the numerical formulation. It should be noted that a general proof for an LBB type relation for all constitutive equations and system formulations is still lacking. In essence, guidance is still often lacking in what type of stress interpolation is admissible.

6.9.5 Equation Change of Type

The work of Joseph et al. [50] demonstrated that the partial differential equations describing viscoelastic flows can sometimes undergo a formal change of type from elliptic to hyperbolic as the Weissenberg number increases. This is analogous to the transonic flow problem where local supercritical flow regions can be embedded in an otherwise subcritical flow. The proof was for a constitutive model without a viscous component, such as the Upper Convected Maxwell model. This work caused a great deal of activity since so much of the early computational work was based on this model. In combination with the work of Dupret, et al. [80] on loss of evolutionary behavior, the change of type analysis spurred development of the EEME method [49]. The proper formulation of the elliptic part of the viscoelastic problem was reinforced by this work.

The change of type was found to occur for the equations describing the fluid vorticity; the occurrence of real characteristics in such an equation admits to the possibility of discontinuities in the vorticity. The occurrence of shear waves has been observed in several transient solutions of viscoelastic flows [47,82–84]. Standard velocity based finite element methods are capable of resolving such flows, as will be demonstrated in the next section.

6.9.6 Closure

With the development of formulations such as the EVSS and its derivative methods, the high Weissenberg number problem is not as formidable as it once was. Solutions at high values of fluid elasticity have been produced and verified through mesh convergence studies. Not all problems have been eliminated however. At very high values of elasticity, numerical methods still have convergence problems. Most investigators believe this is a problem of mesh resolution and thin elastic boundary layers. Until higher resolution simulations are produced, this remains conjecture. It is clear that many simulations, when compared with experiment, fall short of a predictive capability. The constitutive relation is blamed in this case as being not representative of the viscoelastic material. The trend here is to use multimode models (multiple relaxation times) or more sophisticated micro models coupled with the macro flow equations. Finally, though some of the issues described above have been worked around computationally, a clear mathematical understanding of many of these areas is still not available.

6.10 Numerical Examples

6.10.1 Preliminary Comments

In this section a few example problems are described to demonstrate some of the procedures discussed in previous sections. The first examples consider inelastic, non-Newtonian flows and the second group illustrates a couple of viscoelastic applications. Most of the inelastic examples presented here were obtained using the mixed finite element model contained in the NACHOS II code [10], unless otherwise stated. Use of the penalty finite element models for power-law fluids can be found in [85–87]. For further examples, the literature cited in the text should be consulted.

The computations in the viscoelastic section will be limited to a few problems that illustrate some of the non-Newtonian behavior of a viscoelastic fluid. A variety of simulations have been reported in the literature. However, as example problems they suffer from a common shortcoming in that the reported solutions are often little different from those of a Newtonian fluid, mainly due to the “high Weissenberg limit” described above. One exception to the last comment is the work on die swell by Tanner [88], Crochet and Keunings [43,89,90], and others [91], which does show some departure from the Newtonian case for even slightly elastic fluids.

6.10.2 Buoyancy Driven Flow in a Cavity

The first example is concerned with the flow of a power-law fluid inside a rectangular cavity (see Figure 6.10.1). The vertical walls of the cavity are held at uniform but different temperatures, while the horizontal boundaries are insulated. The differentially heated walls produce a buoyancy driven flow in the

cavity. The computational boundary conditions for this problem consist of specified temperatures on the vertical walls, zero fluxes on the horizontal walls, and zero velocities on all four boundaries. The pressure level was set by specifying a pressure value at one point on the boundary of the cavity.

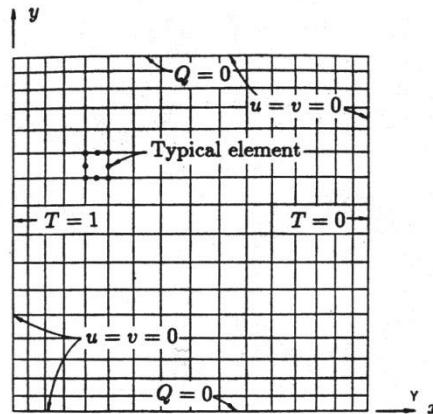


Figure 6.10.1: Schematic of natural convection in a cavity.

Figures 6.10.2 and 6.10.3 contain contour plots of the temperature and stream function obtained for three values of the power-law index. The $n = 1.0$ case corresponds to a Newtonian fluid at a Rayleigh number of $Ra = 10^4$ while the other two cases are for shear thinning fluids at the same nominal Rayleigh number. The effect of the shear thinning viscosity is evident as the fluid motion becomes more vigorous and complex for the same basic driving force. The illustrated cases were solved using a Picard iteration scheme starting from a zero initial solution field and required approximately 10 to 12 iterations to reach convergence.

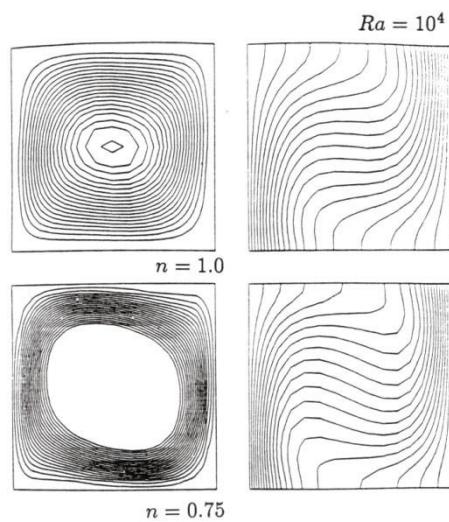


Figure 6.10.2: Streamlines and isotherms, natural convection in a cavity, power law fluid, $n = 0.75$.

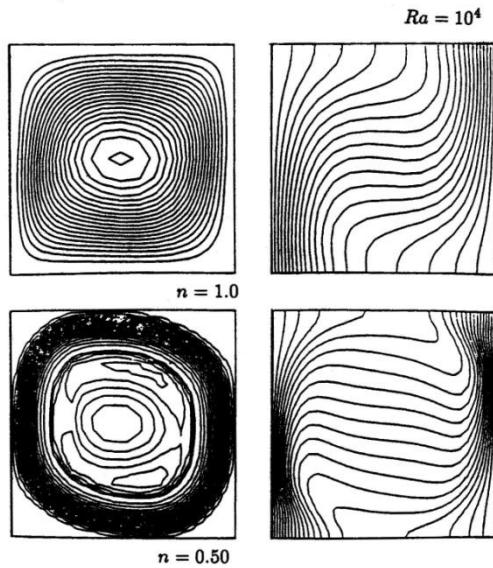


Figure 6.10.3: Streamlines and isotherms, natural convection in a cavity, power law fluid, $n = 0.50$.

6.10.3 Driven Cavity Flow

An isothermal driven cavity flow is considered as a second example. A Bingham fluid is contained in a planar cavity, three sides of which are fixed, with the fourth side moving at unit velocity in its own plane. For purposes of computation, the Bingham parameters are taken to be $\mu = 1.0$ and $\mu_r = 1000.0$ with τ_0 being varied over the range $0 \leq \tau_0 \leq 0.50$. The nominal Reynolds number for the problem is $Re = 100$. The computational mesh for this problem is the same as shown in Figure 6.10.1. Again, the Picard scheme was used to solve the problem with the extrapolation procedure of Section 6.4 also being employed.

Figure 6.10.4 contains contour plots of the stream function at four different values of the yield stress. The $\tau_0 = 0$ case is a Newtonian fluid. With increasing yield stress, the motion in the cavity is reduced and relegated to a small area near the moving lid of the cavity. This effect is seen more clearly in the velocity profiles shown in Figure 6.10.5. Note also that, as shown in Figure 6.10.5, the velocity profiles show a “break” near the vortex center. This is due to the relative magnitude of the deformation rate above and below the vortex center. Actually, the break point should occur at zero velocity but is offset slightly due to the use of a constant viscosity within each element and a relatively coarse mesh.

6.10.4 Squeeze Film Flow

The third example considers the flow of a plastic (Bingham) fluid in a squeeze film apparatus. This type of device is often used to measure the yield parameters for lubricating greases and other “stiff” materials. The geometry and a typical mesh are shown in Figure 6.10.6. Two circular plates are separated by a fluid layer of initial height h . At time zero the upper plate moves downward with a fixed velocity.

The motion of the plate is slow enough that a creeping flow assumption may be used and a series of steady-state problems examined with h as a parameter.

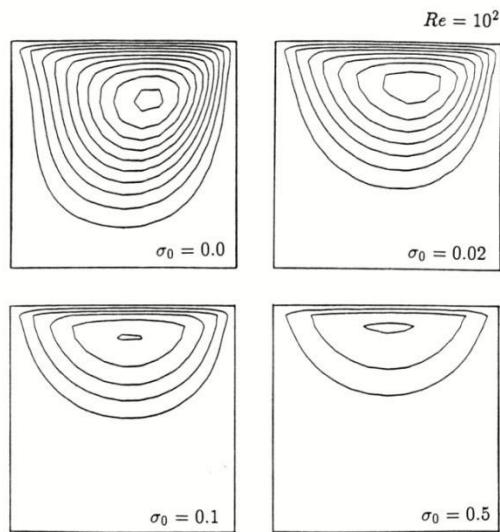


Figure 6.10.4: Streamlines for driven cavity flow; Bingham fluid; various yield stresses.

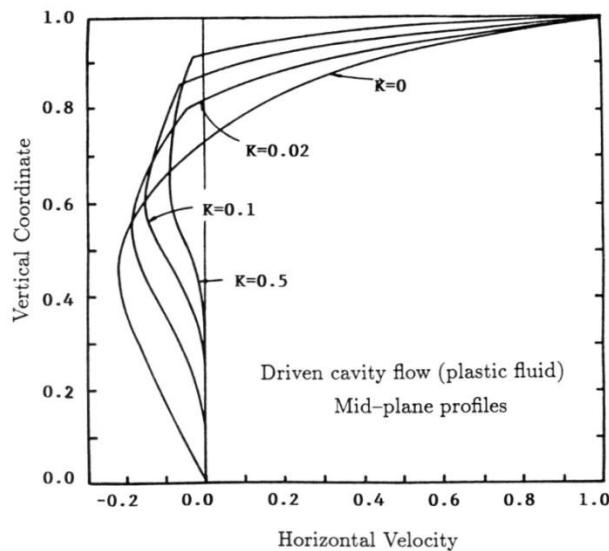


Figure 6.10.5: Midplane velocity profiles for driven cavity flow; Bingham fluid.

Boundary conditions for this simulation consisted of a specified nonzero vertical velocity and a zero horizontal velocity along the top surface of the domain. Along the vertical centerline the horizontal velocity is set to zero and the shear stress (natural boundary condition) vanishes due to symmetry. The horizontal symmetry plane has a zero vertical velocity and a zero shear stress. The outflow boundary assumes a zero vertical velocity (parallel flow assumption) and a constant normal traction. A Picard iteration scheme with extrapolation was used as a solution procedure; up to 12 iterations were required to obtain convergence.

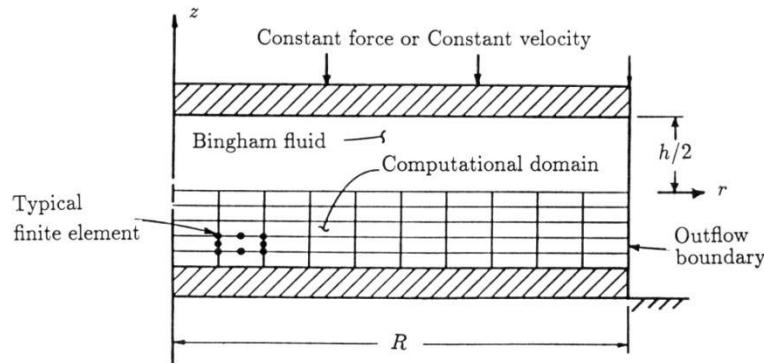


Figure 6.10.6: Schematic of squeeze film flow.

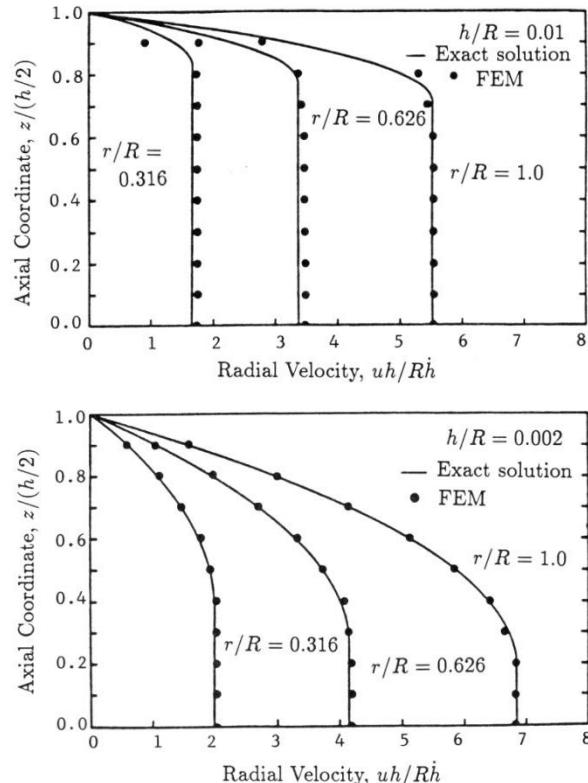


Figure 6.10.7: Radial velocity profiles for the squeeze film flow problem (Bingham fluid). (a) $h/R = 0.01$. (b) $h/R = 0.002$.

Radial velocity profiles for two plate separations are shown in Figure 6.10.7, where they are compared to an analytical solution developed in [17]. The fluid near the center plane of the device flows as an unyielded plug while near the plate a more viscous boundary layer-like flow occurs. The demarcation line between yielded and unyielded material is illustrated in Figure 6.10.8. The agreement with the analytical solution is quite good.

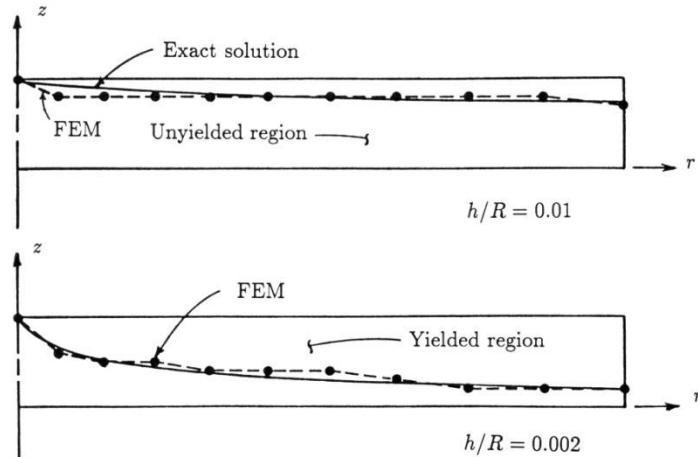


Figure 6.10.8: Plots of the yield surface for the squeeze film flow problem (Bingham fluid). (a) $h/R = 0.01$. (b) $h/R = 0.002$.

6.10.5 Time-Dependent Poiseuille Flow

The first viscoelastic example is a geometrically simple flow with a complex physical response. Consider the problem of an initially quiescent viscoelastic fluid contained between infinite parallel plates. At time zero a specified, constant pressure gradient is imposed. The objective is to compute the time-dependent response of the fluid until it reaches a steady state (Poiseuille flow). A schematic of the problem is shown in Figure 6.10.9.

The above problem was solved analytically by Rivlin [84] for both a Newtonian and an upper-convected Maxwell fluid. The numerical solutions presented here were obtained using the finite element scheme described in [47,48]; results are reported for Newtonian and Maxwell fluids.

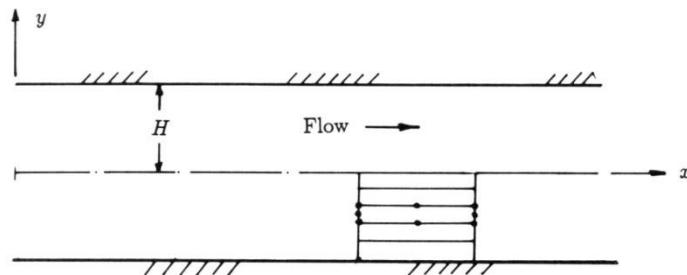


Figure 6.10.9: Schematic of Poiseuille flow.

Shown in Figures 6.10.10 and 6.10.11 are representative time histories and velocity profiles for the response of a Newtonian fluid. The evolution from the rest state to a fully developed parabolic profile is seen to be a smooth, asymptotic process. Comparison of the numerical results with the analytical work of Rivlin shows excellent agreement (less than 1% difference).

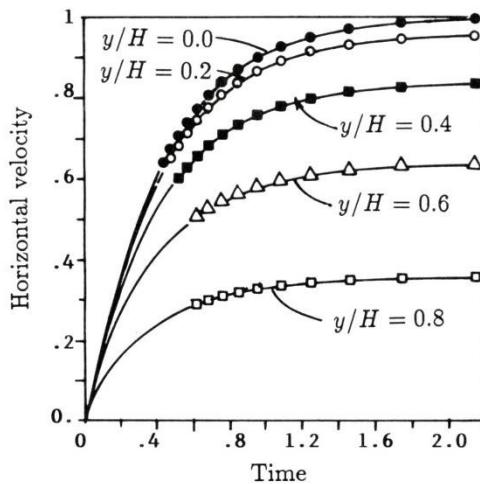


Figure 6.10.10: Velocity histories for Poiseuille flow, Newtonian fluid.

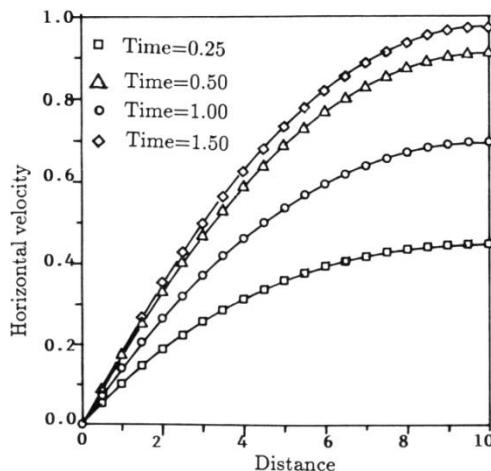


Figure 6.10.11: Velocity profiles for Poiseuille flow, Newtonian fluid.

The response of a viscoelastic fluid, such as a Maxwell material, is quite different. Figure 6.10.12 shows a typical time history for the Maxwell fluid. The damped oscillatory behavior is quite evident and represents a streamwise oscillation of the fluid. This is seen more clearly in Figure 6.10.13, where velocity profiles at various times are illustrated. Early in the start-up process the velocity field shows the propagation of a shear wave from the wall toward the centerline of the channel. The subsequent reflection of the wave causes the oscillatory behavior seen in Figure

6.10.12. At long times the velocity field approaches the expected steady-state profile. A comparison of this solution with the analytical result again shows excellent agreement.

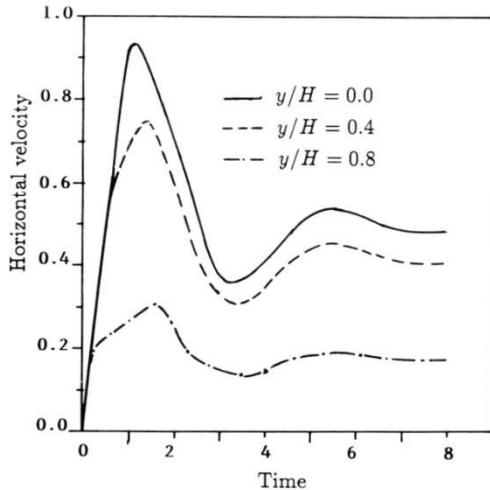


Figure 6.10.12: Velocity histories for Poiseuille flow, Maxwell fluid.

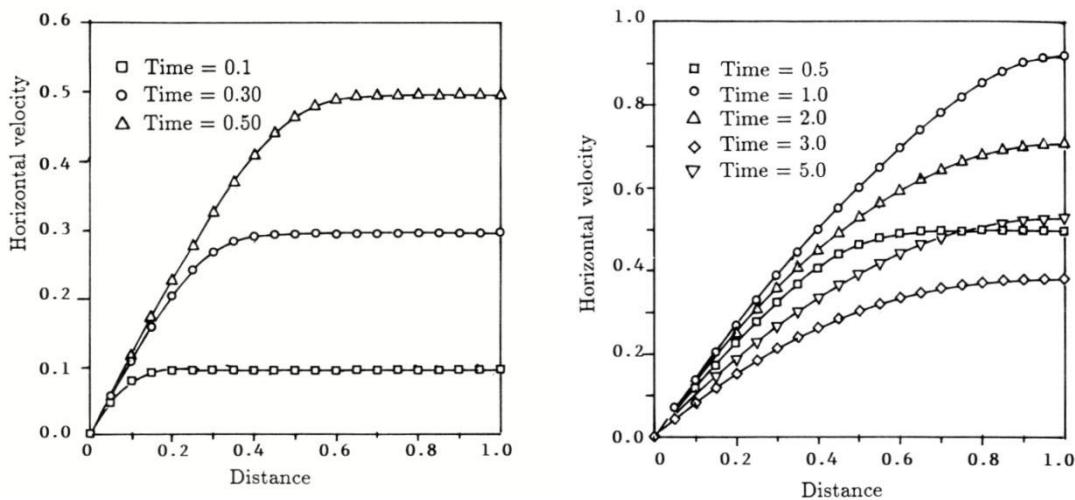


Figure 6.10.13: Velocity profiles for Poiseuille flow, Maxwell fluid.

Figure 6.10.14 shows a series of shear stress profiles that illustrates the early time behavior of the shear wave as it propagates across the channel. The normal stress component has a similar behavior. Though the present example is quite simple, it does illustrate the complex behavior that can occur in fluids with memory.

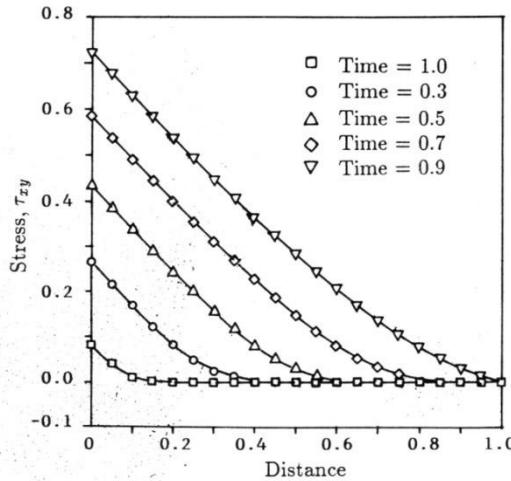


Figure 6.10.14: Shear stress profiles for Poiseuille flow, Maxwell fluid.

6.10.6 Four-to-One Contraction Problem

A standard test problem for the study of viscoelastic flow behavior is the entry flow in axisymmetric, four-to-one contraction. This geometry is popular since it is of technological importance for the polymer processing industry and also exhibits significant flow variation with changes in fluid elasticity. In particular, for several different polymers, as the fluid elasticity is increased (i.e., Weissenberg number is increased) the vortex structure in the largest entry tube undergoes a significant increase in length. The objective of many numerical simulations has been to predict this trend in vortex growth with a suitable constitutive model.

To illustrate typical results for the four-to-one contraction the method proposed by Rao and Finlayson [40] is employed. Figure 6.10.15 contains a schematic of the problem with the velocity boundary conditions shown; a typical quadrilateral mesh is shown in Figure 6.10.16. An Oldroyd B constitutive model was used with a typical mixed method; a cubic approximation for the stress components was used in conjunction with quadratic velocities and a linear pressure. A Newton iteration method with continuation was used to solve the four-to-one contraction for a series of Weissenberg numbers. It was found that to achieve significant values of the Weissenberg number an upwind procedure for the constitutive equation was required. Full details of all the combinations of formulations and algorithms that were tried on this problem are available in [92].

Computed results for the four-to-one contraction in the form of stream function contours are shown in Figure 6.10.17. The plots correspond to increasing values of the recoverable shear, S_R , which is directly proportional to the Weissenberg number. The simulations predict the correct trend in that the vortex length increases as the fluid elasticity increases. Quantitative comparisons with experimental data show that numerical simulations underpredict the vortex growth. This discrepancy is at least partly the result of using approximate constitutive models.

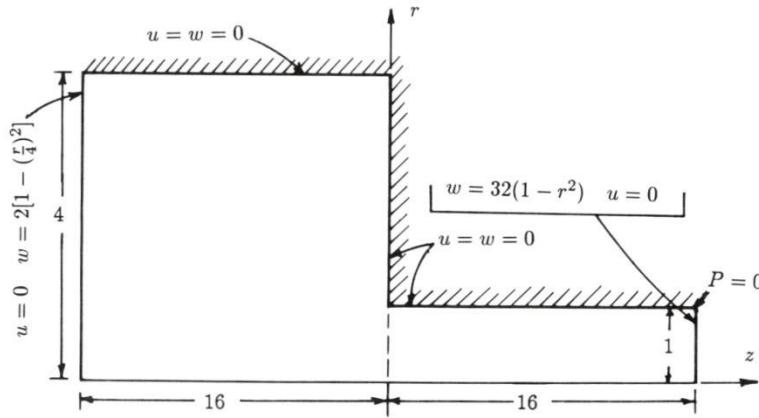


Figure 6.10.15: Schematic for axisymmetric 4:1 contraction problem.

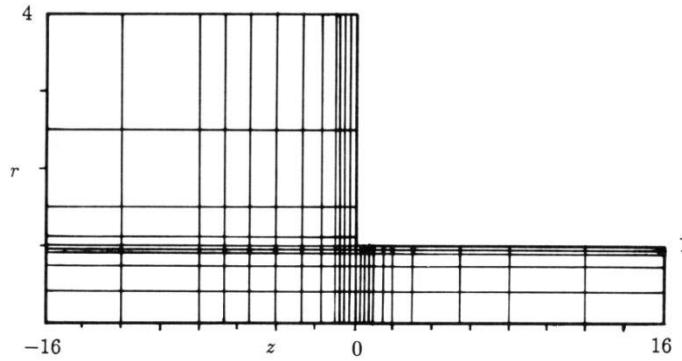


Figure 6.10.16: Quadrilateral mesh for 4:1 contraction. The stress field is approximated using cubic polynomials on each element.

Problems

6.1 The general Oldroyd model is given by (see [5])

$$\begin{aligned} \tau_{ik} + \lambda \overset{\circ}{\tau}_{ik} + \mu \tau_{jj} D_{ik} - \mu_1 (\tau_{ij} D_{jk} + \tau_{kj} D_{ji}) \\ = 2\mu_0 \left(D_{ik} + \lambda_1 \overset{\circ}{D}_{ik} - \mu_2 D_{ij} D_{jk} \right) \end{aligned}$$

where μ_0 is a constant viscosity coefficient, and $(\lambda, \lambda_1, \mu, \mu_1, \mu_2)$ are material constants. Deduce the upper-convected, lower-convected, and corotational Maxwell models by assigning values to the material constants in the Oldroyd model.

6.2 The White–Metzner model [29] is defined by

$$\tau_{ik} + \lambda(I_2) \nabla \tau_{ik} = 2\mu_0(I_2) D_{ik}$$

which describes viscoelastic effects in flow problems that are dominated by the shear viscosity. Develop the finite element model governing viscoelastic flows in two-dimensional Cartesian geometries using the White–Metzner constitutive equation (see [91]).

6.3 Repeat Problem 2 for flows in axisymmetric two-dimensional geometries.

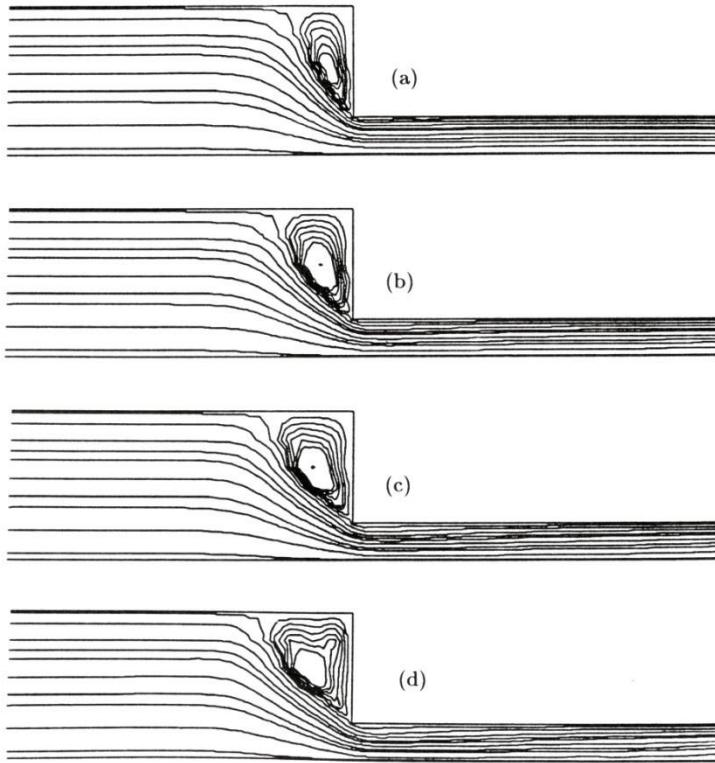


Figure 6.10.17: Streamline for flow in an axisymmetric, 4:1 contraction using an Oldroyd B fluid with recoverable shear values of (a) $S_R = 10$, (b) $S_R = 20$, (c) $S_R = 30$, and (d) $S_R = 40$ (from [91]).

References for Additional Reading

1. R. B. Bird, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids, Vol. 1: Fluid Mechanics*, Second Edition, John Wiley & Sons, New York (1971).
2. A. S. Lodge, *Elastic Liquids*, Academic Press, New York (1964).
3. K. Walters, *Rheometry*, Chapman and Hall, London, U.K. (1975).
4. R. I. Tanner, *Engineering Rheology*, Clarendon Press, Oxford, U.K. (1985).
5. M. J. Crochet, A. R. Davies, and K. Walters, *Numerical Simulation of Non-Newtonian Flow*, Elsevier, Amsterdam, The Netherlands (1984).
6. J. R. A. Pearson, *Mechanics of Polymer Processing*, Elsevier, London, U.K. (1985).
7. J. N. Reddy, *Introduction to Continuum Mechanics with Applications*, Cambridge University Press, New York (2008).
8. R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, John Wiley & Sons, New York (1960).
9. J. G. Oldroyd, “A Rational Formulation of the Equations of Plastic Flow for a Bingham Solid,” *Proceedings of the Cambridge Philosophical Society*, **43**, 100–105 (1947).
10. D. K. Gartling, “NACHOS II-A Finite Element Computer Program for Incompressible Flow Problems,” Sandia National Laboratories Report, SAND86-1816, Albuquerque, NM (1987).
11. R. L. Lee and P. M. Gresho, “Development of a Three-Dimensional Model of the Atmospheric Boundary Layer Using the Finite Element Method,” *Lawrence Livermore National Laboratory Report, UCRL-52366*, Livermore, California (1977).

12. D. K. Gartling, "Finite Element Analysis of Convective Heat Transfer Problems with Change of Phase," in *Computer Methods in Fluids*, K. Morgan, C. Taylor, and C. A. Brebbia (eds.), Pentech Press, London, U.K., 257–284 (1980).
13. D. J. Naylor, "Stresses in Nearly Incompressible Materials by Finite Elements with Application to the Calculation of Excess Pore Pressures," *International Journal for Numerical Methods in Engineering*, **8**, 443–460 (1974).
14. E. Hinton and J. S. Campbell, "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method," *International Journal for Numerical Methods in Engineering*, **8**, 461–480 (1974).
15. E. Hinton, F. C. Scott, and R. E. Ricketts, "Local Least Squares Stress Smoothing for Parabolic Isoparametric Elements," *International Journal for Numerical Methods in Engineering*, **9**, 235–238 (1975).
16. D. K. Gartling, "The Numerical Simulation of Plastic Fluids," in *Proceedings of Third International Conference on Numerical Methods in Laminar and Turbulent Flow*, C. Taylor, C. Johnson, and I. Smith (eds.), Pineridge Press, Swansea, U.K., 669–679 (1983).
17. D. K. Gartling and N. Phan Thien, "A Numerical Simulation of a Plastic Fluid in a Parallel-Plate Plastometer," *Journal of Non-Newtonian Fluid Mechanics*, **14**, 347–360 (1984).
18. R. I. Tanner and J. F. Milthorpe, "Numerical Simulation of the Flow of Fluids with Yield Stresses," in *Proceedings of Third International Conference on Numerical Methods in Laminar and Turbulent Flow*, C. Taylor, C. Johnson, and I. Smith (eds.), Pineridge Press, Swansea, U.K., 680–690 (1983).
19. M. S. Engelman, "Quasi-Newton Methods in Fluid Dynamics," in *The Mathematics of Finite Elements and Applications, IV*, J. R. Whiteman (ed.), Academic Press, London, U.K., 479–487 (1982).
20. C. Truesdell and W. Noll, "The Non-Linear Field Theories of Mechanics," in *Encyclopedia of Physics, Vol III*, S. Flügge (ed.), Springer-Verlag, Berlin, Germany (1965).
21. M. J. Crochet and K. Walters, "Numerical Methods in Non-Newtonian Fluid Mechanics," *Annual Review of Fluid Mechanics*, **15**, 241–260 (1983).
22. A. C. Pipkin and R. I. Tanner, "A Survey of Theory and Experiment in Viscometric Flows of Viscoelastic Liquids," *Mechanics Today, Vol. 1*, S. Nemat-Nasser (ed.), 262–321 (1972).
23. R. I. Tanner, "Recent Progress in Rheology," *Journal of Applied Mechanics*, **105**, 1181–1190 (1983).
24. M. J. Crochet and G. Pilate, "Numerical Study of the Flow of a Fluid of Second Grade in a Square Cavity," *Computers and Fluids*, **3**, 283–291 (1975).
25. K. R. Reddy and R. I. Tanner, "Finite Element Approach to Die-Swell Problems of Non-Newtonian Fluids," *Proceedings of the Sixth Australian Hydraulics and Fluid Mechanics Conference*, Adelaide, Australia, 431–434 (1977).
26. M. A. Mendelson, P. W. Yeh, R. A. Brown, and R. C. Armstrong, "Approximation Error in Finite Element Calculation of Viscoelastic Flow," *Journal of Non-Newtonian Fluid Mechanics*, **10**, 31–54 (1982).
27. M. W. Johnson and D. Segalman, "A Model for Viscoelastic Fluid Behavior Which Allows Non-Affine Deformation," *Journal of Non-Newtonian Fluid Mechanics*, **2**, 255–270 (1977).
28. N. Phan Thien and R. I. Tanner, "A New Constitutive Equation Derived From Network Theory," *Journal of Non-Newtonian Fluid Mechanics*, **2**, 353–365 (1977).
29. J. L. White and A. B. Metzner, "Development of Constitutive Equations for Polymeric Melts and Solutions," *Journal of Applied Polymer Science*, **7**, 1867–1889 (1963).
30. C. J. S. Petrie, "Measures of Deformation and Convected Derivatives," *Journal of Non-Newtonian Fluid Mechanics*, **5**, 147–176 (1979).
31. L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Englewood Cliffs, New Jersey (1969).
32. A. S. Lodge, *Elastic Liquids*, Academic Press, New York (1964).

33. Y. Shimazaki and E. G. Thompson, "Elasto Visco-Plastic Flow With Special Attention to Boundary Conditions," *International Journal for Numerical Methods in Engineering*, **17**, 97–112 (1981).
34. M. Kawahara and N. Takeuchi, "Mixed Finite Element Method for Analysis of Viscoelastic Fluid Flow," *Computers and Fluids*, **5**, 33–45 (1977).
35. M. J. Crochet and M. Bezy, "Numerical Solution for the Flow of Viscoelastic Fluids," *Journal of Non-Newtonian Fluid Mechanics*, **5**, 201–218 (1979).
36. C. J. Coleman, "A Finite Element Routine for Analyzing Non-Newtonian Fluids, Part I: Basic Method and Preliminary Results," *J. of Non-Newtonian Fluid Mechanics*, **7**, 289–301 (1980).
37. P. W. Chang, T. W. Patten, and B. A. Finlayson, "Collocation and Galerkin Finite Element Methods for Viscoelastic Fluid Flow, I. Description of Method and Problems with Fixed Geometries," *Computers and Fluids*, **7**, 267–283 (1979).
38. J. M. Marchal and M. J. Crochet, "Hermitian Finite Elements for Calculating Viscoelastic Flow," *Journal of Non-Newtonian Fluid Mechanics*, **20**, 187–207 (1986).
39. J. M. Marchal and M. J. Crochet, "A New Mixed Finite Element for Calculating Viscoelastic Flow," *Journal of Non-Newtonian Fluid Mechanics*, **26**, 77–114 (1987).
40. R. R. Rao and B. A. Finlayson, "Viscoelastic Flow Simulation Using Cubic Stress Finite Elements," *Journal of Non-Newtonian Fluid Mechanics*, **43**, 61–82 (1992).
41. P. W. Yeh, M. E. Kim, R. C. Armstrong, and R. A. Brown, "Multiple Solutions in the Calculation of Axisymmetric Contraction Flow of an Upper Convected Maxwell Fluid," *Journal of Non-Newtonian Fluid Mechanics*, **16**, 173–194 (1984).
42. D. K. Gartling, "One Dimensional Finite Element Solutions for a Maxwell Fluid," *Journal of Non-Newtonian Fluid Mechanics*, **17**, 203–231 (1985).
43. M. J. Crochet and R. Keunings, "Die Swell of a Maxwell Fluid: Numerical Predictions," *Journal of Non-Newtonian Fluid Mechanics*, **7**, 199–212 (1980).
44. R. Keunings and M. J. Crochet, "Numerical Simulation of the Flow of a Viscoelastic Fluid Through an Abrupt Contraction," *J. of Non-Newtonian Fluid Mechanics*, **14**, 279–299 (1984).
45. P. Gresho, R. Lee, R. Sani, and T. Stullich, "On the Time-Dependent FEM Solution of the Incompressible Navier-Stokes Equations in Two- and Three-Dimensions," in *Recent Advances in Numerical Methods in Fluids*, Vol. 1, Pineridge Press, Swansea, U.K. (1980).
46. S. L. Josse and B. A. Finlayson, "Reflections on the Numerical Viscoelastic Flow Problem," *Journal of Non-Newtonian Fluid Mechanics*, **16**, 13–36 (1984).
47. D. K. Gartling, "A Finite Element Algorithm for Time-Dependent Viscoelastic Flows," Sandia National Laboratories Report, Albuquerque, New Mexico (1988).
48. M. Fortin and A. Fortin, "A New Approach for the FEM Simulation of Viscoelastic Flows," *Journal of Non-Newtonian Fluid Mechanics*, **32**, 295–310 (1989).
49. R. C. King, M. R. Apelian, R. C. Armstrong, and R. A. Brown, "Numerically Stable Finite Element Techniques for Viscoelastic Calculations in Smooth and Singular Geometries," *Journal of Non-Newtonian Fluid Mechanics*, **29**, 147–216 (1988).
50. D. D. Joseph, M. Renardy, and J. C. Saut, "Hyperbolicity and Change of Type in the Flow of Viscoelastic Fluids," *Archive of Rational Mechanics and Analysis*, **87**, 213–251 (1985).
51. D. Rajagopalan, R. C. Armstrong, and R. A. Brown, "Finite Element Methods for Calculation of Steady, Viscoelastic Flow Using Constitutive Equations with a Newtonian Viscosity," *Journal of Non-Newtonian Fluid Mechanics*, **36**, 159–192 (1990).
52. R. Guenette and M. Fortin, "A New Mixed Finite Element Method for Computing Viscoelastic Flows," *Journal of Non-Newtonian Fluid Mechanics*, **60**, 27–52 (1995).
53. J. Sun, N. Phan Thien, and R. I. Tanner, "An Adaptive Viscoelastic Stress Splitting Scheme and Its Applications: AVSS/SI and AVSS/SUPG," *Journal of Non-Newtonian Fluid Mechanics*, **65**, 75–91 (1996).
54. J. Sun, M. D. Smith, R. C. Armstrong, and R. A. Brown, "Finite Element Method for Viscoelastic Flows Based on the Discrete Adaptive Viscoelastic Stress Splitting and the Discontinuous Galerkin Method: DAVSS-G/DG," *Journal of Non-Newtonian Fluid Mechanics*, **86**, 281–307 (1999).

