Distribution
Category **UC-32**

SAND86-1816
Unlimited Release
Printed April 1987
Revised February 2019

NACHOS II -
A FINITE ELEMENT COMPUTER PROGRAM FOR INCOMPRESSIBLE FLOW PROBLEMS

PART I - THEORETICAL BACKGROUND

D. K. Gartling
Fluid Mechanics and Heat Transfer Division I
Sandia National Laboratories
Albuquerque, New Mexico 87185

ABSTRACT

The theoretical and numerical background for the finite element computer program, NACHOS II, is presented in detail. The NACHOS II code is designed for the two-dimensional analysis of viscous incompressible fluid flows, including the effects of heat transfer and/or other transport processes. A general description of the boundary value problems treated by the program is presented. The finite element formulations and the associated numerical methods used in the NACHOS II code are also outlined. Instructions for use of the program are documented in SAND86-1817; examples of problems analyzed by the code are provided in SAND86-1818.

Preface

Since the first version of NACHOS was released in late 1977, a great deal of research and development has occurred in the area of finite element methods for fluid mechanics. Progress in the development of new elements and formulations for incompressible flows, improved iterative and transient solution algorithms and new matrix solution procedures have combined to make current practice significantly different from the accepted methods of a decade ago. These new methods, in concert with significant changes in computer hardware, have dictated the need for a revised and updated version of the NACHOS program. The present report, along with the revised user's manual and new example problem report, document the new code, NACHOS II.

In developing NACHOS II a number of new capabilities and features were added to increase the overall utility of the code. The basic system of balance equations (*i.e.*, mass, momentum and energy) has been augmented with the inclusion of two user defined transport equations. This extension allows problems such as double-diffusion, mass transport and chemically reacting flows to be considered without code modification. Also, a porous flow model was added to the code that permits fluid saturated porous materials to be included in a simulation. The element library was expanded to include a nine-node Lagrange element; a continuous or discontinuous pressure approximation can be used with any element in the library. The discontinuous pressure elements can also be used with a penalty function formulation of the incompressibility constraint. In order to improve the performance of the solution algorithms, all equations in a problem are solved in a fully coupled manner. For steady-state simulations the standard Picard method has been augmented with a full Newton method and a quasi-Newton procedure. Transient analyses are performed with either a backward Euler or trapezoid rule integration procedure. Either method can be run with a fixed timestep or a dynamic time step selection procedure.

Minor improvements in the allowed material models, boundary condition types and dependencies, flux computations and the stream function computation have also been incorporated in NACHOS II. The code has been rewritten in standard FORTRAN 77 to increase its portability. A dynamic dimensioning scheme has been devised to optimize machine usage. Finally, new pre- and post-processing file formats have been developed to permit stand-alone mesh generators and graphics programs to be easily integrated with the program.

Contents

1 Introduction

This document describes the theoretical and numerical background for the finite element computer code NACHOS II. The NACHOS II program is designed for the analysis of two-dimensional, viscous, incompressible fluid flow problems. The basic flows considered may be isothermal, nonisothermal or may involve other physical processes, such as mass transport. Both steady and transient flows may be analyzed.

The present document presents a description of the theoretical fluid mechanics background for the code, as well as a discussion of the most important numerical procedures utilized by NACHOS II. This volume is intended to serve as a background document for the NACHOS II user's manual, [1]. Potential users of NACHOS II are encouraged to become familiar with the present report and the example problem report [2] before attempting to use the program.

During the development of the present report, it has been assumed that the reader has a background in the areas of fluid mechanics and heat transfer. A basic knowledge of numerical methods is also essential for the proper understanding and use of the computer code. Though it has not been assumed that the reader has a background in finite element methods, a general acquaintance with the method is highly advantageous. Since by necessity many of the topics covered here are discussed only in the context of the present code application, the reader interested in general finite element methods is referred to the standard texts by Becker,*et al* [3], Zienkiewicz [4], Huebner and Thornton [5], Reddy [6], or Oden [7].

In the following section, the general class of flow problems treated by NACHOS II is discussed and the equations for the continuum problem are presented. Section 3 presents a brief description of the finite element method (FEM) and the formulations employed for the present class of problems. Sections 4 and 5 discuss computational details of individual element formulations and solution procedures for the discretized equations, respectively. The last section describes some of the procedures used in NACHOS II for auxiliary calculations, *e.g.*, mesh generation, flux computations, computation of the stream function, *etc.*

2 The Continuum Problem

A necessary prerequisite to the development of any computer program is the careful definition of the class of problems to which the code will be applied. In terms of a general categorization, the NACHOS II program is designed for the analysis of nonisothermal, viscous flow problems as described by specific forms of the conservation equations of mass, momentum and energy. Embedded in this general framework are a diversity of specific problem areas such as isothermal flows, forced convection, free convection and mixed convection heat transfer. Solid body heat conduction is also included in this description. In addition to the basic conservation equations, the present formulation in NACHOS II includes one or two "generic" transport equations that may be coupled to the flow equations to describe additional physical processes, *e.g.* mass transport. Though the inclusion of such nonspecific equations makes the precise definition of code applicability somewhat difficult, their utility in many flow simulations is unquestioned.

In the following section the basic equations describing the general nonisothermal flow problem will be developed and discussed. A subsequent section will describe the additional equations available in NACHOS II to simulate other transport or physical processes.

2.1 Viscous Flow Equations

The problem of interest concerns the motion of a viscous fluid and the transport of thermal energy in a material region, Ω. In the general case, the region Ω is composed of a fluid subregion, Ω_f and a solid body subregion, Ω_s. The geometry is limited to two spatial dimensions. All materials are assumed to be homogeneous and isotropic. The fluids of concern are assumed to be Newtonian or inelastic, non-Newtonian (see Section 2.4). The fluid motion is restricted to flows that are incompressible and laminar (see Section 2.4 for comments on turbulence models). For flows involving buoyancy forces, an extended form of the Boussinesq approximation [8,9] is invoked which allows the fluid properties to be functions of the thermodynamic state (*e.g.*, pressure and temperature) and the fluid density to vary with temperature according to

$$\rho = \rho_0 \left[1 - \beta(T - T_0)\right] \tag{1}$$

where β is the coefficient of thermal expansion and the subscript zero indicates a reference condition. The variation of density as given in (1) is permitted only in the description of the body force; the density in all other situations is ρ_0.

Considering the above assumptions and limitations, the basic equations of fluid motion for the region Ω_f are given by the momentum equation

$$\rho_0 \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = \rho_0 g_i - \rho_0 g_i \beta (T - T_0) + \frac{\partial \tau_{ij}}{\partial x_j}. \tag{2}$$

The conservation of mass equation, simplified for incompressible flows, is given by

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{3}$$

The transport of thermal energy in the fluid region Ω_f is described by the energy equation

$$\rho_0 C \left(\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} \right) = -\frac{\partial q_i}{\partial x_i} + Q + \Phi \tag{4}$$

while in the solid region, Ω_s, the energy equation is

$$\rho_s C_s \frac{\partial T}{\partial t} = -\frac{\partial q_i}{\partial x_i} + Q_s. \tag{5}$$

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

$$\tau_{ij} = -\hat{P}\delta_{ij} + 2\mu D_{ij} \tag{6}$$

and

$$q_i = -k \frac{\partial T}{\partial x_i}. \tag{7}$$

The flow kinematics are described by

$$D_{ij} = \tfrac{1}{2}\left(L_{ij} + L_{ji}\right) \quad ; \quad L_{ij} = \frac{\partial u_i}{\partial x_j} \tag{8}$$

where D_{ij} is the rate-of-deformation tensor and L_{ij} is the velocity gradient tensor.

Combining the definitions in (6)-(8) with the conservation equations (2)-(5) produces the following basic set of equations

Momentum:

$$\rho_0 \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left(-P + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) - \rho_0 g_i \beta (T - T_0) \tag{9}$$

Incompressibility:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{10}$$

Energy (Fluid):

$$\rho_0 C \left(\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} \right) = \frac{\partial}{\partial x_i} \left(k \frac{\partial T}{\partial x_i} \right) + Q + \Phi \tag{11}$$

Energy (Solid):

$$\rho_s C_s \frac{\partial T}{\partial t} = \frac{\partial}{\partial x_i} \left(k_s \frac{\partial T}{\partial x_i} \right) + Q_s. \tag{12}$$

Equations (9)-(12) are written for a planar geometry in an Eulerian reference frame with the indices running between 1 and 2 and the usual indicial summation conventions in effect. The axisymmetric form of the equations follows in a straightforward manner. The quantities used in (1)-(12) are defined as follows: u_i is the velocity component in the x_i coordinate direction, t is the time, $\hat{P}$ is the pressure, T is the temperature, τ_{ij} the total stress tensor, q_i the heat flux vector, μ the viscosity, C the specific heat and k the thermal conductivity. Also, Q is the volumetric heat source, Φ the viscous dissipation function ($\Phi = 2\mu D_{ij} D_{ij}$), δ_{ij} is the unit tensor and g_i the gravitational vector. The subscript s denotes a solid material. Finally, in writing (9) a modified pressure P was defined such that $P = \hat{P} + \rho_0 g_i x_i$. The gravitational force is taken to be positive when it is directed opposite to the positive coordinate axes.

Equations (9)-(12) form the basic description of the nonisothermal flows usually considered in the NACHOS II program. Before considering boundary conditions and other aspects of the basic boundary value problem it is useful to consider a modification of the above equations that is also contained in NACHOS II.

2.2 Porous Flow Equations

Let the previously defined fluid domain, Ω_f, be composed of two subregions denoted Ω_{vf} and Ω_{pf} such that $\Omega_f = \Omega_{vf} \cup \Omega_{pf}$. In the subregion Ω_{vf} the viscous flow equations as outlined in (9)-(12) are assumed to hold. The subregion Ω_{pf} is assumed to contain a rigid porous medium that is saturated with a viscous, incompressible fluid. The saturating fluid in Ω_{pf} is the same fluid as in Ω_{vf} if the two regions share a common permeable interface; otherwise the fluids in the two regions may be different. If the porous medium is assumed to be homogeneous and isotropic and the fluid and solid are in thermal equilibrium then the equations describing the fluid motion and energy transport in the region Ω_{pf} are given by

Momentum:

$$\frac{\rho_0}{\phi}\frac{\partial u_i}{\partial t} + \left(\frac{\rho_0 \hat{c}}{\sqrt{\kappa}}\|\vec{u}\| + \frac{\mu}{\kappa}\right) u_i = \frac{\partial}{\partial x_j}\left(-P + \mu_e\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) - \rho_0 g_i \beta (T - T_0) \qquad (13)$$

Incompressibility:

$$\frac{\partial u_i}{\partial x_i} = 0 \qquad (14)$$

Energy (Porous Layer):

$$(\rho_0 C)_e \frac{\partial T}{\partial t} + (\rho_0 C)\, u_j \frac{\partial T}{\partial x_j} = \frac{\partial}{\partial x_i}\left(k_e \frac{\partial T}{\partial x_i}\right) + Q. \qquad (15)$$

In equations (13)-(15), ϕ is the porosity of the porous medium, κ is the permeablility, $\hat{c}$ is the inertia coefficient and $\|\vec{u}\|$ is the magnitude of the velocity. The subscript e indicates an effective property; all other symbols retain their previous definitions. The effective capacitance and conductivity are generally functions of the fluid and solid matrix properties and the porosity. The effective conductivity can also be a function of the fluid velocity when thermal dispersion effects are important. The effective viscosity is often taken to be $\mu_e = \mu$ though other approximations are possible. Suitable empirical models are available for all of these quantities and may be found in references such as [10,11].

The equations in (13)-(15) represent a generalization of the standard Darcy equations for nonisothermal flow in a saturated porous medium. This system is sometimes referred to as the Forchheimer-Brinkman model for porous flow. No attempt will be made here to delineate all the conditions under which such a model is appropriate. However, it can be noted that by selectively including or omitting certain terms a number of other standard porous flow models can be derived. Thus, if $\hat{c} = 0$, a Brinkman model is obtained, while a prescription of $\hat{c} = 0$ and $\mu_e = 0$ produces a standard Darcy formulation. For further discussion of individual models and their regions of applicability the reader is referred to [10,11,12] and the porous media research literature.

The general form of the porous flow equations are very similar to the equations for the viscous flow of a bulk fluid. By simply redefining certain coefficients, equations (13)-(15) can, in fact, be obtained from (9)-(12). As a result of this simple transformation process it is quite straightforward to include a porous flow model in the general framework for nonisothermal flow problems. The equations given in (13)-(15) have been incorporated in NACHOS II and may be accessed at the discretion of the code user. The region Ω_{pf} may exist in conjunction with Ω_{vf} or in place of Ω_{vf}. The solid region, Ω_s, may or may not be included as the problem dictates. In the following sections specific reference will be made to the porous flow equations only when their treatment differs significantly from the equations for a viscous fluid.

2.3 Auxiliary Transport Equations

Though the nonisothermal, viscous (porous) flow problem outlined in the previous sections is very general, there are still a significant number of flow problems that cannot be described by the stated equations. For flows in which transport processes other than those of a thermal nature are important, additional equations are required. The present formulation allows the definition of up to two additional equations to describe such cases. The added equations are of the advection-diffusion type and are generic in the sense that they are not specifically associated with a particular physical process. Possible uses for these additional equations include the description of mass transport in a multicomponent system, the simulation of certain types of chemical reaction and the prediction of particle orientation for flows containing suspended fibers. Problems demonstrating some of these cases are provided in [2].

Consider again the material region Ω. The transport of two scalar quantities, c_1 and c_2, within this region is described by

$$C_\alpha \left(\frac{\partial c_\alpha}{\partial t} + u_j \frac{\partial c_\alpha}{\partial x_j} \right) = \frac{\partial}{\partial x_i} \left(D_\alpha \frac{\partial c_\alpha}{\partial x_i} \right) + Q_\alpha \tag{16}$$

where subscript α is either 1 or 2. In equation (16) C_α is a "capacitance" coefficient (analogous to $\rho_0 C$ in equation (11)), D_α is a diffusion coefficient and Q_α is a volumetric source term. Note that equation (16) is valid in the solid regions of Ω if u_j is set to zero; the transport equations can also be used in a porous region if C_α and D_α are interpreted as "effective" properties of the fluid/matrix system.

In order to fully couple the above transport equations to the general nonisothermal flow problem, certain extensions must be made to the previously stated functional form for the fluid density variation. The possibility of buoyancy forces due to variations in the auxiliary variables c_α requires that the equation of state (1) be modified to

$$\rho = \rho_0 \left[1 - \beta(T - T_0) - \gamma_1(c_1 - c_{1_0}) - \gamma_2(c_2 - c_{2_0}) \right] \tag{17}$$

where γ_α is an expansion coefficient and the subscript zero refers to a reference condition. As before this variation in density is only permitted in the body force term. The body force terms in (9) and (13) will therefore be altered to include the last two terms in (17) whenever the auxiliary transport equations are included in the problem formulation.

In order to complete the specification of the general boundary value problems described in the last three sections, suitable boundary and initial conditions are required. This topic is covered in the next section; a subsequent section will discuss permissable variations in material properties and constitutive equations.