55. P. Lesaint and P. A. Raviart, "On a Finite Element Method for Solving the Neutron Transport Equation," in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, C. de Boor (Ed.), Academic Press, New York (1974).
56. M. Viriyayuthakorn and B. Caswell, "Finite Element Simulation of Viscoelastic Flow," *Journal of Non-Newtonian Fluid Mechanics*, **6**, 245–267 (1980).
57. S. Dupont, J. M. Marchal, and M. J. Crochet, "Finite Element Simulation of Viscoelastic Fluids of the Integral Type," *Journal of Non-Newtonian Fluid Mechanics*, **17**, 157–183 (1985).
58. B. Bernstein, M. K. Kadivar, and D. S. Malkus, "Steady Flow of Memory Fluids with Finite Elements: Two Test Problems," *Computer Methods in Applied Mechanics and Engineering*, **27**, 279–302 (1981).
59. R. C. Armstrong, R. A Brown, and B. Caswell (eds.), "Papers From the Third International Workshop on Numerical Simulation of Viscoelastic Flows," *Journal of Non-Newtonian Fluid Mechanics*, **16** (1984).
60. M. J. Crochet, "Numerical Simulation of Viscoelastic Flow: A Review," *Rubber Reviews*, **62**, 426–455 (1989).
61. R. Keunings, "Progress and Challenges in Computational Rheology," *Rheologica Acta*, **29**, 556–570 (1990).
62. F. P. T. Baaijens, "Mixed Finite Element Methods for Viscoelastic Flow Analysis: A Review," *Journal of Non-Newtonian Fluid Mechanics*, **79**, 361–385 (1998).
63. M. Renardy, "Current Issues in Non-Newtonian Flows: A Mathematical Perspective," *Journal of Non-Newtonian Fluid Mechanics*, **90**, 243–259 (2000).
64. D. Winterscheidt and K. S. Surana, " p -Version Least Squares Finite Element Formulation for Convection Diffusion Problems," *International Journal for Numerical Methods in Engineering*, **36**, 111–133 (1993).
65. D. Winterscheidt and K. S. Surana, " p -Version Least Squares Finite Element Formulation for Burgers Equation," *International Journal for Numerical Methods in Engineering*, **36**, 3629–3646 (1993).
66. K. S. Surana, P. Gupta, P. W. Tenpas, and J. N. Reddy, " h, p, k Least Squares Finite Element Processes for 1-D Helmholtz Equation," *International Journal for Computational Methods in Engineering Science and Mechanics*, **7**(4), 263–291 (2006).
67. K. S. Surana, A. Mohammed, J. N. Reddy, and P. W. Tenpas, " k -Version of Finite Element Method in 2-D Polymer Flows: Oldroyd-B Constitutive Model," *International Journal of Numerical Methods in Fluids*, **52**(2), 119–162 (2006).
68. K. S. Surana, S. Allu, P. W. Tenpas, and J. N. Reddy, " k -Version Finite Element Method in Gas Dynamics: Higher Order Global Differentiability Numerical Solutions," *International Journal for Numerical Methods in Engineering*, **69**(6), 1109–1157 (2007).
69. K. S. Surana, M. K. EngelKemier, J. N. Reddy, and P. W. TenPas, " k -version Least Squares Finite Element Processes for 2-D Generalized Newtonian Fluid Flows," *International Journal of Computational Methods in Engineering Science and Mechanics*, **8**, 243–261 (2007).
70. K. S. Surana, P. Gupta, and J. N. Reddy, "Galerkin and Least-Squares Finite Element Processes for 2-D Helmholtz Equation in h, p, k Framework," *International Journal of Computational Methods in Engineering Science and Mechanics*, **8**(5), 341–361 (2007).
71. K. S. Surana, S. Bhola, J. N. Reddy, and P. W. TenPas, " k -Version of Finite Element Method in 2D-Polymer Flows: Upper Convected Maxwell Model," *Computers and Structures*, **86** (17-18), 1782–1808 (2008).
72. K. S. Surana, S. Allu, J. N. Reddy, and P. W. TenPas, "Least Squares Finite Element Processes in h, p, k Mathematical and Computational Framework for a Non-Linear Conservation Law," *International Journal for Numerical Methods in Fluids*, **57**(10), 1545–1568 (2008).
73. K. S. Surana, K. M. Deshpande, A. Romkes, and J. N. Reddy, "Computations of Numerical Solutions in Polymer Flows Using Giesekus Constitutive Model in the h, p, k Framework with Variationally Consistent Integral Forms," *International Journal of Computational Methods in Engineering Science and Mechanics*, to appear.
74. N. Phan Thien, "Coaxial-Disk Flow of an Oldroyd-B Fluid: Exact Solution and Stability," *Journal of Non-Newtonian Fluid Mechanics*, **13**, 325–340 (1983).

75. N. Phan Thien and W. Walsh, "Squeeze-Film Flow of an Oldroyd-B Fluid: Similarity Solution and Limiting Weissenberg Number," *Journal of Applied Mathematics and Physics (ZAMP)*, **35**, 747–759 (1984).
76. N. Phan Thien, "Cone-and-Plate Flow of the Oldroyd-B Fluid is Unstable," *Journal of Non-Newtonian Fluid Mechanics*, **17**, 37–44 (1985).
77. R. Keunings, "Mesh Refinement Analysis of the Flow of a Maxwell Fluid Through an Abrupt Contraction," *Proceedings of Fourth International Conference on Numerical Methods in Laminar and Turbulent Flow*, C. Taylor et al. (eds.), Pineridge Press, Swansea, U.K. (1985).
78. B. Debbaut and M. J. Crochet, "Further Results on the Flow of a Viscoelastic Fluid Through an Abrupt Contraction," *Journal of Non-Newtonian Fluid Mechanics*, **20**, 173–185 (1986).
79. J. V. Lawler, S. J. Muller, R. A. Brown, and R. C. Armstrong, "Laser Doppler Velocimetry, Measurements of Velocity Fields and Transitions in Viscoelastic Fluids," *Journal of Non-Newtonian Fluid Mechanics*, **20**, 51 (1986).
80. F. Dupret, J. M. Marchal, and M. J. Crochet, "On the Consequence of Discretization Errors in the Numerical Calculation of Viscoelastic Flow," *Journal of Non-Newtonian Fluid Mechanics*, **18**, 173–186 (1985).
81. M. Fortin and R. Pierre, "On the Convergence of the Mixed Method of Crochet and Marchal for Viscoelastic Flows," *Computer Methods in Applied Mechanics and Engineering*, **7**, 1035–1052 (1987).
82. N. Phan Thien, "Coaxial-Disk Flow and Flow About a Rotating Disk of a Maxwell Fluid," *Journal of Fluid Mechanics*, **128**, 427–442 (1983).
83. N. Phan Thien and R. I. Tanner, "Viscoelastic Squeeze-Film Flows – Maxwell Fluids," *Journal of Fluid Mechanics*, **129**, 265–281 (1983).
84. R. S. Rivlin, "Run-Up and Decay of Plane Poiseuille Flow," *Journal of Non-Newtonian Fluid Mechanics*, **14**, 203–217 (1984).
85. M. Iga and J. N. Reddy, "Penalty Finite Element Analysis of Free Surface Flows of Power-Law Fluids," *International Journal of Non-Linear Mechanics*, **24**(5), 383–399 (1989).
86. M. P. Reddy and J. N. Reddy, "Finite Element Analysis of Flows of Non-Newtonian Fluids in Three-Dimensional Enclosures," *International Journal of Non-Linear Mechanics*, **27**(1), 9–26 (1992).
87. M. P. Reddy and J. N. Reddy, "Numerical Simulation of Forming Processes Using a Coupled Fluid Flow and Heat Transfer Model," *International Journal for Numerical Methods in Engineering*, **35**, 807–833 (1992).
88. R. I. Tanner, "Extrudate Swell," in *Computational Analysis of Polymer Processing*, J. R. A. Pearson and S. M. Richardson (eds.), Elsevier, London, U.K., 63–91 (1983).
89. M. J. Crochet and R. Keunings, "On Numerical Die Swell," *Journal of Non-Newtonian Fluid Mechanics*, **10**, 85–94 (1982).
90. M. J. Crochet and R. Keunings, "Finite Element Analysis of Die-Swell of a Highly Elastic Fluid," *Journal of Non-Newtonian Fluid Mechanics*, **10**, 339–356 (1982).
91. J. N. Reddy and V. A. Padhye, "A Penalty Finite Element Model for Axisymmetric Flows of Non-Newtonian Fluids," *Numerical Methods in Partial Differential Equations*, **4**, 33–56 (1988).
92. R. R. Rao, "Adaptive Finite Element Analysis of Non-Newtonian Flow," Ph.D. Dissertation, Department of Chemical Engineering, University of Washington, Seattle, Washington (1990).

Multiphysics Problems

7.1 Introduction

Coupled problems in applied mechanics are generally defined as those requiring the solution of more than one physical process for adequate representation of the overall system. In reality, most engineering problems fall into this category, although it is still a common practice to perform single-physics analysis for many applications. Coupled phenomena are very prevalent in the areas of fluid mechanics and heat transfer as is evident from some of the topics from previous chapters. In Chapter 3, coupled problems involving heat conduction and radiation, and heat conduction and chemical reaction were considered. At the beginning of Chapter 5 it was noted that convective heat transfer was a coupled problem since it involved two different physical phenomena, namely, fluid mechanics and heat transfer within a fluid. Likewise, the conjugate problem of convective heat transfer in a fluid adjacent to a heat conducting solid is a type of coupled problem. In the present chapter we are going to revisit coupled problems, though the emphasis will be on problems involving more than one discipline or branch of mechanics. Specifically, the finite element solution of solid mechanics and electromagnetics problems will first be presented. Subsequent sections will describe how these types of solutions may be coupled to fluid mechanics and heat transfer simulations to provide a more complete analysis.

7.2 Coupled Boundary Value Problems

Boundary value problems describing different types of mechanics may be coupled through a variety of mechanisms and with varying degrees of interaction. Both of these characteristics are difficult to generalize and quantify, and lead to a certain vagueness when discussing coupled problems in generic terms. Before getting to the specific cases of interest here, it is worthwhile to set some terminology and outline some of the complexities that may occur.

The partial differential equations describing different phenomena are coupled when any terms in either equation are functions of the dependent variable or its derivatives from the other equation. This functional dependence may occur directly in source or volume terms, material coefficients, and/or boundary conditions. The dependent variables from one equation may also act more indirectly on the second equation by causing alterations in the geometry of the problem and changes to the temporal behavior of the problem. The degree to which one equation is coupled to

a second equation is particularly difficult to define. The terms “weak” and “strong” coupling are often used without precise definitions. In the present case, a “weakly” coupled problem is defined as one in which the transfer of dependent variable data between equations need not occur at every solution step. In essence, the influence of one physical process on the other is sufficiently mild that only periodic updating is required for an accurate representation of the overall system response. For a weakly coupled problem the data transfer may be bidirectional or unidirectional, i.e., there is no mandate that the processes be equally influential. The definition of a “strongly” coupled problem is obvious from the previous definition. If data must be transferred or shared between equations at each step of the solution to maintain accuracy of overall simulation, the problem is defined as strongly coupled. These definitions are not rigorous but do allow some general classifications for coupled problems.

The degree of coupling will also influence the style and choice of computational algorithms used to solve each equation set and the interaction between discretized equations. In general, our preference is to solve strongly coupled physical processes as fully coupled equation sets as this usually provides the most robust and strongly convergent algorithm. The natural convection problem of Chapter 5 is an example of this type of coupled problem. The fully coupled approach is not, however, always feasible. Both physical and numerical characteristics of the particular coupled problem may render full coupling computationally inefficient or impractical. Large disparities in length and time scales between physical processes are two characteristics that may influence the strong coupling algorithm. The solution procedure for weakly coupled physical processes follows from its definition and is obviously some type of cyclic algorithm with separate solution methods for each equation set intertwined with periodic exchanges of data. The frequency and timing of data updates are very problem dependent and one of the characteristics that makes general coupling methods and software difficult to develop.

7.3 Fluid Mechanics and Heat Transfer

7.3.1 Introduction

To proceed with the coupled problem discussion it is necessary to become specific with regard to physical processes and coupling mechanisms. In the present section the fluid/thermal problem will again be outlined with particular attention paid to possible dependencies on the solid mechanics and electromagnetics problems to be considered subsequently.

7.3.2 Continuum Equations

The boundary value problem for the nonisothermal flow of a viscous, incompressible fluid is described by the standard conservation relations for mass, momentum, and energy. For a Cartesian coordinate system using the Eulerian description, these relations are summarized here.

Conservation of Mass

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (7.3.1)$$

Conservation of Momentum

$$\rho_0 \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left[-P \delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \rho_0 g_i \beta (T - T_0) = 0 \quad (7.3.2)$$

Conservation of Energy

$$\rho_0 C_v \left(\frac{\partial T}{\partial t} + v_j \frac{\partial T}{\partial x_j} \right) - \frac{\partial}{\partial x_i} \left(k_{ij} \frac{\partial T}{\partial x_j} \right) - Q - \Phi = 0 \quad (7.3.3)$$

The parameters and symbols used in (7.3.1)–(7.3.3) are standard and the same as used in Chapter 5. Note that the porous flow problem could also be considered here as part of the coupling problem with little variation in the discussion. The boundary conditions for the nonisothermal flow are

$$v_i = f_i^v(s_k, t) \quad \text{on } \Gamma_v \quad (7.3.4a)$$

$$\mathcal{T}_i \equiv \sigma_{ij}(s_k, t) n_j(s_k) = f_i^T(s_k, t) \quad \text{on } \Gamma_T \quad (7.3.4b)$$

for the fluid mechanics part of the problem, and

$$T = f^T(s_k, t) \quad \text{on } \Gamma_T \quad (7.3.5a)$$

$$-\left(k_{ij} \frac{\partial T}{\partial x_j} \right) n_i \equiv q_i n_i = q_c + q_r + q_a = f^q(s_k, t) \quad \text{on } \Gamma_q \quad (7.3.5b)$$

for the heat transfer part of the problem.

When discussing coupling, solid body thermal analysis will be considered distinct from the fluid mechanics problem since the equation system is substantially different as are the coupling interactions. In the no flow case, Eqs. (7.3.1) and (7.3.2) are neglected and Eq. (7.3.3) reduces to the simple heat conduction problem

$$\rho C \frac{\partial T}{\partial t} = \frac{\partial}{\partial x_i} \left(k_{ij} \frac{\partial T}{\partial x_j} \right) + Q \quad (7.3.6)$$

with boundary conditions as given in Eq. (7.3.5). The addition of radiation and/or chemical reaction to the problem in Eq. (7.3.6) can be anticipated but will not be described in equation form. The radiation and chemical reaction problems were described in detail in Chapter 3.

The general thermal problem described above can be coupled to a solid mechanics problem via several mechanisms. Deformation of the material leads to a new spatial orientation (new coordinates) and new density distribution. The deformation may also produce new surface orientations that influence radiation and/or surface exposures that lead to changes in thermal boundary conditions, e.g., contact. Mechanical failure can also lead to new surface definitions. Dissipation mechanisms during deformation may produce a significant volumetric heat source for the energy balance. Surface tractions in the mechanical problem will influence thermal contact at material interfaces and frictional heating models. Finally, reaction rates may be influenced by the material stress state (pressure).

The usual sources of coupling for a thermal problem to an electromagnetics problem are somewhat fewer in number. These include volumetric heat sources due to Joule heating and the possibility of thermal property dependence on electric and magnetic field strength.

7.3.3 Finite Element Models

The finite element method for the boundary value problems described above have been presented in great detail in previous chapters. For completeness and ease of reference the standard matrix form of the discretized equations will be presented here. For the heat conduction problem, the finite element equations are given by Eqs. (3.6.3) and (3.6.4) as

$$\mathbf{M}\dot{\mathbf{T}} + \hat{\mathbf{K}}\mathbf{T} = \hat{\mathbf{F}} \quad (7.3.7)$$

with

$$\hat{\mathbf{K}} = \mathbf{K} + \mathbf{C} + \mathbf{R} \quad (7.3.8a)$$

$$\hat{\mathbf{F}} = \mathbf{Q} - \mathbf{q}_a + \mathbf{F}_{hc} + \mathbf{F}_{hr} \quad (7.3.8b)$$

In the standard uncoupled problem, the matrices \mathbf{M} , $\hat{\mathbf{K}}$, and $\hat{\mathbf{F}}$ are functions of temperature and/or time. The coupled problem will introduce other dependencies which will be discussed in a later section.

In the case of the nonisothermal, viscous flow problem, the finite element equations are given by Eqs. (5.3.10)–(5.3.12), which are

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (7.3.9)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{v} - \mathbf{Q}\mathbf{P} + \mathbf{B}\mathbf{T} = \mathbf{F} \quad (7.3.10)$$

$$\mathbf{N}\dot{\mathbf{T}} + \mathbf{D}\mathbf{T} + \mathbf{L}\mathbf{T} = \mathbf{G} \quad (7.3.11)$$

and represent the continuity, momentum, and energy equations. Like the heat conduction problem the dependencies of the various terms in (7.3.9)–(7.3.11) will be described in a later section for the various types of coupling.

7.4 Solid Mechanics

7.4.1 Introduction

Computational solid mechanics is a very broad discipline with a well-developed finite element base and widespread use in engineering analysis. Solid mechanics applications that will be of interest in the context of coupled problems are limited to large deformation and inelastic response of solid materials. Dynamics problems are also of interest but are not considered here because of the limited coupling with the diffusion dominated thermal problems and viscous incompressible flow problems that are the main focus of the book. The coverage of even this limited area in solid mechanics must, by necessity, be rather superficial with little explanation of important aspects, such as constitutive behavior and finite element models and solution methods. The interested reader may consult any of several comprehensive texts dedicated to finite elements in solid mechanics [1–8]. The details of theoretical concepts and derivations from solid mechanics may be found in continuum mechanics or elasticity books (see [9–13] and references therein).

7.4.2 Kinematics of Deformation

The starting point for describing the equations of solid mechanics is usually kinematics, which deals with changes in geometry (i.e., deformation) of the body under external loads. Various measures of deformation are introduced in the form of strain tensor and deformation rate tensor. The geometric changes are accompanied by stresses in the body. Suitable measures of stress are introduced and equations of motion are derived using the principles of conservation of linear and angular momenta, as discussed in Section 7.4.3.

The simultaneous positions occupied in space by all material points of the solid body at different instants of time are called configurations. In the study of solid bodies, it is usual to start with a reference configuration, often the undeformed configuration, of a body and describe its subsequent motion and deformation. Suppose that the continuum initially occupies a configuration in which a particle X occupies the position \mathbf{X} , referred to a rectangular Cartesian system (X_1, X_2, X_3) , called the *material coordinates*. After the application of the loads, the continuum changes its geometric shape and thus assumes a new configuration in which the particle X occupies the position \mathbf{x} , as shown in Figure 7.4.1. The mapping χ from the reference configuration to the deformed configuration, called the *deformation mapping*, takes the position vector \mathbf{X} from the reference configuration and places the same point in the deformed configuration as

$$\mathbf{x} = \chi(\mathbf{X}, t) \quad (7.4.1)$$

7.4.2.1 Descriptions of motion

The mathematical description of the deformation of a continuous body follows one of the two approaches: (1) spatial description and (2) material description. The spatial description is known as the *Eulerian description* and the material description is also known as the *Lagrangian description*. In the spatial description, the motion is referred to the current configuration occupied by the continuum, and a typical field variable ϕ is described with respect to the current position \mathbf{x} in space, currently occupied by material particle X :

$$\phi = \phi(\mathbf{x}, t), \quad \mathbf{X} = \mathbf{X}(\mathbf{x}, t) \quad (7.4.2)$$

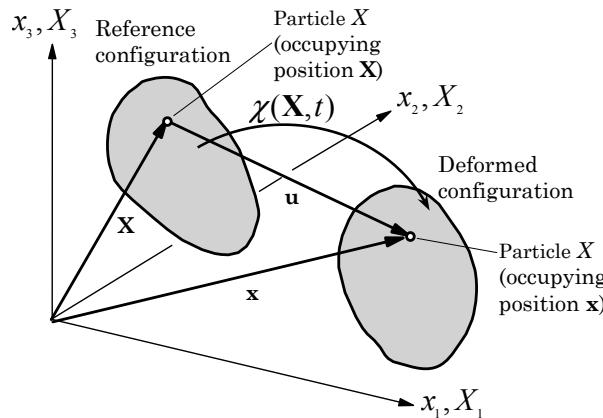


Figure 7.4.1: Reference and deformed configurations of a body.

The coordinates (\mathbf{x}) are termed the *spatial coordinates*. For a fixed value of \mathbf{x} , $\phi(\mathbf{x}, t)$ gives the value of ϕ associated with a fixed point \mathbf{x} in space, which will be the value of ϕ associated with different material points at different times, because different material points occupy the position \mathbf{x} at different times. Thus, a change in time t implies that a different value ϕ is observed at the *same* spatial location \mathbf{x} , now probably occupied by a different material particle X . Hence, attention is focused on a spatial position \mathbf{x} .

In the Lagrangian description, the current coordinates \mathbf{x} are expressed in terms of the reference coordinates \mathbf{X}

$$\mathbf{x} = \chi(\mathbf{X}, t), \quad \chi(\mathbf{X}, 0) = \mathbf{X} \quad (7.4.3)$$

and the variation of a typical variable ϕ over the body is described with respect to the material coordinates \mathbf{X} and time t :

$$\phi = \phi(\mathbf{X}, t) \quad (7.4.4)$$

For a fixed value of \mathbf{X} from the reference configuration, $\phi(\mathbf{X}, t)$ gives the value of ϕ at time t associated with the fixed material point X whose position in the reference configuration is \mathbf{X} , as shown in Fig. 7.4.2. Thus, a change in time t implies that the *same* material particle X occupying position \mathbf{X} has a different value ϕ . Thus the attention is focused on the material particles X of the continuum.

The Eulerian description is the preferred description for the study of motion of fluids because the configuration is known and remains unchanged, and changes in the fluid velocities, pressure, density and so on are determined, as discussed in much of this book. In the study of solid bodies, the Eulerian description is less useful since the deformed configuration is unknown.

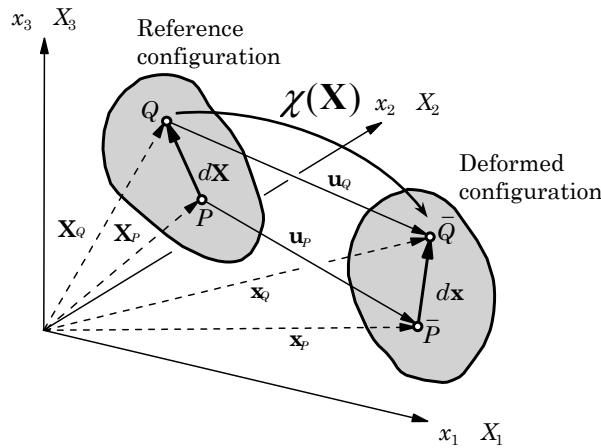


Figure 7.4.2: Reference configuration and deformed configurations in the material description.

7.4.2.2 Displacement vector

The word *deformation* refers to relative displacements and changes in the geometry experienced by the body under the influence of a force system. The displacement of the particle X is given by

$$\mathbf{u} = \mathbf{x} - \mathbf{X} \quad (7.4.5)$$

In the Lagrangian description, the displacements are expressed in terms of the material coordinates X_i

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}. \quad (7.4.6)$$

In the material description, the velocity and acceleration vectors are simply given by [note that $\mathbf{u} = \mathbf{u}(\mathbf{X}, t)$]

$$\mathbf{v}(\mathbf{X}, t) = \frac{\partial \mathbf{u}}{\partial t}, \quad \mathbf{a}(\mathbf{X}, t) = \frac{\partial \mathbf{v}}{\partial t} = \frac{\partial^2 \mathbf{u}}{\partial t^2} \quad (7.4.7)$$

A rigid-body motion is one in which all material particles of the body undergo the same linear and angular displacements. On the other hand, a deformable body is one in which the material particles can move relative to each other. Then the deformation of a body can be determined only by considering the change of distance between any two arbitrary but infinitesimally close points of the body.

7.4.2.3 Deformation gradient tensor

One of the key quantities in deformation analysis is the *deformation gradient* relative to the reference configuration, denoted \mathbf{F} , which gives the relationship of a material line $d\mathbf{X}$ before deformation to the line $d\mathbf{x}$ (consisting of the same material as $d\mathbf{X}$) after deformation. It is defined as [9]

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X} = d\mathbf{X} \cdot \mathbf{F}^T \quad (7.4.8a)$$

$$\mathbf{F} = \left(\frac{\partial \chi}{\partial \mathbf{X}} \right)^T = \left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}} \right)^T \equiv (\nabla_0 \mathbf{x})^T \quad (7.4.8b)$$

and ∇_0 is the gradient operator with respect to \mathbf{X} . By definition, \mathbf{F} is a second-order tensor. The inverse relations are given by

$$d\mathbf{X} = \mathbf{F}^{-1} \cdot d\mathbf{x} = d\mathbf{x} \cdot \mathbf{F}^{-T}, \quad \mathbf{F}^{-T} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \equiv \nabla \mathbf{x} \quad (7.4.9)$$

and ∇ is the gradient operator with respect to \mathbf{x} . In indicial notation, Eqs. (7.4.8b) and (7.4.9) can be written as

$$\mathbf{F} = F_{iJ} \hat{\mathbf{e}}_i \hat{\mathbf{E}}_J, \quad F_{iJ} = \frac{\partial x_i}{\partial X_J} \quad \text{and} \quad \mathbf{F}^{-1} = F_{Ji}^{-1} \hat{\mathbf{E}}_J \hat{\mathbf{e}}_i, \quad F_{Ji}^{-1} = \frac{\partial X_J}{\partial x_i} \quad (7.4.10)$$

The determinant of \mathbf{F} is called the *Jacobian of the motion*, and it is denoted by $J = \det \mathbf{F}$. The equation $\mathbf{F} \cdot d\mathbf{X} = 0$ for $d\mathbf{X} \neq 0$ implies that a material line in the reference configuration is reduced to zero by the deformation. Since this is physically not realistic, we conclude that $\mathbf{F} \cdot d\mathbf{X} \neq 0$ for $d\mathbf{X} \neq 0$. That is, \mathbf{F} is a non-singular tensor, $J \neq 0$. Hence, \mathbf{F} has an inverse \mathbf{F}^{-1} .

7.4.2.4 Green strain tensor

The geometric changes that a solid continuum experiences can be measured in a number of ways. Here, we discuss a general measure of deformation of a solid body. Consider two material particles P and Q in the neighborhood of each other, separated by $d\mathbf{X}$ in the reference configuration. In the current (deformed) configuration the material points P and Q occupy positions \bar{P} and \bar{Q} , and they are separated by $d\mathbf{x}$, as shown in Figure 7.4.2. The change in the squared lengths that occurs as a body deforms from the reference configuration to the current configuration can be expressed relative to the original length as

$$(ds)^2 - (dS)^2 = d\mathbf{x} \cdot d\mathbf{x} - d\mathbf{X} \cdot d\mathbf{X} = d\mathbf{X} \cdot (\mathbf{F}^T \cdot \mathbf{F}) \cdot d\mathbf{X} - d\mathbf{X} \cdot d\mathbf{X} \equiv 2 d\mathbf{X} \cdot \mathbf{E} \cdot d\mathbf{X} \quad (7.4.11)$$

where

$$\begin{aligned} \mathbf{E} &= \frac{1}{2} (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) = \frac{1}{2} [(\mathbf{I} + \nabla_0 \mathbf{u}) \cdot (\mathbf{I} + \nabla_0 \mathbf{u})^T - \mathbf{I}] \\ &= \frac{1}{2} [\nabla_0 \mathbf{u} + (\nabla_0 \mathbf{u})^T + (\nabla_0 \mathbf{u}) \cdot (\nabla_0 \mathbf{u})^T] \end{aligned} \quad (7.4.12)$$

By definition, \mathbf{E} is a symmetric second-order tensor, called the *Green strain tensor*. In index notation, the rectangular Cartesian components of \mathbf{E} are given by

$$E_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right) \quad (7.4.13)$$

The components of \mathbf{E} in other coordinate systems can be obtained using the definition (7.4.13) and writing ∇_0 and \mathbf{u} in that coordinate system.

The Green strain tensor can be expressed as the sum of linear and nonlinear parts

$$\mathbf{E} = \mathbf{e}^1 + \mathbf{e}^2, \quad E_{ij} = e_{ij}^1 + e_{ij}^2 \quad (7.4.14)$$

where

$$\mathbf{e}^1 = \frac{1}{2} [\nabla_0 \mathbf{u} + (\nabla_0 \mathbf{u})^T], \quad \mathbf{e}^2 = \frac{1}{2} (\nabla_0 \mathbf{u}) \cdot (\nabla_0 \mathbf{u})^T \quad (7.4.15a)$$

$$e_{ij}^1 = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right), \quad e_{ij}^2 = \frac{1}{2} \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \quad (7.4.15b)$$

Other strain measures can be defined. For example, the change in the squared lengths that occurs as the body deforms from the initial to the current configuration can be expressed relative to the current length as

$$(ds)^2 - (dS)^2 = 2 d\mathbf{x} \cdot \mathbf{e} \cdot d\mathbf{x} \quad (7.4.16)$$

where \mathbf{e} is called the *Euler strain tensor*, which can be expressed as

$$\mathbf{e} = \frac{1}{2} (\mathbf{I} - \mathbf{F}^{-T} \cdot \mathbf{F}^{-1}) = \frac{1}{2} (\mathbf{I} - \mathbf{B}^{-1}) \quad (7.4.17)$$

where $\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T$ is the *Cauchy strain tensor*, and its inverse is called the *left Cauchy–Green or Finger tensor*.

The choice of a strain measure is mainly one of convenience for the type of constitutive model used. When path-dependent material response is important, it is usual to formulate the equations in terms of strain rates and velocities rather than strains and displacements. Because we will not be exploring constitutive relations in detail, the rate-dependent definitions will not be discussed here.

7.4.3 Kinetics

7.4.3.1 Stress measures

The Cauchy stress tensor σ used in fluid mechanics is defined as the force per unit area in the current configuration. The equations of motion or equilibrium must be derived for the deformed configuration. However, since the geometry of the deformed configuration is not known, the equations must be written in terms of the known reference configuration. In doing so we introduce the second Piola–Kirchhoff stress tensor $\tilde{\sigma}$, which characterizes the current force in the deformed configuration but transformed to undeformed configuration, and measured per unit area in the undeformed configuration. The second Piola–Kirchhoff stress tensor is energetically conjugate to the rate of Green strain tensor in the sense that the rate of internal work done (power) in a continuous medium is given by

$$W = \frac{1}{2} \int_v \sigma : \mathbf{D} d\mathbf{x} = \frac{1}{2} \int_V \tilde{\sigma} : \dot{\mathbf{E}} d\mathbf{X} \quad (7.4.18)$$

where σ is the Cauchy stress tensor and \mathbf{D} is the symmetric part of the *velocity gradient tensor*. This emerges in a natural way as we transform volumes and areas from the deformed configuration \mathcal{C} to undeformed configuration \mathcal{C}^0 (see Reddy [8,9] and Bonet and Wood [10] for details).

The second Piola–Kirchhoff stress tensor $\tilde{\sigma}$ is related to the Cauchy stress σ tensor by

$$\tilde{\sigma} = \frac{\rho_0}{\rho} \mathbf{F}^{-1} \sigma \mathbf{F}^{-T} \quad (7.4.19a)$$

$$\tilde{\sigma}_{ij} = \frac{\rho_0}{\rho} \frac{\partial X_i}{\partial x_k} \sigma_{kl} \frac{\partial X_j}{\partial x_l} \quad (7.4.19b)$$

An inverse relation exists where the Cauchy stress is a function of the Piola–Kirchhoff stress

$$\sigma = \frac{\rho}{\rho_0} \mathbf{F} \tilde{\sigma} \mathbf{F}^T \quad (7.4.20a)$$

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \tilde{\sigma}_{kl} \frac{\partial x_j}{\partial X_l} \quad (7.4.20b)$$

7.4.3.2 Equilibrium statements

The equations of equilibrium for a solid are provided by the principle of conservation of linear momentum and they are the same as described in the previous chapters

$$\nabla \cdot \sigma + \rho \mathbf{b} = \mathbf{0}, \quad \frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i = 0 \quad (7.4.21)$$

where σ is the Cauchy stress tensor, ρ is the density and \mathbf{b} is the body force vector.

The finite element models in solid mechanics are often based on principles of virtual work and the principle of minimum total potential energy [9,13], which are equivalent to conservation of linear momentum. The virtual work statement in the deformed configuration is transformed to one on the undeformed configuration using the equivalence in Eq. (7.4.18). Consequently, the virtual work statement is cast in terms of the second Piola–Kirchhoff stress tensor and the Green strain tensor. We shall return to the discussion of the virtual work statement in Section 7.4.5.

A statement of mass conservation is also required and in material coordinates is given by

$$\rho \mathbf{J} = \rho \left| \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \right| = \rho_0 \quad (7.4.22)$$

which is valid for each material point. In Eq. (7.4.22) the current density is ρ , the initial density is ρ_0 and \mathbf{J} is the determinant of the Jacobian and specifies the ratio of the current configuration to the initial state. The mass conservation equation need not be enforced explicitly in most Lagrangian methods but is used to account for density changes in the material.

Coupling of the solid mechanics problem to a thermal analysis may occur through several phenomena and requires the addition of the energy equation to the equations of equilibrium. Temperature- and/or species-dependent material properties (constitutive parameters) are a common occurrence. Also, the material state, such as the extent of reaction, decomposition or gas fraction, and material addition or removal due to a thermal process, will influence the mechanical response. Changes in mechanical boundary conditions, e.g., pressure loading, may occur due to heat transfer. Mechanical response may be influenced by electromagnetics primarily through Lorentz body forces (magnetic pressures) and field-dependent constitutive response, as illustrated by piezoelectric materials.

7.4.4 Constitutive Relations

Constitutive relations for the stress-strain response of a solid material are quite numerous and range in complexity from simple, linear models to multi-variable, path-dependent, nonlinear models. Here, we will limit the outline to linear and nonlinear elastic constitutive relations and leave the complexities of elastic-plastic, creep, and viscoelastic behavior to the more specialized sources [3,9–12].

A standard elastic constitutive relation (generalized Hooke's law) that is valid for large deformation can be written as

$$\tilde{\sigma} = \mathbf{C} \mathbf{E}, \quad \tilde{\sigma}_{ij} = C_{ijkl} E_{kl} \quad (7.4.23)$$

where $\tilde{\sigma}$ is the second Piola–Kirchhoff stress, \mathbf{E} is the Green strain tensor, \mathbf{C} is the elasticity tensor.

For small strains but large displacements and moderate rotations, the elasticity tensor \mathbf{C} is assumed to be constant. For an isotropic material it can be written in its usual form involving the Lamé constants λ and μ

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad (7.4.24)$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} , \quad \mu = \frac{E}{2(1+\nu)} \quad (7.4.25)$$

where E is the modulus of elasticity and ν is Poisson's ratio.

$$\sigma = \mathbf{C} \mathbf{e}^1 \quad \text{or} \quad \sigma_{ij} = C_{ijkl} e_{kl}^1 = \lambda e_{kk}^1 \delta_{ij} + 2\mu e_{ij}^1 \quad (7.4.26)$$

When strains are not small, the relation in (7.4.23) can still be used though the elasticity tensor \mathbf{C} is no longer a constant. If the tensor \mathbf{C} depends on the strains, the functional dependence is usually expressed in terms of the invariants of the strain tensor. This type of material is usually termed hyperelastic and the elasticity tensor is most often derived from a strain energy functional. Another nonlinear elastic material model defines the elasticity tensor to be a function of a number of variables including the strain, stress, load, damage, and so on. This type of model is termed hypoelastic and is usually written in an incremental form. Note that the above constitutive relations can be augmented with a term that produces a thermal stress. This additional effect is quite common and leads directly to coupling with the energy equation.

7.4.5 Boundary Conditions

The boundary conditions for the momentum equation in (7.4.9) are standard and consist of specified displacements on part of the boundary

$$u_i = f_i^u(s_k, t) \quad \text{on} \quad \Gamma_u \quad (7.4.27)$$

and specified tractions on the remainder of the boundary

$$\tau_{ij} n_j = f_i^\tau(s_k, t) \quad \text{on} \quad \Gamma_\tau \quad (7.4.28)$$

The displacement conditions are often homogeneous and may be used to eliminate the rigid body motions from the deforming geometry. Specified displacements or velocities are also used to define the boundary motion while the region undergoes deformation. Applied tractions, often in the form of normal pressures, provide another method for loading the structure. The contact boundary condition, either frictionless or with friction, is in widespread use but is difficult to implement. In quasi-static problems it is often difficult to uniquely define the geometric aspects of contact and to know how to apportion the forces and deformation between contacting bodies of comparable strength.

7.4.6 Finite Element Models

The development of a finite element model for solid mechanics is normally accomplished through the principles of virtual work or the principle of minimum total potential energy [8,13]. These are equivalent to the weak forms or variational problems, and they form the basis of most finite element models in solid mechanics. The weak form formulations are also equivalent to these two principles. In this section the principle of virtual displacements is used to derive a finite element model.

The total virtual work done by actual forces in moving through virtual displacements, δW , for a solid body consists of the virtual internal (strain) energy δW_I and the virtual work done δW_e by the volume and surface forces. The virtual internal energy is taken to be only the virtual strain energy for the solid region and

it is expressed in terms of the Piola–Kirchhoff stress components $\tilde{\sigma}_{ij}$ and the virtual Green strain components δE_{ij} as

$$\delta W_I = \int_{\Omega_0} \tilde{\sigma}_{ij} \delta E_{ij} d\mathbf{X} \quad (7.4.29)$$

where Ω_0 is the volume occupied by the undeformed or reference configuration \mathcal{C}^0 . Using the constitutive relation (7.4.23) and the decomposition of the strain into linear and quadratic terms, the strain energy can be written as [8,13]

$$\begin{aligned} \delta W_I &= \int_{\Omega_0} C_{ijkl}(e_{kl}^1 + e_{kl}^2)(\delta e_{ij}^1 + \delta e_{ij}^2) d\mathbf{X} \\ &= \int_{\Omega_0} [C_{ijkl} e_{kl}^1 \delta e_{ij}^1 + C_{ijkl} (e_{kl}^1 \delta e_{ij}^2 + e_{ij}^2 \delta e_{kl}^1) + C_{ijkl} e_{kl}^2 \delta e_{ij}^2] d\mathbf{X} \end{aligned} \quad (7.4.30)$$

The strain components appearing in Eq. (7.4.30) can be expressed in terms of the displacement derivatives using the definitions in (7.4.15b). The virtual work done by the applied forces is given by

$$\delta W_E = - \left[\int_{\Omega_0} b_i \delta u_i d\mathbf{X} + \int_{\Gamma_0} f_i^\tau \delta u_i dS \right] \quad (7.4.31)$$

where Γ_0 is the boundary of the region Ω_0 and u_i are the displacement components. Note that the total virtual work done $\delta W = \delta W_I + \delta W_E$ is only a function of the displacements. The principle of virtual displacements, $\delta W = 0$, yields the equilibrium equations and force boundary conditions of the problem.