2.4 Boundary Conditions

Boundary conditions for the nonisothermal, viscous flow problem outlined in Section 2.1 are most easily understood and described by considering the fluid mechanics separate from the energy transport. Reference to Figure 1 will also aid the discussion. For the fluid dynamic part of the problem either the velocity components (Dirichlet or essential boundary conditions) or the total surface stress or traction (Neumann or natural boundary conditions) are generally specified on the boundary of the fluid region. Symbolically these conditions are given by

$$u_i = f_i^u(s,t) \qquad on\ \Gamma_u$$

$$\mathcal{T}_i = \tau_{ij}(s,t) n_j(s) = f_i^{\mathcal{T}}(s,t) \qquad on\ \Gamma_{\mathcal{T}} \tag{18}$$

where s is the coordinate 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,

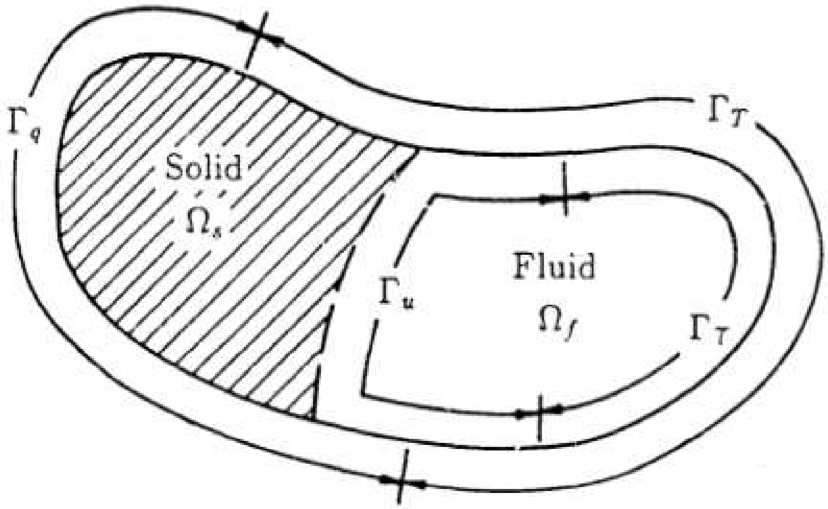

Figure 1: Schematic for boundary condition definitions.

with $\Gamma_f = \Gamma_u \cup \Gamma_\mathcal{T}$. Note that the conditions written in (18) are in vector form, *i.e.*, there must be a condition on each component of the velocity. Though shown as being applied to separate regions, the specified velocity and traction conditions can be mixed so that over a boundary segment each velocity component is subject to a different type of boundary condition. The specified functions f_i^u and $f_i^\mathcal{T}$ are generally simple expressions for most boundaries of interest (*e.g.*, $f_i^u = 0$ for stationary solid boundaries, $f_i^\mathcal{T} = 0$ on symmetry and some outflow boundaries). Under certain special flow conditions, such as those involving free surfaces or frictional (slip) boundaries, the $f_i^\mathcal{T}$ functions can be more complex and even involve a mixed or Robin type of boundary condition. The specific forms of $f_i^\mathcal{T}$ implemented in NACHOS II are described in Section 4.6.2.

The flow conditions that can be applied to the boundary of a fluid-saturated porous medium depend on the specific model used to describe the problem. For the general Forchheimer-Brinkman equation listed in (13), the admissable boundary conditions are the same as those given for the Navier-Stokes equations, namely, equation (18). However,in cases where the Brinkman terms are excluded from the momentum equation ($\mu_e = 0$) the order of the equation is reduced and fewer boundary conditions are needed. A Forchheimer or simple Darcy model allows the following types of boundary conditions

$$u_i n_i = f^u(s,t) \qquad on\ \Gamma_u$$

$$\mathcal{T}_i n_i = -\hat{P}(s,t)\delta_{ij} n_j(s) n_i = -\hat{P}(s,t) = f^\mathcal{T}(s,t) \qquad on\ \Gamma_\mathcal{T}. \tag{19}$$

In essence these conditions state that the fluid velocity normal to the boundary may be specified (outflow/inflow) or the normal force (pressure) on the boundary can be imposed. These simplified models do not allow a tangential velocity to be imposed (*e.g.*, no-slip walls are not admissable) nor the specification of viscous forces. Also, note that only one type of boundary condition can be specified at any point on the boundary. At an interface between a saturated porous material and a bulk fluid the continuity of the velocity field is required; continuity of the stress components depends on the specific model used in the porous region.

The thermal part of the boundary value problem for the fluid, porous or solid region requires the temperature (Dirichlet or essential condition) or the heat flux (Neumann or natural condition) to be specified on all parts of the boundary enclosing the heat transfer region. In equation form these conditions are

$$T = f^T(s,t) \qquad on\ \Gamma_T$$

$$\left(k\frac{\partial T}{\partial x_i}\right) n_i + q_c + q_r = f^q(s,t) \qquad on\ \Gamma_q \tag{20}$$

where Γ_{ht} is the total boundary enclosing the heat transfer region and $\Gamma_{ht} = \Gamma_T \cup \Gamma_q$. The quantities q_c and q_r refer to the convective and radiative components of the heat flux and are given by

$$q_c = h_c(s,T,t)(T - T_c)$$

$$q_r = h_r(s,T,t)(T - T_r)\ . \tag{21}$$

In equation (21), h_c is the convective heat transfer coefficient, which, in general, depends on location on the boundary, temperature and time, and T_c is a reference (or sink) temperature for convective transfer. Also, the effective radiation heat transfer coefficient is $h_r = \mathcal{F}\sigma(T+T_r)(T^2+T_r^2)$ where σ is the Stefan-Boltzmann constant, $\mathcal{F}$ is a form factor and T_r the reference temperature for radiative transfer. The form factor, $\mathcal{F}$, is related to the boundary emissivity, ϵ , and the position of the boundary relative to surrounding surfaces (see *e.g.*, [13]). As in the case of the fluid conditions, the functions f^T and f^q are simple functions for the cases normally encountered in practical problems.

As a final point on the fluid and thermal boundary condition specification note that the boundaries Γ_f and Γ_{ht} need not coincide (see Figure 1). Indeed, for problems involving regions of convection and solid body conduction the boundary regions are not the same. When both fluid and solid and/or porous material regions exist in a problem, continuity conditions on the temperature and heat flux must generally prevail along the interface of the different materials. The interface conditions along a phase change boundary (melting or solidification) are somewhat different though this case is not considered here. Procedures for incorporating phase change conditions in convection problems are given in [14,15]

When the auxiliary transport equations are included in the problem formulation the boundary conditions for these equations are directly analogous to those for the previously described energy equation. The Dirichlet condition requires that c_α be specified on the boundary while the Neumann condition prescribes the flux of c_α to be known on the boundary. In equation form these are expressed by

$$c_\alpha = f_\alpha^c(s,t) \qquad on \;\; \Gamma_{c_\alpha}$$

$$\left(D_\alpha \frac{\partial c_\alpha}{\partial x_i} \right) n_i + q_{c_\alpha} = f_\alpha^{q_c}(s,t) \;\; on \;\; \Gamma_{q_c} \tag{22}$$

where Γ is the total boundary enclosing the transport region and $\Gamma = \Gamma_{c_\alpha} \cup \Gamma_{q_c}$. In general, the flux condition q_{c_α} is given by

$$q_{c_\alpha} = h_{c_\alpha}(s, c_\alpha, t)(c_\alpha - c_{\alpha,c}) \tag{23}$$

where h_{c_α} is a general transfer coefficient and $c_{\alpha,c}$ is a reference value. At material interfaces continuity of the auxiliary variable and its flux is normally required.

For time-dependent problems, a set of initial conditions are necessary for the dependent variables. Very often these conditions consist of a solid body at a uniform temperature and a quiescent fluid at a uniform temperature and hydrostatic pressure. An initial state for the auxiliary variables is also required if they are present in the problem. A second common possibility is for the transient motion to be initiated from an established steady-state flow field with known temperature and auxiliary variable fields. In all cases, the dependent variables must be known for all x_i and must satisfy the basic conservation equations (*e.g.*, the initial fluid velocity field must be divergence free and have a compatible pressure field).

2.5 Material Properties and Constitutive Equations

2.5.1 Property Variations

For the general nonisothermal flow problems defined in Sections 2.1 and 2.2 the fluid properties, such as μ, C, β and k are explicit functions of, at most, two thermodynamic variables such as pressure and temperature. Due to the assumption of incompressibility the dependence on pressure variations is usually negligible, thus leaving the material properties as functions of the temperature only. Solid materials may also have specific heats and conductivities that vary with temperature. The volumetric heat source for the fluid and/or solid may be a function of temperature, time and spatial location. The basic material properties involved in the porous flow equations follow the same basic pattern and are generally functions of temperature and perhaps time and spatial location.

The property dependencies described above can be altered by the inclusion of the auxiliary transport equations. In some cases the basic fluid properties may be functions of one or both of the auxiliary variables; such dependencies are determined by the specific nature of the auxiliary variables. The material coefficients in the auxiliary transport equations will, in general, be functions of temperature as well as the additional dependent variables. The volumetric source terms associated with the c_i variables will have a functional form similar to the volumetric heat source.

The variation of the fluid viscosity with temperature (and perhaps c_α) is appropriate for the description of laminar flows of a Newtonian fluid. With suitable modifications to this functional dependence, the flow of certain types of non-Newtonian materials may be described as well as some simplified turbulent flows.

2.5.2 Non-Newtonian Models

Consider the case of an inelastic, non-Newtonian fluid (*i.e.*, a so-called generalized Newtonian fluid) for which the total stress tensor can be written as

$$\tau_{ij} = -\hat{P}\delta_{ij} + 2\mu(D_{ij})D_{ij} \tag{24}$$

which is similar to (6) except for the dependence of the viscosity, μ, on the rate-of-deformation tensor, D_{ij}. Since μ is a scalar, its dependence on D_{ij} must be of a particular form [16],

$$\mu = \mu(D_{ij}) = \mu(I_1, I_2, I_3) \tag{25}$$

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

$$I_1 = tr(\mathbf{D}) = \sum_i D_{ii} \tag{26}$$

$$I_2 = \tfrac{1}{2} tr(\mathbf{D}^2) = \tfrac{1}{2} \sum_i \sum_j D_{ij} D_{ji} \tag{27}$$

$$I_3 = \tfrac{1}{3} tr(\mathbf{D}^3) = \tfrac{1}{3} \sum_i \sum_j \sum_k D_{ij} D_{jk} D_{ki} \tag{28}$$

and tr indicates the trace. For an incompressible fluid, $I_1 = \nabla \cdot u = 0$. Also, there is no experimental evidence to suggest that the viscosity depends on I_3; thus, the dependence on the third invariant is generally neglected. Equation (25) then reduces to

$$\mu = \mu(D_{ij}) = \mu(I_2) \tag{29}$$

for the description of a generalized Newtonian fluid. The viscosity in (29) can of course depend on the temperature and other dependent variables, as required. Though the definition of I_2 given in (27) is quite standard, it is often convenient to multiply this definition by a factor of four. This alteration allows I_2 to reduce to the shear rate for simple flows. NACHOS II uses the altered definition in all computations.

A variety of non-Newtonian models of the form given in (29) have been proposed on the basis of experimental observation and empiricism. Some of the most useful and popular of these models are cataloged below. A more extensive discussion of viscosity functions is provided in [17].

Power Law Model

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

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

where n and K are constants (or perhaps functions of temperature) and are termed the power law index and consistency, respectively. 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 are often useful

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

$$K = K_0 exp\left(-A[T - T_0]/T_0\right)$$

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

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

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

In the above equation μ_0 and μ_∞ are the zero and infinite shear rate viscosities, respectively, and λ is a time constant. The remaining parameters were defined previously.

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, σ_0, such that when the applied stresses are below σ_0 no motion occurs; when the applied stresses exceed σ_0 the material flows with the deformation rate being proportional to the excess of the stress over the yield condition. Typically, the fluid response after yield is taken to be linear in the deformation rate (Bingham model) though other forms such as a power law equation (Herschel-Buckley model) are possible.

In general form the Bingham model is expressed as [16,18]

$$\sigma_{ij} = \left(\frac{\sigma_0}{\sqrt{I_2}} + 2\mu \right) D_{ij} \qquad \text{when} \;\; \tfrac{1}{2} tr(\sigma^2) \geq \sigma_0^2$$

$$D_{ij} = 0 \qquad \text{when} \;\; \tfrac{1}{2} tr(\sigma^2) < \sigma_0^2$$

where σ_{ij} is the deviatoric part of the stress tensor given in (24). From the first relation the apparent viscosity of the material beyond the yield point is $\left(\sigma_0/2\sqrt{I_2} + \mu\right)$. The inequalities in the above formulas describe a von Mises yield criteria. The implementation of the Bingham model into a computational procedure requires a slight modification due to the condition $D_{ij} = 0$. The necessary changes are described in detail in references [2,19,20].

The above models serve to illustrate some of the typical viscosity functions that are available to model inelastic, non-Newtonian fluids. All of these viscosity functions can be used in the present version of NACHOS II.

2.5.3 Turbulence Models

The turbulence models that may be employed in NACHOS II are the simplest possible and involve the use of an apparent or "eddy" viscosity function and a mixing length model. These so-called algebraic or zero equation models do not involve the solution of any additional differential equations and may be implemented in the same manner as the previously described non-Newtonian models. No extensive theoretical development of turbulence or turbulence modeling will be given here and no attempt to justify the use or utility of the algebraic model will be undertaken. The discussion will be limited to a simple statement of the relevant equations and the form of the permissable computational models. For a more extensive discussion the reader is referred to standard texts, *e.g.*, [21,22].

It is normally assumed that the dependent variables in a turbulent flow can be represented by the sum of a mean and a fluctuating component. The use of such an assumption, in conjunction with suitable time averaging rules, permits the balance laws of mass, momentum, energy, *etc.* to be developed in terms of the mean quantities in the flow. The mean flow equations are identical to the previously cited laminar flow equations except for the presence of some additional stress terms in the momentum equation and additional flux terms in the energy equations. For the momentum equation these additional terms, called the Reynolds stresses, arise due to the presence of velocity fluctuations in the flow; the added flux terms stem from fluctuations in the velocity and the temperature. The effective stress tensor for a turbulent flow can be written as

$$\tau_{ij} = -\hat{P}\delta_{ij} + 2\mu D_{ij} - \tau'_{ij} \tag{30}$$

with the Reynolds stresses τ'_{ij} being defined by

$$\tau'_{ij} = \rho_0 \overline{u'_i u'_j} \tag{31}$$

where u'_i is a component of the fluctuating velocity field and the overbar denotes a time average.

If the stress tensor in (30) is to be used in the momentum equation then a closure of the system must be developed by relating the Reynolds stresses to mean flow quantitites. This is one of the fundamental problems involving turbulent flows. For the present application, closure is achieved through Prandtl's mixing length hypothesis [21] which in general form is

$$\tau'_{ij} = \rho_0 \overline{u'_i u'_j} = 2\mu_t D_{ij} \tag{32}$$

where μ_t is the apparent or "eddy" viscosity and D_{ij} is the usual rate-of-deformation tensor for the mean velocity field. The apparent viscosity is

$$\mu_t = \rho_0 l^2 \sqrt{I_2} \tag{33}$$

where I_2 is the second invariant of D_{ij} as given in equation (27) and l is Prandtl's mixing length. The similarity of this viscosity function with the non-Newtonian models is quite evident. The mixing length l is an empirically determined parameter that varies for different types of flows, *e.g.*, jets, wakes and boundary layers. Values and models for l are available in the literature. Note that for use in a turbulent flow simulation the apparent viscosity μ_t should be combined with the usual molecular viscosity μ and the sum used in the standard momentum equation (9).

When considering nonisothermal flows, the thermal conductivity must also be modified to reflect the turbulent nature of the flow. With the model outlined above, Reynolds analogy [21] can be used to obtain an apparent thermal conductivity, k_t. This value is combined with the normal conductivity and the result used in the standard energy equation. Similar analogies can be used to derive apparent diffusion coefficients for other types of transport equations, such as those described in a previous section.

2.6 Summary of Equations

Though the previous sections have described the various equations and boundary conditions in some detail, it is useful to summarize the complete boundary value problem in one section. Therefore, the complete set of equations are presented here by physical process.