A finite element approximation is defined for the nodal values of the displacement components u_i as

$$u_i(\mathbf{x}, t) = \Phi^T \mathbf{u}_i \quad (7.4.32)$$

where Φ are the shape functions. The strain-displacement relations can then be expressed in matrix form as

$$\mathbf{e}^1 = \mathbf{D}^1 \mathbf{u} \quad , \quad \mathbf{e}^2 = \mathbf{u}^T \mathbf{D}^2 \mathbf{u} \quad (7.4.33)$$

where \mathbf{D}^i are matrix differential operators as implied by Eq. (7.4.15b). Substituting (7.4.32) and (7.4.33) into the expression for $\delta W = \delta W_I + \delta W_E$, we obtain the displacement finite element model of the problem in the form

$$[\mathbf{K}^0(\mathbf{u}) + \mathbf{K}^1(\mathbf{u}) + \mathbf{K}^2(\mathbf{u})] \mathbf{u} = \mathbf{F} \quad (7.4.34)$$

where each of the stiffness terms corresponds to the terms in (7.4.30), and the possible dependency of the elasticity tensor on the strain has been indicated. The linear, small strain and small deformation part is embodied in coefficient matrix \mathbf{K}^0 ; material nonlinearity may also be present in this term. The second matrix \mathbf{K}^1 is usually identified with geometric stiffness since it contains the quadratic strain terms. The matrix \mathbf{K}^2 is a nonlinear term that contains the quadratic strains and possible material nonlinearities.

7.4.7 Solution Methods

The finite element equations developed in the previous section for the quasi-static response of a solid generally represent a system of highly nonlinear algebraic equations. Two basic methods of solution may be considered. In the first case, the full matrix system in (7.4.34) is constructed and solved using any of the fixed point iterative methods, such as Newton's method. There are generally numerous difficulties with this approach, though when the problem is suitable the method may be very effective. The biggest drawbacks to direct iteration are nonconvergence of the iterative process and construction of the Jacobian for Newton's method. With any nonlinear problem, the starting point for the iterative procedure is crucial in achieving convergence (see Appendix C). The starting guess for most problems involving geometric and/or material nonlinearities will be far enough from the final, equilibrium solution that convergence will not be possible. The construction of the Jacobian is also difficult for many material models that are not analytic; the geometric nonlinearities can be incorporated analytically into the Jacobian through a tangent stiffness matrix.

The second approach to solving (7.4.34) involves an incremental or imbedding process, in which the load is applied in a series of increments and a sequence of equilibrium problems is solved. This method has the distinct advantage of keeping the next solution in the sequence "close" to the previously converged solution. Iterative methods, like Newton's method, are more robust in this incremental algorithm and convergence is more rapid. Though not formulated here, path- or history-dependent material models must be approached with an incremental strategy. The variety of incremental procedures is quite large and outside the scope of this discussion. Additional details on solution algorithms for solid mechanics problems may be found in [3,9-12].

7.5 Electromagnetics

7.5.1 Introduction

Problems involving the coupling of electromagnetic (EM) fields with fluid and thermal transport have a broad spectrum of applications ranging from astrophysics to manufacturing and to electromechanical devices and sensors. Here we will limit the discussion to the interaction of electromagnetic fields with solid bodies or incompressible fluids that are good electrical conductors. This eliminates many of the interesting problems involving plasmas, and concentrates the applications in the area of metals and liquid metal flows. In many of these problems an applied or induced magnetic field provides an additional body force to the fluid, which results in a convective motion. For high current applications, resistive heating may also be important as a volumetric energy source.

In the following section an outline of the field equations for electromagnetics is given along with their coupling to the nonisothermal, viscous flow problem and the solid body heat conduction problem. Subsequent sections show how the EM problem is redefined in terms of potential functions that are more suitable for finite element model development, and then describe some of the numerical issues associated with EM field simulation. A good introduction to coupled fluid-EM problems is available in [14]; general EM field theory is available in texts such as the one by Jackson

[15]. Finite element models for EM applications are well covered in the texts by Jin [16], Sadiku [17], Binns, et al. [18] and Silvester [19]. The coverage of the electromagnetics problem is substantially more detailed than the solid mechanics problem mainly because it is less familiar to practitioners of thermal sciences.

7.5.2 Maxwell's Equations

The appropriate mathematical description of electromagnetic phenomena in a conducting material region, Ω_C , is given by Maxwell's equations. In rational MKSA notation these equations are expressed as (see [9,10])

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (7.5.1)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \quad (7.5.2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (7.5.3)$$

$$\nabla \cdot \mathbf{D} = \rho \quad (7.5.4)$$

where the field variables are the electric field intensity, \mathbf{E} , the magnetic field intensity, \mathbf{H} , the magnetic flux density, \mathbf{B} , the electric flux (displacement) density, \mathbf{D} , the conduction current density, \mathbf{J} , and the source charge density, ρ . Typically, Eq. (7.5.1) is referred to as Faraday's law, Eq. (7.5.2) as Ampere's law (as modified by Maxwell), and Eq. (7.5.4) as Gauss's law. A continuity condition on the current density is also defined by

$$\nabla \cdot \mathbf{J} = \frac{\partial \rho}{\partial t} \quad (7.5.5)$$

Note that only three of the above five equations are independent; the combinations (7.5.1), (7.5.2) and (7.5.4) or (7.5.1), (7.5.2) and (7.5.5) form valid descriptions of the fields.

7.5.2.1 Constitutive relations

To complete the formulation, the constitutive relations for the material are required. The fluxes are functionally related to the field variables by

$$\mathbf{D} = f_D(\mathbf{E}, \mathbf{B}) \quad (7.5.6)$$

$$\mathbf{H} = f_H(\mathbf{E}, \mathbf{B}) \quad (7.5.7)$$

$$\mathbf{J} = f_J(\mathbf{E}, \mathbf{B}) \quad (7.5.8)$$

where the functions (f_D, f_H, f_J) may also depend on external variables such as temperature or mechanical stress. The form of the material response to applied \mathbf{E} or \mathbf{B} fields can vary strongly depending on the state of the material, its microstructure and the strength, and time-dependent behavior of the applied field.

Conductive and dielectric materials

For conducting materials, the standard f_J relation is Ohm's law which relates the current density \mathbf{J} to the electric field intensity \mathbf{E}

$$\mathbf{J} = \sigma \cdot \mathbf{E} \quad (7.5.9)$$

where σ is the conductivity tensor. For isotropic materials σ is a scalar. In general, the conductivity may be a function of \mathbf{E} or an external variable such as temperature. This form of Ohm's law applies to stationary conductors. If the conductive material is moving in a magnetic field, then Eq. (7.5.9) is modified to read

$$\mathbf{J} = \sigma \cdot \mathbf{E} + \sigma \cdot (\mathbf{v} \times \mathbf{B}) \quad (7.5.10)$$

where \mathbf{v} is the velocity vector describing the motion of the conductor and \mathbf{B} is the magnetic flux vector.

For dielectric materials, the standard f_D function relates the electric flux density \mathbf{D} to the electric field \mathbf{E} and polarization vector \mathbf{P}

$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P} \quad (7.5.11)$$

where ϵ_0 is the permittivity of free space. The polarization is generally related to the electric field through

$$\mathbf{P} = \epsilon_0 \chi_e \mathbf{E} + \mathbf{P}_0 \quad (7.5.12)$$

where χ_e is the electric susceptibility tensor that accounts for the different types of polarization, and \mathbf{P}_0 is the remnant polarization that may be present in some materials. The electric susceptibility is always positive and may be a simple scalar (linearly polarizable material) or a field-dependent scalar or tensor. In some situations the polarization exhibits a hysteretic behavior with respect to the electric field (ferroelectric material) and the susceptibility is defined as the local slope of the \mathbf{P} versus \mathbf{E} curve. Combining Eqs. (7.5.11) and (7.5.12), we obtain

$$\mathbf{D} = (\epsilon_0 \mathbf{I} + \epsilon_0 \chi_e) \cdot \mathbf{E} + \mathbf{P}_0 = \epsilon_0 (\mathbf{I} + \chi_e) \cdot \mathbf{E} + \mathbf{P}_0 = \epsilon \cdot \mathbf{E} + \mathbf{P}_0 \quad (7.5.13)$$

where \mathbf{I} is the unit tensor and $\epsilon = \epsilon_0 (\mathbf{I} + \chi_e)$. Note that if \mathbf{P}_0 exists, it only needs to be explicitly defined for linear materials (i.e., when ϵ is independent of \mathbf{E}); for nonlinear constitutive behavior, \mathbf{P}_0 is usually absorbed in the functional form for ϵ .

Magnetic materials

For magnetic materials, the standard f_H function relates the magnetic field intensity \mathbf{H} to the magnetic flux \mathbf{B}

$$\mathbf{H} = \frac{1}{\mu_0} \mathbf{B} - \mathbf{M} \quad (7.5.14)$$

where μ_0 is the permeability of free space and \mathbf{M} is the magnetization vector. The magnetization can be related to either the magnetic flux or magnetic intensity

$$\mathbf{M} = \frac{\chi_m}{(\mathbf{I} + \chi_m)} \frac{1}{\mu_0} \cdot \mathbf{B} + \mathbf{M}_0 \quad (7.5.15a)$$

or

$$\mathbf{M} = \chi_m \cdot \mathbf{H} + (\mathbf{I} + \chi_m) \cdot \mathbf{M}_0 \quad (7.5.15b)$$

where χ_m is the magnetic susceptibility for the material and \mathbf{M}_0 is the remnant magnetization. If the susceptibility is negative, the material is diamagnetic; while a positive susceptibility defines a paramagnetic material. Generally, these

susceptibilities are quite small and are often neglected. Ferromagnetic materials have large positive susceptibilities and produce a nonlinear (hysteretic) relationship between \mathbf{B} and \mathbf{H} . These materials may also exhibit spontaneous and remnant magnetization. Combining Eq. (7.5.14) with either Eq. (7.5.15a) or Eq. (7.5.15b) leads to

$$\mathbf{H} = \frac{1}{\mu_0(\mathbf{I} + \chi_m)} \cdot \mathbf{B} - \mathbf{M}_0 = \frac{1}{\mu} \cdot \mathbf{B} - \mathbf{M}_0 = \nu \cdot \mathbf{B} - \mathbf{M}_0 \quad (7.5.16)$$

Here $\mu = \mu_0(\mathbf{I} + \chi_m)$ is the permeability tensor for the material. The relative permeability is $\mu_r = \mu/\mu_0 = (\mathbf{I} + \chi_m)$ and the reluctivity ν , is defined as the inverse of the permeability. The remnant magnetization, \mathbf{M}_0 , need only be explicitly defined for linear magnetic materials (e.g., permanent magnets); in the nonlinear case \mathbf{M}_0 is usually absorbed into the $\mu(\mathbf{B})$ function.

7.5.2.2 Electromagnetic forces and volume heating

The coupling of electromagnetic fields with a fluid or thermal problem occurs through the dependence of material properties on EM field quantities and the production of EM-induced body forces and volumetric energy production. The Lorentz body force in a conductor due to the presence of electric currents and magnetic fields is given by

$$\mathbf{F}_B = \rho \mathbf{E} + \mathbf{J} \times \mathbf{B} \quad (7.5.17)$$

where, in the general case, the current is defined by Eq. (7.5.10). The first term on the right-hand side of Eq. (7.5.17) is the electric field contribution to the Lorentz force; the magnetic term $\mathbf{J} \times \mathbf{B}$ is usually of more interest in applied mechanics problems. The energy generation or Joule heating in a conductor is described by

$$Q_J = \mathbf{J} \cdot \mathbf{E} \quad (7.5.18)$$

which takes on a more familiar form if the simplified ($\mathbf{u} = \mathbf{0}$) form of Eq. (7.5.10) is used to produce

$$Q_J = \sigma^{-1}(\mathbf{J} \cdot \mathbf{J}) \quad (7.5.19)$$

The above forces and heat source occur in the fluid momentum and energy equations, respectively.

7.5.2.3 Quasi-static approximation

For good conductors, the conduction current \mathbf{J} is large compared to the displacement current \mathbf{D} for most frequencies of interest. Neglecting the displacement current allows Ampere's law (7.5.2) to be simplified and Coulomb's law (7.5.4) to be omitted. Also, the continuity relation is simplified since the divergence of $\nabla \times \mathbf{H}$ is zero by a vector identity. The omission of the displacement current is termed a quasi-static approximation (pre-Maxwell equations) since the propagation of electromagnetic waves is precluded. Under this assumption Maxwell's equations for a conducting region become

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (7.5.20)$$

$$\nabla \times \mathbf{H} = \mathbf{J} \quad (7.5.21)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (7.5.22)$$

The continuity condition is

$$\nabla \cdot \mathbf{J} = 0 \quad (7.5.23)$$

The simplified forms of the required constitutive relations are

$$\mathbf{B} = \mu \cdot \mathbf{H} \text{ or } \mathbf{H} = \nu \cdot \mathbf{B} \quad (7.5.24)$$

$$\mathbf{J} = \sigma \cdot \mathbf{E} + \sigma \cdot (\mathbf{v} \times \mathbf{B}) \text{ or } \mathbf{J} = \sigma \cdot \mathbf{E} \quad (7.5.25)$$

7.5.3 Electromagnetic Potentials

For many static and quasi-static applications it is usual to introduce a set of potential functions to represent the electric and magnetic field variables and reduce the number of partial differential equations requiring solution. Two basic systems of potentials may be considered: (a) the electric scalar potential V and a magnetic vector potential \mathbf{A} , and (b) the electric vector potential \mathbf{T} and the scalar magnetic potential ψ . The \mathbf{T} - ψ formulation is of limited value for a general analysis since there are significant difficulties in representing multiply-connected domains. Though the \mathbf{A} – V formulation generally leads to a larger number of differential equations, it is preferred for numerical computation due to its complete generality.

From Gauss' relation $\nabla \cdot \mathbf{B} = 0$, it follows that \mathbf{B} is derivable from a vector potential. By definition, then

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (7.5.26)$$

where \mathbf{A} is the *magnetic vector potential*. In addition, from Faraday's law (7.5.20), one has

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} = -\frac{\partial(\nabla \times \mathbf{A})}{\partial t} \quad (7.5.27a)$$

Rearranging the above equation, we obtain

$$\nabla \times \left(\mathbf{E} + \frac{\partial \mathbf{A}}{\partial t} \right) = \mathbf{0} \quad (7.5.27b)$$

For a scalar V , the vector identity $\nabla \times \nabla V = \mathbf{0}$ holds and allows the definition

$$-\nabla V \equiv \mathbf{E} + \frac{\partial \mathbf{A}}{\partial t}$$

or

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t} \quad (7.5.28)$$

The scalar function V is known as the *electric potential*. The definitions in Eqs. (7.5.26) and (7.5.28) may be used in the appropriate forms of Ampere's law and the current continuity equation to produce the needed field equations for \mathbf{A} and V in conductors and in free-space regions. Using Eqs. (7.5.24)–(7.5.26) and (7.5.28) in Eq. (7.5.21), one obtains

$$\nabla \times [\nu \cdot (\nabla \times \mathbf{A})] = -\sigma \cdot \frac{\partial \mathbf{A}}{\partial t} - \sigma \cdot \nabla V \quad (7.5.29)$$

Also, in the conduction regions the electric field is described by the continuity equation (7.5.23), which is rewritten in terms of \mathbf{A} and V by the use of Eqs. (7.5.25b) and (7.5.28)

$$\nabla \cdot \left(-\sigma \cdot \nabla V - \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} \right) = 0 \quad (7.5.30)$$

Equations (7.5.29)–(7.5.30) describe the general, quasi-static electromagnetic field problem for free space and conductors in terms of the magnetic vector potential \mathbf{A} and electric scalar potential V . The above derivation has been executed for stationary conductors with simple material behavior; the formulation is altered slightly if conductors are moving and/or constitutive behavior is more complex. These equations are appropriate for solution by the finite element method.

The general problem outlined above may now be specialized to particular material regions and types of current (\mathbf{J}) specifications. For conduction regions without source currents, both Eqs. (7.5.29) and (7.5.30) are generally required to describe both the electric and magnetic fields. Conduction regions that have specified currents may require some alteration of Eqs. (7.5.29)–(7.5.30), depending on the form of the current source (7.5.8). Electric currents described by distributions of the electric potential require no alteration to the equation set since this specification would appear as a boundary condition on the variable V . However, when current densities are assumed known and are specified directly, the total current density in Ohm's law must be rewritten as the combination of two parts. Define the total current density as

$$\mathbf{J} = \mathbf{J}_s + \mathbf{J}_e \quad (7.5.31)$$

where the known source current \mathbf{J}_s is associated with the electric potential term $-\sigma \cdot \nabla V$, and the induced or self-induced current is defined by the time-dependent magnetic potential term $-\sigma \cdot (\partial \mathbf{A} / \partial t)$. Therefore, in a conductor with a known current density \mathbf{J}_s , Eq. (7.5.29) is rewritten as

$$\nabla \times (\nu \cdot \nabla \times \mathbf{A}) + \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} = \mathbf{J}_s \quad (7.5.32)$$

Also, Eq. (7.5.30) is no longer required in the source region since the imposed current density is assumed to be divergence free; the induced currents in the source conductor are also divergence free due to the gauge condition that will be discussed in a subsequent section. Note that the total current through the source conductor is given by

$$I = \int_{\Gamma} \mathbf{J} \cdot \hat{\mathbf{n}} d\Gamma = \int_{\Gamma} \mathbf{J}_s \cdot \hat{\mathbf{n}} d\Gamma - \int_{\Gamma} \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} \cdot \hat{\mathbf{n}} d\Gamma \quad (7.5.33)$$

where $\hat{\mathbf{n}}$ is the vector normal to the cross-sectional area Γ of the conductor. For the general time-dependent problem, the total current, \mathbf{J} , cannot be specified a priori, since the self-induced portion of the current is obtained as a part of the solution. This implies that an iterative procedure is needed if specified current problems are defined.

The equations needed for free space or dielectric regions are simply those of (7.5.29) with σ set to zero; Eq. (7.5.30) is not required as no electric fields are

considered. Finally, note that simplification of the equations for all regions is possible for problems of reduced dimensionality. If the geometry is two-dimensional (planar or axisymmetric) and the currents and potential gradients are oriented orthogonal to the plane, then Eq. (7.5.30) is not required and Eq. (7.5.29) reduces to a single equation for the remaining (axial or circumferential) component of the magnetic potential.

The general time-dependent equations given in (7.5.29)–(7.5.30) are applicable to any type of time varying field problem. However, in the often encountered special case of a single frequency, time-harmonic excitation (e.g., alternating current), the equations may be simplified through use of a phasor representation. Let any specified current densities be represented as a time harmonic excitation and assume that the electric scalar and magnetic vector potentials have a time-harmonic form that can be represented as

$$\begin{aligned}\mathbf{J}_s &= \mathbf{J}_{s_0} e^{i\omega t} = (\mathbf{J}_s^R + i\mathbf{J}_s^I) e^{i\omega t} \\ V &= V_0 e^{i\omega t} = (V^R + iV^I) e^{i\omega t} \\ \mathbf{A} &= \mathbf{A}_0 e^{i\omega t} = (\mathbf{A}^R + i\mathbf{A}^I) e^{i\omega t}\end{aligned}\quad (7.5.34)$$

where ω is the circular frequency ($= 2\pi f$, f is the imposed AC frequency in Hz), $i = \sqrt{-1}$ and t is the time. The superscripts R and I denote the real and imaginary components of a variable. Then substituting (7.5.34) into (7.5.29) and (7.5.30) and eliminating the common exponential factor produces

$$\nabla \times (\nu \cdot \nabla \times \mathbf{A}_0) + i\omega\sigma\mathbf{A}_0 + \sigma \cdot \nabla V_0 = 0 \quad (7.5.35)$$

$$\nabla \cdot (\sigma \cdot \nabla V_0 + i\omega\sigma \cdot \mathbf{A}_0) = 0 \quad (7.5.36)$$

and for conduction regions with source currents (7.5.32) produces

$$\nabla \times (\nu \cdot \nabla \times \mathbf{A}_0) + i\omega\sigma\mathbf{A}_0 = \mathbf{J}_{s_0} \quad (7.5.37)$$

These complex equations describe the amplitudes for the potentials. Note that the boundary conditions for V and \mathbf{A} must also be expressed in terms of the harmonic approximation given in (7.5.34). Implicit in the use of the phasor representation is the assumption that material properties are independent of the temporal behavior of the electromagnetic fields. This is a particularly stringent requirement that is violated when considering high field applications such as induction heating or most types of ferromagnetic materials.

7.5.4 Boundary and Interface Conditions

Boundary and interface conditions for the quasi-static, electromagnetic field problem are most easily described by reference to the generic domain shown in Figure 7.5.1. The region Ω is composed of a number of different materials ($\Omega = \Omega_C \cup \Omega_J \cup \Omega_D$), several of which are illustrated in the figure. The boundary or interface between two conductors is denoted by Γ_{CC} while the boundary between a dielectric or free space region and a conductor is labelled Γ_{DC} . Note that since the equations for Ω_J and Ω_D are the same except for a source function, no specific designation for an interface between these regions is required. The external boundary of the entire domain Ω

is defined by Γ which may be composed of one or more well-defined segments. A two-dimensional representation of the region is used for simplicity.

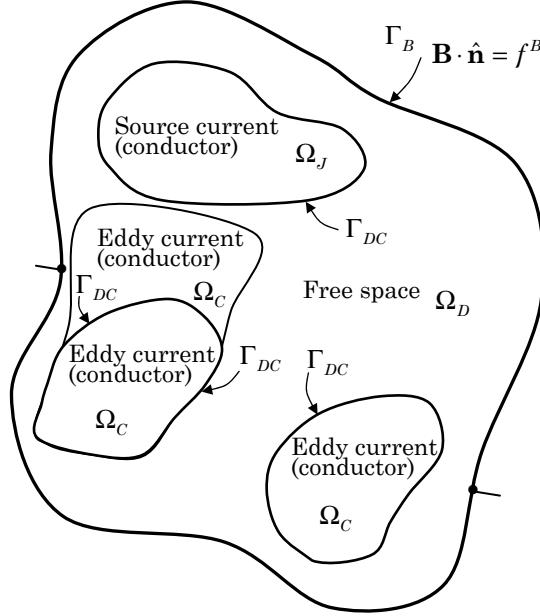


Figure 7.5.1: Schematic of regions for general electromagnetic field problem.

The electromagnetic problem requires that on the exterior free space boundary either the magnetic flux or the magnetic field be specified at all points of the boundary, Γ . In equation form these conditions are given by

$$\mathbf{B} \cdot \hat{\mathbf{n}} = (\nabla \times \mathbf{A}) \cdot \hat{\mathbf{n}} = f^B(s_k, t) \text{ on } \Gamma_B \quad (7.5.38)$$

$$\mathbf{H} \times \hat{\mathbf{n}} = (\nu \cdot \mathbf{B}) \times \hat{\mathbf{n}} = (\nu \cdot \nabla \times \mathbf{A}) \times \hat{\mathbf{n}} = \mathbf{f}^H(s_k, t) \text{ on } \Gamma_H \quad (7.5.39)$$

In Eqs. (7.5.38) and (7.5.39) the f^B and \mathbf{f}^H functions are specified values of the known boundary magnetic flux and magnetic field. Also, $\hat{\mathbf{n}}$ is the outward unit normal to the boundary Γ , s_k are coordinates defined on the boundary, t is the time, and $\Gamma = \Gamma_B \cup \Gamma_H$. The functions f^B and \mathbf{f}^H are generally simple expressions for most boundaries of practical interest. When a conductor forms part of the external boundary the above conditions are augmented with a condition on the current flux or the electric field. That is,

$$\mathbf{J} \cdot \hat{\mathbf{n}} = \sigma \cdot \mathbf{E} \cdot \hat{\mathbf{n}} = \left(-\sigma \cdot \frac{\partial \mathbf{A}}{\partial t} - \sigma \cdot \nabla V \right) \cdot \hat{\mathbf{n}} = f^J(s_k, t) \text{ on } \Gamma_J \quad (7.5.40)$$

or

$$\mathbf{E} \times \hat{\mathbf{n}} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t} \times \hat{\mathbf{n}} = \mathbf{f}^E(s_k, t) \text{ on } \Gamma_E \quad (7.5.41)$$

Along a material interface, such as Γ_{CC} or Γ_{DC} , the usual assumption is that the normal component of the magnetic flux is continuous and the tangential component of the magnetic field is discontinuous by an amount equal to the surface current, \mathbf{J}_{surf} . These conditions are specified by

$$(\mathbf{B}_2 - \mathbf{B}_1) \cdot \hat{\mathbf{n}} = (\nabla \times \mathbf{A}_1 - \nabla \times \mathbf{A}_2) \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_{CC}, \Gamma_{DC} \quad (7.5.42)$$

$$(\mathbf{H}_2 - \mathbf{H}_1) \times \hat{\mathbf{n}} = (\nu_1 \cdot \nabla \times \mathbf{A}_1 - \nu_2 \cdot \nabla \times \mathbf{A}_2) \times \hat{\mathbf{n}} = \mathbf{J}_{surf} \text{ on } \Gamma_{CC}, \Gamma_{DC} \quad (7.5.43)$$

where the subscripts 1 and 2 indicate variables evaluated on either side of the interface. In many cases the surface current is not important and may be neglected.

The divergence and curl relations for the electric field also provide two conditions at a material interface. In this case the normal component of the current density is continuous and the tangential components of the electric field are continuous. That is,

$$(\mathbf{J}_2 - \mathbf{J}_1) \cdot \hat{\mathbf{n}} = (\sigma_2 \cdot \mathbf{E}_2 - \sigma_1 \cdot \mathbf{E}_1) \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_{CC} \quad (7.5.44)$$

$$(\mathbf{E}_2 - \mathbf{E}_1) \times \hat{\mathbf{n}} = \mathbf{0} \text{ on } \Gamma_{CC} \quad (7.5.45)$$

for the boundary between two conductors and

$$\mathbf{J}_2 \cdot \hat{\mathbf{n}} = \sigma_2 \cdot \mathbf{E}_2 \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_{DC} \quad (7.5.46)$$

$$\mathbf{E}_2 \times \hat{\mathbf{n}} = \mathbf{0} \text{ on } \Gamma_{DC} \quad (7.5.47)$$

for the boundary between a very good conductor and a dielectric where the subscript 2 refers to the conducting region.

7.5.5 Gauge Conditions

The quasi-static form of Maxwell's equations is given by Eqs. (7.5.29)–(7.5.30) in terms of the magnetic and electric potentials. The original or primitive variable form of Maxwell's Eqs. (7.5.20)–(7.5.22) can be shown to provide unique solutions for the \mathbf{B} and \mathbf{E} fields when appropriate boundary conditions are specified. However, with the introduction of the potential variables, uniqueness of the solution is not retained, i.e., Eqs. (7.5.29) and (7.5.30) define the curl of \mathbf{A} , but \mathbf{A} itself is only defined up to the gradient of an arbitrary scalar function. Typically this arbitrariness in \mathbf{A} is resolved by defining the divergence of \mathbf{A} and supplying appropriate boundary conditions for \mathbf{A} (rather than boundary conditions for the curl of \mathbf{A}). The incorporation of a $\nabla \cdot \mathbf{A}$ constraint, termed gauging, may be accomplished in any of several ways. The Coulomb gauge is one particular choice that has found extensive use in numerical simulation methods. In this case the magnetic vector potential is made unique by the constraint

$$\nabla \cdot \mathbf{A} = 0 \quad (7.5.48)$$

Other choices for the gauge condition, such as the Lorentz gauge, select a nonhomogeneous form for Eq. (7.5.48). Note that in some cases of reduced dimensionality, the explicit use of Eq. (7.5.48) is not required since vector \mathbf{A} is automatically divergence free.

The implementation of Eq. (7.5.48) in a numerical method may take any of several forms, including modification of the field equations, penalty methods, projection methods, and the construction of divergence free basis functions. All of these techniques incur a computational penalty in terms of either additional work or the modification of the equation system to a less desirable form.

For some applications a unique value of the magnetic potential is not required and the above gauge condition may be neglected. In particular, if the magnetic field is time independent, then all field quantities are related to the curl of \mathbf{A} and a unique value of \mathbf{A} is not required. However, for the time-dependent case, the electric field \mathbf{E} is related to the time derivative of \mathbf{A} in Eq. (7.5.28) and the \mathbf{A} field must be unique if \mathbf{E} is required.

7.5.6 Static Field Problems

Within the general framework established above, a number of simpler static problems may also be defined. Each of these problem classes may have importance in the context of coupling with other mechanics problems or as a stand alone analysis in electromagnetics. As subclasses of the general formulation, they may be solved with many of the same numerical techniques as the general quasi-static problem.

7.5.6.1 Electrostatics

The electrostatic problem is described by Gauss's law (7.5.4) and the definition of the electric flux (displacement current). Combining Eq. (7.5.4) with the simple conductive form of the constitutive relation in (7.5.11) leads to

$$\nabla \cdot \mathbf{D} = \nabla \cdot (\epsilon \mathbf{E}) = \rho \quad (7.5.49)$$

where ρ is the spatial distribution of (free) electric charge. Faraday's law (7.5.1) under steady conditions ($\nabla \times \mathbf{E} = 0$) implies that \mathbf{E} is derivable from a scalar potential, $\mathbf{E} = -\nabla V$, and thus (7.5.49) becomes

$$\nabla \cdot (\epsilon \cdot \nabla V) = -\rho \quad (7.5.50)$$

which is valid for electrically conductive materials. Note that constant potential regions (conductors without a specified charge distribution) may be removed from the analysis domain or approximated as a high permittivity material. For dielectrics, the free charge is zero and the more general form of the constitutive relation in (7.5.11) can be used to produce

$$\nabla \cdot (\epsilon \cdot \nabla V) = \nabla \cdot \mathbf{P}_0 \quad (7.5.51)$$

Nonlinear dielectrics would ignore the source term in (7.5.51) and use $\epsilon(\mathbf{E})$. Boundary conditions for electrostatics generally involve specification of the scalar potential, V , or the definition of the electric flux normal to the boundary (i.e., the normal derivative of the potential). When the spatial variation of the potential has been determined, the electric field \mathbf{E} and the electric flux may be found from the relevant definitions.

7.5.6.2 Steady current flow

For time-independent problems the system in (7.5.29) and (7.5.30) becomes decoupled and the current continuity condition (7.5.30) may be written as

$$\nabla \cdot (-\sigma \cdot \nabla V) = 0 \quad (7.5.52)$$

with

$$\mathbf{J} = -\sigma \cdot \nabla V \quad (7.5.53)$$

Equations (7.5.52) and (7.5.53) describe steady electric currents within a conductor. Boundary conditions on the system would normally include specification of the electric potential (voltage) over part of the boundary and/or the current density normal to the boundary, i.e., $\frac{\partial V}{\partial n}$. Once the electric potential and current distributions have been found then the Joule heating could be recovered from the definition in Eq. (7.5.19).

7.5.6.3 Magnetostatics

Ampere's law (7.5.30) or (7.5.32) for the time-independent case becomes

$$\nabla \times (\nu \cdot \nabla \times \mathbf{A}) = -\sigma \cdot \nabla V = \mathbf{J}_s \quad (7.5.54)$$

This describes the magnetic field due to specified current distributions \mathbf{J}_s . Note that the conduction currents could be specified directly or computed from the steady current flow Eq. (7.5.52) for the electric potential V . When the magnetic potential is known, then the magnetic field \mathbf{B} may be computed from its definition in (7.5.26). In addition, the Lorentz forces can be found from Eq. (7.5.17).

7.5.7 Finite Element Models for EM Fields

The potential equations in (7.5.29) and (7.5.30) represent the three components of the vector magnetic potential and the scalar electric potential for the general quasi-static field problem. Using the standard method of weighted residual techniques these equations can be converted to appropriate weak forms for subsequent use with a finite element approximation. This process will be outlined here for the case of a conducting material; simplifications for free space or source current regions are obvious.

7.5.7.1 Quasi-static potential equations

The weak or weighted integral forms corresponding to (7.5.29) and (7.5.30) are obtained by defining a vector weighting function \mathbf{W} and a scalar weighting function W , multiplying (7.5.29) and (7.5.30) by the appropriate function and integrating each equation over the conducting region. That is,

$$\int_{\Omega_C} \mathbf{W} \cdot \nabla \times (\nu \cdot \nabla \times \mathbf{A}) d\Omega + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} d\Omega + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \nabla V d\Omega = 0 \quad (7.5.55)$$

and

$$\int_{\Omega_C} W \nabla \cdot (-\sigma \cdot \nabla V) d\Omega + \int_{\Omega_C} W \nabla \cdot (-\sigma \cdot \frac{\partial \mathbf{A}}{\partial t}) d\Omega = 0 \quad (7.5.56)$$

The weak forms in (7.5.55) and (7.5.56) may be further transformed by utilizing Gauss's theorem to reduce the highest order derivative terms. Proceeding first with the magnetic potential equation, use the definition $\mathbf{H} = \nu \cdot \nabla \times \mathbf{A}$ and rewrite (7.5.55) as

$$\int_{\Omega_C} \mathbf{W} \cdot \nabla \times \mathbf{H} d\Omega + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} d\Omega + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \nabla V d\Omega = 0 \quad (7.5.57)$$

Using the vector identity

$$\nabla \cdot (\mathbf{W} \times \mathbf{H}) = \mathbf{H} \cdot \nabla \times \mathbf{W} - \mathbf{W} \cdot \nabla \times \mathbf{H} \quad (7.5.58)$$

Eq. (7.5.57) is written as

$$\begin{aligned} & \int_{\Omega_C} \mathbf{H} \cdot \nabla \times \mathbf{W} d\Omega - \int_{\Omega_C} \nabla \cdot (\mathbf{W} \times \mathbf{H}) d\Omega + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} d\Omega \\ & + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \nabla V d\Omega = 0 \end{aligned} \quad (7.5.59)$$

Introducing the divergence theorem for the second integral and the vector identity $(\mathbf{W} \times \mathbf{H}) \cdot \mathbf{n} = \mathbf{W} \cdot (\mathbf{H} \times \mathbf{n})$ then (7.5.59) becomes

$$\begin{aligned} & \int_{\Omega_C} \mathbf{H} \cdot \nabla \times \mathbf{W} d\Omega - \int_{\Gamma_C} \mathbf{W} \cdot (\mathbf{H} \times \mathbf{n}) d\Gamma + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} d\Omega \\ & + \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \nabla V d\Omega = 0 \end{aligned} \quad (7.5.60)$$

where \mathbf{n} is the outward normal to the boundary Γ_C . Finally, reintroducing the definition of \mathbf{H} , rearranging and introducing the natural boundary condition for the magnetic potential, lead to the required form of the integral statement

$$\begin{aligned} & \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} d\Omega + \int_{\Omega_C} \nabla \times \mathbf{W} \cdot (\nu \cdot \nabla \times \mathbf{A}) d\Omega \\ & = - \int_{\Omega_C} \mathbf{W} \cdot \sigma \cdot \nabla V d\Omega + \int_{\Gamma_C} \mathbf{W} \cdot \mathbf{f}^H d\Gamma \end{aligned} \quad (7.5.61)$$

The electric potential equation (7.5.56) is transformed in a similar manner. Using the divergence theorem on the first two integrals in (7.5.56) leads to

$$\begin{aligned} & - \int_{\Omega_C} \nabla W \cdot (-\sigma \cdot \nabla V) d\Omega - \int_{\Omega_C} \nabla W \cdot \left(-\sigma \cdot \frac{\partial \mathbf{A}}{\partial t} \right) d\Omega \\ & = - \int_{\Gamma_C} W \left(-\sigma \cdot \frac{\partial \mathbf{A}}{\partial t} - \sigma \cdot \nabla V \right) \cdot \mathbf{n} d\Gamma \end{aligned} \quad (7.5.62)$$

This may be rearranged into a standard form, with the natural boundary conditions for the potential included to produce

$$\int_{\Omega_C} \nabla W \cdot \sigma \cdot \nabla V d\Omega + \int_{\Omega_C} \nabla W \cdot \sigma \cdot \frac{\partial \mathbf{A}}{\partial t} d\Omega = \int_{\Gamma_C} W f^J d\Gamma \quad (7.5.63)$$

Equations (7.5.61) and (7.5.63) are weak forms of the potential equations for electromagnetics and are suitable for use with a finite element approximation. Though not shown explicitly, the above formulation can be easily extended to include the case of motion of the conductor and/or more complex material behavior, e.g., remnant magnetization. To proceed with the finite element model, the region Ω_C is discretized into an assemblage of finite elements and the weighted integral statements are applied to each element. Within each element the vector magnetic potential and scalar electric potential are approximated by expansions of the form

$$\mathbf{A}(\mathbf{x}, t) = \Phi^T(\mathbf{x}) \mathbf{A}_x(t) \mathbf{e}_x + \Phi^T(\mathbf{x}) \mathbf{A}_y(t) \mathbf{e}_y + \Phi^T(\mathbf{x}) \mathbf{A}_z(t) \mathbf{e}_z \quad (7.5.64a)$$

$$V(\mathbf{x}, t) = \Psi^T(\mathbf{x}) \mathbf{V}(t) \quad (7.5.64b)$$

which are written here for the three-dimensional Cartesian case with the \mathbf{e}_i being unit vectors for the coordinate system; similar expressions can be constructed for the axisymmetric and two-dimensional cases. In Eqs. (7.5.64a,b) Φ and Ψ represent vectors of interpolation functions, \mathbf{A}_i and \mathbf{V} are vectors of nodal point unknowns and superscript T indicates a vector transpose. Note that the assumed approximations for the dependent variables are, as usual, semi-discrete with the spatial dependence being discretized through interpolation and the temporal dependence remaining continuous. For the Galerkin method the weight functions \mathbf{W} and W are selected to be the same functions as used to represent the variables. That is,

$$\mathbf{W}(\mathbf{x}) = \Phi(\mathbf{x}) \mathbf{e}_x + \Phi(\mathbf{x}) \mathbf{e}_y + \Phi(\mathbf{x}) \mathbf{e}_z \quad (7.5.65a)$$

$$W(\mathbf{x}) = \Psi(\mathbf{x}) \quad (7.5.65b)$$

Substituting the definitions in (7.5.64) and (7.5.65) into the weighted residual equations in (7.5.61) and (7.5.63) produces the following set of discrete equations for each element

$$\mathbf{M} \dot{\mathbf{A}} + \mathbf{K} \mathbf{A} + \mathbf{N} \mathbf{V} = \mathbf{F}_A \quad (7.5.66)$$

$$\mathbf{N}^T \dot{\mathbf{A}} + \mathbf{L} \mathbf{V} = \mathbf{F}_V \quad (7.5.67)$$

where the superposed dot indicates a time derivative. Equation (7.5.66) represents the three component equations for the magnetic potential.