Viscous Flow:

$$\rho_0 \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left(-P + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right)$$

$$-\rho_0 g_i \left[\beta(T-T_0)+\gamma_1(c_1-c_{1_0})+\gamma_2(c_2-c_{2_0})\right] \tag{34}$$

$$\frac{\partial u_i}{\partial x_i}=0 \tag{35}$$

Porous Flow:

$$\frac{1}{\phi}\frac{\partial u_i}{\partial t}+\left(\frac{\rho_0 \hat{c}}{\sqrt{\kappa}}\|\vec{u}\|+\frac{\mu}{\kappa}\right)u_i=\frac{\partial}{\partial x_j}\left(-P+\mu_e\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\right)$$
$$-\rho_0 g_i \left[\beta(T-T_0)+\gamma_1(c_1-c_{1_0})+\gamma_2(c_2-c_{2_0})\right] \tag{36}$$

$$\frac{\partial u_i}{\partial x_i}=0 \tag{37}$$

Heat Transfer:

$$\rho_0 C\left(\frac{\partial T}{\partial t}+u_j\frac{\partial T}{\partial x_j}\right)=\frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right)+Q+\Phi \tag{38}$$

Auxiliary Transport:

$$C_\alpha\left(\frac{\partial c_\alpha}{\partial t}+u_j\frac{\partial c_\alpha}{\partial x_j}\right)=\frac{\partial}{\partial x_i}\left(D_\alpha\frac{\partial c_\alpha}{\partial x_i}\right)+Q_\alpha \tag{39}$$

Equations (34)-(39) with the noted boundary and initial conditions form a complete set for the determination of the velocity, pressure, temperature and auxiliary variables in a fluid or fluid/solid region. Embedded in this description are numerous classes of flow problems that can be recovered through the omission or inclusion of specific equations, the neglect of certain terms within an equation and the variations of material properties. The general problem as outlined here is the basis for the present version of NACHOS II.

3 Finite Element Formulation

The boundary value problems outlined in the previous sections are generally not amenable to closed form solution except in cases where problem geometry may be regularized and/or physical phenomena neglected. For the solution of most realistic problems, one is forced to consider approximate solution methods of which the computerized numerical schemes are the most powerful. The currently popular numerical methods are generally divided into two groups– finite difference methods (FDM) and finite element methods (FEM). The objective of both approaches is to reduce the continuous problem (infinite number of degrees of freedom) described by a partial differential equation to a discrete problem (finite number of degrees of freedom) described by a system of algebraic equations. Though the ultimate results of both numerical methods are very similar, the procedures are sufficiently different in their philosophies and implementation to be considered distinct.

It is beyond the scope of the present discussion to describe either method in general. Rather, the approach followed here will be to describe in some detail the particular FEM used in developing the NACHOS II program.

3.1 Nonisothermal Flow Problem

To simplify the ensuing derivation only the equations describing nonisothermal, viscous flow will be considered initially. The extension of the derivation to the porous flow equations and the auxiliary transport equations will be treated in subsequent sections.

The development of the FEM for a particular boundary value problem may be approached from several viewpoints. The derivation presented here is a pragmatic one and is oriented toward the development of a computational method. More mathematical approaches involving discusions of variational methods, weak forms of the boundary value problem and approximation theory are also possible. The reader interested in these more theoretical aspects of the method should consult the indicated texts [3-7] and the large body of archival literature.

In the present case the finite element procedure begins with the division of the continuum region of interest Ω into a number of simply shaped regions called finite elements, as shown in Figure 2. Since the Eulerian

description of the fluid motion was used in the field equations (34), (35) and (38), these elements are assumed to be fixed in space. Within each element, the dependent variables (u_i,P and T) are interpolated by continuous functions of compatible order, in terms of values to be determined at a set of nodal points. For purposes of developing the equations for these nodal point unknowns, an individual element can be separated

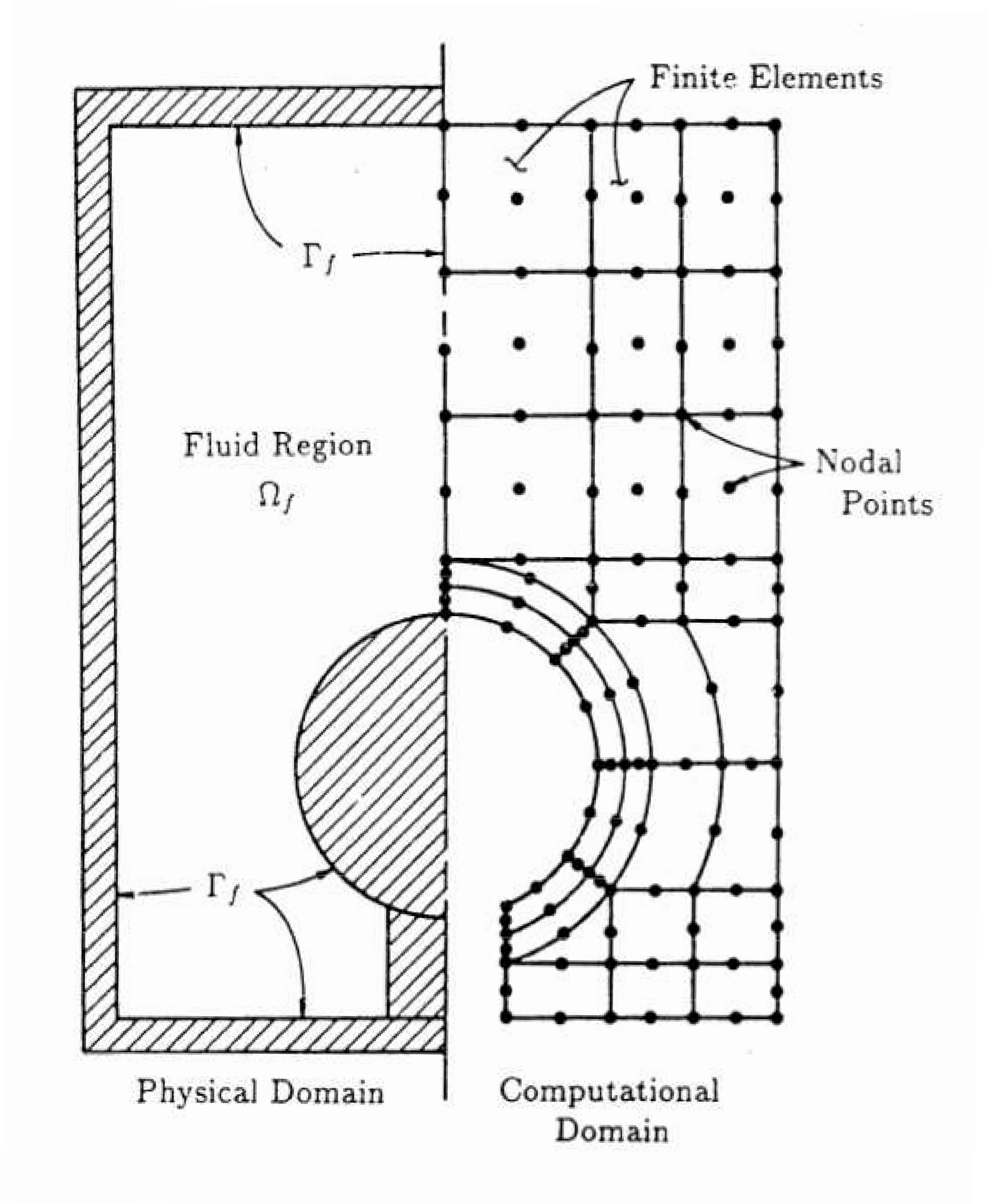

Figure 2: Typical finite element discretization.

from the assembled system. (Note: It is not necessary to develop the equations on an element by element basis; the equations may be derived for the assembled system directly. The present approach is utilized since it mimics the procedure found in the computer program.)