The matrix system shown above is unsymmetric and is of an undesirable form from the standpoint of time integration. To restore symmetry, the following definition proposed by Chari, et al. [15] may be employed

$$\mathbf{V} \equiv \frac{\partial \mathbf{v}}{\partial t} = \dot{\mathbf{v}} \quad (7.5.68)$$

Using this definition Eqs. (7.5.66) and (7.5.67) can be rewritten as

$$\mathbf{M} \dot{\mathbf{A}} + \mathbf{K} \mathbf{A} + \mathbf{N} \dot{\mathbf{v}} = \mathbf{F}_A \quad (7.5.69)$$

$$\mathbf{N}^T \dot{\mathbf{A}} + \mathbf{L} \dot{\mathbf{v}} = \mathbf{F}_V \quad (7.5.70)$$

and in a completely assembled form

$$\begin{bmatrix} \mathbf{M} & \mathbf{N} \\ \mathbf{N}^T & \mathbf{L} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{A}} \\ \dot{\mathbf{v}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{A} \\ \mathbf{v} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_A \\ \mathbf{F}_V \end{Bmatrix} \quad (7.5.71)$$

The component matrices defined in (7.5.71) are defined by the following integrals that arise from the weighted residual statements. The matrices are written here in terms of vector notation; the explicit forms involving derivatives on the element and the Jacobian for the transformation to the master element can be obtained by use of obvious substitutions from previous chapters

$$\begin{aligned}\mathbf{M} &= \int_{\Omega_e} \Phi \cdot \sigma \cdot \Phi^T d\Omega, & \mathbf{K} &= \int_{\Omega_e} \nabla \times \Phi \cdot \nu \cdot \nabla \times \Phi^T d\Omega \\ \mathbf{N} &= \int_{\Omega_e} \Phi \cdot \sigma \cdot \nabla \Psi^T d\Omega, & \mathbf{N}^T &= \int_{\Omega_e} \nabla \Psi \cdot \sigma \cdot \Phi^T d\Omega \\ \mathbf{L} &= \int_{\Omega_e} \nabla \Psi \cdot \sigma \cdot \nabla \Psi^T d\Omega, & \mathbf{F}_A &= \int_{\Gamma_e} \Phi \cdot \mathbf{f}^H d\Gamma \\ \mathbf{F}_M &= \int_{\Omega_e} \nabla \times \Phi \cdot \mathbf{M}_0 d\Omega, & \mathbf{F}_V &= \int_{\Gamma_e} \Psi f^J d\Gamma\end{aligned}\quad (7.5.72)$$

7.5.7.2 Gauge condition

When the magnetic potential must be made unique, the Coulomb gauge defined in (7.5.48) is added to the field equations as a constraint. From a computational standpoint there are several methods available for developing a discrete form of (7.5.48). One of the simplest implementations is the penalty method. The penalty method is most easily invoked by casting the original quasi-static field problem in terms of a functional to be minimized. To simplify the derivation the specified current form of the magnetostatics problem will be considered; the more general quasi-static or eddy current problem is developed in an analogous manner. Consider the static form of (7.5.29) or more precisely, Eq. (7.5.54)

$$\nabla \times (\nu \cdot \nabla \times \mathbf{A}) = \mathbf{J}_s \quad (7.5.73)$$

The functional associated with (7.5.73) is

$$I(\mathbf{A}) = \frac{1}{2} \int_{\Omega} \nabla \times \mathbf{A} \cdot \nu \cdot \nabla \times \mathbf{A} d\Omega - \int_{\Omega} \mathbf{J}_s \cdot \mathbf{A} d\Omega \quad (7.5.74)$$

A finite element approximation for \mathbf{A} may be used in (7.5.74), which when minimized with respect to the dependent variables produces a discrete system that is the same as the appropriate magnetostatics subset of the weak-form Galerkin equations in (7.5.71). To invoke the Coulomb gauge on this system, the functional in (7.5.74) is augmented with a least squares penalty term

$$I_C(\mathbf{A}) = I(\mathbf{A}) + \frac{\lambda}{2} \int_{\Omega} (\nabla \cdot \mathbf{A})^2 d\Omega \quad (7.5.75)$$

where λ is the penalty parameter. When a finite element approximation, such as (7.5.64) is used in the augmented functional in (7.5.75), and variations are taken with respect to the components of \mathbf{A} , the result is a discrete system of the following form

$$(\mathbf{K} + \lambda \mathbf{K}_{\text{div}}) \mathbf{A} = \mathbf{F}_A + \mathbf{F}_J \quad (7.5.75)$$

where most of the matrices and vectors are defined in (7.5.72) and the penalty term is defined by

$$\mathbf{K}_{\text{div}} = \int_{\Omega_e} \nabla \cdot \boldsymbol{\Phi} \nabla \cdot \boldsymbol{\Phi}^T d\Omega \quad (7.5.76)$$

This process is very similar to the penalty method used to enforce incompressibility in the viscous flow problem. The penalty enforcement of the Coulomb gauge requires the construction of an additional matrix that is added to the appropriate terms in the magnetic vector potential equation. The penalty parameter λ is set to a large number with the free space reluctivity being a good choice ($\lambda = 1/\mu_0 = \nu_0$). In order to avoid an over-constrained system, the penalty matrix must be singular. The standard method for achieving this situation is to reduce the order of the quadrature rule used to evaluate the integral in (7.5.76).

7.5.7.3 Static field equations

The finite element equations for the simplified static field problems are developed in the same manner as outlined in the previous sections. For some situations, the equations are merely subsets of the more complex quasi-static problem.

The basic equation for electrostatics is either (7.5.50) or (7.5.51) depending on the type of material. If combined, the relevant equation is

$$\nabla \cdot (\epsilon \cdot \nabla V) = -\rho \nabla \cdot \mathbf{P}_0 \quad (7.5.77)$$

The weak form of this equation is given by

$$\int_{\Omega} \nabla W \cdot \epsilon \cdot \nabla V d\Omega = - \int_{\Omega} W \rho d\Omega + \int_{\Omega} \nabla W \cdot \mathbf{P}_0 d\Omega + \int_{\Gamma} W f^D d\Gamma \quad (7.5.78)$$

where the natural boundary condition is a specification of the electric flux (gradient of the potential) normal to the boundary. Let the electric potential be approximated by the steady form of the finite element representation given in (7.5.64b)

$$V(\mathbf{x}) = \boldsymbol{\Psi}^T(\mathbf{x}) \mathbf{V} \quad (7.5.79)$$

Using the Galerkin definition and the scalar weight function in (7.5.65b), Eq. (7.5.78) leads to the matrix equation

$$\mathbf{L}_\epsilon \mathbf{V} = \mathbf{F}_\rho + \mathbf{F}_P + \mathbf{F}_D \quad (7.5.80)$$

where

$$\begin{aligned} \mathbf{L}_\epsilon &= \int_{\Omega_e} \nabla \boldsymbol{\Psi} \cdot \epsilon \cdot \nabla \boldsymbol{\Psi}^T d\Omega, & \mathbf{F}_\rho &= \int_{\Omega_e} \boldsymbol{\Psi} \rho d\Omega \\ \mathbf{F}_P &= \int_{\Omega_e} \nabla \boldsymbol{\Psi} \cdot \mathbf{P}_0 d\Omega, & \mathbf{F}_D &= \int_{\Gamma_e} \boldsymbol{\Psi} f^D d\Gamma \end{aligned} \quad (7.5.81)$$

The steady current flow problem is a subset of the general quasi-static formulation and can be defined immediately as

$$\mathbf{L}\mathbf{V} = \mathbf{F}_V \quad (7.5.82)$$

where the matrix and vector are defined in (7.5.72).

The magnetostatics problem is also a subset of the general quasi-static problem and corresponds to the equations for a free space region with specified currents. The relevant finite element equations are derived from (7.5.69)

$$\mathbf{KA} = \mathbf{F}_A + \mathbf{F}_J \quad (7.5.83)$$

where the vector \mathbf{F}_J is a weighted integral over the element volume of the specified current density.

7.5.8 Solution Methods – EM Fields

The finite element equations for the general quasi-static electromagnetics problem, as given in Eq. (7.5.71), represent a set of ordinary differential equations not unlike the equation sets that were encountered in the heat conduction and viscous flow formulations. Because of this similarity, virtually all of the solution methods used in the transport problems can be used in the electromagnetics problem without modification.

For static problems, the equations in (7.5.71) reduce to nonlinear algebraic equations that can be linearized and solved via iteration using the Picard or Newton methods. The general time-dependent case is well suited to implicit time integration methods such as backward Euler or the trapezoid rule. Both constant time step and adaptive time step, predictor/corrector methods, have been used successfully with these types of integrators.

The only problem type that differs significantly from the transport equations is the time harmonic field problem. If the nodal point variables are defined through a phasor representation with frequency ω [as was done for the continuum case in Eq. (7.5.34)], the equations in (7.5.69) and (7.5.70) become

$$i\omega\mathbf{MA}_0 + \mathbf{KA}_0 + i\omega\mathbf{Nv}_0 = \mathbf{F}_{A_0} \quad (7.5.84)$$

$$i\omega\mathbf{N}^T\mathbf{A}_0 + i\omega\mathbf{Lv}_0 = \mathbf{F}_{v_0} \quad (7.5.85)$$

or in the combined matrix form

$$\begin{bmatrix} i\omega\mathbf{M} & i\omega\mathbf{N} \\ i\omega\mathbf{N}^T & i\omega\mathbf{L} \end{bmatrix} \begin{Bmatrix} \mathbf{A}_0 \\ \mathbf{v}_0 \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{A}_0 \\ \mathbf{v}_0 \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_A \\ \mathbf{F}_V \end{Bmatrix} \quad (7.5.86)$$

where the unknowns are now complex and defined by $V_0 = (V^R + iV^I)$ and $\mathbf{A}_0 = (\mathbf{A}^R + i\mathbf{A}^I)$. Recall also that $V_0 = v_0 = i\omega v_0$.

Through the use of the phasor representation, the time harmonic problem has been reduced to a steady, linear matrix problem; the matrix coefficients and nodal point variables are complex.

The linear algebra problem associated with each of the steady, time-dependent or harmonic solution methods may be solved using either direct matrix solvers or iterative solvers of the conjugate gradient or Krylov type. Note that the matrix systems are usually unsymmetric. The gauge condition alters the matrix structure and may cause convergence difficulties for iterative methods. The complex coefficients in the time harmonic case can be used directly in solvers set up for complex arithmetic. An alternative is to separate the complex matrix into real and imaginary parts and solve twice the number of equations using only real coefficients.

7.6 Coupled Problems in Mechanics

7.6.1 Introduction

The equations from the three previous sections may be combined in a variety of ways to describe a large class of problems in mechanics. The fact that there is such a large spectrum of problems implies that meaningful discussions of coupling will be difficult to accomplish in a general way. Therefore, we will revert to a simple and expeditious method of description that considers the pairwise combinations of five mechanics areas: heat conduction (HC), nonisothermal viscous flow (VF), quasi-static solid mechanics (SM), electric fields (EF), and magnetic fields (MF). The symmetric coupling matrix shown in Figure 7.6.1 is used to organize and focus the discussion. Note that at the intersection of each pair of mechanics areas, one or more of the usual types of coupled problems are listed. No attempt is made to be thorough in this listing and only the most common interactions are indicated. In the matrix and in the following sections, some of the possible data dependencies for each interaction will be described, the typical strength of the interaction will be cited, and the types of algorithms that may be used will be suggested. Since this is a book on fluid mechanics and heat transfer, the focus will be on the first two rows of the matrix.

	1 Heat Conduction/ Radiation	2 Viscous/ Porous Flow	3 Quasi- Static Solids	4 Electric Fields	5 Magnetic Fields
1 Heat Conduction/ Radiation		1-2 Conjugate Heat Transfer B (S)	1-3 Thermal Stress V.P.G. (W-S)	1-4 Resistive Heating V.P. (W-S)	1-5 Inductive Heating V.P. (W-S)
2 Viscous/ Porous Flow	2-1		2-3 Fluid/Solid Interaction B.G. (S)	2-4 EHD Electro- Rheology V.P. (S-W)	2-5 MHD Inductive Stirring V.P. (W-S)
3 Quasi- Static Solids	3-1	3-2		3-4 Electro- Striction Piezoelectric B.P.G.(S-W)	3-5 Magnetic Stress V.G. (W-S)
4 Electric Fields	4-1	4-2	4-3		4-5 Eddy Currents V.P. (S)
5 Magnetic Fields	5-1	5-2	5-3	5-4	

Types of coupling:

- B – Boundary conditions/surface flux
- V – Volume terms; G – Geometry
- P – Property/constitutive dependence

Strength of coupling:

- S – Strong
- W – Weak

Figure 7.6.1: Matrix of possible mechanics interactions.

7.6.2 Heat Conduction – Viscous Fluid Interactions 1 & 2

For completeness of the exposition, some of the coupled problem situations from previous chapters are reviewed here in summary form. Consider the interaction of heat conduction and radiation, and conduction and chemical reaction from Chapter 3. These are strongly coupled problems with conduction/radiation being a boundary condition interaction and possibly a thermophysical property interaction if the emissivity is a function of temperature. The conduction/chemical reaction problem is primarily a volume source interaction on the conduction side of the interaction and a (kinetics) property interaction for the reaction equations. Because of the strong coupling, the preferred method of solution would be a simultaneous solution of each pair of equations.

The completely coupled solution process is often not practical in these two cases because of the characteristics of the radiation and chemical reaction equations. In the conduction/radiation case, the matrix structure of the radiation equation is full due to the view factor (surface-to-surface) coupling while the conduction matrix is sparse. The size of the radiation matrix and its precipitous growth with increase in mesh resolution often make the combined matrix problem not tractable. The combined conduction/reaction problem suffers mainly from a problem of disparate time scales and stiffness in the reaction equations. The excessively small time steps required in the chemistry solution are an unnecessary computational penalty for the thermal diffusion problem. Cyclic or operator splitting methods may not be optimal from a coupling perspective, but are practical from an overall algorithm point of view as detailed in Chapter 3.

The conjugate problem that couples solid body conduction with a nonisothermal flow is another strongly coupled problem with an interaction at the boundary. This class of problems is typically solved as a fully coupled matrix problem. The addition of conducting regions (scalar equation) to a primarily viscous flow problem (vector equation) presents no real increase in computational burden. Time constants for the two equation sets are usually of the same order of magnitude so that integration methods may follow the smaller time scale without too much of a penalty and coupling is straightforward. Note that it is possible to treat the conjugate problem in a decoupled or cyclic manner. In this case, the temperature and flux at the common boundary are used as the coupling variables. One variable is used as a boundary condition while the second variable is used to measure the convergence or agreement with the other equation set. Though feasible, this is not a recommended solution method for this type of problem.

7.6.3 Heat Conduction – Quasi-Static Solid Interactions 1 & 3

Thermal stress problems are one of the most heavily studied and commonly encountered types of coupled mechanics problems. In most cases the coupling is relatively weak, with the thermal field providing a thermal strain through the mechanical constitutive relation; in some cases a temperature dependence of the parameters in the constitutive equation adds to the interaction. When the interaction is limited to these effects, the problem can be easily solved via a one-way or decoupled procedure. With Eq. (7.3.7) representing the finite element heat conduction problem

$$\mathbf{M}\dot{\mathbf{T}} + \hat{\mathbf{K}}\mathbf{T} = \hat{\mathbf{F}} \quad (7.6.1)$$

and Eq. (7.4.22) defining the mechanical equilibrium

$$\mathbf{K}(\mathbf{u}, \mathbf{T})\mathbf{u} = \mathbf{F} \quad (7.6.2)$$

the solution procedure is fairly evident. The conduction problem, being independent of any mechanical variables, is solved first over the time interval of interest. Temperature data are passed to the solid mechanics equations and these are solved for the displacement, strain, and stress fields over the temperature history. Since (7.6.2) is a quasi-static model, increments of the structural load is substituted for a time integration procedure. The temperature field at any given load step can be found by interpolation of the finite element solution. This type of problem could also be solved through a fully coupled procedure though this is not a computationally attractive alternative. Since the thermal diffusion process has the smaller time constant (compared to the infinite time constant for the solid mechanics problem), the thermal problem must be solved a fairly large number of times over the time interval. A fully coupled algorithm would thus force the solid mechanics problem to be solved many more times than would normally be required to achieve an accurate stress solution. The additional solutions of (7.6.2) make this approach ill-advised.

The coupling between Eqs. (7.6.1) and (7.6.2) increases in strength when mechanical dissipation is important and/or very large deformations occur in the structure such that the geometric definition of the heat conduction problem is altered. Mechanical dissipation appears as a volumetric source term in the conduction equation and is directly dependent on the stress and strain rate fields. That is, the force vector in (7.6.1) is $\hat{\mathbf{F}}(\sigma, \dot{\epsilon})$. Depending on the dissipation rate the problem may be either weakly or strongly coupled. If the heat generation rate is slow and temperature effects in the solid negligible, the solid mechanics problem could be solved over the loading history of interest to produce a dissipation history. The dissipation history could then be used to solve the conduction problem for the temperature response of the region. As the dissipation rate increases and temperature dependence increases, the coupling is more pronounced and Eqs. (7.6.1) and (7.6.2) must be solved more implicitly. A fully coupled method can be used, though a cyclic algorithm is more cost effective. Again, the conduction problem sets the time scale (time step) for the integration; the solid mechanics problem is solved after each thermal time step or whenever the fields change sufficiently to require an update in the dissipation. This decision on the frequency of solution for a subordinate process is one of the more difficult judgments to make in the use of cyclic or staggered solution strategies.

The coupling between (7.6.1) and (7.6.2) is very strong when large changes in geometry occur in the solid and the boundary conditions in the thermal problem are affected. If radiation is part of the thermal problem geometric changes imply changes in the radiation view factors and a redistribution of surface flux. Also, if deformation leads to new contacts between surfaces or the separation of previously contacting surfaces, the thermal conditions may be significantly altered. This type of coupling may be effectively treated with a cyclic algorithm, though the structural solution and geometrical updates would usually have to be computed at every thermal time step or whenever deformation was significant. The recomputing of radiation view factors for a dynamic geometry is a significant computational burden and makes this type of problem quite challenging.

7.6.4 Heat Conduction - Electric Field Interactions 1 & 4

Resistive or Joule heating occurs when energy is dissipated by an electric current flowing through a conductor. This type of problem would be represented by the conduction equation in (7.3.7)

$$\mathbf{M}\dot{\mathbf{T}} + \hat{\mathbf{K}}\mathbf{T} = \hat{\mathbf{F}}(\mathbf{J}) \quad (7.6.3)$$

and the steady current flow equation (7.5.52) and its finite element model (7.5.82)

$$\mathbf{L}(\mathbf{T})\mathbf{V} = \mathbf{F}_V \quad (7.6.4)$$

The variable dependencies are shown in (7.6.3) and (7.6.4). The volume heating in (7.6.3) is given by $Q_J = \sigma^{-1}\mathbf{J}^2$ and \mathbf{J} is related to the gradient of the electric potential, \mathbf{V} , which is the unknown in (7.6.4). The electrical conductivity is usually a function of temperature, thus making the diffusion operator in (7.6.4) dependent on \mathbf{T} .

If these dependencies are both present, Eqs. (7.6.3) and (7.6.4) can be most easily solved as a fully coupled system. An implicit time integration procedure applied to (7.6.3) results in a generally nonlinear matrix problem as described in Chapter 3. This matrix can be combined with (7.6.4) to represent the two scalar equations. At each time step (or iteration, if the problem is steady), the combined nonlinear problem can be solved using either the Picard iteration if the nonlinearities are fairly mild or Newton's method for stronger nonlinear behavior. This problem can also be solved via a decoupled or cyclic procedure. Because (7.6.3) is time dependent, the conduction problem is the master process and sets the time scale for the problem. At a given time step, the known temperature field can be passed to the current flow equation, the temperature-dependent conductivity evaluated, and the current density and Joule heating computed. The Joule heating is transferred back to the conduction equation to allow the next update on the temperature field. This process may be repeated until convergence occurs at the current time; in many cases the changes that occur over a time step are sufficiently small that only one iteration through the data exchange is necessary to maintain acceptable accuracy.

7.6.5 Heat Conduction – Electromagnetic Field Interactions 1 & 4 & 5

Isolated magnetic fields do not generate any significant interactions with a thermal field, though the full coupling of both electric and magnetic fields with temperature effects describe a number of engineering processes of importance. Induction heating for melting and surface treatment (hardening) are commonly used processes that depend on this type of coupling. Eddy current analysis for the performance of electromagnetic devices, such as motors and generators, is another source of coupled problems. In the present description the focus will be on induction heating where extremes in the interaction are observed; the less demanding thermal performance problems will be subsets of this type of coupling.

To describe the nonisothermal eddy current problem, the conduction equation (7.3.7) is required

$$\mathbf{M}\dot{\mathbf{T}} + \hat{\mathbf{K}}\mathbf{T} = \hat{\mathbf{F}}(\mathbf{J}) \quad (7.6.5)$$

as is the quasi-static, electromagnetic system defined by the field problem in Eqs. (7.5.29) and (7.5.30) and the corresponding finite element model in equations (7.5.69) and (7.5.70)

$$\mathbf{M}(\mathbf{T})\dot{\mathbf{A}} + \mathbf{K}(\mathbf{T}, \mathbf{B})\mathbf{A} + \mathbf{N}(\mathbf{T})\dot{\mathbf{v}} = \mathbf{F}_A \quad (7.6.6)$$

$$\mathbf{N}^T(\mathbf{T})\dot{\mathbf{A}} + \mathbf{L}(\mathbf{T})\dot{\mathbf{v}} = \mathbf{F}_V \quad (7.6.7)$$

The variable dependencies are shown in Eqs. (7.6.5)–(7.6.7), which are written for the general case where the electrical problem is potential or voltage driven. When the problem is current driven, Eq. (7.6.7) is no longer needed and the $\dot{\mathbf{v}}$ term in (7.6.6) is known and associated with the known current density. These subcases were covered previously in Section 7.5.3. The volume heating in (7.6.5) is again given by the Joule heating relation

$$Q_J = \sigma^{-1} \mathbf{J}^2$$

and \mathbf{J} is related to the combination of the gradient of the electric potential, \mathbf{V} , if it is present, and the time derivative of the magnetic vector potential [see Eq. (7.5.28) and following]. The time derivative term is the induced or eddy current due to the fluctuating magnetic field. The dependencies in Eqs. (7.6.6) and (7.6.7) correspond to the possible variation of the electrical conductivity with temperature and the magnetic permeability with both temperature and the magnetic flux density.

In ferromagnetic materials the variation of the magnetic permeability is a major complicating factor in the analysis since it is a hysteretic function of \mathbf{B} and \mathbf{H} which is scaled by the temperature. The magnetic permeability may vary by three orders of magnitude over typical magnetic field strengths (at moderate temperatures) while above the Curie temperature the magnetic permeability reverts to a constant equal to free space permeability. A final complication in this type of coupled problem stems from the limited penetration into the material of the induced current fields and Joule heating when the applied fields are moderate to high frequencies. This skin depth effect implies that the simulation regions that are critical to the heat transfer part of the problem do not necessarily overlap all of the electromagnetic regions of importance.

In many applications the electromagnetic fields are time harmonic and the phasor representation of (7.6.6) and (7.6.7) [Eqs. (7.5.84) and (7.5.85)] are the appropriate finite element equations with similar functional dependencies. Even though the electromagnetic problem is nonlinear, a phasor representation can be used with single frequency driving fields if material properties are treated appropriately. This method is illustrated in [19]. Though the coupling is very strong, the appropriate solution strategy for Eqs. (7.6.5)–(7.6.7), or their time harmonic equivalents, is a cyclic procedure due to the vastly different time scales for most applications. The diffusion time constant for the heat conduction problem is significantly longer than equilibration of the induced electromagnetic fields; the electromagnetic fields are virtually constant over the diffusion time interval even if they are periodic. The heat conduction problem, therefore, is the master process and sets the time integration procedure. At a given time step, the current temperature field is passed to the electromagnetic problem. If the time-dependent forms of (7.6.6) and (7.6.7) are used, the fields are integrated forward to the end of the time interval using

an appropriately small time step. The Joule heating history is computed from the time-dependent fields and integrated to provide an energy rate over the time step for each resistively heated finite element. When the time harmonic forms of (7.6.6) and (7.6.7) are used, the procedure is similar except that a time independent, often nonlinear field problem must be solved and the resulting complex variables must be manipulated to generate the Joule heating. After the electromagnetic fields are updated, the new Joule heating data are used as a source term for the next conduction step. Note that in some applications, the geometry of the problem may change with time. This is not usually a difficulty because both the conduction and electromagnetics problems can be solved in a Lagrangian reference frame making geometric updates straightforward in the coupled situation. When radiation is a consideration, changing geometries will increase the computational work for the heat transfer solution step.

Even though the electromagnetic problem is the subordinate process and is simply providing the “loads” for the thermal problem it is, in this case, the more difficult problem. The inclusion of the gauge condition (7.5.48) makes the solution of the magnetic vector potential equation very difficult in much the same way as the incompressibility constraint hinders the solution of the viscous flow problem. In addition, the large changes in magnetic permeability with both field strength and temperature, plus the hysteretic nature of the property variation, generate strong material nonlinearities that inhibit convergence. Finally, though the time harmonic case is simplified in the time domain, the use of complex variables increases significantly the computational burden due to the matrix and variable vector sizes.

7.6.6 Viscous Flow – Quasi-Static Solid Interactions 2 & 3

Fluid structure interaction problems are quite common and this type of coupling has been well studied. In many simulations the fluid model can be simplified to a potential flow where only pressure loads on the structure are of interest and fluid separation is not an important feature of the flow. The coupling of the fluid potential equation with the solid mechanics problem will not be considered here as the main interest remains the viscous flow problem. The major difficulties in coupled viscous fluid and solid mechanics problems stem from the inherently different coordinate descriptions used in the two fields and the possible variations in defining a dominant physical process.

Because the preferred description for solid mechanics is Lagrangian and the usual description for fluids is an Eulerian coordinate system, some compromise must be found for a useful coupled problem description. The standard solution is to work in the usual Lagrangian reference frame for the solid and specify the fluid motion in an Arbitrary Lagrangian Eulerian (ALE) framework, as was outlined in Section 4.10.3. This approach has the advantage of maintaining an accurate definition of the fluid/solid interface which is the primary coupling mechanism in this type of problem. The possible large deformation of the solid is well captured by this approach and the use of appropriate strain measures in the solid mechanics formulation. Also, the ALE method with a mesh moving scheme coupled to the interface motion (which is determined from the solid mechanics problem), provides a robust and accurate algorithm for the fluid motion. Many of the details associated with ALE methods were described in Section 4.10.3 and will not be repeated here.

The second area of difficulty is to determine the primary physical process so that a solution algorithm for the coupled system can be constructed. Unfortunately, the fluid motion can drive the solid deformation or the solid motion and deformation can drive the fluid motion. Since neither process is always dominant a fully coupled algorithm is suggested.

To describe the fluid structure interaction problem the viscous flow model from equations (7.3.9) and (7.3.10) is required,

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (7.6.8)$$

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{C}(\mathbf{v}^m)\mathbf{v} + \mathbf{K}\mathbf{v} - \mathbf{Q}\mathbf{P} = \mathbf{F}(\mathbf{v}^m) \quad (7.6.9)$$

which is written here in its isothermal form. Coupling with a nonisothermal flow is certainly possible, in which case the energy equation from (7.3.11) would be added to the above and a heat conduction equation would be required for the solid region. This complication adds little to the overall algorithm and therefore will not be explicitly considered. The quasi-static solid mechanics problem is again given by Eq. (7.4.22)

$$\mathbf{K}(\mathbf{v}, \mathbf{v}^m, \mathbf{T})\mathbf{v} = \mathbf{F}(\mathbf{P}, \mathbf{u}) \quad (7.6.10)$$

For ease of notation it is assumed that the solid mechanics problem is written in terms of velocities; the use of the same matrix and vector symbols in (7.6.9) and (7.6.10) does not imply that these are the same matrices. The variable dependencies are indicated in (7.6.8)–(7.6.10) and are seen to be a strong function of the mesh movement \mathbf{v}^m (interface geometry) and the fluid loads on the structure.

As noted above, the most general solution procedure for this type of problem is a fully coupled method in which Eqs. (7.6.8)–(7.6.10) are solved in a single matrix problem. A time integration scheme would be applied to the fluid equations while the static solid equations would have the appropriate loads applied for the current time. The Picard or Newton's method could be used to converge the solution at the current time. The fluid and solid mechanics regions do not overlap; the entire domain is meshed by a single finite element discretization. If a pseudo-structural mesh movement scheme is employed as part of the ALE method, then solid mechanics equations are solved over the entire domain. The mesh movement equations may be fully coupled with the fluid and solid equations or may be decoupled and used to update the fluid mesh region in a subcycle. For efficiency, this fully coupled method assumes that the time scales for the fluid and solid regions are not too different, which they probably cannot be, due to the strong coupling at the boundary. This algorithm does pay a penalty in the matrix solution because the fluid equations are nonsymmetric while the solid equations are symmetric and could be solved with less computational effort.

7.6.7 Viscous Flow – Electric Field Interactions 2 & 4

There are a number of problems that involve a viscous fluid and an electric field, most of which are unfamiliar to the applied mechanics community. Electrohydrodynamics involves the motion of a fluid due to an electric pressure which is proportional to the square of the applied electric field. The fluid may be either conducting or a dielectric. The process of flow electrification involves a dielectric fluid and the convection of charge from the double layers that form near boundaries. This type of problem is

common in electrochemical devices such as batteries. A more familiar problem, and one that will be considered here, is the flow of an electrorheological fluid. In this case an applied electric field causes a large increase in the viscosity of the fluid, which can act as a control mechanism for the flow. The electrorheological application is not a particularly difficult type of coupled problem since the coupling is essentially in one direction with the electric field influencing the flow.

The equations for this type of problem include the isothermal, viscous flow equations from (7.3.9) and (7.3.10) or

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (7.6.11)$$

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{C}\mathbf{v} + \mathbf{K}(\mathbf{E})\mathbf{v} - \mathbf{Q}\mathbf{P} = \mathbf{F} \quad (7.6.12)$$

and the potential equation for the electric field from Eq. (7.5.82)

$$\mathbf{L}\mathbf{V} = \mathbf{F}_V \quad (7.6.13)$$

The viscous term in Eq. (7.6.12) is a function of the electric field, which is computed from the gradient of the electric potential, the unknown in (7.6.13). The master process is the viscous flow problem which is integrated in time by standard methods. When a voltage is applied in some region of the problem, the potential equation must be solved for the instantaneous electric field. These field data are then transferred to the flow equation to allow the viscosity function to be evaluated for the next time step. The process is continued as long as a voltage is applied to the problem. The subordinate process in this case involves a series of steady solutions with the boundary condition (voltage) having a time variation.

7.6.8 Viscous Flow – Electromagnetic Field Interactions 2 & 4 & 5

The topical area of magnetohydrodynamics (MHD) brings together the descriptions of nonisothermal, fluid mechanics and the electromagnetics of conducting materials. The MHD field includes a variety of problem types many of which are quite complex especially with regard to coupling and property variations. Rather than attempt a superficial coverage of general MHD problems, we will narrow the focus here to a class of problems that is important in material processing and manufacturing applications. Processes such as high current melting and inductive stirring fall into this problem category. The major assumption involved in this type of problem is that the magnetic Reynolds number is small. The magnetic Reynolds number is usually defined as $Re_m = UL\sigma\mu_m$, where σ is the electrical conductivity, μ_m is the magnetic permeability, and U and L are a representative flow velocity and length scale, respectively. The magnetic Reynolds number represents a ratio of magnetic convection to magnetic diffusion. For many common processes involving liquid metals, the magnetic Reynolds number is small, which implies that the magnetic field lines are unaffected by the flow field. The opposite extreme of a high magnetic Reynolds number ensures that the magnetic field lines are strongly convected by the flow field. This limit is typical of fusion applications and problems in astrophysics.

With the small magnetic Reynolds number assumption, the equations required for the MHD problem include the nonisothermal, viscous flow equations

$$-\mathbf{Q}^T \mathbf{v} = \mathbf{0} \quad (7.6.13)$$

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{v} - \mathbf{Q}\mathbf{P} + \mathbf{B}\mathbf{T} = \mathbf{F}(\mathbf{J}, \mathbf{B}) \quad (7.6.14)$$

$$\mathbf{N}\dot{\mathbf{T}} + \mathbf{D}\mathbf{T} + \mathbf{L}\mathbf{T} = \mathbf{G}(\mathbf{J}) \quad (7.6.15)$$

and the potential forms of the electromagnetic equations

$$\mathbf{M}(\mathbf{T})\dot{\mathbf{A}} + \mathbf{K}(\mathbf{T})\mathbf{A} + \mathbf{N}(\mathbf{T})\dot{\mathbf{v}} = \mathbf{F}_A \quad (7.6.16)$$

$$\mathbf{N}^T(\mathbf{T})\dot{\mathbf{A}} + \mathbf{L}(\mathbf{T})\dot{\mathbf{v}} = \mathbf{F}_V \quad (7.6.17)$$

In the flow equations (7.6.13)–(7.6.15), the main electromagnetic coupling is through the resistive or Joule heating in the energy equation (which is proportional to the square of the current density) and the Lorentz body force in the momentum equation (which is proportional to the current density and the magnetic flux density). The electromagnetic equations are coupled to the flow problem primarily through temperature-dependent properties; a change of phase in the flow region may require some adjustment in geometry and boundary conditions to account for expansion or contraction of the fluid region. The small magnetic Reynolds number assumption produces the reduced coupling of the electromagnetics to the flow problem. If this assumption is not invoked, the magnetic equation (7.6.16) would have an additional term that depends on the fluid velocity.

Equations (7.6.13)–(7.6.17) form a strongly coupled set when all of the dependencies are present. Though a fully coupled algorithm is attractive from the convergence point of view, the very large size of the system and the generally differing time scales make such an approach impractical. In many applications, the electromagnetic fields are slowly varying or steady and the flow process will set the time scale and be the dominant process. The fact that the fluid and thermal “loads” come from the electromagnetics, while only properties vary in the other coupling direction, also argues for this choice of master and slave process. The standard staggered algorithm would thus proceed with the current temperature field being transferred to the electromagnetics equations (7.6.16) and (7.6.17). A time integration or steady solution for the electric and magnetic fields would produce the needed Joule heating and Lorentz forces. Transferring these data back to the nonisothermal flow equations would allow the next fluid and thermal fields to be computed. In the limit of temperature-independent electromagnetic properties and steady EM fields, Eqs. (7.6.16) and (7.6.17) reduce to a magnetostatics description, which only needs to be solved once for resistive heating and body forces. Slowly varying electric and magnetic fields could also be treated as a series of steady states that may be precomputed and made available to the flow problem as a loading history.

7.6.9 Quasi-Static Solid – Electromagnetic Field Interactions 3 & 4 & 5

The interaction of solid mechanics with electromagnetics will not be treated here in detail because it is outside the scope of the text. It is sufficient to note that these types of interactions can produce coupled problems that are similar in type and complexity to those described in the previous sections. Electric fields may cause stresses and deformations in a solid (electrostriction) and deformation may induce an electrical response (piezoelectric effect). These types of interactions are primarily through the constitutive relations for the material. Also, a strong magnetic field may produce a magnetic stress (magnetostriiction) as described by the Maxwell stress tensor. The algorithms for coupling these types of interactions would certainly be similar to the fluid and thermal procedures outlined above.

7.7 Implementation of Coupled Algorithms

A primary feature of coupled field problems in mechanics is the almost endless variety of possible interactions. This diversity plus the accompanying distribution of workable numerical algorithms, as illustrated in the previous sections, complicates the construction of finite element software. Though this text is not oriented toward the descriptions of code architectures and finite element implementation issues, it is important to discuss some general aspects of coupled problem implementation.

The first choice in implementing a coupled or multiphysics solution method is whether the algorithm will be in a single code or multiple software packages. For strongly coupled problems that will be solved in a fully coupled method, the choice is obviously a single code. In many ways this is the simplest type of implementation since only one matrix problem needs to be considered and all the equations are solved at once. Finite element interpolation order for the various fields, equation ordering within the matrix, type of integration and/or iterative method demand consideration but are not major impediments to implementation. The major drawback to this approach is that the code does one and only one type of coupled problem.

A second approach to coupling involves a single code or software package that is composed of modules for the different types of physical phenomena that are to be modeled. The coupling or required data exchanges are then orchestrated from a driver routine. This type of design would work for either strongly or weakly coupled problems since any strongly coupled interaction can be solved iteratively. The interactions in this case are predefined in terms of what data will be transferred from one process to another; new types of interactions require code modifications that may be quite extensive if the alterations are in constitutive behavior rather than boundary or source terms. An advantage in this design is the sharing of finite element infrastructure between modules. An element library, solver package, and data management routines can be made common between physics modules and reduce code maintenance. Since the individual modules may be run independently, the overall utility of the software is increased significantly over the previous, fully coupled approach. The disadvantage of the multiple module implementation is the size of the overall code, the number of modules that may be required for general applications, and designing enough flexibility into the driver to anticipate the numerous types and strengths of interactions that may be needed.

A final methodology that strongly resembles the multiple module concept introduces the idea of coupling separately maintained mechanics codes into an interacting package. No driver routine is needed though the code containing the dominant mechanics process will act as the master routine. The secondary mechanics codes will act as slave processes and provide data on the demand of the master. For such a design to be feasible, the individual codes must be sufficiently general to make proper use of any externally supplied data and return needed quantities to the master. Also, since no assumptions about which physical process will be dominant in any particular application, all codes should be constructed to function in either the master or slave role. The communication between individual codes could be achieved through data file transfers. Another alternative is the use of message passing software developed for parallel computing environments. Message passing utilities, such as the PVM (Parallel Virtual Machine) libraries [21] or the MPI (Message Passing Interface) libraries [22], allow the memory to memory

transfer of data, while each code is running on its own processor. The advantage of the independent code design is that the very substantial investment in stand alone mechanics code development can be readily applied to coupled problems with relatively minor software modifications. Stand alone mechanics codes also tend to contain more extensive capabilities and more robustness for an application than a module in a more monolithic code. The seamless coupling of such specialized codes provides an increase in overall capability. The drawbacks to this design are the need to set some standards for data transfer protocols and develop some general interface routines that use either file reads and writes or the message passing utilities.

In addition to the overall architecture, there are a number of algorithm details that must be resolved and implemented during the development of a coupled problem or multiphysics capability. Many of these items have been mentioned throughout this chapter but the major issues will be collected here as a series of questions to summarize the subject.

1. How many physical processes will be allowed in any simulation? Though coupling of only two processes have been described above, there are a variety of important problems in which three physical phenomena can be coupled.
2. What types of coupling will be allowed? Will the data dependencies be in source terms, constitutive relations, boundary conditions, geometric changes, or all of these areas?
3. What types of data will be exchanged between processes? Will the coupling be limited to nodal variables, or will element and global data also be permitted in the exchange?
4. How will the dominant or master process be selected? Will this be predetermined so that only certain types of physics may drive the simulation or will any process be selectable as the master?
5. What assumptions will be made regarding the domains for the different physics? Will these areas be forced to coincide or will they be completely arbitrary in extent and overlap? Will the mesh discretizations be the same for each physical process or will data interpolation be required?
6. How will the solution procedures for each process be synchronized? Will only similar time dependencies, both steady or time dependent, be allowed in the coupling or will any useful combination be permitted? How will the time integration processes in each process be allowed to influence each other? How will the frequency of data exchange be determined?

As was outlined in the individual discussions of the various types of coupled heat transfer and fluid mechanics problems, the answers to the above questions can vary considerably. In any given situation the answers are fairly obvious but when planning a general capability the decisions are more complicated. Though complete generality in multiphysics simulations has not as yet been achieved, considerable progress has been made in linking the most important types of interactions together to obtain more realistic finite element solutions.

7.8 Numerical Examples

7.8.1 Introduction

The coupled problems illustrated in this section are relatively complex engineering analyses that are intended to demonstrate some of the multiphysics areas described in this chapter. Due to space limitations, each problem is simply outlined and some representative results described. The finite element codes used in these computations and the method of code coupling vary from example to example. In the first example of a thermal stress problem, the COYOTE [23] heat transfer code was connected via subroutine calls to the quasi-static solid mechanics code, JAS [24]. The second and fourth examples involving electromagnetics employed individual codes linked by the PVM [21] message passing utilities. The second problem of induction heating used the thermal code COYOTE [23] and the electromagnetics code TORO [25], while the remelting problem was solved with the flow code NACHOS II [26] connected to TORO. The third example was solved with the flow code GOMA [27], which has a solid mechanics and ALE capability included as part of a fully coupled algorithm.

7.8.2 Thermal-Stress Example

The thermal-structural interaction example is a geometrically simple problem with a fairly complex material response. A cylindrical canister is filled with a foam-like material; located within the foam-like material is a solid of rectangular cross section. The canister is subjected to a thermal boundary condition that is representative of a heat source at one end of the canister. As the temperature of the container rises, the foam-like material begins to decompose, producing an off-gas and the eventual disappearance of the foam-like material. The container is pressurized by the gas production causing deformation of the canister wall and changes in the geometry for the thermal analysis. Figure 7.8.1 shows a meshed, cut away view of the canister and its contents at the initial time.

A full three-dimensional simulation of this problem was produced using a thermal analysis code as a master process and coupled to the quasi-static, solid mechanics code through a subroutine interface. A chemical kinetic description was used to detail the heat and mass transfer due to the decomposition of the foam-like material. The chemistry was solved in conjunction with the temperature field via an operator splitting technique. As the foam-like material completed the decomposition reaction in each element, the element was removed from the simulation (element death). Mass transfer from the decomposition reaction was accounted for in the expanding clear space of the container through use of a bulk node (see Section 3.11.4). Radiation inside the container was included in the analysis with view factors being constantly updated due to element death and changes in the shape of the container walls. The pressure computed in the bulk node was passed to the solid mechanics code for use as a boundary condition on the container walls. The temperature field was also transferred to the solid mechanics code for use in the temperature-dependent, elastic-plastic constitutive model of the container wall. The foam-like material was not modelled in the structural computation because of its very limited load carrying capacity. An updated, deformed geometry was transferred back to the thermal analysis code at the completion of each solution step.

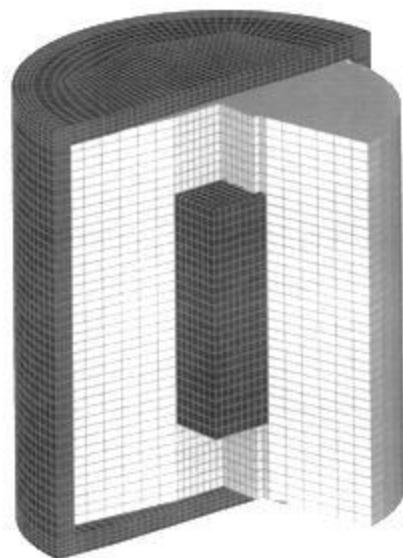


Figure 7.8.1: Schematic and mesh for heated canister.

The deformed geometries for the canister at several times during the heating process are shown in Figure 7.8.2. The foam-like material begins to disappear (element death) at 250 seconds after heating begins and creates a clear space at the bottom of the canister. Visible deformation begins at approximately 300 seconds and continues to a time of 450 seconds after which additional deformations are relatively small. The deformations in Figure 7.8.2 are scaled by a factor of 1.5.

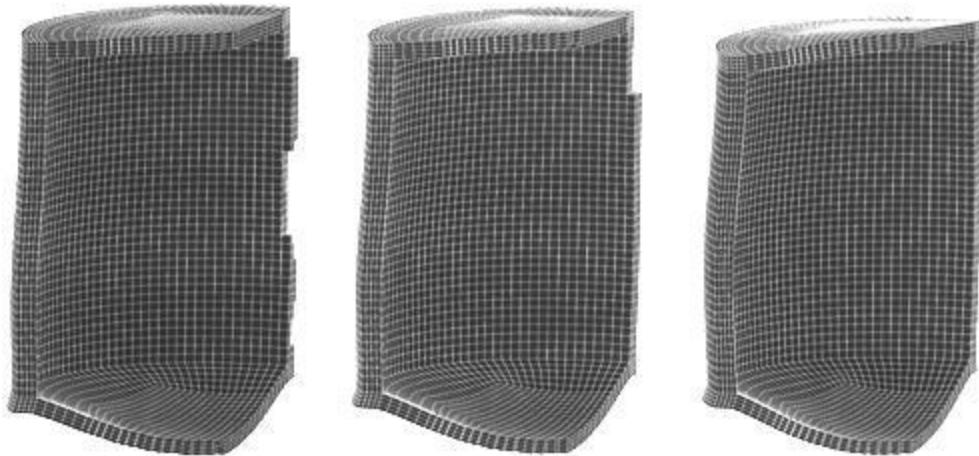


Figure 7.8.2: Deformed geometry for heated/pressurized canister.

The state of the foam within the canister is illustrated in the cut away plots in Figure 7.8.3 which are viewed from the heated end of the container. The recession of the foam-like material is easily seen in this figure, as only the currently active elements are plotted; an undeformed configuration of the canister is used for reference. At the last time shown, the block buried in the foam-like material is just starting to be exposed. Though the mesh discretization is not excessively refined, the complexity of the physical processes modeled required that a parallel computation be utilized to reduce the analysis time. This analysis is due to R. G. Schmitt (personal communication).

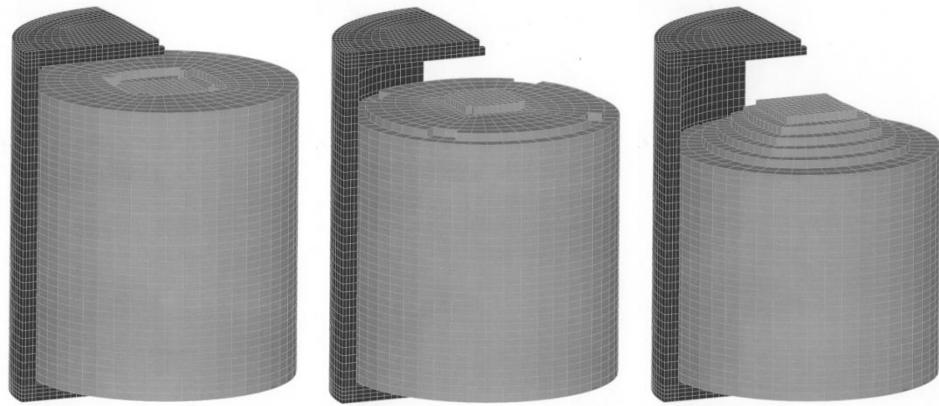


Figure 7.8.3: Material decomposition and removal for heated/pressurized canister; currently active elements are shown.

7.8.3 Thermal-Electromagnetic Example

Induction heating is used extensively in manufacturing processes, especially in the surface treatment (hardening) of ferromagnetic materials. In general terms, induction hardening consists of a part placed in the vicinity of a alternating current coil. The time varying coil current induces an eddy current in the part which in turn produces a resistive heating and temperature rise in the part. The high frequency of the applied coil current and shallow penetration of the magnetic field (skin depth effect) in the part, limit the thermal effect to the part surface. Complexities in this process include strongly temperature-dependent electromagnetic properties, a strongly nonlinear magnetic permeability and the relative motion between the part and the coil when a realistic industrial process is simulated.

Shown in Figure 7.8.4 is a schematic of a simplified induction hardening process in which a cylindrical part is heated within an annular coil of square cross section. The finite element mesh used for the axisymmetric analysis is shown in Figure 7.8.5. The time harmonic electromagnetics problem was solved over the domain shown in the figure where the current density was specified for the coil. The thermal problem was solved over the cylindrical part and the coil, each of which was subjected to

Joule heating; the air space was not included in the thermal analysis. Radiation between the part and coil could be included in the analysis but was omitted for this demonstration. For this application the thermal problem is the master process since the time constant for thermal diffusion is significantly longer than the time constant for induction. The thermal problem was integrated over the heating cycle of 5.4 seconds using an implicit integration method. At every time step, the current temperature field was transferred to the electromagnetics code via PVM and the (nonlinear) time harmonic eddy current problem solved for the Joule heating in the part and coil. For the present example, the coil current was 8 kA and was run at a frequency of 7.6 kHz. These data were returned to the thermal code via PVM and used in the next temperature solution. At the conclusion of the heating cycle the part was subjected to a spray quench which was applied via a convective heat transfer boundary condition on the vertical surface of the part.

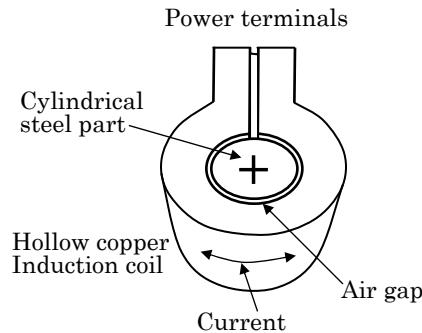


Figure 7.8.4: Schematic of induction hardening problem.

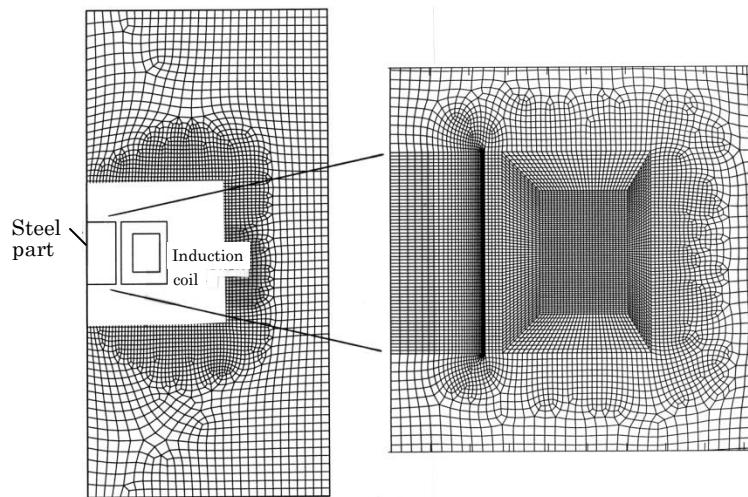


Figure 7.8.5: Finite element mesh for axisymmetric induction hardening problem.

Figure 7.8.6 shows computed and measured temperature histories for various locations within the midplane of the cylinder. The agreement between model and experiment is reasonably good and shows the rapid rise in surface temperature. Note that the surface temperature exceeds the Curie temperature at which point the magnetic permeability falls to a free space value. At quench, the surface temperature drops immediately while the internal temperatures continue to rise due to conduction.

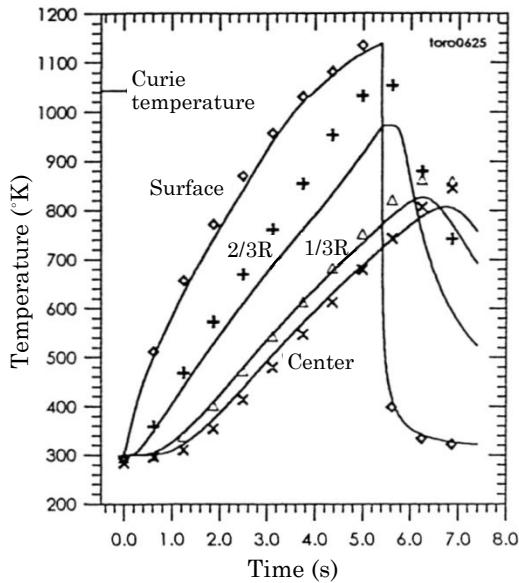


Figure 7.8.6: Predicted and measured temperature histories for induction heated cylindrical part.

In the real analysis setting, the thermal problem is augmented by a reaction kinetics solution that describes the phase transformation of the material and the hardness of the processed part. Ultimately, the coupled solution described here is also passed to a quasi-statics solid mechanics code that evaluates residual stresses and deformation due to the heat treatment. Further details on this problem are available in [28].

The same type of problem with a longer cylinder that translates along its axis, through the coil, was solved. A sliding surface was placed within the (air space) mesh between the coil and the cylinder. Along the slide surface a multipoint constraint was employed to enforce continuity of the magnetic vector potential. The mesh inside the slide surface moved with the translating cylinder while the mesh outside the slide surface remained stationary with the coil. The time-dependent motion of the cylinder was specified. The thermal problem required no constraint since the air was not included in the analysis. If radiation had been included in the analysis, continual updating of the view factors would be necessary to account for the motion of the cylinder.

Figure 7.8.7 contains a series of contour plots which illustrate the temperature field at several times during the motion of the rod through the coil. Motion of the cylinder begins 0.5 second after the coil current is applied and continues for 1

second after heating is terminated; the coil is energized for a period of 5 seconds. The process is sufficiently rapid so that the heat loss from the cylinder is minimal and no thermal boundary conditions were applied; no quench process was modeled in this example.

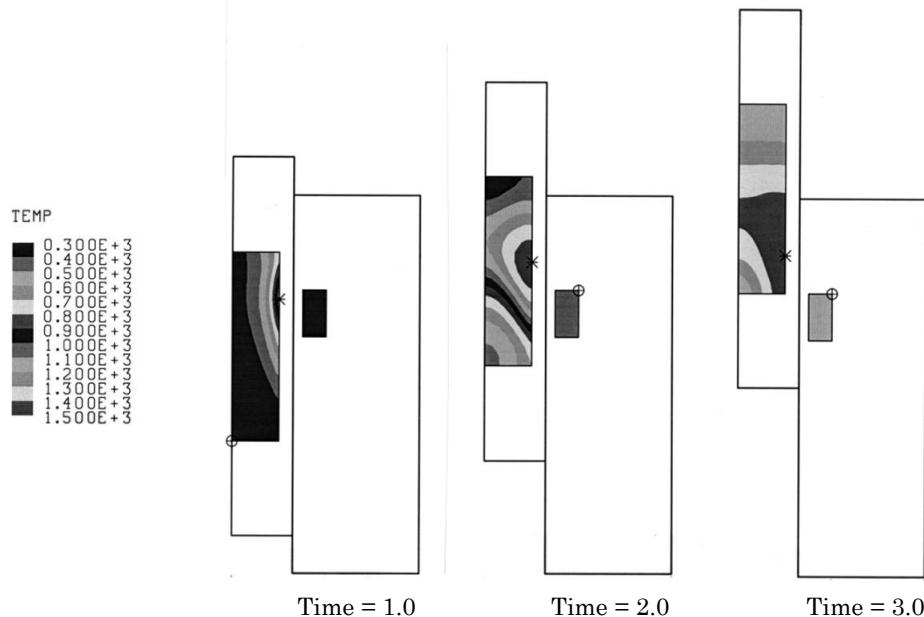


Figure 7.8.7: Temperature contours for a cylinder moving through an alternating current coil.

7.8.4 Fluid-Solid Interaction Example

As an example of the interaction between a viscous fluid and a solid, the problem of creating a drop by pushing fluid through an orifice is considered. A cross section of a typical axisymmetric geometry is shown in Figure 7.8.8 along with an initial mesh for the simulation. A piston is located in a small fluid reservoir and is moved toward the orifice at an almost constant rate. Fluid is pushed through the orifice forming an axisymmetric drop (no gravity). The boundary behind the piston is an inflow boundary that simulates the connection to a large reservoir. During the piston motion, the piston and reservoir wall are slightly deformed due to the relative thinness of the solid sections and the fluid pressure.

The drop formation problem was simulated using a fully coupled ALE technique to track the motion of the fluid free surface (with surface tension) and motion/deformation of the solid regions. Due to the large geometric change in the fluid region, mesh distortion became unacceptable at several times during the transient and remeshing of the fluid region was required. The solid was treated as an elastic material.

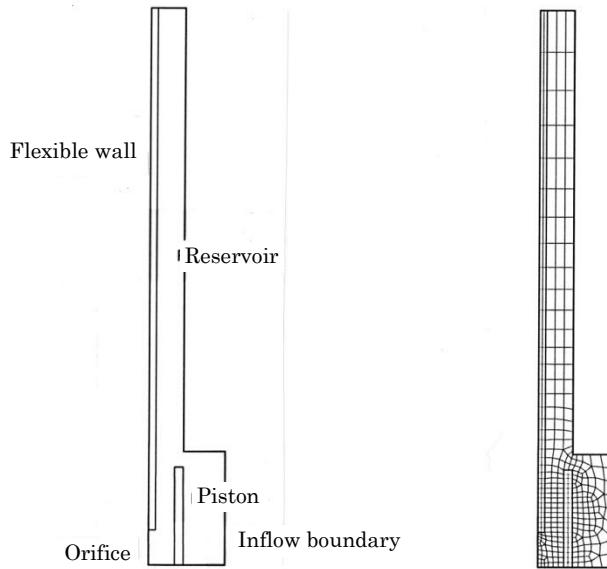


Figure 7.8.8: Schematic and initial finite element mesh for drop formation problem.

Shown in Figure 7.8.9 is a series of deformed mesh plots that illustrate the motion of the piston and the formation of a fluid drop. Necking of the drop is apparent in the second plot and reaches incipient pinch-off by the last plot. No mechanism for pinch-off is available in this method and the simulation was terminated at the last time shown. The deflection of the reservoir wall is not visible in the unmagnified plots of Figure 7.8.9; the deformation is shown at one time in Figure 7.8.10 where the displacements have been magnified by a factor of five. The deflection of the wall is obvious since the wall was initially parallel with the piston. Small deflections of the piston were observed early in the transient as the piston started to accelerate the fluid; these deflections were not as large as the bending of the reservoir wall.

7.8.5 Fluid-Electromagnetic Example

The last example in this section is a simulation of the vacuum arc remelt (VAR) process that involves nonisothermal viscous fluid flow, flow in a porous medium, solidification and Joule heating, and Lorentz forces from a magnetostatic field. The problem and solution techniques are described in detail in [29] and will only be summarized here. Figure 7.8.11 shows a schematic of the VAR process and a typical finite element mesh for the axisymmetric analysis of the solidifying ingot. In the VAR process, a high current is passed through a consumable electrode, creating a metal vapor plasma arc between the electrode and the melt pool contained within the crucible. The arc provides the energy for melting the small electrode.

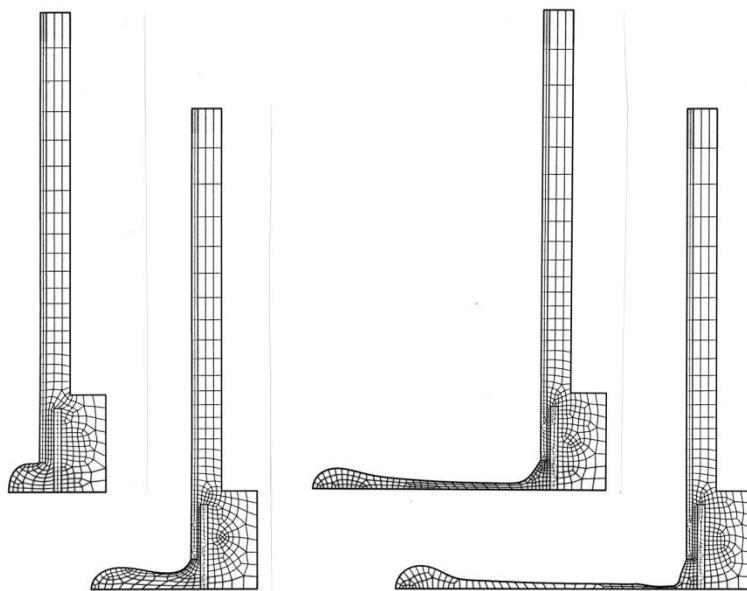


Figure 7.8.9: Finite element solution for drop formation problem. Remeshing occurs at three times during the simulation.

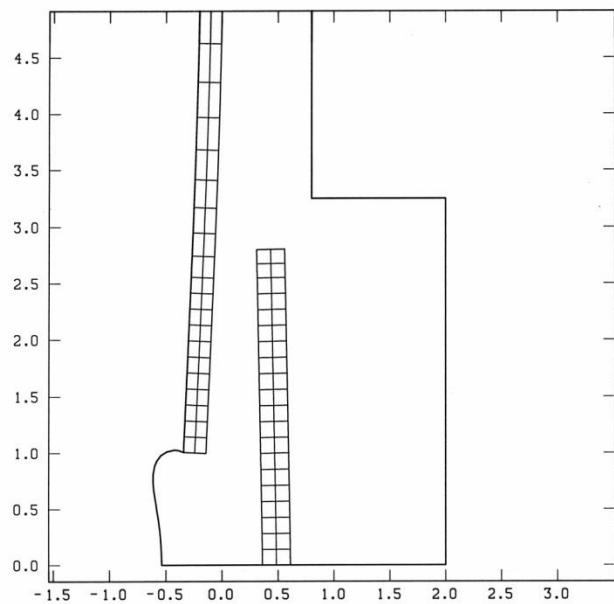


Figure 7.8.10: Reservoir wall deformation during drop formation; the displacements are magnified by a factor of five.

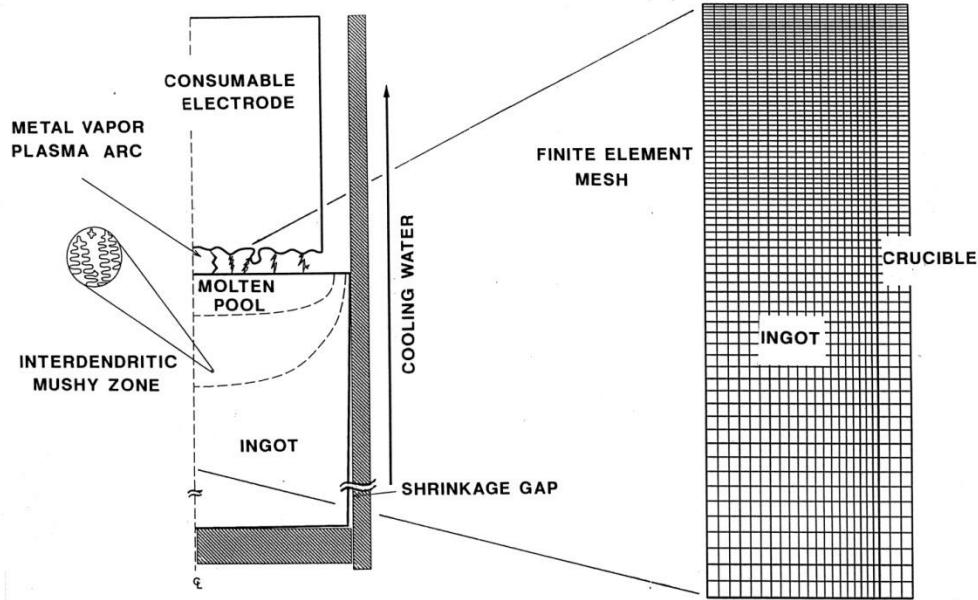


Figure 7.8.11: Schematic for vacuum arc remelt process and typical finite element mesh for crucible.

The objective is to create a larger ingot with better and more uniform metallurgical properties. In the finite element simulation of the solidifying ingot, the melt pool is represented as a viscous incompressible fluid with significant buoyancy forces; the mushy zone, which is typically present in alloy solidification, is modeled as a porous media using the Darcy–Brinkman equations of Section 4.4. Solid body conduction describes heat flow in the solidified portion of the ingot. The melt pool fluid is a conducting medium and the interaction of the current and magnetic field produces body forces and resistive heating within the crucible. A low magnetic Reynolds number is typical for this flow and the magnetostatics equations from Section 7.5 provide the appropriate description of the fields. The electromagnetic problem is weakly coupled to the flow problem since the properties are temperature and phase dependent. Boundary conditions are fairly complex and represent mass, energy, and current input from the arc and interface conditions with the crucible wall. As the ingot solidifies it shrinks in diameter and pulls away from the wall. This effect is modeled by increasing the resistance to heat transfer and electrically insulating the wall below the shrinkage point. A coupled solid mechanics solution would be required to eliminate this model parameter.

The algorithm used to solve this type of problem is a cyclic procedure with the fluid mechanics and heat transfer code being the master process; the magnetostatics code was invoked whenever an update on the volume heating or body force was required. Because time-independent solutions were of primary interest, a solution strategy involving zeroeth order continuation and combinations of the Picard and Newton iteration for the flow problem was adopted. The solution process was

delicate with small increments in surface heat flux from the arc being used to “advance” the solution while buoyancy and Lorentz forces were scaled to maintain a balance between these opposing effects. Data transfer between the two finite element solutions occurred at user-specified intervals and was processed by the PVM utilities. Contour plots for the main variables in a typical VAR simulation are shown in Figure 7.8.12.

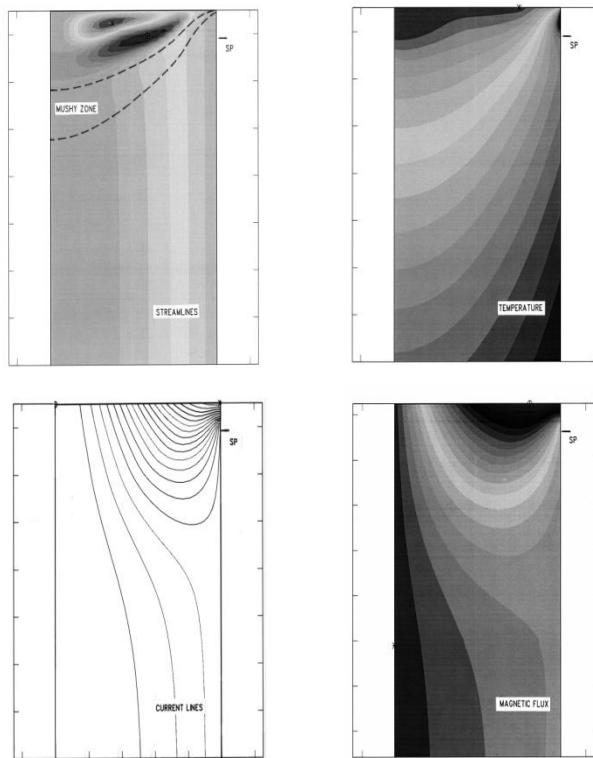


Figure 7.8.12: Contour plots for a typical VAR solution with buoyancy and Lorentz effects included.

Of primary interest is the stream function plot that illustrates the motion of the liquid metal from its small inflow at the top surface of the melt pool, through the clockwise rotation of the electromagnetically driven, Lorentz cell, the counterclockwise rotation of the thermally driven cell, through the porous layer, and finally as a solid body translation with the solid ingot. The temperature field shows some influence of the cell structure and the separation of the ingot from the wall. The electromagnetic fields are also strongly influenced by the wall boundary conditions. These results are for a fairly large current applied to the ingot. As the current is reduced, the thermal cell grows in size and limits the Lorentz cell to a thin region at the top of the melt pool. Without the Lorentz force, very large thermal cell solutions with unrealistically large melt pools are predicted. Additional details on modeling of the VAR process can be found in [29,30].

References for Additional Reading

1. O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 6th ed., Elsevier, New York (2005).
2. R. D. Cook, D. S. Malkus, and M. E. Plesha, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, New York (1989).
3. K. J. Bathe, *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ (1996).
4. M. A. Crisfield, *Non-linear Finite Element Analysis of Solids and Structures, Volume 1 - Essentials*, John Wiley & Sons, Chichester, U.K. (1991).
5. M. A. Crisfield, *Non-Linear Finite Element Analysis of Solids and Structures, Vol. 2: Advanced Topics*, John Wiley, Chichester, UK (1997).
6. T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*, John Wiley, Chichester, UK (2000).
7. J. N. Reddy, *An Introduction to the Finite Element Method*, 3rd ed., McGraw-Hill, New York (2006).
8. J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis*, Oxford University Press, Oxford (2004).
9. J. N. Reddy, *An Introduction to Continuum Mechanics with Applications*, Cambridge University Press, New York (2008).
10. J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press, New York (1997).
11. L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall, Englewood Cliffs, New Jersey (1969).
12. ABAQUS Theory Manual, Version 5.8, Hibbit, Karlsson & Sorenson, Inc., Pawtucket, Rhode Island (1998).
13. J. N. Reddy, *Energy Principles and Variational Methods in Applied Mechanics*, 2nd ed., John Wiley & Sons, New York (2002).
14. W. F. Hughes and F.J. Young, *The Electromagnetodynamics of Fluids*, John Wiley & Sons, New York (1966).
15. J. D. Jackson, *Classical Electrodynamics*, Second Edition, John Wiley & Sons, New York (1975).
16. J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York (1993).
17. M. N. O. Sadiku, *Numerical Techniques in Electromagnetics*, CRC Press, Boca Raton, Florida (1992).
18. K. J. Binns, P. J. Lawrence, and C. W. Trowbridge, *The Analytical and Numerical Solution of Electric and Magnetic Fields*, John Wiley & Sons, New York (1992).
19. P. P. Silvester and P. L. Ferrari, *Finite Elements for Electrical Engineers*, Third Edition, Cambridge University Press, Cambridge, U.K. (1996).
20. M. V. K. Chari, A. Konrad, M. A. Palmo, and J. D'Angelo, "Three-Dimensional Vector Potential Analysis for Machine Field Problems," *IEEE Transactions on Magnetics*, **MAG-18**, 436–446 (1982).
21. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, *PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Network Parallel Computing*, MIT Press, Cambridge, Massachusetts (1996).
22. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI, Portable Parallel Programming with the Message Passing Interface*, MIT Press, Cambridge, Massachusetts (1995).

23. D. K. Gartling, R. E. Hogan, and M. W. Glass, "COYOTE - A Finite Element Computer Program for Nonlinear Heat Conduction Problems," Sandia National Laboratories Report, SAND94-1173 and SAND94-1179, Albuquerque, New Mexico (2000).
24. M. L. Blanford, "JAS3D A Multi-Strategy Iterative Code fro Solid Mechanics Analysis, User's Instructions, Sandia National Laboratories Report, in progress, Albuquerque, New Mexico (2000).
25. D. K. Gartling, "TORO - A Finite Element Computer Program for Nonlinear Quasi-Static Problems in Electromagnetics," Sandia National Laboratories Report, SAND95-2472 and SAND96-0903, Albuquerque, New Mexico (2000).
26. D. K. Gartling, "NACHOS II - A Finite Element Computer Program for Incompressible Flow Problems," Sandia National Laboratories Report, SAND86-1816 and SAND86-1817, Albuquerque, New Mexico (1986).
27. P. R. Schunk, P. A. Sackinger, R. R. Rao, K. S. Chen, R. A. Cairncross, T. A. Baer, and D. A. Labreche, "GOMA 2.0 - A Full-Newton Finite Element Program for Free and Moving Boundary Problems with Coupled Fluid/Solid Momentum, Energy, Mass, and Chemical Species Transport: User's Guide," Sandia National Laboratories Report, SAND97-2404, Albuquerque, New Mexico (1998).
28. D. R. Adkins, D. K. Gartling, J. B. Kelly, and P. M. Kahle, "TORO II Simulations of Induction Heating in Ferromagnetic Materials," in *Proceedings of the 1st International Induction Heat Treating Symposium*, Indianapolis, Indiana (1997).
29. D. K. Gartling and P. A. Sackinger, "Finite Element Simulation of Vacuum Arc Remelting," *International Journal for Numerical Methods in Fluids*, **24**, 1271-1289 (1997).
30. F. J. Zanner and L. A. Bertram, "Vacuum Arc Remelting - An Overview," in *Proceedings of the 8th International Conference on Vacuum Metallurgy*, **1**, 512-552 (1985).

Parallel Processing

8.1 Introduction

Parallel processing is still a relatively new facet of computing, with a significant history in computational mechanics of less than two decades. However, the technology is a certainty and will continue to have a major impact on computational mechanics for the foreseeable future. Many of the currently popular computing systems are based on the sequential processing, von Neumann architecture, in which instructions are executed in sequence, one at a time, on a single data element. For computational algorithms that have a sequential character, this type of processing is optimal. However, most computational problems are not strictly sequential, but have processes that are independent of each other (e.g., element matrices), and could therefore be executed simultaneously. For these parallel tasks or algorithms, sequential processing is obviously not optimal. In the past, increases in computer performance have been primarily achieved through increases in speed of the central processing unit (CPU). However, as the rate of increase in CPU speed slows down, and costs for complex CPUs increase, computer developers have turned to alternative architectures to achieve high computing performance. A logical choice is to use multiple processors and exploit the parallelism inherent in most computational algorithms. This hardware development has led to the rapidly growing field of parallel computing.

This chapter provides a brief introduction to the topic of parallel processing and its relation to finite element algorithms. In previous chapters we have mostly avoided discussions of finite element implementation since there are many ways to achieve the required algorithmic result. Some texts [1–4] have illustrated the finite element implementation process by providing detailed descriptions and listings of source codes for a variety of applications. Appendix A provides a minimal introduction of this type for some two-dimensional problems of interest in this text. The implementation of finite element procedures in a parallel computing environment involves a substantial increase in complexity with many algorithms requiring a heavy dependence on areas in computer science. A good introduction to parallel computing is found in [5].

In the next section an introductory description of parallel systems is provided. We will then outline some of the major algorithmic areas that must be addressed if a finite element procedure is going to be adapted to or designed for a parallel computer. It is not our intention to be extremely detailed in this chapter but it is important to call attention to methods and code structures that have been tested and put into use in engineering simulations. Likewise, it is important to point out where finite element methods and parallel processing are the most conflicted and a substantial code development effort may be required.

8.2 Parallel Systems

8.2.1 Classification

The standard terminology used for classifying parallel computers is the scheme proposed by Flynn [6]. Even though this scheme does not include many new developments in classification [7], it still forms a good basis. Following are the four classes in the scheme proposed by Flynn [6]:

1. Single Instruction Single Data (SISD)
2. Single Instruction Multiple Data (SIMD)
3. Multiple Instruction Single Data (MISD)
4. Multiple Instruction Multiple Data (MIMD).

An instruction stream is the sequence of instructions executed by a processing element (PE) and a data stream is the sequence of data on which instructions are performed.

The sequential computers (Micro and Mini computers) fall under the class of the SISD system. They have a single processing element executing instructions in sequence on a single stream of data. The SIMD systems have many processing elements and execute the same set of instructions in each of the processing elements but on different data streams. Vector computers fall under this class. The MISD computers execute a different set of instructions in each processing element but on the same data stream. These systems are useful in applications like signal processing and are often termed pipeline processors. The MIMD machines are the most general type among the parallel processing systems. They have many processing elements executing their own sets of instructions on different data streams.

Parallel processing systems can be further classified based on the characteristics listed below. Most of the parallel machines of interest in computational mechanics come under the MIMD or SIMD class of computers. The classification discussed here is relevant primarily to these systems.

8.2.1.1 Granularity of the processing elements

Granularity of the system can be either fine or coarse, where a coarse grain system has fewer processors than the fine grain system. A coarse grain system typically operates with a few (say, 100) powerful processing elements, while a fine grain system would have many (say, 1,000) ordinary processing elements. The boundary between fine and coarse grain computers is not rigidly defined. Systems with more than a 1,000 processors are usually termed *massively parallel* computers.

8.2.1.2 Topology of interconnections

The topology of interconnections is the manner in which the individual processing elements are connected. In terms of hardware, the connections may be arranged as a ring, tree, pipeline, or a hypercube. With the advent of new systems and with a virtual communication facility, the programmer can assume that each processor is connected to every other processor. The system will take care of the data communication through the intermediate processors based on the actual physical topology. Efficiency of such a system will be high if the actual and virtual layouts are identical. Figure 8.1.1 shows the commonly used interconnect topologies. These types of processor topologies are common to multiprocessors, i.e., computers with all of the processors located in a single system.

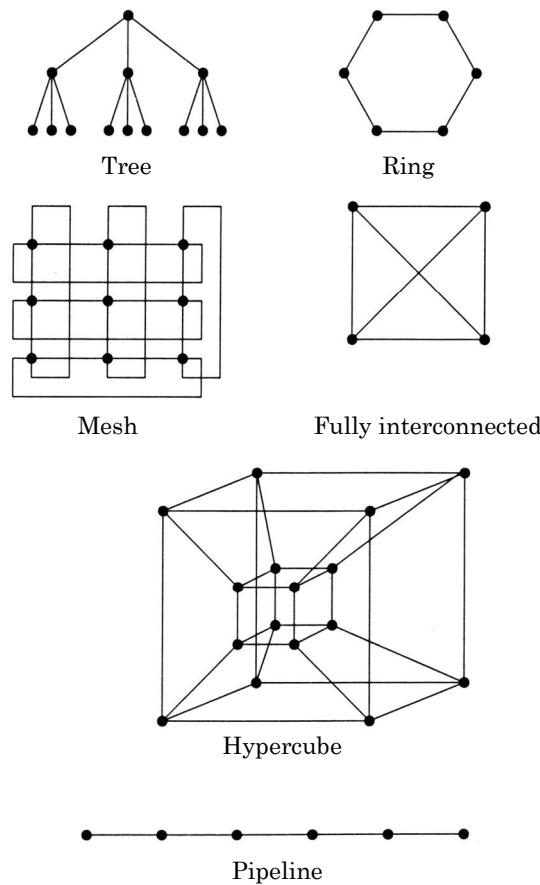


Figure 8.1.1: Topology of interconnections.

Networks of single processor computers can be arranged to provide a parallel computing system; a heterogeneous network of multiprocessor machines can be used to form a virtual parallel machine. These types of networked architectures are usually referred to as *distributed computing*. The major difference between a parallel or massively parallel computer and a distributed computing system is the variability of the processing unit. On a multiprocessor every processor is essentially the same, whereas in a distributed (virtual) machine each processor on the network could be very different in terms of speed and capability. Likewise, data formats may vary significantly between processors on a network. These differences lead to software portability issues and concerns with the basic parallel paradigm, language selection, and communication protocols.

8.2.1.3 Distribution of control across the processing elements

The control of the system may be with a single processor, which directs the system, as is often the case in SIMD machines. Also, in MIMD machines, even though a master (or main) processor is identified, each processor operates on its own, independently. However, interprocessor communication demands synchronous operation.

8.2.1.4 Memory access

Each processing element can have its own local memory and/or share the global memory of the system. Both types of architectures have advantages and disadvantages. In a local memory system, synchronous communication becomes vital and deadlocks/delays in communication may cause drastic deterioration of the system efficiency. In a shared memory system, processing elements compete with each other in addressing the memory locations. Bottlenecks in communication remain one of the major problems in a shared memory system. Again, the most common architectures used in computational mechanics are the shared memory MIMD machines and the distributed memory MIMD computers with the latter being the most popular. A detailed discussion of these characteristics is available in [5-8].

8.2.2 Languages and Communication Utilities

There are many types of parallelism in computing. For some applications, special computer languages or language extensions have been developed to simplify program development. A *data parallel* model seeks to exploit parallel capabilities in the performance of even simple arithmetic tasks such as a vector-matrix product. In this case, corresponding components of the matrix and vector are distributed to different processors for multiplication and returned for summation. Microtasking or multitasking of this type originally required compiler directives to indicate sections of the code that could be spread across multiple processors. Currently, languages such as High Performance (HP) Fortran [8] and Fortran 90 [10] implicitly support data parallel constructs. For systems based on transputers [11], the language of choice is Occam, which was developed specifically for this type of processor [11,12].

The usual approach to parallel computing of mechanics problems is a domain decomposition or partitioning paradigm with a single program, multiple data (SPMD) software implementation. In this situation, the problem data are divided or partitioned equally among the available processors. A copy of the finite element mechanics code runs on each processor and computes its locally assigned part of the problem. Crucial to the success of this paradigm is the requirement to efficiently communicate (transfer) data between processors such that the global problem can be solved from its distributed pieces. The two traditional methods for processor to processor communication on standard MIMD machines are the use of shared memory or the use of an explicit message passing utility. Message passing has become something of a standard because of its generality, especially considering the difficulties associated with distributed computing.

In the standard SPMD implementation, no special languages or extensions are required. The use of ANSI standard Fortran 77 and C is widespread. Finite element programs require a mixture of data processing procedures, some of which are best suited to C or Fortran 90 constructs. Other parts of the code are optimally constructed using the unit stride facilities of Fortran. Most current codes are a mixture of the two languages. The use of Object Oriented designs and the C++ language [13] is gaining in popularity.

Utilization of a distributed memory parallel system requires the ability to perform two basic functions: create tasks and set communication channels between processors. Portable, system independent libraries with standardized interfaces have

been developed to accomplish these functions. The two most popular packages of this type are the Parallel Virtual Machine (PVM) library [14] and the Message Passing Interface (MPI) software [15]. Other systems, such as Linda [16] and P4 [17] exist but do not have the overall popularity of PVM and MPI. PVM is the older of the two communication packages and is best suited for systems with a few processors and an architecture that operates with a master processor and a number of slave processors. Distributed computing architectures with heterogeneous network processors are well aligned with PVM. The MPI software has found the widest utility and is heavily used on multiprocessor machines and networks, as well as massively parallel machines. MPI is especially convenient for the SPMD models found in most mechanics applications. Some language extensions that support task and channel functionality are also available, though not widely used. The CC++ [18] language is an extension of C++ and supports parallelism from within the language; Fortran M [19] is an extension of Fortran with similar functionality.

8.2.3 Performance

The performance of a parallel code is generally estimated by its parallel efficiency, which indicates how much is gained by solving the problem in parallel. The best way to estimate this is to compare it with a sequential version of the same code. This can be done in two different ways [20].