Within each element, the velocity, pressure, and temperature fields are approximated by,

$$\begin{aligned} u_i(x_i,t) &= \mathbf{\Phi}^T(x_i)\mathbf{u}_i(t) \\ P(x_i,t) &= \mathbf{\Psi}^T(x_i)\mathbf{P}(t) \\ T(x_i,t) &= \mathbf{\Theta}^T(x_i)\mathbf{T}(t) \end{aligned} \tag{40}$$

where the $\mathbf{u}_i, \mathbf{P}$ and $\mathbf{T}$ are vectors of element nodal point unknowns, $\mathbf{\Phi}, \mathbf{\Psi}$ and $\mathbf{\Theta}$ are vectors of interpolation (basis) functions and superscript T denotes a vector transpose.

Substitution of these approximations into the field equations (34), (35) and (38), yields a set of functional equations of the form

Momentum :

$$\begin{aligned} f_{u_1}(\mathbf{\Phi},\mathbf{\Psi},\mathbf{\Theta},\mathbf{u_i},\mathbf{P},\mathbf{T}) &= R_{u_1} \\ f_{u_2}(\mathbf{\Phi},\mathbf{\Psi},\mathbf{\Theta},\mathbf{u_i},\mathbf{P},\mathbf{T}) &= R_{u_2} \end{aligned}$$

Incompressibility :

$$f_P(\mathbf{\Phi},\mathbf{u_i}) = R_P$$

Energy :

$$f_T(\mathbf{\Phi},\mathbf{\Theta},\mathbf{u_i},\mathbf{T}) = R_T$$

where the R's denote the residuals (errors) resulting from the use of the approximations in equation (40).

The Galerkin form of the method of weighted residuals [23] seeks to reduce these errors to zero, in a weighted sense, by making the residuals orthogonal to the interpolation functions ($i.e.$,$\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Theta}$ in equation (40)) over each element. These orthogonality conditions are expressed by,

$$\begin{aligned} \langle \mathbf{\Phi}, f_{u_1}\rangle &= \langle \mathbf{\Phi}, R_{u_1}\rangle = 0 \\ \langle \mathbf{\Phi}, f_{u_2}\rangle &= \langle \mathbf{\Phi}, R_{u_2}\rangle = 0 \\ \langle \mathbf{\Psi}, f_P\rangle &= \langle \mathbf{\Psi}, R_P\rangle = 0 \\ \langle \mathbf{\Theta}, f_T\rangle &= \langle \mathbf{\Theta}, R_T\rangle = 0 \end{aligned} \tag{41}$$

where $\langle a,b\rangle$, denotes the inner product, defined by,

$$\langle a,b\rangle = \int_\Omega a\cdot b\; d\Omega$$

with Ω being the volume of the element.

The detailed manipulations involving the integrals defined in equation (41) are presented in Appendix A. The results of those computations can be expressed by the following matrix equations,

Momentum :

$$\mathbf{M}\dot{\mathbf{U}} + \mathbf{C}(\mathbf{U})\mathbf{U} - \mathbf{Q}\mathbf{P} + \mathbf{K}(\mathbf{U},\mathbf{T})\mathbf{U} + \mathbf{B}(\mathbf{T})\mathbf{T} = \mathbf{F}(\mathbf{T}) \tag{42}$$

Incompressibility :

$$-\mathbf{Q}^T\mathbf{U} = \mathbf{0} \tag{43}$$

Energy :

$$\mathbf{N}\dot{\mathbf{T}} + \mathbf{D}(\mathbf{U})\mathbf{T} + \mathbf{L}(\mathbf{T})\mathbf{T} = \mathbf{G}(\mathbf{T},\mathbf{U}) \tag{44}$$

with

$$\mathbf{U}^T = \{\mathbf{u}_1^T, \mathbf{u}_2^T\}.$$

The component equations (42)-(44) can be combined into a single matrix equation

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{U}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix}$$

$$+ \begin{bmatrix} \mathbf{C}(\mathbf{U}) + \mathbf{K}(\mathbf{U},\mathbf{T}) & -\mathbf{Q} & \mathbf{B}(\mathbf{T}) \\ -\mathbf{Q}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}(\mathbf{U}) + \mathbf{L}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(\mathbf{T}) \\ \mathbf{0} \\ \mathbf{G}(\mathbf{T},\mathbf{U}) \end{Bmatrix} \tag{45}$$

or in a more symbolic format as

$$\overline{\mathbf{M}}\dot{\mathbf{V}} + \overline{\mathbf{K}}(\mathbf{U},\mathbf{T})\mathbf{V} = \overline{\mathbf{F}}(\mathbf{U},\mathbf{T}) \tag{46}$$

where

$$\mathbf{V}^T = \{\mathbf{u}_1^T, \mathbf{u}_2^T, \mathbf{P}^T, \mathbf{T}^T\}.$$

The superposed dot indicates a time derivative.

The matrix equations given by (42)-(46) represent the discrete analogues of the conservation equations for an individual fluid finite element. Note that the **C** and **D** matrices represent the advection (convection) of momentum and energy, respectively; the **K** and **L** matrices represent the diffusion of momentum and energy. The **Q** matrix is a gradient operator and $\mathbf{Q}^T$ is the divergence operator. The **M** and **N** matrices represent the mass and capacitance terms in the field equations. The **B** matrix represents the buoyancy term while the **F** and **G** vectors provide the forcing functions for the system in terms of volume forces (body force,volumetric heating) and surface forces (stress, heat flux).

For the case where a solid (non-flowing) material is to be represented, then only the capacitance, diffusion, and force terms of equation (44) need to be considered, *i.e.*,

$$\mathbf{N}\dot{\mathbf{T}} + \mathbf{L}(\mathbf{T})\mathbf{T} = \mathbf{G}(\mathbf{T}) \tag{47}$$

which is the discrete analog for the transient heat conduction equation.

The above derivation has been concerned with a single finite element and the limited portion of the continuum it represents. The discrete representation of the entire continuum region of interest is obtained through an assemblage of elements such that interelement continuity of the approximate velocity, pressure, and temperature is enforced. This continuity requirement is met through the appropriate summation of equations for nodes common to adjacent elements (the so-called "direct stiffness" approach) [3-6]. The result of such an assembly process is a system of matrix equations of the form given by equations (45) through (47).

3.2 Penalty Function Formulation

The previous development was based on a so-called mixed FEM in which all of the dependent variables were directly approximated and retained in the global matrix problem. For the incompressible flow problem a second method, the penalty method, can be formulated in which the pressure is eliminated from the matrix problem. Consider an incompressibility condition of the following form

$$\frac{\partial u_i}{\partial x_i} = -\epsilon P \tag{48}$$

where ϵ is the penalty parameter and P is the pressure. The penalty parameter is typically a small constant $\sim 10^{-6}$. Under a suitable set of conditions [24] it can be shown that the solution of the usual momentum equations (34) with the above constraint converges to the solution of the true incompressible problem as $\epsilon \to 0$. Thus, equation (48) is a useful approximation to the original constraint condition.

Using the above approximation for the continuity equation and the usual Galerkin method of weighted residuals produces

$$\mathbf{Q}^T\mathbf{U} = -\epsilon\mathbf{M_p}\mathbf{P}. \tag{49}$$

The details of deriving (49) are given in Appendix A. Solving (49) for the pressure $\mathbf{P}$ and substituting into the discretized form of the momentum equations (42) produces

$$\mathbf{M}\dot{\mathbf{U}} + \mathbf{C}(\mathbf{U})\mathbf{U} + \mathbf{K_p}\mathbf{U} + \mathbf{K}(\mathbf{U},\mathbf{T})\mathbf{U} + \mathbf{B}(\mathbf{T})\mathbf{T} = \mathbf{F}(\mathbf{T}) \tag{50}$$

where

$$\mathbf{K_p} = \frac{1}{\epsilon}\mathbf{Q}\mathbf{M_p}^{-1}\mathbf{Q}^T. \tag{51}$$

The matrix system for the nonisothermal flow problem then becomes

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{U}} \\ \dot{\mathbf{T}} \end{Bmatrix}$$

$$+ \begin{bmatrix} \mathbf{C}(\mathbf{U}) + \mathbf{K_p} + \mathbf{K}(\mathbf{U},\mathbf{T}) & \mathbf{B}(\mathbf{T}) \\ \mathbf{0} & \mathbf{D}(\mathbf{U}) + \mathbf{L}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(\mathbf{T}) \\ \mathbf{G}(\mathbf{T},\mathbf{U}) \end{Bmatrix}. \qquad (52)$$

This system again has the same general form as shown in equation (45).