8.2.3.1 Algorithmic efficiency

This is defined as the ratio of CPU time taken for running the parallel code on N_p processing elements to the time taken when the code is run on a single processing element. The sequential code used here is essentially a parallel code run on a single processor. Hence the sequential code will also have the delays in communication reflected in its timing. This will indicate the parallel efficiency of the algorithm rather than the problem. The formula for algorithmic efficiency is

$$\eta_{alg} = \frac{t_{par}}{t_{seq1}N_p} 100 \quad (8.2.1)$$

where t_{par} is the time taken to execute the program on a parallel machine with N_p processors and t_{seq1} is the time taken to run the same code on a single processor.

8.2.3.2 Actual/beneficial efficiency

Another measure, the actual or beneficial efficiency is defined as the ratio of CPU time taken for running the parallel code on N_p processing elements to the time taken by the fastest sequential code run on one processing element (t_{seq2}). This efficiency will reflect the parallel efficiency of the problem, or in other words it will reflect both the level of parallelism in the problem under consideration and the efficiency of the parallel algorithm used. It indicates the actual benefit derived by resorting to parallel processing. It is given by the following equation:

$$\eta_{act} = \frac{t_{par}}{t_{seq2}N_p} 100 \quad (8.2.2)$$

The actual efficiency will always be less than or equal to the algorithmic efficiency. If these efficiencies are less than 50%, then it is not usually worth considering the

case. Sometimes a speed-up ratio is also used to rate a program. This is just the ratio of time taken by the parallel program to the time taken by the sequential program. The speed-up ratio should be greater than one. The speed-up factor can be calculated by multiplying the efficiency by $N_p/100$.

8.2.3.3 Scalability

Another important issue for performance is scalability. It is expected that systems with increasing numbers of processors will continue to be developed. Software that maintains performance with an increasing number of processors is therefore important for effective use of a parallel system. Early definitions of scalability sought to decrease computational time by increasing the number of processors on a fixed work load. It was believed that the serial portions of the algorithm and communication costs would ultimately limit performance as the number of processors increased. This is a statement of Amdahl's law which has as its focus a decreasing work load per processor. A current view of scalability asks for a constant work load per processor, in which case the overall work load would increase with the number of processors. In this situation, any degradation in performance can be assigned to communication cost.

8.3 FEM and Parallel Processing

8.3.1 Preliminary Comments

Previous chapters have outlined the steps/procedures for finite element algorithms for a number of specific problem types. The implementation of all of these finite element methods follows a series of generic steps. In the first part of this section, these generic finite element procedures will be discussed in the context of a parallel computing implementation. A later section will describe some of the issues that arise due to specific computational and modeling techniques such as multipoint constraints and code coupling.

As noted previously, the most common parallel architecture of interest in the computational mechanics community is the MIMD architecture with a shared memory, or more preferably, with a distributed memory. The MIMD architecture argues for a decomposition or partitioning strategy when considering approaches to finite element solutions in a parallel environment. Note that here we are considering only the solution to a single finite element model; other, more coarse grained levels of parallelism are possible when considering multiphysics applications. If the computer has N processors, then the finite element mesh is decomposed into N roughly equal groups of (contiguous) nodes or elements with each group assigned to a processor. Each processor computes the finite element algorithm for its group of nodes or elements with interprocessor communication required for completion of the algorithm at processor boundaries where nodes or elements are shared. Input and output from the process may be handled through a special, dedicated front end processor, by one of the N processors that is assigned the duty, or through all the processors and a parallel I/O hardware system. Often the finite element software is structured such that a copy of the executable code is running on each processor; at appropriate points in the execution, the processors are synchronized and data are exchanged between processors. This is the SPMD model mentioned previously.

As sketched here, the domain decomposition or partitioning approach to parallel finite element solutions is the most common paradigm currently in use. The discussion to follow will be oriented toward this type of procedure. Note that the term domain decomposition is also used to describe parallel solution methods for large systems of equations. The two areas are obviously related but not necessarily coincident. We use the term here only to describe an approach to parallel implementation of finite element methods.

8.3.2 Generic FEM Steps

A finite element algorithm or program is naturally divided into the following major steps or units:

- External preprocessing
- Internal preprocessing
- Solution processing
 - Element matrix building
 - Matrix solving
 - Solution control
- Internal postprocessing
- External postprocessing

The first and last processing steps usually occur outside of the basic finite element program in specialized mesh generators and graphics packages. However, these external processes must be included when discussing a parallel application because they play a significant role in the problem setup. Each of these processing steps is considered individually in the next several sections.

8.3.3 External Preprocessing

Preprocessing for a finite element model usually refers to the generation of the element mesh from some type of geometric description of the problem region. In many cases the problem geometry resides in a computer-aided design (CAD) file. Mesh generation sets the location of the nodal points (global coordinates) and defines the elements in terms of the nodes through construction of the nodal connectivity. The connectivity is an ordered list relating the nodes in an element to the global node numbering in the assembled model. Elements in the mesh are normally assigned material i.d.s at this point and element surfaces and nodes are flagged for boundary conditions. The output from this step is a file, in some standard format, that provides all the required data for each element in the mesh.

For parallel applications, preprocessing involves several additional steps. The distribution of the finite element computations to a number of processors is accomplished through domain decomposition or mesh partitioning algorithm. Within this procedure, the overall problem is divided into N sets of elements for execution on N processors of the parallel platform. Efficient execution of the parallel problem demands that the computational work on each processor be as equal as possible and that the communication between processors be minimized. This part of the process is referred to as “load-balancing.” Various software packages, such as Chaco [21] and Parmetis [22], have been developed to perform this type of mesh decomposition and processor load-balancing. The mesh decomposition algorithm

is often based on viewing the mesh as a graph and utilizing sophisticated graph partitioning methods. Recursive bisection and octree methods have also been used as partitioning procedures. In each case, the partitioning method must consider the general case where the number of unknowns per node and per element may vary between elements. This situation is usually handled by weighting the vertices of the graph to reflect the work associated with a node or element. Also, partitioning may be nodal or element based. In a node-based decomposition, nodes are assigned to processors and elements may be split between processors. An element-based decomposition assigns elements to a processor with nodes then being shared between processors. The type of needed decomposition and load balance is thus dependent on the architecture of the finite element code.

The domain decomposition and load-balancing software typically functions by reading the mesh generation file, decomposing the mesh for the stated number of processors, and producing its own output file or files. The form of the partitioned output again depends on the structure of the finite element code. If the finite element code is designed to have a master process and a number of slave processes, the domain decomposition data can be stored in a single scalar file. This file would contain the load balance data needed by the master process to distribute the node, element, and communication data to the individual processors. Finite element codes designed to run with a copy of the code on each processor (the SPMD model) require the decomposition data in a set of parallel files. Each of the parallel files would contain some global information plus the nodal, element, and communication data specific to a processor.

Figure 8.3.1 shows the finite element mesh of a casting with a partial gating system that was partitioned for use in a heat conduction/radiation simulation using eight processors. The original geometry was meshed using a standard mesh generator with the output being processed by a graph partitioning method within the Chaco code [21]. Additional utilities were used to reformat the output file into a series of eight parallel files that were subsequently read by a heat transfer code. Note that the original mesh contained 3,297 elements (6,661 nodes) and the element-based partitioning yielded 412 elements on seven processors and 413 elements on one processor. The number of nodes on each processor varied from 790 to 944.

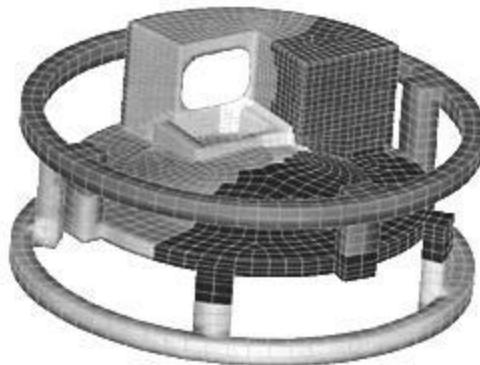


Figure 8.3.1: A finite element mesh showing a partitioning for use on a parallel computer.

In the course of this discussion, it has been implied that the mesh generation occurs on a serial computer and the decomposition and load-balancing follow as a utility process. In many situations this type of procedure is adequate. For extremely large and/or complex finite element models, it may not be possible to generate the mesh on a serial computer. Parallel mesh generation is an area of current investigation and will be required as simulations become more sophisticated. Also, static domain decompositions are adequate for static element topologies; adaptive meshing or remeshing will require dynamic load-balancing capabilities [23].

8.3.4 Internal Preprocessing

The first steps within the finite element code usually entail the reading of input data, reading of mesh data, the allocation of memory for the problem and the setting of execution and control pointers and flags. These same steps are required in a parallel code though the source of the data may be different. As noted in the previous section, mesh data may be available from a parallel input file for some types of codes or may be passed from the master processor and a serial file. Input data that specify material properties, boundary condition values, solution control, and postprocessing requirements usually come from a single, serial file and must be broadcast to the appropriate processors. Processor communication utilities, such as MPI, must be initialized. Assuming a distributed memory machine, local memory on a processor is allocated based on the number of elements or nodes assigned to the processor. Print or text output from the code is generally written from a single processor, even if the code design does not specifically include a master or front-end processor. At the conclusion of initialization and data checking, the processors are synchronized and the first step of the solution is ready for processing.

8.3.5 Solution Processing

A solution step for a finite element method consists of two major parts – element matrix building and solution of the global matrix system. For time-dependent simulations or nonlinear steady problems this step will be repeated for each time step and/or each iteration. The solution of the equations is usually the most time-consuming part of the analysis and can benefit substantially from a parallel implementation. For time-dependent solutions obtained via explicit methods, no matrix solution is required though a parallel implementation is still very effective.

8.3.5.1 Element matrices

The finite element method lends itself naturally to parallel computing especially in the equation building part of the algorithm. The usual structure for a serial finite element code contains a loop over the elements during which element matrices are computed (via numerical quadrature) and added to the global matrix. In most cases, the evaluation of the matrices for an individual element is independent of other elements. Hence, generation of element matrices can be carried out simultaneously, i.e., in parallel. The parallel finite element code, therefore, has a loop over all the elements on the processor, where element matrices are computed and added into the part of the global matrix represented on the processor. This architecture presumes that an element-based decomposition has been used in the mesh partitioning so that all the data for an element reside on the processor. The part of the global matrix that is assembled on each processor represents a combination of both fully summed

equations and partially summed equations for nodes shared between processors. At the completion of the element loop, this submatrix is passed to the parallel matrix solver for further processing and solution. Note that the element-based decomposition is the natural choice for element construction since an element is the primary data entity.

In the case where the partitioning is node based, equation construction is slightly more complex. The looping is over the nodes on the processor. An inverse connectivity is used to determine which elements (and nodes) are connected to the node. The integration of the shape functions for the nodal equations will require coordinate data from other processors and therefore communication between processors. At the completion of the loop, a submatrix will be available to the parallel solution routine. This type of decomposition is well suited for most parallel, iterative matrix solution methods but is not well aligned with finite element data processing.

8.3.5.2 Matrix solvers

The matrix solver used for a parallel code implementation can be either iterative or direct; the serial versions of these two types of solvers are discussed in Appendix B. Direct solvers obtain the solution to the system of simultaneous equations in a fixed number of steps. On the other hand, iterative solvers do not converge to the solution in a fixed number of iterations. The number of iterations required depends on the mathematical structure of the system of equations being solved. Nonlinearity of the problem and the diagonal dominance of the system of equations affect the convergence rate. In some cases, it may not even be possible to obtain a solution using an iterative solver, especially without a preconditioning of the coefficient matrix. The major advantage of most iterative solvers is that they are amenable to the element-by-element (EBE) or block EBE solution approach. The system of equations is solved without direct assembly of the element matrices. These algorithms have very high parallel efficiency when compared with the direct solvers.

Most direct solvers use a Gauss elimination scheme as their base algorithm. This algorithm consists of two steps, the forward elimination and the backward substitution. The forward elimination can be programmed in parallel with a good parallel efficiency, but with a significant increase in solver complexity. The parallel forward elimination is much like a substructure process where the interior nodes (fully summed equations) on a processor are eliminated, leaving the border nodes (partially summed equations). For efficiency, the summation and subsequent elimination of groups of border nodes should be redistributed among all the processors. The continuation of this process to its conclusion involves substantial data tracking and communication. Finally, the backward substitution is sequential in nature, and this will decrease the overall parallel efficiency of the solver. Also, the forward elimination will be less efficient for banded systems.

The Gauss–Jordan solver is a less efficient solver in a sequential mode, and it is rarely used in sequential programming. If N is the number of unknowns, Gauss elimination performs $O(N^3/3)$ multiplications and the same number of additions during the elimination operation (multiplication and division and addition and subtraction are taken as identical operations). The back substitution scheme involves $O(N^2/2)$ multiplications and additions. On the other hand, the

Gauss–Jordan solver performs $O(N^3/2)$ multiplications and additions during the elimination step, and there is no back substitution. Since there are more operations in the Gauss–Jordan method than the Gauss elimination scheme, it is not preferred; but the issues take different priorities in parallel processing. The Gauss–Jordan algorithm is devoid of the back substitution step, which is highly sequential. This is a major advantage. Also, more computation is performed for a unit of communication in the Gauss–Jordan method than the Gauss elimination scheme. This will result in an improved parallel efficiency [23]. For a complete system, Gauss–Jordan will break even with Gauss elimination, even in terms of CPU time, with only a few processors. However, for a banded system, a greater number of processors will be required to achieve parity.

Among the many iterative solvers, like the Jacobi iteration method and the Gauss–Seidel method, the Conjugate Gradient Method (CGM) is very efficient and effective for many finite element equation systems. In theory, it will yield convergent solution in less than N iterations, where N is the number of unknowns. In practical computations, round-off errors will affect convergence and it often requires more than N iterations to achieve a solution. Hence, restarting procedures have to be used. Even with this penalty, it is faster than the Gauss–Seidel iteration method by an order of magnitude.

The conjugate gradient method is often used for minimizing a function. It uses the gradient of the function to minimize it, without evaluating the function itself. This method was originally developed for solving systems of simultaneous equations. The original method can solve only symmetric, positive-definite systems. Using an appropriate preconditioning matrix, other systems can also be solved. Details of the conjugate gradient method with preconditioning can be found in [24] as well as in Appendix B.

In developing a parallel implementation for an iterative matrix solution method, two key features are exploited. Algorithms like the conjugate gradient method involve matrix–vector products and vector dot products. These steps can be easily split into parallel operations by expressing the product of a matrix with a vector as the sum of the products of their components. The key idea is

$$[A]\{x\} = \left[\sum_{i=1}^{N_{elem}} [A_i] \right] \{x\} \quad (8.3.1)$$

$$[A]\{x\} = \sum_{i=1}^{N_p} \left[\sum_{j=M_{i1}}^{M_{i2}} [A_j] \right] \{x\} \quad (8.3.2)$$

where N_p is the number of processors and N_{elem} is the number of elements. The values of M_{i1} and M_{i2} can be calculated in different ways. Only the matrix $[A]$ is expressed as a sum and not the load vector. Therefore, the load vector generated by the processors has to be added to obtain the final vector, which involves some communication. If the load vector is also expressed as a sum, the process will be even more complicated. Also, it will be computationally intensive with a heavy communication load. This can be readily seen from the following equations:

$$A \cdot x = (A_1 + A_2)(x_1 + x_2) = A_1x_1 + A_1x_2 + A_2x_1 + A_2x_2 \quad (8.3.3)$$

$$A \cdot x = \left[\sum_{i=1}^N A_i \right] \left[\sum_{j=1}^N x_j \right] = \sum_{i=1}^N \sum_{j=1}^N A_i x_j \quad (8.3.4)$$

If there are N values of A and x , each value of A is associated with all the N components of x . This places a requirement on the processors to know all the values, and there is a larger number multiplications as well. It is therefore not advisable to express the vectors involved in the matrix-vector product as a sum of its element contributions.

The other product frequently encountered is a vector dot product. Consider two vectors $\{x\}$ and $\{y\}$ of dimension N . The dot product can be written as

$$\{x\} \cdot \{y\} = \sum_{i=1}^N x_i y_i \quad (8.3.5)$$

$$\{x\} \cdot \{y\} = \sum_{i=1}^{N_p} \left[\sum_{j=M_{i3}}^{M_{i4}} x_j y_j \right] \quad (8.3.6)$$

The values of M_{i3} and M_{i4} are calculated from the element connectivity of the elements present in the i th processor. Care should be taken to make sure that a particular node is identified with only one processor in calculating the dot product, even though it may be associated with many processors. This is due to the fact that the vectors are assembled and they are global. Depending upon the particular algorithm under consideration, the methodology can be restructured.

Although the matrix solution process occupies a large majority of the execution time in a finite element code, it is no longer necessary to understand the implementation details of the various matrix solvers. A number of software projects have produced solver libraries with convenient, standardized interfaces that can be used easily with many finite element formulations. Many of these libraries have both direct and iterative matrix solvers and usually have available a number of preconditioning options. Typical of these solver packages are LAPACK [25] for shared memory machines and ScaLAPACK [26], PETSC [27] and Aztec [28] for distributed memory architectures. The use of well-documented solver libraries is an effective method for quickly getting a finite element algorithm running in parallel. As noted above, much of a finite element computation is naturally parallel and easy to code. The matrix solution is the difficult part of the parallel implementation and involves most of the “computer science” aspects of a finite element code. Though solver libraries may be conveniently used for expediting program development, it is important to understand the relationship between various matrix solver methods and characteristics of the finite element equations (matrices).

8.3.5.3 Solution control

At the conclusion of a matrix solution, various computations must be completed that influence the overall progression of the algorithm. In particular, norms on the solution field must be computed and checked to determine if convergence has occurred. Also, for time-dependent simulations tests for reaching steady state are computed and a new time step may be required. Each of these computations requires

some communication between processors though these tasks are minor compared to the data exchange that occurs within the matrix solver.

8.3.6 Internal Postprocessing

At the completion of the finite element solution, some postprocessing of the data may be required. The computation of auxiliary quantities, such as flux data or globally integrate values, may occur at the end of each time step or nonlinear iteration or at the completion of the steady state solution. In a serial algorithm these data are again processed within a loop on the elements. The parallel implementation of this type of computation is similar though some communication may be required.

If fluxes are computed at the integration points of an element, then the computation is completely parallel if the decomposition was element based. When continuous flux data are required, the usual process is to extrapolate the integration point fluxes to the nodes and average the nodal data between elements attached to the node. This type of averaging requires some interprocessor communication. Using communication utilities such as MPI makes the computation straightforward. Quantities that are integrated over surfaces or volumes likewise require data exchange between processors but nothing outside the scope of MPI-like capabilities.

Output of postprocessing data may be accommodated by either of the two methods used to input data. For many systems, data would be collected on the master processor or a designated processor and written to a printed output file and/or a postprocessing file. Codes using the SPMD model would more likely collect data on a single processor for printing while using parallel output files for postprocessing data. The parallel files are usually concatenated into a single file prior to use with a graphics package.

8.3.7 External Postprocessing

External postprocessing refers to the use of specialized graphics programs to investigate the solution fields generated by the finite element code. These software packages have a variety of methods for graphically displaying data, manipulating data, and probing the computed fields. For applications generated by a parallel algorithm the concerns related to postprocessing stem mainly from the size of the computed data sets. It is a significant challenge to graphically render nodal or element data from time-dependent finite element meshes containing several million elements. Not only is rendering time an issue but temporary and permanent storage of the data sets is a formidable problem. Work on parallel graphics software is in progress and will be mandatory as finite element simulations continue to increase in size and complexity.

8.3.8 Other Parallel Issues

It is clear from the previous section that most of the parallel issues in a standard finite element algorithm are contained within the matrix or linear algebra solution procedure. However, there are some finite element modeling capabilities that complicate the normal parallel implementation based on mesh partitioning. As described in Chapter 3, contact and multipoint constraint boundary conditions require data external to the current element. Algorithms that require the staggered or cyclic solution of multiple equation sets also add some complexity to a parallel implementation. These topics are considered briefly in the following sections.

8.3.8.1 Nonlocal data

The efficient use of a domain decomposition method requires that data used by a processor be local to the processor. This is the usual situation when an element-based decomposition is used and the looping is over the elements on a processor. Nonlocal or off-processor data can occur in the implementation of certain types of boundary conditions, such as node-to-surface contact and multipoint constraints (see Section 3.11.3). Implementation of these types of conditions requires two steps: the location of the slave node on the master surface and the construction of an equation relating the slave node to the nodes on the master surface. When the slave node is on a different processor from the master surface, additional interprocessor communication is required during both steps of the procedure. Note that preprocessing data is not available for establishing these communication paths since the boundary condition may be dynamic and change location during the course of the solution.

The search for the location of one or more (slave) nodes within a list of master elements is a procedure that occurs in several computational situations. The transfer (interpolation) of data from one mesh to another requires this type of procedure as does the implementation of the boundary conditions mentioned above. In a parallel setting the local search process is most effectively done after the slave nodes and master elements that are geometrically close are redistributed to common processors. This amounts to a slave node-based repartitioning of the data. An effective method for load balancing this problem is a recursive bisection algorithm. After completion of the local, on-processor search procedure, the location data for each slave must be returned to its original processor.

The actual implementation of the multipoint constraint requires a second step in which a constraint matrix is constructed between the slave node and the nodes on the master surface. The contact boundary condition requires this same type of construction if the implementation is fully implicit; an explicit formulation for contact can be constructed without this second step. After the constraint equations are constructed on the processor containing the slave node, the constraint coefficients must be distributed to the processors containing the master surface nodes and added to their portion of the global matrix. These interprocessor communication paths are flagged during the search process since they may change with time and are not included as part of the original mesh partition.

Another type of nonlocal data occurs for modeling features such as a bulk node (see Section 3.11.4). Here, the equation for a bulk node is not part of the original mesh partition and in fact is owned by all the processors. Because the bulk node may be connected to a large number of elements (and nodes) located on different processors, updating of the bulk node equation occurs on every processor. This local update is followed by a global accumulation of data and integration of the bulk node equation on each processor. The bulk node parameters are then available to each processor for use in boundary conditions on the processor.

8.3.8.2 Multiphysics simulations

The use of cyclic or staggered solution methods for multiphysics or multiple systems of equations may introduce additional complexities into the parallel implementation. The two issues of main concern are load balancing and data transfer between

equation sets. When the equation sets are solved on the same element topology (and same domain decomposition), these problems do not occur.

As a specific example of the load-balancing problem, consider the fairly common situation of solving the coupled heat conduction, enclosure radiation problem or the heat conduction, chemical kinetics problem outlined in Chapter 3. These types of problems would normally be solved with a single software package, so that code coupling is not a complicating factor. A recommended method in both cases is the staggered or alternating solution of the two equation sets as a function of time. In a parallel implementation, the finite element conduction problem is partitioned across the N processors and solved by the standard method. However, when the second equation set is solved it is probable that a number of processors will be idle because its computational domain does not necessarily coincide with the conduction domain. In the radiation case, the view factors and radiosity problem are computed for a subset of element surfaces. In the chemical kinetics problem it is usual for only one material in the problem to be reactive, while the other materials require no chemistry solution. Note that it is quite possible to have both radiation and chemistry in the same problem in which case none of the equation domains are coincident.

The solution to this dilemma is of course to have individual decompositions for each equation set in the problem. The individual partitions may be generated externally if the equation sets are complex, e.g., two or more partial differential equation models. For less complex secondary equation systems, the partitioning may be done internally with a simple algorithm. In the case of radiation view factors or chemistry on an element, the number of equations or elements divided by the number of processors provides an adequate partition. This simple method works in these cases because there is no interaction between the equations within the set. Regardless of the method used for partitioning the second equation set, some additional interprocessor communication is required. In essence, the communication relationship between the partitions must be established to allow the exchange of dependent variables between the sets of equations.

The previous load-balancing discussion centered on the situation encountered when a single finite element code was required to handle multiple equation sets. It was assumed that the variables were defined at the same spatial locations (nodes and elements) in each equation, though the spatial extent of the domains did not have to coincide. The coupling of independent finite element codes, as described in Chapter 7, can present another difficulty for parallel implementation. When the coupled codes use independent finite element meshes and different domain decompositions, the transfer of variable data from one mesh to the other involves added complexity. The basic problem is similar to the multipoint constraint problem described above. To transfer nodal data from mesh A to mesh B, the locations of the mesh B nodes within mesh A must be found. Once located, the nodal data for mesh B can be found from shape function interpolation on mesh A. The search procedure is most effectively done in parallel by redistributing the data across the processors such that groups of mesh B nodes and mesh A elements are geometrically close. A recursive bisection procedure is fairly optimal for this process. Once the interpolated nodal data are computed, the variables are transferred back to the original processor so that the next solution step in the algorithm can be completed. In general, the search

and interpolation process must be completed in each direction if the two solution fields are mutually dependent. When the geometries are static the search need only be done once, though the interpolation would have to be carried out at each data exchange step. The complexity of this process obviously increases substantially for each coupled code added to the simulation.

8.4 Summary

The future of computational mechanics will be influenced to a great extent by the developments in parallel processing. Computational fluid dynamics, in particular, can and will benefit significantly from the developments in parallel computing. Currently, most of the parallel algorithms are the parallel versions of existing sequential algorithms. It may be more useful to conceive and develop algorithms which are fundamentally parallel. Other than CFD, problems involving artificial vision, signal processing, neural networks, optimization, and general finite element analysis will benefit with the advances in parallel computing. The scope is much broader than what is mentioned here.

References for Additional Reading

1. C. Taylor and T. G. Hughes, *Finite Element Programming of the Navier-Stokes Equations*, Pineridge Press, Swansea, U.K. (1981).
2. T. J. R. Hughes, *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*, Prentice Hall, Englewood Cliffs, New Jersey (1987).
3. J. N. Reddy, *An Introduction to the Finite Element Method*, 3rd ed., McGraw-Hill, New York (2006).
4. J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis*, Oxford University Press, Oxford (2004).
5. I. Foster, *Designing and Building Parallel Programs*, Addison-Wesley, Reading, Massachusetts (1995).
6. M. J. Flynn, “Very High Speed Computing Systems,” *Proceedings of IEEE*, **54**, 1901–1909 (1966).
7. B. Davidson, “A Parallel Processing Tutorial,” *IEEE Antennas and Propagation Society Magazine*, **32**, 6–19 (1990).
8. M. R. Hord, *Parallel Supercomputing in SIMD Architecture*, CRC Press, Boca Raton, Florida (1990).
9. C. Koebel, D. Loveman, R. Schreiber, G. Steele, and M. Zosel, *The High Performance Fortran Handbook*, MIT Press, Cambridge, Massachusetts (1994).
10. J. Adams, W. Brainerd, J. Martin, B. Smith, and J. Wagener, *The Fortran 90 Handbook*, McGraw-Hill, New York (1992).
11. A. Carling, *Parallel Processing: the Transputer and Occam*, Sigma Press, London, U.K. (1992).
12. J. Galletly, *Occam 2*, Krieger Publishing, Melbourne, Florida (1990).
13. J. Barton and L. Nackman, *Scientific and Engineering C++*, Addison-Wesley, Reading, Massachusetts (1994).
14. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, *PVM: Parallel Virtual Machine, A User’s Guide and Tutorial for Network Parallel Computing*, MIT Press, Cambridge, Massachusetts (1996).

15. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI, Portable Parallel Programming with the Message Passing Interface*, MIT Press, Cambridge, Massachusetts (1995).
16. N. Carriero and D. Gelernter, "Linda in Context," *Communications of the ACM*, **32**, 444–458 (1989).
17. R. Butler and E. Lusk, "Monitors, message, and clusters: The *p4* parallel programming system," *Parallel Computing*, **20**, 547–564 (1994).
18. P. Sivilotti and P. Carlin, "A Tutorial for CC++," California Institute of Technology Technical Report, CS-TR-94, Pasadena, California (1994).
19. I. Foster, R. Olson, and S. Tuecke, "Programming in Fortran M," Argonne National Laboratory, Mathematics and Computer Sciences Division Technical Report, ANL-93/26, Argonne, Illinois (1993).
20. T. L. Freeman, *Parallel Numerical Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey (1992).
21. B. Hendrickson and R. Leland, "The Chaco User's Guide - Version 2.0," Sandia National Laboratories Report, SAND94-2692, Albuquerque, New Mexico (1994).
22. G. Karypis and V. Kumar, "Parmetis: Parallel Graph Partitioning and Sparse Matrix Ordering Library," University of Minnesota, Department of Computer Science Technical Report, 97-060, Minneapolis, Minnesota (1997).
23. J. J. Modi, *Parallel Algorithms and Matrix Computations*, Oxford University Press, Oxford, U.K. (1988).
24. O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, U.K. (1996).
25. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbau, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, "LAPACK Users' Guide, Second Edition," Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania (1994).
26. L. S. Blackford, J. Choi, A. Cleary, E. D'azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, "ScaLAPACK Users' Guide," Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania (1997).
27. S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "PETSc 2.0 Users Manual," Argonne National Laboratory Report, ANL-95/11 - Revision 2.0.28, Argonne, Illinois (2000).
28. S. Hutchinson, L. Prevost, J. Shadid, C. Tong, and R. Tuminaro, "Official Aztec User's Guide, Version 2.1," Sandia National Laboratories Report, SAND99-8801, Albuquerque, New Mexico (1999).

Appendix A

Computer Program *FEM2DHT*

A.1 Introduction

The computer program *FEM2DHT* is a finite element analysis program for the solution of two-dimensional heat transfer and viscous, incompressible fluid flow problems. The program is a modification of the computer program FEM2D from the book by Reddy [1]. The program is discussed here to aid the readers with the basic computer implementation aspects of the finite element method. The program is educational in nature and it does not contain powerful pre- or post-processors found in a commercial finite element software.

A.2 Heat Transfer and Related Problems

The heat transfer part of the program is based on a slightly general form of equation (2.2.1) on page 44 of this book ($a_{11} = k_{xx}$ and $a_{22} = k_{yy}$):

$$-\left[\frac{\partial}{\partial x} \left(a_{11} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(a_{22} \frac{\partial u}{\partial y} \right) \right] + a_{00}u = f \quad \text{in } \Omega \quad (\text{A.2.1})$$

with the boundary conditions [c.f., equation (2.2.5a,b)]

$$u = \hat{u}(s) \quad \text{on } \Gamma_u \quad (\text{A.2.2a})$$

$$\left(a_{11} \frac{\partial u}{\partial x} n_x + a_{22} \frac{\partial u}{\partial y} n_y \right) + \beta(u - u_\infty) = q \quad \text{on } \Gamma_q \quad (\text{A.2.2b})$$

where Γ_u and Γ_q are disjoint portions of the boundary Γ such that $\Gamma = \Gamma_u \cup \Gamma_q$. For a heat transfer problem, $a_{11} = k_{xx}$ and $a_{22} = k_{yy}$ denote conductivities in the x and y directions, respectively, $f(x, y)$ denotes the known internal heat generation per unit volume, (n_x, n_y) denote the direction cosines of the unit normal vector $\hat{\mathbf{n}}$ on the boundary, β denotes the convective heat transfer coefficient, and u_∞ is the ambient temperature.

Equation (A.2.1) arises in a variety of other fields and the program *FEM2DHT* can be used to analyze them as long as the governing equation of the problem solved is a special case of Eq. (A.2.1). For example, slow flows of inviscid fluids are often described in terms of either the *velocity potential* ϕ

$$v_x \equiv -\frac{\partial \phi}{\partial x}, \quad v_y \equiv -\frac{\partial \phi}{\partial y} \quad (\text{A.2.3})$$

or the *stream function* ψ

$$v_y \equiv -\frac{\partial \psi}{\partial x}, \quad v_x \equiv \frac{\partial \psi}{\partial y} \quad (\text{A.2.4})$$

Here (v_x, v_y) denote components of the velocity vector \mathbf{v} . Thus, the program *FEM2DHT* can be used to solve the inviscid flow problems in terms of the velocity potential or the stream function, and then post-compute the velocity field as per Eqs. (A.2.3) and (A.2.4).

A.3 Flows of Viscous Incompressible Fluids

The finite element model of viscous, incompressible flows used in the *FEM2DHT* program is the *reduced integration penalty model* discussed in Section 4.3.3. The flows considered here have no inertia and are therefore termed Stokes flows. The governing equations of the model are a combination of the continuity equation (4.3.1) and momentum equations (4.3.2):

$$-\frac{\partial}{\partial x} \left(2\mu \frac{\partial v_x}{\partial x} \right) - \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \right] - \gamma \frac{\partial}{\partial x} \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) = 0 \quad (\text{A.3.1a})$$

$$-\frac{\partial}{\partial x} \left[\mu \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \right] - \frac{\partial}{\partial y} \left(2\mu \frac{\partial v_y}{\partial y} \right) - \gamma \frac{\partial}{\partial y} \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) = 0 \quad (\text{A.3.1b})$$

with the boundary conditions

$$v_x = \hat{v}_x \text{ on } \Gamma_{v_x} \text{ or } \mathcal{T}_1 = \hat{\mathcal{T}}_1 \text{ on } \Gamma_{\mathcal{T}_1} \quad (\text{A.3.3a})$$

$$v_y = \hat{v}_y \text{ on } \Gamma_{v_y} \text{ or } \mathcal{T}_2 = \hat{\mathcal{T}}_2 \text{ on } \Gamma_{\mathcal{T}_2} \quad (\text{A.3.3b})$$

where Γ_{v_x} and $\Gamma_{\mathcal{T}_1}$, and Γ_{v_y} and $\Gamma_{\mathcal{T}_2}$ are disjoint portions of the boundary Γ such that $\Gamma = \Gamma_{v_x} \cup \Gamma_{\mathcal{T}_1}$ and $\Gamma = \Gamma_{v_y} \cup \Gamma_{\mathcal{T}_2}$. The boundary stress components \mathcal{T}_1 and \mathcal{T}_2 are given by

$$\mathcal{T}_1 = \left(2\mu \frac{\partial v_x}{\partial x} - P \right) n_x + \mu \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) n_y \quad (\text{A.3.4a})$$

$$\mathcal{T}_2 = \mu \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) n_x + \left(2\mu \frac{\partial v_y}{\partial y} - P \right) n_y \quad (\text{A.3.4b})$$

In equations (A.3.3a) and (A.3.3b) γ denotes the *penalty parameter*, and in equations (A.3.4a) and (A.3.4b) P denotes the pressure.

A.4 Description of the Input Data

The following table describes the input data required for the solution of heat transfer and related problems and viscous fluid flow problems using program *FEM2DHT*. Sample input data files are included. For additional examples, the reader may consult the book by Reddy [1, Chapter 13].

Table A.4.1: Description of input variables to computer program *FEM2DHT*.*** Data Card 1**

TITLE - Title of the problem being solved (80 characters)

*** Data Card 2**

ITYPE - Problem type

ITYPE = 0, Heat transfer (and like) problems

ITYPE = 1, Viscous incompressible flow problems

IGRAD - Flag for computing the gradient of the solution

IGRAD = 0, No postprocessing is required

IGRAD > 0, Postprocessing is required

When ITYPE=0 and IGRAD=1, the gradient is computed as in Eq. (A.3); for ITYPE = 0 and IGRAD > 1, the gradient is computed as in Eq. (A.4).

ITEM - Indicator for transient analysis:

ITEM = 0, Steady-state analysis is required

ITEM > 0, Transient analysis is required

*** Data Card 3**

IELTYP - Element type used in the analysis:

IELTYP= 0, Triangular elements

IELTYP> 0, Quadrilateral elements

NPE - Nodes per element:

NPE = 3, Linear triangle (IELTYP=0)

NPE = 4, Linear quadrilateral (IELTYP>0)

NPE = 6, Quadratic triangle (IELTYP=0)

NPE = 8 or 9, Quadratic quadrilateral (IELTYP>0)

MESH - Indicator for mesh generation by the program:

MESH = 0, Mesh is not generated by the program

MESH = 1, Mesh is generated by the program for rectangular domains (by MSH2DR)

MESH > 1, Mesh is generated by the program for general domains (by MSH2DG)

NPRNT - Indicator for printing certain output:

NPRNT = 0, Not print array NOD, element matrices, or global matrices

NPRNT = 1, Print array NOD and Element 1 matrices:

[ELK] and {ELF}

NPRNT = 2, Print array NOD and assembled matrices, [GLK] and {GLF}

NPRNT > 2, Combination of NPRNT=1 and NPRNT=2

*** Data Card 4:** SKIP the card if MESH.EQ.1

NEM - Number of elements in the mesh when the user inputs the mesh or the mesh is generated by MSH2DG

NNM - Number of nodes in the mesh when the user inputs the mesh or the mesh is generated by MSH2DG

* **Data Card 5** SKIP the card if MESH.NE.0; otherwise, the card is
read in a loop on the number of elements (N=1, NEM)
NOD(N,I)-Connectivity for the Nth element (I=1,NPE)

* **Data Card 6** SKIP the card if MESH.NE.0
GLXY(I,J)-Global x and y coordinates of Ith global node in
the mesh (J=1, x-coordinate, J=2, y-coordinate)
Loops on I and J are: ((J=1,2), I=1,NNM); the NNM
pairs of (x,y)--coordinates are read sequentially

_____ The next FOUR data cards are read in subroutine MSH2DG _____
* * * * * SKIP Cards 7, 8, 9, and 10 unless MESH.GT.1 * * * * *

* **Data Card 7**
NRECL - Number of line records to be read in the mesh
* **Data Card 8** Read the following variables NRECL times
NOD1 - First global node number of the line segment
NODL - Last global node number of the line segment
NODINC - Node increment on the line
X1 - The global x-coordinate of NOD1
Y1 - The global y-coordinate of NOD1
XL - The global x-coordinate of NODL
YL - The global y-coordinate of NODL
RATIO - Ratio of the first to the last element lengths

* **Data Card 9**
NRECEL - Number of rows of elements to be read in the mesh
* **Data Card 10** Read the following variables NRECEL times
NEL1 - First element number of the row
NELL - Last element number of the row
IELINC - Increment of element number in the row
NODINC - Increment of global node number in the row
NPE - Number of nodes in each element
NODE(I) - Connectivity array of the first element in the row

* **Data Card 11** SKIP the card if MESH.NE.1
NX - Number of elements in the x-direction
NY - Number of elements in the y-direction
* **Data Card 12** SKIP the card if MESH.NE.1
X0 - The x-coordinate of global node 1
DX(I) - The x-dimension of the Ith element (I=1,NX)
* **Data Card 13** SKIP the card if MESH.NE.1
Y0 - The y-coordinate of global node 1
DY(I) - The y-dimension of the Ith element (I=1,NY)

* **Data Card 14**
NSPV - The number of specified primary variables

* **Data Card 15** SKIP the card if NSPV.EQ.0
ISPV(I,J)-Node number and LOCAL degree of freedom number of the I-th specified
primary variable:
ISPV(I,1)=Node number; ISPV(I,2)=Local DOF number
The loops on I and J are: ((J=1,2),I=1,NSPV)

* **Data Card 16** SKIP the card if NSPV.EQ.0

VSPV(I) - Specified value of the Ith primary variable

* **Data Card 17**

NSSV - Number of (nonzero) specified secondary variables

* **Data Card 18** SKIP the card if NSSV.EQ.0

ISSV(I,J)-Node number and LOCAL degree of freedom number of
the Ith specified secondary variable:

ISSV(I,1)=Node number; ISSV(I,2)=Local DOF number

The loops on I and J are: ((J=1,2),I=1,NSSV)

* **Data Card 19** SKIP the card if NSSV.EQ.0

VSSV(I) - Specified value of the Ith secondary variable
(I=1,NSSV)

Data Cards 20 - 26 are for HEAT TRANSFER PROBLEMS (ITYPE = 0) only

* **Data Card 20** SKIP the card if ITYPE.NE.0

A10 |

A1X | Coefficients of the differential equation

A1Y | a11 = A10 +A1X*X + A1Y*Y

* **Data Card 21** SKIP the card if ITYPE.NE.0

A20 |

A2X | Coefficients of the differential equation

A2Y | a22 = A20 +A2X*X + A2Y*Y

* **Data Card 22** SKIP the card if ITYPE.NE.0

A00 - Coefficient of the differential equation

* **Data Card 23** SKIP the card if ITYPE.NE.0

ICONV - Indicator for convection boundary conditions

ICONV = 0, No convection boundary conditions

ICONV > 0, Convection boundary conditions present

* **Data Card 24** SKIP the card if ITYPE.NE.0 or ICONV.EQ.0

NBE - Number elements with convection

* **Data Card 25** SKIP the card if ITYPE.NE.0 or ICONV.EQ.0

The following cards are read for each I, I=1,NBE

IBN(I) - Ith element number with convection

BETA(I) - Film coefficient for convection on Ith element

TINF(I) - Ambient temperature of the Ith element

* **Data Card 26** SKIP the card if ITYPE.NE.0 or ICONV.EQ.0

INOD(I,J)- Local node numbers of the side with convection

(J=1,2; for quadratic elements, give end nodes)

Loops on I and J are: ((J=1,2), I=1,NBE)

_____ Data Card 27 is for VISCOUS FLUID FLOWS (ITYPE = 1) only _____

* **Data Card 27** SKIP the card if ITYPE.EQ.0

AMU - Viscosity of the fluid

PENLTY - Penalty parameter used

_____ Remaining data cards are for ALL problem types _____

*** Data Card 28**

F0 |
FX | Coefficients to define the source term:
FY | $f = F0 + FX*x + FY*y$

_____ Cards 29 thru 32 are for TRANSIENT ANALYSIS only _____

*** Data Card 29** SKIP the card if ITEM.EQ.0

C0 | Coefficients defining the temporal parts of the
CX | differential equations:
CY | $CT = C0 + CX*x + CY*y$

*** Data Card 30** SKIP the card if ITEM.EQ.0

NTIME - Number of time steps for the transient solution
NSTP - Time step number at which the source is removed
INTVL - Time step interval at which to print the solution
INTIAL - Indicator for nature of initial conditions:
INTIAL=0, Zero initial conditions are used
INTIAL>0, Non-zero initial conditions are used

*** Data Card 31** SKIP the card if ITEM.EQ.0

DT - Time step used for the transient solution
ALFA - Parameter in the alfa-family of time approximation
used:
ALFA=0, The forward difference scheme (C.S.)@
ALFA=0.5, The Crank-Nicolson scheme (stable)
ALFA=2/3, The Galerkin scheme (stable)
ALFA=1, The backward difference scheme (stable)
@C.S.=Conditionally Stable; For all schemes with
ALFA < 0.5, the time step DT is restricted to:
DT < 2/[MAXEGN*(1-2*ALFA)], where MAXEGN is the
maximum eigenvalue of the discrete problem.
EPSLN - A small parameter to check if the solution has
reached a steady state

*** Data Card 32** SKIP the card if ITEM or INTIAL.EQ.0

GLU(I) - Vector of initial values of the primary variables
(I=1,NEQ, NEQ=Number of nodal values in the mesh)

Example 1: Convective heat transfer in a square region (see Figure A.4.1) is solved using 2×2 mesh of nine-node rectangular elements. The conductivities are taken to be $a_{11} = a_{22} = 10 \text{ W}/(\text{m } ^\circ\text{C})$. Echo of the input data along with the output from program *FEM2DHT* is given below and on the next couple of pages.