By use of (49) the pressure has been removed from the basic matrix problem and the overall size of the problem is thereby reduced. Note that to recover the pressure the inverted form of (49) is used with a known velocity

$$\mathbf{P} = -\frac{1}{\epsilon}\mathbf{M_p}^{-1}\mathbf{Q}^T\mathbf{U}. \qquad (53)$$

The particular form of the penalty method described here is commonly termed a consistent penalty method since the penalty approximation was used on the discretized form of the constraint equation. Other forms are possible such as the substitution of (48) into the original partial differential form of the momentum equation. A Galerkin form of the resulting “perturbed” partial differential equation leads to a system similar to the above but with some subtle differences that affect the numerical implementation. For details on the so-called “reduced integration penalty” (RIP) methods see for example [24,25].

It is important to note that the penalty method provides an approximate method for satisfying a strong constraint condition. The basis for its use comes from the field of constrained minimization problems. The successful numerical implementation of the method described here relies on the ability to efficiently construct the $\mathbf{K_p}$ matrix, *i.e.* invert $\mathbf{M_p}$ at the element level. This requirement restricts the choices for the basis functions, $\boldsymbol{\Psi}$, used to represent the pressure. Further comment on this aspect of the penalty method will be delayed until the discussion on element matrix construction.

3.3 Porous Flow Problem

The derivation of the finite element equations for the porous flow models follows the previous sections in a completely analogous manner. The dependent variables are again approximated by the relations in (40). A Galerkin weighted residual procedure is applied to each equation in (36)-(38). The result of such a procedure is a set of matrix equations of the following form (see Appendix A for details)

Momentum :

$$\tilde{\mathbf{M}}\dot{\mathbf{U}} + \tilde{\mathbf{C}}(\mathbf{U})\mathbf{U} + \tilde{\mathbf{A}}\mathbf{U} - \tilde{\mathbf{Q}}\mathbf{P} + \tilde{\mathbf{K}}\mathbf{U} + \tilde{\mathbf{B}}(\mathbf{T})\mathbf{T} = \tilde{\mathbf{F}}(\mathbf{T}) \qquad (54)$$

Incompressiblity :

$$-\tilde{\mathbf{Q}}^T\mathbf{U} = \mathbf{0} \tag{55}$$

Energy :

$$\tilde{\mathbf{N}}\dot{\mathbf{T}} + \tilde{\mathbf{D}}(\mathbf{U})\mathbf{T} + \tilde{\mathbf{L}}(\mathbf{T})\mathbf{T} = \tilde{\mathbf{G}}(\mathbf{T}) \tag{56}$$

where again

$$\mathbf{U}^T = \{\mathbf{u}_1^T, \mathbf{u}_2^T\}.$$

The porous flow equations (54)-(56) can be arranged into a single matrix equation of the same form as given previously for the Navier-Stokes system (45)

$$\begin{bmatrix} \tilde{\mathbf{M}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{N}} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{U}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \tilde{\mathbf{C}}(\mathbf{U}) + \tilde{\mathbf{A}} + \tilde{\mathbf{K}} & -\tilde{\mathbf{Q}} & \tilde{\mathbf{B}}(\mathbf{T}) \\ -\tilde{\mathbf{Q}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{D}}(\mathbf{U}) + \tilde{\mathbf{L}}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \tilde{\mathbf{F}}(\mathbf{T}) \\ \mathbf{0} \\ \tilde{\mathbf{G}}(\mathbf{T}) \end{Bmatrix}. \tag{57}$$

The above equation may take on a somewhat different structure when the various porous flow models are invoked. When the inertia terms are unimportant then $\tilde{\mathbf{C}} = 0$. The ommission of the Brinkman extension forces $\tilde{\mathbf{K}} = 0$. For the familiar Darcy flow case the above equation simplifies to

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{N}} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{U}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \tilde{\mathbf{A}} & -\tilde{\mathbf{Q}} & \tilde{\mathbf{B}}(\mathbf{T}) \\ -\tilde{\mathbf{Q}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{D}}(\mathbf{U}) + \tilde{\mathbf{L}}(\mathbf{T}) \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \tilde{\mathbf{F}}(\mathbf{T}) \\ \mathbf{0} \\ \tilde{\mathbf{G}}(\mathbf{T}) \end{Bmatrix}. \tag{58}$$

Finally, note that the penalty function formulation of Section 3.2 can also be used with the porous flow equations since the incompressibility constraint and pressure terms appear in the same form in (57) as in the Navier-Stokes system.

3.4 Auxiliary Transport Equations

The finite element formulation for the auxiliary transport equations need not be performed explicitly since it is virtually identical to the energy equation derivation discussed in Section 3.1. The dependent variables c_α are approximated by the following expansion

$$c_\alpha = \mathbf{\Pi}^T\mathbf{c}_\alpha. \tag{59}$$

The Galerkin procedure applied to (39) produces a matrix equation of the form

$$\mathbf{N}_\alpha\dot{\mathbf{c}}_\alpha + \mathbf{D}_\alpha(\mathbf{U})\mathbf{c}_\alpha + \mathbf{L}_\alpha(\mathbf{c}_\alpha)\mathbf{c}_\alpha = \mathbf{G}_\alpha(\mathbf{c}_\alpha). \tag{60}$$

The definitions for the matrices in (60) may be obtained by the appropriate substitution of the shape function $\mathbf{\Pi}$ for the function $\mathbf{\Theta}$ in the relevant terms of equation (A7) in Appendix A.

When the auxiliary equations are employed in a simulation, they are appended directly to the basic nonisothermal flow problem as given in Section 3.1. Recalling the basic matrix equation (45) and adding (60) to this set produces

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N}_\alpha \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{U}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{T}} \\ \dot{\mathbf{c}}_\alpha \end{Bmatrix}$$

$$+ \begin{bmatrix} \mathbf{C}(\mathbf{U}) + \mathbf{K}(\mathbf{U},\mathbf{T}) & -\mathbf{Q} & \mathbf{B}(\mathbf{T}) & \mathbf{B}_\alpha(\mathbf{c}_\alpha) \\ -\mathbf{Q}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}(\mathbf{U}) + \mathbf{L}(\mathbf{T}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_\alpha(\mathbf{U}) + \mathbf{L}_\alpha(\mathbf{c}_\alpha) \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{T} \\ \mathbf{c}_\alpha \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(\mathbf{T},\mathbf{c}_\alpha) \\ \mathbf{0} \\ \mathbf{G}(\mathbf{T},\mathbf{U}) \\ \mathbf{G}_\alpha(\mathbf{c}_\alpha) \end{Bmatrix}. \tag{61}$$

Note that other forms of this equation are possible depending on the use of a penalty approximation or the porous flow equations. Also, all of the possible dependencies on c_α have not been indicated in (61) since these are highly dependent on the physical definition for c_α.

4 Elements and Element Matrix Constuction

The previous section in conjunction with Appendix A has presented a formalism whereby the continuum boundary value problem of interest can be reduced to a finite dimensional system of matrix equations. In the present section some of the details involving the actual construction of the required element matrices will be discussed.

Of central importance to the development of a finite element procedure is the choice of particular elements to be included in the element library. For the class of problems considered here, the choice of interpolation functions for the dependent variables within an element is constrained by the special role the pressure plays in incompressible flows. Basically, the pressure can be interpreted as a Lagrange multiplier that serves to enforce incompressibility. From equations (41) and (A2) it is seen that the pressure approximation function $\mathbf{\Psi}$ is the weighting function for the incompressibility constraint. It has been shown, both numerically [26] and theoretically [27], that in order to prevent an overconstrained system of discrete equations, the pressure interpolation must be at least one order lower than the approximation order for the velocity. Numerous element types satisfying this condition have been studied and evaluated. In the present version of NACHOS II three basic elements have been selected : a six-node triangle, an eight-node quadrilateral and a nine-node quadrilateral. The interpolation functions for each of these elements are described below along with the permissable approximations that can be used for the pressure in each element.