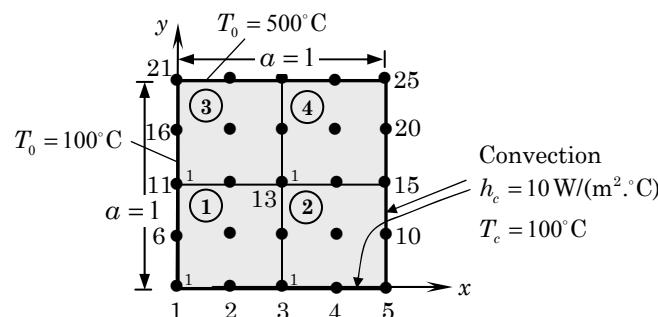


Figure A.4.1: Domain, boundary conditions, and mesh used for the convective heat transfer problem in Example 1.

Input data and output for Example 1: _____

Example 1: CONVECTIVE heat transfer in a square region

```

0 1 0 0          ITYPE,IGRAD,ITEM,NEIGN
2 9 1 0          IEL, NPE, MESH, NPRNT
2 2              NX, NY
0.0 0.5 0.5      X0, DX(1), DX(2)
0.0 0.5 0.5      Y0, DY(1), DY(2)
9                NSPV
1 1 6 1 11 1 16 1 21 1 22 1 23 1 24 1 25 1      ISPV(I,J)
100.0 100.0 100.0 500.0 500.0 500.0 500.0 500.0 500.0  VSPV(I)
0                NSSV
10.0 0.0 0.0     A10, A1X, A1Y
10.0 0.0 0.0     A20, A2X, A2Y
0.0              A00
1                ICONV
4                NBE
1 10.0 100.0 2 10.0 100.0 2 10.0 100.0 4 10.0 100.0    IBM,BETA,TINF
1 2 1 2 2 3 2 3 INOD(I,J)
0.0 0.0 0.0       F0, FX, FY

```

Example 1: CONVECTIVE heat transfer in a square region

OUTPUT FROM PROGRAM *FEM2DHT* BY J. N. REDDY

ANALYSIS OF A HEAT TRANSFER PROBLEM

COEFFICIENTS OF THE DIFFERENTIAL EQUATION:

```

Coefficient, A10 .....= 0.1000E+02
Coefficient, A1X .....= 0.0000E+00
Coefficient, A1Y .....= 0.0000E+00
Coefficient, A20 .....= 0.1000E+02
Coefficient, A2X .....= 0.0000E+00
Coefficient, A2Y .....= 0.0000E+00
Coefficient, A00 .....= 0.0000E+00

```

CONVECTIVE HEAT TRANSFER DATA:

Number of elements with convection, NBE .= 4

Elements, their LOCAL nodes and convective
heat transfer data:

Ele. No.	End Nodes	Film Coeff.	T-Infinity
----------	-----------	-------------	------------

1	1 2	0.10000E+02	0.10000E+03
2	1 2	0.10000E+02	0.10000E+03
2	2 3	0.10000E+02	0.10000E+03
4	2 3	0.10000E+02	0.10000E+03

CONTINUOUS SOURCE COEFFICIENTS:

```

Coefficient, F0 .....= 0.0000E+00
Coefficient, FX .....= 0.0000E+00
Coefficient, FY .....= 0.0000E+00

```

***** A STEADY-STATE PROBLEM is analyzed *****

*** A mesh of QUADRILATERALS is chosen by user ***

FINITE ELEMENT MESH INFORMATION:

Element type: 0 = Triangle; >0 = Quad.).....= 2
 Number of nodes per element, NPE= 9
 No. of primary deg. of freedom/node, NDF= 1
 Number of elements in the mesh, NEM= 4
 Number of nodes in the mesh, NNM= 25
 Number of equations to be solved, NEQ= 25
 Half bandwidth of the matrix GLK, NHBW ...= 13
 Mesh subdivisions, NX and NY= 2 2
 No. of specified PRIMARY variables, NSPV ...= 9

Node	x-coord.	y-coord.	Speci. primary & secondary variables (0, unspecified; >0, specified)	
			Primary DOF	Secondary DOF
1	0.0000E+00	0.0000E+00	1	0
2	0.2500E+00	0.0000E+00	0	0
3	0.5000E+00	0.0000E+00	0	0
4	0.7500E+00	0.0000E+00	0	0
5	0.1000E+01	0.0000E+00	0	0
6	0.0000E+00	0.2500E+00	1	0
7	0.2500E+00	0.2500E+00	0	0
8	0.5000E+00	0.2500E+00	0	0
9	0.7500E+00	0.2500E+00	0	0
10	0.1000E+01	0.2500E+00	0	0
11	0.0000E+00	0.5000E+00	1	0
12	0.2500E+00	0.5000E+00	0	0
13	0.5000E+00	0.5000E+00	0	0
14	0.7500E+00	0.5000E+00	0	0
15	0.1000E+01	0.5000E+00	0	0
16	0.0000E+00	0.7500E+00	1	0
17	0.2500E+00	0.7500E+00	0	0
18	0.5000E+00	0.7500E+00	0	0
19	0.7500E+00	0.7500E+00	0	0
20	0.1000E+01	0.7500E+00	0	0
21	0.0000E+00	0.1000E+01	1	0
22	0.2500E+00	0.1000E+01	1	0
23	0.5000E+00	0.1000E+01	1	0
24	0.7500E+00	0.1000E+01	1	0
25	0.1000E+01	0.1000E+01	1	0

NUMERICAL INTEGRATION DATA:

Full quadrature (IPDF x IPDF) rule, IPDF = 3
 Reduced quadrature (IPDR x IPDR), IPDR = 2
 Quadrature rule used in postproc., ISTR = 2

SOLUTION :

Node	x-coord.	y-coord.	Temperature
1	0.00000E+00	0.00000E+00	0.10000E+03
2	0.25000E+00	0.00000E+00	0.11380E+03
3	0.50000E+00	0.00000E+00	0.13273E+03
4	0.75000E+00	0.00000E+00	0.13350E+03
5	0.10000E+01	0.00000E+00	0.11697E+03
6	0.00000E+00	0.25000E+00	0.10000E+03
7	0.25000E+00	0.25000E+00	0.14268E+03
8	0.50000E+00	0.25000E+00	0.17135E+03
9	0.75000E+00	0.25000E+00	0.17561E+03
10	0.10000E+01	0.25000E+00	0.15241E+03
11	0.00000E+00	0.50000E+00	0.10000E+03
12	0.25000E+00	0.50000E+00	0.18453E+03
13	0.50000E+00	0.50000E+00	0.24111E+03
14	0.75000E+00	0.50000E+00	0.24392E+03
15	0.10000E+01	0.50000E+00	0.21538E+03
16	0.00000E+00	0.75000E+00	0.10000E+03
17	0.25000E+00	0.75000E+00	0.29562E+03
18	0.50000E+00	0.75000E+00	0.34509E+03
19	0.75000E+00	0.75000E+00	0.35267E+03
20	0.10000E+01	0.75000E+00	0.30641E+03
21	0.00000E+00	0.10000E+01	0.50000E+03
22	0.25000E+00	0.10000E+01	0.50000E+03
23	0.50000E+00	0.10000E+01	0.50000E+03
24	0.75000E+00	0.10000E+01	0.50000E+03
25	0.10000E+01	0.10000E+01	0.50000E+03

The orientation of gradient vector is measured from the positive x-axis
x-coord. y-coord. -a11(du/dx) -a22(du/dy) Flux Mgntd Orientation

0.1057E+00	0.1057E+00	-0.9881E+03	-0.5666E+03	0.1139E+04	-150.17
0.1057E+00	0.3943E+00	-0.2681E+04	-0.7906E+03	0.2795E+04	-163.57
0.3943E+00	0.1057E+00	-0.8220E+03	-0.1403E+04	0.1626E+04	-120.37
0.3943E+00	0.3943E+00	-0.1632E+04	-0.2457E+04	0.2950E+04	-123.59
0.6057E+00	0.1057E+00	-0.1830E+03	-0.1566E+04	0.1577E+04	-96.67
0.6057E+00	0.3943E+00	-0.2553E+03	-0.2872E+04	0.2884E+04	-95.08
0.8943E+00	0.1057E+00	0.8494E+03	-0.1499E+04	0.1723E+04	-60.46
0.8943E+00	0.3943E+00	0.1152E+04	-0.2709E+04	0.2944E+04	-66.97
0.1057E+00	0.6057E+00	-0.7097E+04	-0.1718E+04	0.7302E+04	-166.39
0.1057E+00	0.8943E+00	-0.5095E+04	-0.1272E+05	0.1370E+05	-111.83
0.3943E+00	0.6057E+00	-0.2009E+04	-0.4743E+04	0.5151E+04	-112.95
0.3943E+00	0.8943E+00	-0.7522E+03	-0.6432E+04	0.6476E+04	-96.67
0.6057E+00	0.6057E+00	-0.4085E+03	-0.4237E+04	0.4257E+04	-95.51
0.6057E+00	0.8943E+00	-0.2876E+03	-0.5918E+04	0.5925E+04	-92.78
0.8943E+00	0.6057E+00	0.1909E+04	-0.3846E+04	0.4293E+04	-63.60
0.8943E+00	0.8943E+00	0.1194E+04	-0.6904E+04	0.7006E+04	-80.19

Example 2: Flow of an incompressible viscous fluid in a slider bearing (see Figure A.4.2) is analyzed using a mesh of six nine-node elements. The mesh used is to illustrate the input data, and it is not representative of the mesh required to obtain an accurate solution. The input data and an edited output from program *FEM2DHT* are given below.

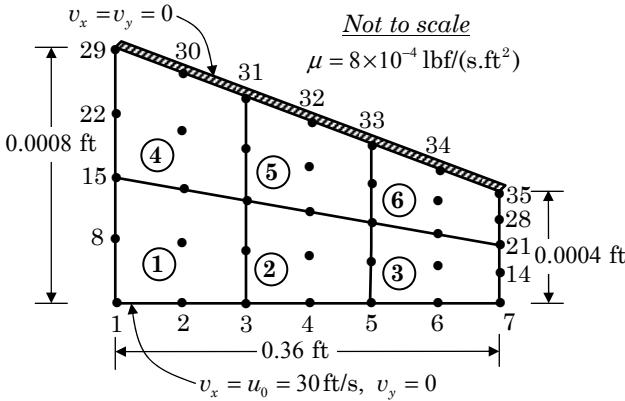


Figure A.4.2: Domain and mesh used for the flow of a viscous incompressible fluid in a slider bearing (Example 2).

Input data and output for Example 2:

```

Example 2: Flow of a viscous fluid in a slider bearing
1 1 0          ITYPE,ISTRS,ITEM
1 9 2 0        IELTYP,NPE,MESH,NPRNT
6 35           NEM, NNM
5              NRECL
1 7 1 0.0 0.0   0.36 0.0    1.0  NOD1,NODL,NODINC,...
8 14 1 0.0 5.0E-5 0.36 5.0E-5 1.0
15 21 1 0.0 1.0E-4 0.36 1.0E-4 1.0
22 28 1 0.0 3.5E-4 0.36 1.5E-4 1.0
29 35 1 0.0 8.0E-4 0.36 4.0E-4 1.0
2              NRECEL
1 3 1 2 9 1 3 17 15 2 10 16 8 9 NEL1,NELL,..
4 6 1 2 9 15 17 31 29 16 24 30 22 23
28             NSPV
1 1 1 2 2 1 2 2 3 1 3 2 4 1 4 2 5 1 5 2
6 1 6 2 7 1 7 2 29 1 29 2 30 1 30 2 31 1 31 2
32 1 32 2 33 1 33 2 34 1 34 2 35 1 35 2 ISPV(I,J)
30.0 0.0 30.0 0.0 30.0 0.0 30.0 0.0 30.0 0.0
30.0 0.0 30.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0      VSPV(I)
0              NSSV
8.0E-4 8.0E12  AMU, PENLTY
0.0 0.0 0.0     F0, FX, FY

```

Example 2: Flow of a viscous fluid in a slider bearing

OUTPUT FROM PROGRAM *FEM2DHT* BY J. N. REDDY

A VISCOUS INCOMPRESSIBLE FLOW IS ANALYZED

PARAMETERS OF THE FLUID FLOW PROBLEM:

Viscosity of the fluid, AMU= 0.8000E-03
 Penalty parameter, PENLTY= 0.8000E+13

CONTINUOUS SOURCE COEFFICIENTS:

Coefficient, F0= 0.0000E+00
 Coefficient, FX= 0.0000E+00
 Coefficient, FY= 0.0000E+00

***** A STEADY-STATE PROBLEM is analyzed *****
 *** A mesh of QUADRILATERALS is chosen by user ***

FINITE ELEMENT MESH INFORMATION:

Element type: 0 = Triangle; >0 = Quad.).....= 1
 Number of nodes per element, NPE= 9
 No. of primary deg. of freedom/node, NDF= 2
 Number of elements in the mesh, NEM= 6
 Number of nodes in the mesh, NNM= 35
 Number of equations to be solved, NEQ= 70
 Half bandwidth of the matrix GLK, NHBW ..= 34
 No. of specified PRIMARY variables, NSPV ..= 28

Node	x-coord.	y-coord.	Speci. primary & secondary variables (0, unspecified; >0, specified)			
			Primary DOF	Secondary DOF		

1	0.0000E+00	0.0000E+00	1	2	0	0
2	0.6000E-01	0.0000E+00	1	2	0	0
3	0.1200E+00	0.0000E+00	1	2	0	0
4	0.1800E+00	0.0000E+00	1	2	0	0
5	0.2400E+00	0.0000E+00	1	2	0	0
6	0.3000E+00	0.0000E+00	1	2	0	0
7	0.3600E+00	0.0000E+00	1	2	0	0
8	0.0000E+00	0.5000E-04	0	0	0	0
9	0.6000E-01	0.5000E-04	0	0	0	0
10	0.1200E+00	0.5000E-04	0	0	0	0
11	0.1800E+00	0.5000E-04	0	0	0	0
12	0.2400E+00	0.5000E-04	0	0	0	0
13	0.3000E+00	0.5000E-04	0	0	0	0
14	0.3600E+00	0.5000E-04	0	0	0	0
15	0.0000E+00	0.1000E-03	0	0	0	0
16	0.6000E-01	0.1000E-03	0	0	0	0
17	0.1200E+00	0.1000E-03	0	0	0	0
18	0.1800E+00	0.1000E-03	0	0	0	0
19	0.2400E+00	0.1000E-03	0	0	0	0
20	0.3000E+00	0.1000E-03	0	0	0	0
21	0.3600E+00	0.1000E-03	0	0	0	0
22	0.0000E+00	0.3500E-03	0	0	0	0

23	0.6000E-01	0.3167E-03	0	0	0	0
24	0.1200E+00	0.2833E-03	0	0	0	0
25	0.1800E+00	0.2500E-03	0	0	0	0
26	0.2400E+00	0.2167E-03	0	0	0	0
27	0.3000E+00	0.1833E-03	0	0	0	0
28	0.3600E+00	0.1500E-03	0	0	0	0
29	0.0000E+00	0.8000E-03	1	2	0	0
30	0.6000E-01	0.7333E-03	1	2	0	0
31	0.1200E+00	0.6667E-03	1	2	0	0
32	0.1800E+00	0.6000E-03	1	2	0	0
33	0.2400E+00	0.5333E-03	1	2	0	0
34	0.3000E+00	0.4667E-03	1	2	0	0
35	0.3600E+00	0.4000E-03	1	2	0	0

NUMERICAL INTEGRATION DATA:

Full quadrature (IPDF x IPDF) rule, IPDF = 3

Reduced quadrature (IPDR x IPDR), IPDR = 2

Quadrature rule used in postproc., ISTR = 2

SOLUTION:

Node	x-coord.	y-coord.	Velocity, u	Velocity, v
1	0.00000E+00	0.00000E+00	0.30000E+02	0.00000E+00
2	0.60000E-01	0.00000E+00	0.30000E+02	0.00000E+00
3	0.12000E+00	0.00000E+00	0.30000E+02	0.00000E+00
4	0.18000E+00	0.00000E+00	0.30000E+02	0.00000E+00
5	0.24000E+00	0.00000E+00	0.30000E+02	0.00000E+00
6	0.30000E+00	0.00000E+00	0.30000E+02	0.00000E+00
7	0.36000E+00	0.00000E+00	0.30000E+02	0.00000E+00
8	0.00000E+00	0.50000E-04	0.26008E+02	-0.45304E-02
9	0.60000E-01	0.50000E-04	0.25983E+02	0.22683E-02
10	0.12000E+00	0.50000E-04	0.26085E+02	-0.46442E-02
11	0.18000E+00	0.50000E-04	0.26308E+02	0.21300E-02
12	0.24000E+00	0.50000E-04	0.26982E+02	-0.50555E-02
13	0.30000E+00	0.50000E-04	0.28270E+02	0.14714E-02
14	0.36000E+00	0.50000E-04	0.31533E+02	-0.68447E-02
15	0.00000E+00	0.10000E-03	0.22292E+02	0.10406E-01
16	0.60000E-01	0.10000E-03	0.22259E+02	-0.51970E-02
17	0.12000E+00	0.10000E-03	0.22421E+02	0.10016E-01
18	0.18000E+00	0.10000E-03	0.22847E+02	-0.57272E-02
19	0.24000E+00	0.10000E-03	0.23933E+02	0.86438E-02
20	0.30000E+00	0.10000E-03	0.26237E+02	-0.81796E-02
21	0.36000E+00	0.10000E-03	0.31469E+02	0.26246E-02
22	0.00000E+00	0.35000E-03	0.87401E+01	0.14297E-01
23	0.60000E-01	0.31667E-03	0.10255E+02	-0.80107E-02
24	0.12000E+00	0.28333E-03	0.12079E+02	0.12563E-01
25	0.18000E+00	0.25000E-03	0.14350E+02	-0.99138E-02
26	0.24000E+00	0.21667E-03	0.17197E+02	0.97258E-02
27	0.30000E+00	0.18333E-03	0.20956E+02	-0.13382E-01
28	0.36000E+00	0.15000E-03	0.25982E+02	0.38986E-02
29	0.00000E+00	0.80000E-03	0.00000E+00	0.00000E+00
30	0.60000E-01	0.73333E-03	0.00000E+00	0.00000E+00

31	0.12000E+00	0.66667E-03	0.00000E+00	0.00000E+00
32	0.18000E+00	0.60000E-03	0.00000E+00	0.00000E+00
33	0.24000E+00	0.53333E-03	0.00000E+00	0.00000E+00
34	0.30000E+00	0.46667E-03	0.00000E+00	0.00000E+00
35	0.36000E+00	0.40000E-03	0.00000E+00	0.00000E+00

x-coord.	y-coord.	sigma-x	sigma-y	sigma-xy	pressure
0.2536E-01	0.2113E-04	-0.2325E+04	-0.2325E+04	-0.6464E+02	0.2325E+04
0.2536E-01	0.7887E-04	-0.2324E+04	-0.2324E+04	-0.5930E+02	0.2324E+04
0.9464E-01	0.2113E-04	-0.8666E+04	-0.8666E+04	-0.6391E+02	0.8666E+04
0.9464E-01	0.7887E-04	-0.8666E+04	-0.8666E+04	-0.5883E+02	0.8666E+04
0.1454E+00	0.2113E-04	-0.1311E+05	-0.1311E+05	-0.6235E+02	0.1311E+05
0.1454E+00	0.7887E-04	-0.1311E+05	-0.1311E+05	-0.5731E+02	0.1311E+05
0.2146E+00	0.2113E-04	-0.1771E+05	-0.1771E+05	-0.5387E+02	0.1771E+05
0.2146E+00	0.7887E-04	-0.1771E+05	-0.1771E+05	-0.5184E+02	0.1771E+05
0.2654E+00	0.2113E-04	-0.1858E+05	-0.1858E+05	-0.4341E+02	0.1858E+05
0.2654E+00	0.7887E-04	-0.1859E+05	-0.1859E+05	-0.4380E+02	0.1859E+05
0.3346E+00	0.2113E-04	-0.9608E+04	-0.9608E+04	-0.2520E+00	0.9608E+04
0.3346E+00	0.7887E-04	-0.9607E+04	-0.9607E+04	-0.1734E+02	0.9607E+04
0.2536E-01	0.1753E-03	-0.2323E+04	-0.2323E+04	-0.4786E+02	0.2323E+04
0.2536E-01	0.5632E-03	-0.2323E+04	-0.2323E+04	-0.1607E+02	0.2323E+04
0.9464E-01	0.1590E-03	-0.8666E+04	-0.8666E+04	-0.4855E+02	0.8666E+04
0.9464E-01	0.5025E-03	-0.8667E+04	-0.8667E+04	-0.2187E+02	0.8667E+04
0.1454E+00	0.1471E-03	-0.1311E+05	-0.1311E+05	-0.4832E+02	0.1311E+05
0.1454E+00	0.4580E-03	-0.1311E+05	-0.1311E+05	-0.2752E+02	0.1311E+05
0.2146E+00	0.1309E-03	-0.1771E+05	-0.1771E+05	-0.4730E+02	0.1771E+05
0.2146E+00	0.3973E-03	-0.1771E+05	-0.1771E+05	-0.3830E+02	0.1771E+05
0.2654E+00	0.1190E-03	-0.1859E+05	-0.1859E+05	-0.4532E+02	0.1859E+05
0.2654E+00	0.3529E-03	-0.1859E+05	-0.1859E+05	-0.4934E+02	0.1859E+05
0.3346E+00	0.1027E-03	-0.9605E+04	-0.9605E+04	-0.6196E+02	0.9605E+04
0.3346E+00	0.2922E-03	-0.9606E+04	-0.9606E+04	-0.7193E+02	0.9606E+04

A.5 Source Listings of Selective Subroutines

The computer implementation of the finite element formulations presented in Chapters 2 and 3 can be found in Chapter 13 of [1]. A complete listing of the FORTRAN source of FEM2D is also included in [1]. The program *FEM2DHT* consists of the following subroutines:

BOUNRY: Imposes applied boundary conditions on primary and secondary variables.

CONCT: Assembles element coefficient matrices and vectors.

ECHO: Echoes the input file.

ELKMF: Generates element coefficient matrices and vectors for rectangular (quadrilateral) elements.

ELKMFT: Generates element coefficient matrices and vectors for triangular elements.

MSH2DG: Generates finite element meshes for some general domains (see [1] for details).

MSH2DR: Generates finite element meshes for only rectangular domains (see [1] for details).

PSTPRC: Postprocessor

SHPRCT: Evaluates shape (or interpolation) functions and their global derivatives for rectangular (isoparametric) elements.

SHPTRI: Evaluates shape (or interpolation) functions and their global derivatives for triangular (isoparametric) elements.

SOLVER: Solves symmetric, banded system of algebraic equations.

TEMPORAL: Sets up coefficient matrices and column vectors for transient analysis.

To aid the reader in the computer implementation of the finite element formulations presented in Chapters 2 and 3, listings of subroutines ELKMFR, SHPRCT, and TEMPORAL from program *FEM2DHT*, which illustrate the computation of element matrices for quadrilateral elements, are included in the remaining pages of this appendix. The complete FORTRAN source of program *FEM2DHT* or *FEM2D* from [1] may be obtained for a small charge from the first author.

Reference for Additional Reading

1. J. N. Reddy, *An Introduction to the Finite Element Method*, 3rd ed., McGraw-Hill, New York (2006).

Listings of Subroutines *ELKMFR*, *SHPRCT*, and *TEMPORAL*

```

SUBROUTINE ELKMFR(NPE,NN,ITYPE)
C
C   Element calculations based on linear and quadratic rectangular
C   elements and isoparametric formulation are carried out for the
C   heat transfer and penalty model of fluid flow.
C
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON/STF/ELF(18),ELK(18,18),ELM(18,18),ELXY(9,2),ELU(18),A1,A2
COMMON/PST/A10,A1X,A1Y,A20,A2X,A2Y,A00,C0,CX,CY,F0,FX,FY,
1      AMU,PENLTY
COMMON/SHP/SF(9),GDSF(2,9),SFH(16)
COMMON/PNT/IPDF,IPDR,NIPF,NIPR
DIMENSION GAUSPT(5,5),GAUSWT(5,5)
COMMON/IO/IN,ITT
DATA GAUSPT/5*0.0D0, -0.57735027D0, 0.57735027D0, 3*0.0D0,
2 -0.77459667D0, 0.0D0, 0.77459667D0, 2*0.0D0, -0.86113631D0,
3 -0.33998104D0, 0.33998104D0, 0.86113631D0, 0.0D0, -0.90617984D0,
4 -0.53846931D0, 0.0D0, 0.53846931D0, 0.90617984D0/
      DATA GAUSWT/2.0D0, 4*0.0D0, 2*1.0D0, 3*0.0D0, 0.55555555D0,
2 0.8888888D0, 0.55555555D0, 2*0.0D0, 0.34785485D0,
3 2*0.65214515D0, 0.34785485D0, 0.0D0, 0.23692688D0,
4 0.47862867D0, 0.56888888D0, 0.47862867D0, 0.23692688D0/

```

```

NDF = NN/NPE
NET=NPE
C
C Initialize the arrays
C
DO 120 I = 1,NN
ELF(I) = 0.0
DO 120 J = 1,NN
IF(ITEM.NE.0) THEN
    ELM(I,J)= 0.0
ENDIF
120 ELK(I,J)= 0.0
C
C Do-loops on numerical (Gauss) integration begin here. Subroutine
C SHPRCT (SHaPe functions for ReCTangular elements) is called here
C
DO 200 NI = 1,IPDF
DO 200 NJ = 1,IPDF
XI = GAUSPT(NI,IPDF)
ETA = GAUSPT(NJ,IPDF)
CALL SHPRCT (NPE,XI,ETA,DET,ELXY,NDF,ITYPE)
CNST = DET*GAUSWT(NI,IPDF)*GAUSWT(NJ,IPDF)
X=0.0
Y=0.0
DO 140 I=1,NPE
X=X+ELXY(I,1)*SF(I)
140 Y=Y+ELXY(I,2)*SF(I)
C
SOURCE=F0+FX*X+FY*Y
IF(ITEM.NE.0) THEN
    CT=C0+CX*X+CY*Y
ENDIF
IF(ITYPE.LE.0) THEN
    A11=A10+A1X*X+A1Y*Y
    A22=A20+A2X*X+A2Y*Y
ENDIF
C
II=1
DO 180 I=1,NET
JJ=1
DO 160 J=1,NET
S00=SF(I)*SF(J)*CNST
S11=GDSF(1,I)*GDSF(1,J)*CNST
S22=GDSF(2,I)*GDSF(2,J)*CNST
S12=GDSF(1,I)*GDSF(2,J)*CNST
S21=GDSF(2,I)*GDSF(1,J)*CNST
IF(ITYPE.EQ.0) THEN

```

472 THE FINITE ELEMENT METHOD IN HEAT TRANSFER AND FLUID DYNAMICS

```

C
C Heat transfer and like problems (i.e. single DOF problems):_____
C
    ELK(I,J) = ELK(I,J) + A11*S11 + A22*S22 + A00*S00
    IF(ITEM.NE.0) THEN
        ELM(I,J) = ELM(I,J) + CT*S00
    ENDIF
    ELSE
C
C Viscous incompressible fluids:_____
C Compute coefficients associated with viscous terms (full integ.)
C
        ELK(II,JJ)      = ELK(II,JJ)      + AMU*(2.0*S11 + S22)
        ELK(II+1,JJ)   = ELK(II+1,JJ)   + AMU*S12
        ELK(II,JJ+1)   = ELK(II,JJ+1)   + AMU*S21
        ELK(II+1,JJ+1)= ELK(II+1,JJ+1) + AMU*(S11 + 2.0*S22)
        IF(ITEM.NE.0) THEN
            ELM(II,JJ)      = ELM(II,JJ)      + CT*S00
            ELM(II+1,JJ+1)= ELM(II+1,JJ+1) + CT*S00
        ENDIF
    ENDIF
    ENDIF
160 JJ = NDF*j+1
C
C Source of the form fx = F0 + FX*X + FY*Y is assumed
C
    L=(I-1)*NDF+1
    ELF(L) = ELF(L)+CNST*SF(I)*SOURCE
180 II = NDF*I+1
200 CONTINUE
C
    IF(ITYPE.GT.0) THEN
C
C Use reduced integration to evaluate coefficients associated with
C penalty terms for flows of viscous incompressible fluids.
C
        DO 280 NI=1,IPDR
        DO 280 NJ=1,IPDR
        XI = GAUSPT(NI,IPDR)
        ETA = GAUSPT(NJ,IPDR)
        CALL SHPRCT (NPE,XI,ETA,DET,ELXY,NDF,ITYPE)
        CNST=DET*GAUSWT(NI,IPDR)*GAUSWT(NJ,IPDR)
C
        II=1
        DO 260 I=1,NPE
        JJ = 1
        DO 240 J=1,NPE
        S11=GDSF(1,I)*GDSF(1,J)*CNST
        S22=GDSF(2,I)*GDSF(2,J)*CNST
        S12=GDSF(1,I)*GDSF(2,J)*CNST
        S21=GDSF(2,I)*GDSF(1,J)*CNST
        ELK(II,JJ) = ELK(II,JJ) + PENLTY*S11
        ELK(II+1,JJ) = ELK(II+1,JJ) + PENLTY*S21
        ELK(II,JJ+1) = ELK(II,JJ+1) + PENLTY*S12
        ELK(II+1,JJ+1)= ELK(II+1,JJ+1) + PENLTY*S22
240    JJ=NDF*j+1
260    II=NDF*I+1
280    CONTINUE
    ENDIF

```

```

IF(ITEM.NE.0) THEN
C
C Compute the coefficient matrices of the final algebraic equations
C (i.e., after time approximation) in the transient analysis:_____
C
CALL TEMPORAL(NN)
ENDIF
RETURN
END

```

```

SUBROUTINE SHPRCT(NPE,XI,ETA,DET,ELXY,NDF,ITYPE)
C _____
C
C The subroutine evaluates the interpolation functions (SF(I)) and
C their derivatives with respect to global coordinates (GDSF(I,J))
C for Lagrange linear & quadratic rectangular elements, using the
C isoparametric formulation. The subroutine also evaluates Hermite
C interpolation functions and their global derivatives using the
C subparametric formulation.
C
C SF(I).....Interpolation function for node I of the element
C DSF(J,I).....Derivative of SF(I) with respect to XI if J=1 and
C and ETA if J=2
C GDSF(J,I)....Derivative of SF(I) with respect to X if J=1 and
C and Y if J=2
C XNODE(I,J)..J-TH (J=1,2) Coordinate of node I of the element
C NP(I).....Array of element nodes (used to define SF and DSF)
C GJ(I,J).....Jacobian matrix
C GJINV(I,J)...Inverse of the jacobian matrix
C _____
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ELXY(9,2),XNODE(9,2),NP(9),DSF(2,9),GJ(2,2),GJINV(2,2)
COMMON/SHP/SF(9),GDSF(2,9)
COMMON/IO/IN,ITT
DATA XNODE/-1.0D0, 2*1.0D0, -1.0D0, 0.0D0, 1.0D0, 0.0D0, -1.0D0,
*      0.0D0, 2*-1.0D0, 2*1.0D0, -1.0D0, 0.0D0, 1.0D0, 2*0.0D0/
DATA NP/1,2,3,4,5,7,6,8,9/
C
FNC(A,B) = A*B
IF(NPE.EQ.4) THEN
C
C LINEAR Lagrange interpolation functions for FOUR-NODE element
C
DO 10 I = 1, NPE
XP = XNODE(I,1)
YP = XNODE(I,2)
XI0 = 1.0+XI*XP
ETA0=1.0+ETA*YP
SF(I) = 0.25*FNC(XI0,ETA0)
DSF(1,I)= 0.25*FNC(XP,ETA0)
10   DSF(2,I)= 0.25*FNC(YP,XI0)
ELSE
IF(NPE.EQ.8) THEN

```

```

C
C QUADRATIC Lagrange interpolation functions for EIGHT-NODE element
C
DO 20 I = 1, NPE
NI = NP(I)
XP = XNODE(NI,1)
YP = XNODE(NI,2)
XI0 = 1.0+XI*XP
ETA0 = 1.0+ETA*YP
XI1 = 1.0-XI*XI
ETA1 = 1.0-ETA*ETA
IF(I.LE.4) THEN
SF(NI) = 0.25*FNC(XI0,ETA0)*(XI*XP+ETA*YP-1.0)
DSF(1,NI) = 0.25*FNC(ETA0,XP)*(2.0*XI*XP+ETA*YP)
DSF(2,NI) = 0.25*FNC(XI0,YP)*(2.0*ETA*YP+XI*XP)
ELSE
IF(I.LE.6) THEN
SF(NI) = 0.5*FNC(XI1,ETA0)
DSF(1,NI) = -FNC(XI,ETA0)
DSF(2,NI) = 0.5*FNC(YP,XI1)
ELSE
SF(NI) = 0.5*FNC(ETA1,XI0)
DSF(1,NI) = 0.5*FNC(XP,ETA1)
DSF(2,NI) = -FNC(ETA,XI0)
ENDIF
ENDIF
20 CONTINUE
ELSE
C
C QUADRATIC Lagrange interpolation functions for NINE-NODE element
C
DO 30 I=1,NPE
NI = NP(I)
XP = XNODE(NI,1)
YP = XNODE(NI,2)
XI0 = 1.0+XI*XP
ETA0 = 1.0+ETA*YP
XI1 = 1.0-XI*XI
ETA1 = 1.0-ETA*ETA
XI2 = XP*XI
ETA2 = YP*ETA
IF(I .LE. 4) THEN
SF(NI) = 0.25*FNC(XI0,ETA0)*XI2*ETA2
DSF(1,NI)= 0.25*XP*FNC(ETA2,ETA0)*(1.0+2.0*XI2)
DSF(2,NI)= 0.25*YP*FNC(XI2,XI0)*(1.0+2.0*ETA2)
ELSE
IF(I .LE. 6) THEN
SF(NI) = 0.5*FNC(XI1,ETA0)*ETA2
DSF(1,NI) = -XI*FNC(ETA2,ETA0)
DSF(2,NI) = 0.5*FNC(XI1,YP)*(1.0+2.0*ETA2)
ELSE
IF(I .LE. 8) THEN
SF(NI) = 0.5*FNC(ETA1,XI0)*XI2
DSF(2,NI) = -ETA*FNC(XI2,XI0)
DSF(1,NI) = 0.5*FNC(ETA1,XP)*(1.0+2.0*XI2)
ELSE
SF(NI) = FNC(XI1,ETA1)
DSF(1,NI) = -2.0*XI*ETA1
DSF(2,NI) = -2.0*ETA*XI1
ENDIF
ENDIF

```

```

          ENDIF
          ENDIF
30      CONTINUE
          ENDIF
          ENDIF
C
C Compute the Jacobian matrix [GJ] and its inverse [GJINV]
C
        DO 40 I = 1,2
        DO 40 J = 1,2
        GJ(I,J) = 0.0
        DO 40 K = 1,NPE
40    GJ(I,J) = GJ(I,J) + DSF(I,K)*ELXY(K,J)
C
        DET = GJ(1,1)*GJ(2,2)-GJ(1,2)*GJ(2,1)
        GJINV(1,1) = GJ(2,2)/DET
        GJINV(2,2) = GJ(1,1)/DET
        GJINV(1,2) = -GJ(1,2)/DET
        GJINV(2,1) = -GJ(2,1)/DET
C
C Compute the derivatives of the interpolation functions with
C respect to the global coordinates (x,y): [GDSF]
C
        DO 50 I = 1,2
        DO 50 J = 1,NPE
        GDSF(I,J) = 0.0
        DO 50 K = 1, 2
50    GDSF(I,J) = GDSF(I,J) + GJINV(I,K)*DSF(K,J)
        RETURN
        END

```

SUBROUTINE TEMPORAL(NN)

C

C

C The subroutine computes the algebraic equations associated with

C the parabolic differential equations by using the *alfa-family* of

C approximations. A *constant* source is assumed.

C

IMPLICIT REAL*8(A-H,O-Z)

COMMON/STF/ELF(18),ELK(18,18),ELM(18,18),ELXY(9,2),ELU(18),A1,A2

C

C The alfa-family of time approximation for parabolic equations

C

```

        DO 20 I=1,NN
        SUM=0.0
        DO 10 J=1,NN
        SUM =SUM+(ELM(I,J)-A2*ELK(I,J))*ELU(J)
10    ELK(I,J)=ELM(I,J)+A1*ELK(I,J)
20    ELF(I) =(A1+A2)*ELF(I)+SUM
        RETURN
        END

```

Appendix B

Solution of Linear Equations

B.1 Introduction

All of the steady-state iteration methods and most of the time-dependent solution methods described throughout the text lead ultimately to the point where at least one set (matrix) of linear algebraic equations must be solved. In general terms, the system

$$\tilde{\mathbf{K}}\mathbf{U}^* = \tilde{\mathbf{F}} \quad (\text{B.1.1})$$

must be solved where $\tilde{\mathbf{K}}$ is a banded, sparse matrix that may be either symmetric or unsymmetric depending on the characteristics of the partial differential equation describing the physical problem and perhaps the nonlinear solution algorithm. The solution vector (nodal point variables) \mathbf{U}^* may contain several degrees of freedom per node or a single variable, again depending on the specific problem. Other characteristics for $\tilde{\mathbf{K}}$, such as whether or not it is positive definite, and the size of its condition number, are also very dependent on the type of finite element model and the particular solution algorithm. However, these characteristics are extremely important, since they strongly influence the type of matrix solution procedure that can be successfully used for any given problem.

A linear set of equations can be solved by either a direct or iterative method. Direct solvers, like the Gauss elimination method, are often used to solve systems of algebraic equations. They provide the solution after a fixed number of steps and are less sensitive to the conditioning of the matrix. However, the main deficiency of the direct solvers is that they require the coefficient matrix be stored in an ordered format to enhance the matrix band structure and reduce storage requirements. In recent years, the direct solvers have been refined to reduce this deficiency through innovative data management techniques (e.g., frontal solvers, skyline solvers, and others; see Carey and Oden [1]). These improvements enable users to solve moderately large systems of equations efficiently. However, they have been found to be unsuitable for solving very large systems of equations (especially in three-dimensional problems) because they demand out-of-core storage of the equations and hence require large data transfers. In addition, direct methods are difficult to organize for efficient use on multiprocessor, parallel computers and are therefore seeing reduced utilization.

The limitations on CPU time and storage requirements make the use of direct solvers not economical, even on present-day high speed and high memory computers, and using direct solvers for complex problems with more than a quarter million equations is impractical. For large systems, iterative methods are more efficient in that they require less storage and CPU time while giving comparable accuracy in the solution. This is due to the fact that the global matrix formation may be avoided, and the major operation is the matrix-vector multiplication as compared to the matrix reduction in direct methods. Another advantage of the various iterative methods is that the solution algorithm can be effectively performed in parallel on an array of processors. Iterative methods are well suited to many of the linear algebraic equation sets generated by finite element models. However, the methodology is not universal in its application. The matrix problems arising from constrained finite element models, such as incompressible flow or quasi-static electromagnetics, present significant challenges to iterative methods with the major failing being the lack of a suitable preconditioner.

In the following sections a brief outline of some methods for the solution of linear algebraic equations will be given. Further details for this topic, which is outside the major focus of this book, can be found in references such as [2–4].

B.2 Direct Methods

By *direct methods*, we mean those in which simultaneous linear algebraic equations are solved exactly, assuming negligible computational round-off error, by successive elimination of variables and back substitution. The Gauss elimination method is a direct technique [2–4], and frontal [5] and skyline [6] solution methods are examples of direct solution methods that use the Gauss elimination technique efficiently. Direct solution methods involve a fixed number of steps to determine the solution. The direct techniques are useful when the number of equations involved is not too large. The number of operations (i.e., multiplication, division, etc.) for Gauss elimination is of the order $n^3/3 + O(n^2)$, where n denotes the number of operations.

For a comparable direct method, we refer to the *frontal* solution procedure [5]. Because it is faster than most direct solvers, it requires less core space as long as active variables can be kept in the core, and it allows for partial pivoting. An additional advantage is that no stringent node numbering scheme is needed, though a judicious element numbering helps to minimize the front width. Details of the frontal method will not be given here as it is a sophisticated implementation of the simple Gauss elimination method which is adequately studied in numerous texts. Implementation issues for the frontal method can be found in [5].

Due to the fact that the approximation functions are defined only within an element, the coefficient matrix in the finite element method is banded, i.e., $K_{ij} = 0$ for $j > i + m_b$, where m_b is the half bandwidth of the matrix $[K]$. This greatly reduces the number of operations, if we make note of the fact that elements outside the bandwidth are zero. Of course, the bandwidth size depends on the global node numbering. The skyline technique is one in which bandedness of the finite element equations is exploited by storing the row number m_j of the first nonzero element in column j . The variables $m_i, i = 1, 2, \dots, n$, define the skyline of the matrix. For additional details see Bathe [6].

B.3 Iterative Methods

B.3.1 General Comments

Among the various iterative methods, the Conjugate Gradient (CG) method [7] is most widely used because it is a finite step method (i.e., apart from round-off errors, the solution is achieved in a fixed number of iterations) and it can be used to determine the inverse. However, the number of iterations required depends on the condition number of the coefficient matrix (the condition number of the coefficient matrix is the ratio of the largest to smallest eigenvalue). The convergence of the conjugate gradient method, and iterative methods in general, can be improved by preconditioning and/or scaling the equations [8–10].

The limitations on storage can be overcome by solving the equations at the element level, i.e., using the Gauss–Seidel iteration idea for the set of variables associated with the element. This approach avoids assembly of element matrices to form the global coefficient matrix. This idea of using the element-by-element data structure of the coefficient matrix was first pointed out by Fox and Stanton [11] and Fried [12–14]. The phrase *element-by-element* refers to a particular data structure for finite element techniques wherein information is stored and maintained at the element level rather than assembled into a global data structure. In this method the matrix–vector multiplications are carried out at the element level and the assembly is carried out on the resultant vector. This idea proves to be very attractive when solving large problems, because the matrix–vector multiplication can be done in parallel on a series of processors. Another advantage of this method is that the resultant savings in storage, compared to direct solvers, allows solution of large problems on small computers. A review of the literature on element-by-element algorithms is presented in [15], and they have been investigated by numerous investigators [16–34].

In summary, for iterative solution methods the advantages of the element-by-element data structure over assembling the global coefficient matrix are

1. the need for formation and storage of a global matrix is eliminated, and therefore the total storage and computational costs are low,
2. the amount of storage is independent of the node numbering and mesh topology and depends on the number and type of elements in the mesh, and
3. the element-by-element solution algorithms can be vectorized for efficient use on supercomputers.

The major disadvantage of the element-by-element data structure is the limited number of preconditioners that can be formulated from the unassembled matrices. This becomes of critical importance when the linear system is not well-conditioned as in the mixed method, incompressible flow model.

B.3.2 Solution Algorithms

In this section, we review three iterative solvers from [15] that are applicable to nonsymmetric, positive definite equation systems that are typical of isothermal flow algorithms. The three iterative solution schemes used here are the Biorthogonal Conjugate Gradient method [9], the Lanczos ORTHORES [9], and the GMRES [31].

The conjugate gradient method for solving a system of equations can be interpreted as the search for the minimum of the energy E of the system. The energy

of the system is a minimum when the residual vector $\mathbf{r} = \tilde{\mathbf{F}} - \tilde{\mathbf{K}}\mathbf{U}^*$ vanishes. The algorithm for the biorthogonal conjugate gradient method (also known as two-term form of the steepest descent method, Lanczos/ORTHOMIN) for unsymmetric systems of equations [9,15] is given in Table B.3.1, and the steps involved in the Lanczos ORTHORES solution algorithm [9,15] are given in Table B.3.2.

Table B.3.1: Steps involved in using biorthogonal conjugate gradient method (Lanczos/ORTHOMIN solver).

Repeat the following steps for each nonlinear iteration:

I. *Initial Calculations*

- (1) Form the element stiffness matrix $\tilde{\mathbf{K}}^e$ and force vector $\tilde{\mathbf{F}}^e$.
- (2) Apply essential and/or natural boundary conditions, and modify $\tilde{\mathbf{K}}^e$ and $\tilde{\mathbf{F}}^e$.
- (3) Store the element matrices in $\bar{\mathbf{A}}$ (whose dimensions are nem, neleq, neleq).†
- (4) Store the inverse of the diagonal terms of the global system in \mathbf{W} ($W_{ii} = \sum_{e=1}^{\text{nem}} \tilde{K}_{ii}^{-1}$).
- (5) Assemble the global force vector.

II. *Preconditioning*

Form the preconditioned system of equations

$$\bar{\mathbf{K}}\bar{\mathbf{U}} = \bar{\mathbf{F}}; \quad \bar{\mathbf{K}} = \mathbf{W}^{-1/2}\tilde{\mathbf{K}}\mathbf{W}^{-1/2}, \quad \bar{\mathbf{U}} = \mathbf{W}^{1/2}\mathbf{U}^*, \quad \bar{\mathbf{F}} = \mathbf{W}^{-1/2}\tilde{\mathbf{F}}$$

III. *Lanczos ORTHOMIN Algorithm*

- (1) For known initial solution vector $\bar{\mathbf{U}}^0$, compute:

$$\begin{aligned} \mathbf{r}^0 &= \bar{\mathbf{F}} - \bar{\mathbf{K}}\bar{\mathbf{U}}^0, \quad \mathbf{P}^0 = \mathbf{r}^0, \quad \tilde{\mathbf{r}}^0 = \tilde{\mathbf{P}}^0 = \mathbf{r}^0, \\ \alpha_0 &= 0, \quad \lambda_0 = \frac{(\mathbf{r}^0, \tilde{\mathbf{r}}^0)}{(\bar{\mathbf{K}}\mathbf{P}^0, \tilde{\mathbf{r}}^0)}, \quad \bar{\mathbf{U}}^1 = \bar{\mathbf{U}}^0 + \lambda_0 \mathbf{P}^0 \end{aligned}$$

- (2) For each ORTHOMIN iteration $m = 1, 2, 3, \dots$, compute[§]:

$$\begin{aligned} \lambda_m &= \frac{(\mathbf{r}^m, \tilde{\mathbf{r}}^m)}{(\bar{\mathbf{K}}\mathbf{P}^m, \tilde{\mathbf{r}}^m)}, \quad \alpha_m = \frac{(\mathbf{r}^m, \tilde{\mathbf{r}}^m)}{(\mathbf{r}^{m-1}, \tilde{\mathbf{r}}^{m-1})}, \\ \mathbf{P}^m &= \mathbf{r}^m + \alpha_m \mathbf{P}^{m-1}, \quad \tilde{\mathbf{P}}^m = \tilde{\mathbf{r}} + \alpha_m \tilde{\mathbf{P}}^{m-1}, \\ \mathbf{r}^{m+1} &= \mathbf{r}^m - \lambda_m \bar{\mathbf{K}}\mathbf{P}^m, \quad \tilde{\mathbf{r}}^{m+1} = \tilde{\mathbf{r}}^m - \lambda_m \bar{\mathbf{K}}^T \tilde{\mathbf{P}}^{m-1}, \\ \bar{\mathbf{U}}^{m+1} &= \bar{\mathbf{U}}^m + \lambda_m \mathbf{P}^m \end{aligned}$$

- (3) Convergence criterion: $\|\bar{\mathbf{U}}^{m+1}\|/\|\mathbf{r}^0\| \leq 10^{-6}$
 - (4) If convergence criterion is satisfied $\bar{\mathbf{U}}^* = \mathbf{W}^{-1/2}\bar{\mathbf{U}}^{m+1}$
-

† nem = number of elements in the finite element mesh, neleq = number of element equations.

§ $(a, b) = \sum a_i b_i$.

The third iterative solver uses the GMRES solution algorithm. For an approximate solution of the form $\bar{\mathbf{U}}_0 + \mathbf{z}$, where $\bar{\mathbf{U}}_0$ is the initial guess vector and \mathbf{z} is a member of the Krylov space \mathcal{K} of dimension k , the GMRES algorithm determines the vector \mathbf{z} such that $\| \tilde{\mathbf{F}} - \tilde{\mathbf{K}}(\bar{\mathbf{U}}_0 + \mathbf{z}) \|$ is minimized, where $\| \cdot \|$ denotes the L_2 -norm. The Krylov space is given by $\mathcal{K} = \text{span}\{\bar{\mathbf{U}}_0, \tilde{\mathbf{K}}\bar{\mathbf{U}}_0, \tilde{\mathbf{K}}^2\bar{\mathbf{U}}_0, \dots, \tilde{\mathbf{K}}^{k-1}\bar{\mathbf{U}}_0\}$. Therefore, when solving large systems of equations, as the value of k increases, the amount of storage required also increases. This drawback can be overcome by employing the GMRES algorithm iteratively by using a smaller value for k and restarting the algorithm after every k steps. The restart version of the GMRES algorithm [31,15] is explained in Table B.3.3.

Table B.3.2: Steps involved in using Lanczos/ORTHORES solver.

Repeat the following steps for each nonlinear iteration:

I. Initial Calculations

- (1) Form the element stiffness matrix $\tilde{\mathbf{K}}^e$ and force vector $\tilde{\mathbf{F}}^e$.
- (2) Apply the boundary conditions, and modify $\tilde{\mathbf{K}}^e$ and $\tilde{\mathbf{F}}^e$.
- (3) Store the element matrices in $\bar{\mathbf{A}}$ (whose dimensions are nem, neleq, neleq).
- (4) Store the inverse of the diagonal terms of the global system in \mathbf{W} ($W_{ii} = \sum_{e=1}^{\text{nem}} \tilde{K}_{ii}^{-1}$).
- (5) Assemble the global force vector.

II. Preconditioning

Form the preconditioned system of equations

$$\bar{\mathbf{K}}\bar{\mathbf{U}} = \bar{\mathbf{F}}, \quad \bar{\mathbf{K}} = \mathbf{W}^{-1/2}\tilde{\mathbf{K}}\mathbf{W}^{-1/2}, \quad \bar{\mathbf{U}} = \mathbf{W}^{1/2}\tilde{\mathbf{U}}, \quad \bar{\mathbf{F}} = \mathbf{W}^{-1/2}\tilde{\mathbf{F}}$$

III. Lanczos ORTHORES Algorithm

- (1) For known initial solution vector $\bar{\mathbf{U}}^0$, compute: $\mathbf{r}^0 = \bar{\mathbf{F}} - \bar{\mathbf{K}}\bar{\mathbf{U}}^0$
- (2) For each ORTHORES iteration $m = 0, 1, 2, \dots$, compute ($\tilde{\mathbf{r}} = \mathbf{r}_0, \lambda^0 = 0$):

$$\begin{aligned} \lambda^{m+1} &= \frac{(\mathbf{r}^m, \tilde{\mathbf{r}}^m)}{(\bar{\mathbf{K}}\mathbf{r}^m, \tilde{\mathbf{r}}^m)} \\ \beta^{m+1} &= \begin{cases} 1 & ; \text{ if } m = 0 \\ \left[1 - \frac{\lambda^{m+1}}{\lambda^m} \frac{(\mathbf{r}^m, \tilde{\mathbf{r}}^m)}{(\mathbf{r}^{m-1}, \tilde{\mathbf{r}}^{m-1})} \frac{1}{\beta_m} \right]^{-1} & ; \text{ if } m \geq 1 \end{cases} \\ \mathbf{r}^{m+1} &= \beta^{m+1} \left(\mathbf{r}^m - \lambda^{m+1} \bar{\mathbf{K}}\mathbf{r}^m \right) + (1 - \beta^{m+1}) \mathbf{r}^{m-1} \\ \tilde{\mathbf{r}}^{m+1} &= \beta^{m+1} \left(\tilde{\mathbf{r}}^m - \lambda^{m+1} \bar{\mathbf{K}}\tilde{\mathbf{r}}^m \right) + (1 - \beta^{m+1}) \tilde{\mathbf{r}}^{m-1} \\ \bar{\mathbf{U}}^{m+1} &= \beta^{m+1} \left(\bar{\mathbf{U}}^m + \lambda^{m+1} \mathbf{r}^m \right) + (1 - \beta^{m+1}) \bar{\mathbf{U}}^{m-1} \end{aligned}$$

- (3) Convergence criterion: $\|\bar{\mathbf{U}}^{m+1}\|/\|\mathbf{r}^0\| \leq 10^{-6}$
 - (4) If convergence criterion is satisfied: $\mathbf{U}^* = \mathbf{W}^{-1/2}\bar{\mathbf{U}}^{m+1}$
-

Table B.3.3: Steps involved in using the GMRES solver.

Repeat the following steps for each nonlinear iteration:

I. *Initial Calculations*

- (1) Form the element stiffness matrix $\tilde{\mathbf{K}}^e$ and force vector $\tilde{\mathbf{F}}^e$.
- (2) Apply the boundary conditions, and modify $\tilde{\mathbf{K}}^e$ and $\tilde{\mathbf{F}}^e$.
- (3) Store the element matrices in $\bar{\mathbf{A}}$ (whose dimensions are nem, neleq, neleq).
- (4) Store the inverse of the diagonal terms of the global system in \mathbf{W}
 $(W_{ii} = \sum_{e=1}^{nem} \hat{K}_{ii}^{-1})$.
- (5) Assemble the global force vector.

II. *Preconditioning* Form the preconditioned system of equations

$$\bar{\mathbf{K}}\bar{\mathbf{U}} = \bar{\mathbf{F}}, \quad \bar{\mathbf{K}} = \mathbf{W}^{-1/2}\tilde{\mathbf{K}}\mathbf{W}^{-1/2}, \quad \bar{\mathbf{U}} = \mathbf{W}^{1/2}\tilde{\mathbf{U}}, \quad \bar{\mathbf{F}} = \mathbf{W}^{-1/2}\tilde{\mathbf{F}}$$

III. *GMRES Algorithm*

- (1) *Start* Choose $\bar{\mathbf{U}}^0$ and compute $\mathbf{r}^0 = \bar{\mathbf{F}} - \bar{\mathbf{K}}\bar{\mathbf{U}}^0$, and $\mathbf{v}^1 = \bar{\mathbf{U}}^0 / \| \bar{\mathbf{U}}^0 \|$
- (2) *Iterate* For $j = 1, 2, \dots, k$, do: $h_{i,j} = (\bar{\mathbf{K}}v_j, v_i)$, $i = 1, 2, \dots, j$, $\hat{v}_{j+1} = \bar{\mathbf{K}}v_j - \sum_{i=1}^j h_{i,j}v_i$, $h_{j+1,j} = \| \hat{v}_{j+1} \|$, and $v_{j+1} = \hat{v}_{j+1}/h_{j+1,j}$
- (3) *Form approximate solution* $\bar{\mathbf{U}}^k = \bar{\mathbf{U}}^0 + \mathbf{V}\mathbf{y}$, where \mathbf{y} minimizes $\| \mathbf{e} - \bar{\mathbf{H}}\mathbf{y} \|$, $\mathbf{y} \in \mathbf{R}^k$.
- (4) *Restart* Compute $\mathbf{r}^k = \bar{\mathbf{F}} - \bar{\mathbf{K}}\bar{\mathbf{U}}^k$; check convergence; if satisfied stop; otherwise, compute $\bar{\mathbf{U}}^0 := \bar{\mathbf{U}}^k$, $\mathbf{v}_1 := \bar{\mathbf{U}}^k / \| \bar{\mathbf{U}}^k \|$, and go to step 2.
- (5) *Convergence criterion* $\| \bar{\mathbf{U}}^{m+1} \| / \| \mathbf{r}^0 \| \leq 10^{-6}$
- (6) *If convergence criterion is satisfied* $\mathbf{U}^* = \mathbf{W}^{-1/2}\bar{\mathbf{U}}$

where \mathbf{V} is a $N \times k$ matrix whose columns are 1–2 orthonormal basis vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$, $\mathbf{e} = \{\| \bar{\mathbf{U}}^0 \|, 0, \dots, 0\}$, and $\bar{\mathbf{H}}$ is the upper $k \times k$ Hessenberg matrix whose entries are the scalars $h_{i,j}$. When using the restart version of the GMRES algorithm, the total number of iteration m can be computed from the number of restarts and the dimension of k .

The presence of a penalty matrix in the global coefficient matrix of the penalty model spoils the condition number. This results in slow convergence when using iterative solvers. However, the convergence of the iterative solvers can be improved by preconditioning the system. In [15], the system of equations is transformed using diagonal scaling matrix (Jacobi/diagonal preconditioning). Accordingly, the system of equations

$$\tilde{\mathbf{K}}\mathbf{U}^* = \tilde{\mathbf{F}} \tag{B.3.1}$$

becomes

$$\bar{\mathbf{K}}\bar{\mathbf{U}} = \bar{\mathbf{F}} \tag{B.3.2}$$

$$\bar{\mathbf{K}} = \mathbf{W}^{-1/2}\tilde{\mathbf{K}}\mathbf{W}^{-1/2}; \quad \bar{\mathbf{U}} = \mathbf{W}^{1/2}\mathbf{U}^*; \quad \bar{\mathbf{F}} = \mathbf{W}^{-1/2}\tilde{\mathbf{F}} \tag{B.3.3}$$

where $\mathbf{W}_{ii} = \tilde{\mathbf{K}}_{ii}^{-1}$ is a diagonal matrix. During the matrix multiplication, the element-by-element data structure is exploited and the multiplications are carried out at the element level and the residuals are then assembled to form the global vector. The diagonal terms are always positive because of the viscous and penalty terms.

References for Additional Reading

1. G. F. Carey and J. T. Oden, *Finite Elements: Computational Aspects*, Prentice Hall, Englewood Cliffs, New Jersey (1984).
2. K. E. Atkinson, *An Introduction to Numerical Analysis*, Wiley, New York (1978).
3. B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*, Wiley, New York (1969).
4. D. K. Fadeev and V. N. Fadeeva, *Computational Methods of Linear Algebra*, Freeman, San Francisco, California (1963).
5. P. Hood, “Frontal Solution Program for Unsymmetric Matrices,” *International Journal for Numerical Methods in Engineering*, **10**, 379–399 (1976); also see **10**, 1055 (1976) for a correction.
6. K. J. Bathe, *Finite Element Procedures*, Prentice Hall, Englewood Cliffs, New Jersey (1998).
7. M. R. Hestenes and E. L. Stiefel, “Methods of Conjugate Gradients for Solving Linear Systems,” *National Bureau of Standards Journal of Research*, **49**, 409–436 (1952).
8. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Second Edition, The Johns Hopkins University Press, Baltimore, Maryland (1989).
9. K. C. Jea and D. M. Young, “On the Simplification of Generalized Conjugate-Gradient Methods for Nonsymmetrizable Linear Systems,” *Linear Algebra Applications*, **52**, 399–417 (1983).
10. D. J. Evans, “Use of Preconditioning in Iterative Methods for Solving Linear Equations with Symmetric Positive Definite Matrices,” *Computer Journal*, **4**, 73–78 (1961).
11. R. L. Fox and E. L. Stanton, “Developments in Structural Analysis by Direct Energy Minimization,” *AIAA Journal*, **6**, 1036–1042 (1968).
12. I. Fried, “Gradient Methods for Finite Element Eigenproblems,” *AIAA Journal*, **7**, 739–741 (1969).
13. I. Fried, “More on Generalized Iterative Methods in Finite Element Analysis,” *AIAA Journal*, **7**, 565–567 (1969).
14. I. Fried, “A Gradient Computational Procedure for the Solution of Large Problems Arising from the Finite Element Discretization Method,” *International Journal for Numerical Methods in Engineering*, **2**, 477–494 (1970).
15. M. P. Reddy, J. N. Reddy and H. U. Akay, “Penalty Finite Element Analysis of Incompressible Flows Using Element by Element Solution Algorithms,” *Computer Methods in Applied Mechanics and Engineering*, **100**, 169–205 (1992).
16. A. J. Wathen, “An Analysis of Some Element-By-Element Techniques,” *Computer Methods in Applied Mechanics and Engineering*, **74**, 271–287 (1989).

17. T. J. R. Hughes, I. Levit, and J. Winget, "An Element-By-Element Implicit Algorithm for Heat Conduction," *Journal of the Engineering Mechanics Division, ASCE*, **109**(2), 576–585 (1983).
18. J. M. Winget and T. J. R. Hughes, "Solution Algorithms for Nonlinear Transient Heat Conduction Analysis Employing Element-By-Element Iterative Strategies," *Computer Methods in Applied Mechanics and Engineering*, **52**, 711–815 (1985).
19. G. F. Carey and B. Jiang, "Element-By-Element Linear and Nonlinear Solution Schemes," *Communications in Applied Numerical Methods*, **2**, 145–153 (1986).
20. J. L. Hayes and P. Devloo, "An Element-By-Element Block Iterative Method for Large Non-Linear Problems," in *Innovative Methods for Nonlinear Behavior*, W. K. Liu et al. (eds.), Pineridge Press, Swansea, U.K., 51–62 (1985).
21. E. K. Buratynski, "An Element-By-Element Method for Heat Conduction CAE Including Composite Problems," *International Journal for Numerical Methods in Engineering*, **26**, 199–215 (1988).
22. I. Levit, "Element By Element Solvers of Order N," *Computers & Structures*, **27**(3), 357–360 (1987).
23. A. de La Bourdinaye, "The Element By Element Method as a Preconditioner for Linear Systems Coming from Finite Element Models," *International Journal of Supercomputer Applications*, **3**, 60–68 (1989).
24. K. V. G. Prakhya, "Some Conjugate Gradient Methods for Symmetric and Nonsymmetric Systems," *Communications in Applied Numerical Methods*, **4**, 531–539 (1988).
25. G. F. Carey, E. Barragy, R. McLay, and M. Sharma, "Element-by-Element Vector and Parallel Computations," *Communications in Applied Numerical Methods*, **4**, 299–307 (1988).
26. L. J. Hayes and P. Devloo, "A Vectorized Version of a Sparse Matrix-Vector Multiplication," *International Journal for Numerical Methods in Engineering*, **23**, 1043–1056 (1986).
27. F. Shakib, T. J. R. Hughes, and Z. Johan, "A Multi-Element Group Preconditioned GMRES Algorithm for Nonsymmetric Systems Arising in Finite Element Analysis," *Computer Methods in Applied Mechanics and Engineering*, **75**, 415–456 (1989).
28. T. E. Tezduyar and J. Liou, "Element-by-Element and Implicit-Explicit Finite Element Formulations for Computational Fluid Dynamics," *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux (eds.), SIAM, Philadelphia, Pennsylvania, 281–300 (1988).
29. T. E. Tezduyar and J. Liou, "Grouped Element-By-Element Iteration Schemes for Incompressible Flow Computations," *Computational Physics Communications*, **53**, 441–453 (1989).
30. J. Liou and T. E. Tezduyar, "A Clustered Element-By-Element Iteration Method for Finite Element Computations," University of Minnesota Supercomputer Institute Research Report, 90/116 (1990).
31. Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computations*, **7**(3), 856–869 (1986).
32. J. A. Mitchell and J. N. Reddy, "A Multilevel Hierarchical Preconditioner for Thin Elastic Solids," *International Journal for Numerical Methods in Engineering*, **43**, 1383–1400 (1998).
33. J. A. Mitchell and J. N. Reddy, "A High Performance Iterative Solution Procedure for the Analysis of Structural Problems," *Journal of High Performance Computing*, **5**(1), 3–13 (1999).
34. J. A. Mitchell and J. N. Reddy, "A Hierarchical Iterative Procedure for the Analysis of Composite Laminates," *Computer Methods in Applied Mechanics and Engineering*, **181**, 237–260 (2000).

Appendix C

Fixed Point Methods and Contraction Mappings

C.1 Fixed Point Theorem

The discussion in this section will focus on the use of fixed point iteration schemes for the solution of the system of nonlinear equations of the form

$$\mathcal{R}(\mathbf{x}) = \mathbf{0} \quad (\text{C.1.1})$$

or its fixed point form

$$\mathbf{x} = \mathcal{G}(\mathbf{x}) \equiv \mathbf{x} - \mathcal{R}(\mathbf{x}) \quad (\text{C.1.2})$$

where \mathbf{x} represents a vector of unknowns. Solutions of Eq. (C.1.2) are called *fixed points* of the mapping \mathcal{G} , because they are unchanged by the operation of \mathcal{G} . The form of (C.1.2) immediately suggests the following iteration scheme [1]:

$$\mathbf{x}^{n+1} = \mathcal{G}(\mathbf{x}^n) \quad (\text{C.1.3})$$

where $n = 0, 1, \dots$. Equation (C.1.3) is the method of successive substitutions known as the Picard iteration method. The convergence of the method in (C.1.3) is guaranteed by the *Banach Fixed Point Theorem*, which gives the sufficient conditions for the existence and uniqueness of solutions (see Reddy [1], pp. 202–208). Before we present the theorem, it is necessary to define a contraction mapping.

An operator \mathcal{G} is termed a *contraction mapping* if there exists a positive number λ , $0 < \lambda < 1$, such that

$$\| \mathcal{G}(\mathbf{x}) - \mathcal{G}(\mathbf{y}) \| \leq \lambda \| \mathbf{x} - \mathbf{y} \| \quad (\text{C.1.4})$$

holds for all \mathbf{x}, \mathbf{y} in the closed ball $\mathcal{N} = \{ \mathbf{x} : \| \mathbf{x} - \mathbf{x}_0 \| \leq r \}$ of a normed space \mathcal{V} , with center at \mathbf{x}_0 and radius r . Here λ denotes the contraction factor. The notation $\| \cdot \|$ is an appropriate norm for the operator in \mathcal{N} (see [1]). It is clear from Eq. (C.1.4) that a contraction mapping “contracts distances”: the distance between the images $\mathcal{G}(\mathbf{x})$ and $\mathcal{G}(\mathbf{y})$ is smaller by a scale factor of λ than the distance between the elements \mathbf{x} and \mathbf{y} .

Theorem C.1: Let $\mathcal{G}(\mathbf{x})$ be a contraction mapping on a closed subset \mathcal{N} of a closed normed space (or a Banach space) \mathcal{V} , with a contraction factor λ and let \mathbf{x}_0 be such that

$$\frac{1}{1 - \lambda} \| \mathcal{G}(\mathbf{x}_0) - \mathbf{x}_0 \| = r_0 \leq r$$

Then

1. The sequence defined by $\mathbf{x}^{n+1} = \mathcal{G}(\mathbf{x}^n)$ converges to a point \mathbf{x}^* in \mathcal{N} .
2. \mathbf{x}^* is a fixed point of the operator \mathcal{G} .
3. \mathbf{x}^* is the unique fixed point of \mathcal{G} in \mathcal{N} .

The proof of this theorem and a detailed background on nonlinear solution methods are available in [1,2].

Based on the above theorem, a further result can be obtained if a few restrictions are placed on \mathcal{G} . Assume that (C.1.3) has a solution \mathbf{x}^* and that \mathcal{G} has a continuous Jacobian that satisfies

$$\| \mathcal{G}'(\mathbf{x}) \|_{\infty} \leq \lambda < 1 \quad (\text{C.1.5})$$

where $\| \cdot \|_{\infty}$ denotes the *sup-norm* (“sup” stands for supremum). Then it can be shown that for any initial estimate \mathbf{x}^0 satisfying

$$\| \mathbf{x}^0 - \mathbf{x}^* \|_{\infty} \leq r \quad (\text{C.1.6})$$

the iterative scheme in Eq. (C.1.3) will converge to the unique solution \mathbf{x}^* . The conditions (C.1.5) and (C.1.6) basically state that the norm of the Jacobian of the iteration operator must be less than unity and the initial estimate for the solution must be within the ball of contraction, if the fixed point scheme is to converge. The quantity r is also referred to as the radius of convergence of the iteration scheme.

Generally, it is quite difficult to prove that a given operator \mathcal{G} is a contraction mapping, especially for the systems generated by finite element applications. It is somewhat easier to use (C.1.5) and estimate the norm of \mathcal{G}' to test for a convergent process. However, the real benefit from the ideas of fixed point operators is derived from the iteration scheme in (C.1.3) and its generalization,

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \alpha \mathbf{A}(\mathbf{x}^n) \mathcal{F}(\mathbf{x}^n) \quad (\text{C.1.7})$$

where α is a constant and \mathbf{A} is a nonsingular matrix function of \mathbf{x} . In the following, a few of the basic iteration schemes will be defined based on the definition of \mathbf{A} .

C.2 Chord Method

The simplest choice for \mathbf{A} is a constant which defines the parallel chord method. For a one degree-of-freedom problem, this becomes

$$x^{n+1} = x^n - af(x^n) \quad (\text{C.2.1})$$

which is shown graphically in Figure C.2.1. In the multi-dimensional case, the line with slope $\frac{1}{a}$ is replaced by a hyperplane. The behavior of this method depends strongly on the form of \mathbf{A} . One possibility is to set $\mathbf{A} = \alpha \mathbf{I}$ (\mathbf{I} is the identity matrix) in which case each component of \mathbf{x} is corrected in proportion to α . If $\mathbf{A} = \alpha^T \mathbf{I}$ where α is a vector, then each component of \mathbf{x} can be corrected individually. The problem remains, however, as to how to select α .

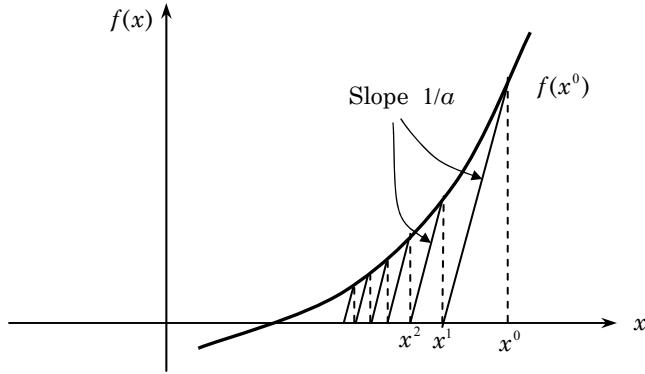


Figure C.2.1: Illustration of the chord method for one degree-of-freedom problem.

C.3 Newton's Method

Extending the idea that \mathbf{A} is in some sense an inverse “slope,” it is appropriate to relate \mathbf{A} to the derivative of \mathcal{F} . Therefore, let

$$\mathbf{A} = \left[\frac{\partial \mathcal{F}}{\partial \mathbf{x}} \right]^{-1} = \mathbf{J}^{-1} \quad (\text{C.3.1})$$

in which case Eq. (C.1.7) becomes

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \alpha \mathbf{J}^{-1}(\mathbf{x}^0) \mathcal{F}(\mathbf{x}^n) \quad (\text{C.3.2})$$

where the Jacobian is evaluated at \mathbf{x}^0 . For the one degree-of-freedom case, this becomes ($\alpha = 1$)

$$x^{n+1} = x^n - \frac{1}{f'(x^0)} f(x^n) \quad (\text{C.3.3})$$

which is illustrated in Figure C.3.1. This is recognized as a modified Newton's method since the Jacobian is not updated at each iteration.

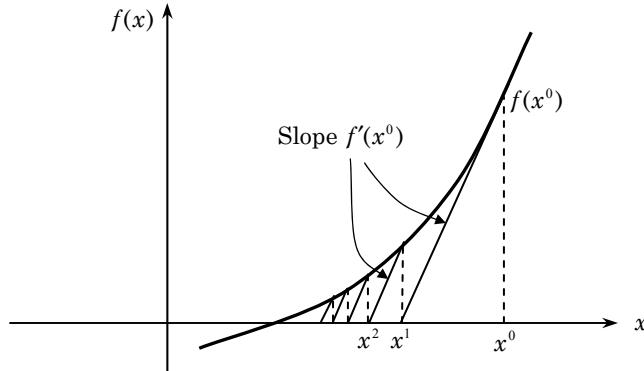


Figure C.3.1: Illustration of modified Newton's method.

C.4 The Newton–Raphson Method

The modified Newton's scheme suggests a rational manner in which to select \mathbf{A} . However, its convergence is little better than the chord method due to the use of a constant Jacobian. Allowing the Jacobian to change at each iteration produces the standard Newton (or Newton–Raphson) procedure

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \alpha \mathbf{J}^{-1}(\mathbf{x}^n) \mathcal{F}(\mathbf{x}^n) \quad (\text{C.4.1})$$

This is shown graphically in Figure C.4.1 for a one-component system. Note that if $\alpha \neq 1$ a damped Newton scheme is produced.

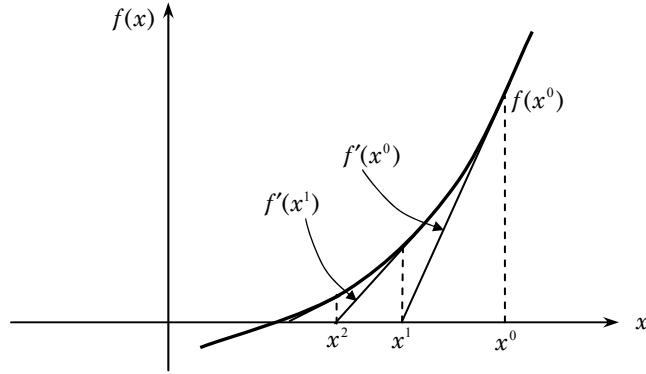


Figure C.4.1: Illustration of Newton's method.

C.5 Descent Methods

Descent methods reinterpret the general algorithm in (C.1.7) as

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \alpha \mathbf{D}^n \quad (\text{C.5.1})$$

where \mathbf{D} is a vector that determines the direction of the correction and α controls the magnitude of the correction. In addition, rather than directly finding the solution to (C.1.1), descent methods work by trying to minimize the function

$$R(\mathbf{x}) = \mathcal{F}^T(\mathbf{x}) \mathcal{F}(\mathbf{x}) \quad (\text{C.5.2})$$

The simplest descent methods are the univariant or relaxation schemes in which \mathbf{D} is selected so that only one component of \mathbf{x} is corrected at each iteration. A better approach involves the gradients of R , such as the method of steepest descent. In this case

$$\mathbf{D}^n = \mathbf{g}^n = \nabla R(\mathbf{x}^n) \quad (\text{C.5.3})$$

The method of steepest descent has a linear rate of convergence.

Finally, the conjugate gradient method is a sequential method in which the correction vectors are computed from the recursion relation

$$\mathbf{D}^n = \nabla R(\mathbf{x}^n) + \beta^{n-1} \mathbf{D}^{n-1} \quad (\text{C.5.4})$$

Here the new correction vector is a combination of the gradient at \mathbf{x}^n plus the old correction direction. The parameter β is defined to insure that the new and old correction vectors are properly orthogonal (conjugate).

Full details for all of the above algorithms and the general concepts behind nonlinear solution methods can be found in [3–6].

References for Additional Reading

1. J. N. Reddy, *Applied Functional Analysis and Variational Methods in Engineering*, McGraw-Hill, New York (1986); reprinted by Krieger Publishers, Melbourne, Florida (1991).
2. M. R. Hestenes, *Optimization Theory: The Finite Dimensional Case*, John Wiley & Sons, New York (1975).
3. E. B. Becker, G. F. Carey, and J. T. Oden, *Finite Elements, An Introduction*, Vol. I, Prentice Hall, Englewood Cliffs, New Jersey (1981).
4. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).
5. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, UK (1987).
6. J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis*, Oxford University Press, Oxford, UK (2004).

The Finite Element Method in Heat Transfer and Fluid Dynamics

As computational fluid dynamics (CFD) and computational heat transfer (CHT) evolve and become increasingly important in standard engineering design and analysis practice, users require a solid understanding of mechanics and numerical methods to make optimal use of available software. **The Finite Element Method in Heat Transfer and Fluid Dynamics, Third Edition** illustrates what a user must know to ensure the optimal application of computational procedures—particularly the finite element method (FEM)—to important problems associated with heat conduction, incompressible viscous flows, and CHT.

This book follows the tradition of the bestselling previous editions, noted for their concise explanation and powerful presentation of useful methodology tailored for use in simulating CFD and CHT. The authors update research developments while retaining the previous editions' key material and popular style in regard to text organization, equation numbering, references, and symbols.

This updated third edition features new or extended coverage of

- Coupled problems and parallel processing
- Mathematical preliminaries and low-speed compressible flows
- Mode superposition methods and a more detailed account of radiation solution methods
- Variational multi-scale methods (VMM) and least-squares finite element models (LSFEM)
- Application of the finite element method to non-isothermal flows
- Formulation of low-speed, compressible flows

With its presentation of realistic, applied examples of FEM in thermal and fluid design analysis, this proven masterwork is an invaluable tool for mastering basic methodology, competently using existing simulation software, and developing simpler special-purpose computer codes. It remains one of the very best resources for understanding numerical methods used in the study of fluid mechanics and heat transfer phenomena.



CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

85980

ISBN: 978-1-4200-8598-3
90000



9 781420 085983