4.1 Triangular Elements

The triangular element used in NACHOS II is a six node element with nodes located at the vertices and midsides as shown in Figure 3. The velocity components, temperature and auxiliary variables are approximated by quadratic functions defined on the triangle by

$$\mathbf{\Phi} = \mathbf{\Theta} = \mathbf{\Pi} = \left\{ \begin{array}{c} L_1(2L_1 - 1) \\ L_2(2L_2 - 1) \\ L_3(2L_3 - 1) \\ 4L_1L_2 \\ 4L_2L_3 \\ 4L_3L_1 \end{array} \right\} \tag{62}$$

where the ordering of the functions corresponds to the ordering of the nodes shown in Figure 3. The functions in (62) are expressed in terms of the area or natural coordinates for a triangle [3,4]. Note that the coordinates L_i, which range from 0 to 1, are not independent but are related by $L_1 + L_2 + L_3 = 1$.

Two different pressure approximations are available with the six-node triangle. The first is a continuous, linear approximation in which the pressure is defined at the vertex

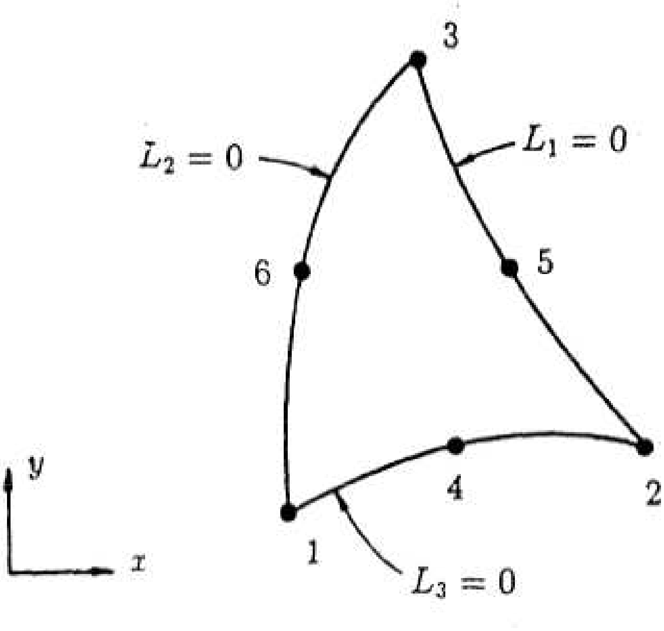

Figure 3: Six-node triangular element.

nodes and the interpolation function is defined by

$$\mathbf{\Psi} = \left\{ \begin{array}{c} L_1 \\ L_2 \\ L_3 \end{array} \right\} . \tag{63}$$

A second approximation involves a discontinuous, linear function defined on the element by

$$\mathbf{\Psi} = \left\{ \begin{array}{c} 1 \\ x \\ y \end{array} \right\} \tag{64}$$

in which the pressure unknowns are not nodal point values of the pressure but correspond to the coefficients in $P = a + bx + cy$. In equation (64) the coordinates are defined in the global coordinate system for the problem.

When the element interpolation functions are written in terms of the area coordinates L_i the relationship between the physical coordinates x, y (or r, z) and the element coordinates is obtained from the parametric mapping concept originally discussed by Ergatoudis, *et al* [28]. That is, the coordinate transformation is given by

$$x = \mathbf{\Upsilon}^T \mathbf{x} \quad ; \quad y = \mathbf{\Upsilon}^T \mathbf{y} \tag{65}$$

where $\mathbf{\Upsilon}$ is a vector of interpolation functions on the triangle and the $\mathbf{x}, \mathbf{y}$ are vectors of coordinates describing the geometry of the element (generally, nodal point coordinates).

The transformation given in (65) is quite general and allows for the description of curved-sided elements. In the present case, if $\mathbf{\Upsilon} = \mathbf{\Phi}$ (equation (62)), a quadratic interpolation of the element boundary is possible and the element is termed isoparametric (*i.e.*, the functions defining the primary dependent variables are the same order as the functions defining the element geometry). If $\mathbf{\Upsilon} = \mathbf{\Psi}$ (equation (63)), a linear interpolation of the element boundary is possible and the element is subparametric. The present version of NACHOS II incorporates both iso and subparametric versions of the six-node triangle.

4.2 Quadrilateral Elements

Two types of quadrilateral elements are used in NACHOS II – an eight node element and a nine node element. For the eight-node element the velocity components, temperature and auxiliary variables are approximated by the "serendipity", biquadratic functions [4] given by

$$\mathbf{\Phi} = \mathbf{\Theta} = \mathbf{\Pi} = \left\{ \begin{array}{c} 1/4(1-s)(1-t)(-s-t-1) \\ 1/4(1+s)(1-t)(s-t-1) \\ 1/4(1+s)(1+t)(s+t-1) \\ 1/4(1-s)(1+t)(-s+t-1) \\ 1/2(1-s^2)(1-t) \\ 1/2(1+s)(1-t^2) \\ 1/2(1-s^2)(1+t) \\ 1/2(1-s)(1-t^2) \end{array} \right\} . \tag{66}$$

The ordering of the functions in (66) corresponds to the ordering of the nodes shown in Figure 4. The above interpolation functions are written in terms of the normalized or natural coordinates for the element, s, t, which vary from -1 to $+1$ as shown in the figure.

Like the triangle, the eight-node quadrilateral permits two different pressure approximations. The continuous, bilinear pressure is defined at the corner nodes of the element by an interpolation function of the form

$$\mathbf{\Psi} = \left\{ \begin{array}{c} 1/4(1-s)(1-t) \\ 1/4(1+s)(1-t) \\ 1/4(1+s)(1+t) \\ 1/4(1-s)(1+t) \end{array} \right\} . \tag{67}$$

A discontinuous, linear approximation for the pressure is given by

$$\mathbf{\Psi} = \left\{ \begin{array}{c} 1 \\ x \\ y \end{array} \right\} \tag{68}$$

with $P = a + bx + cy$ within the element. Again, the coefficients a, b, c are not associated with the nodal points.

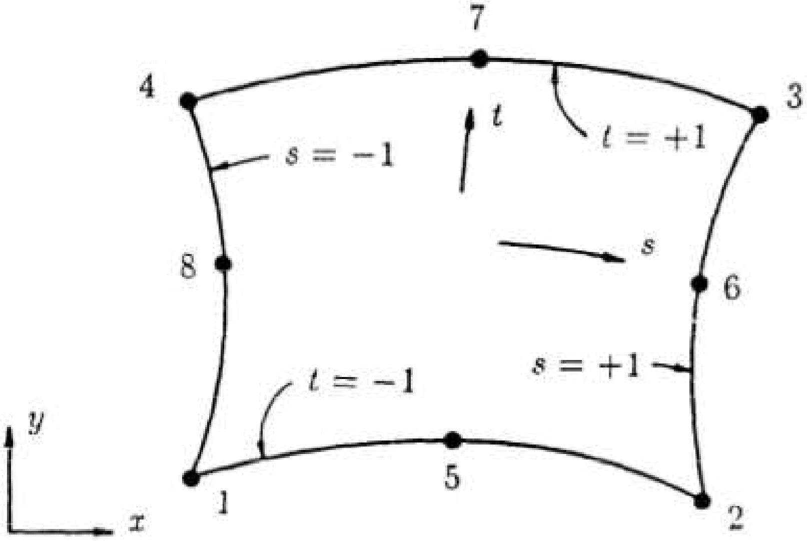

Figure 4: Eight-node quadrilateral element.

The nine-node element is similar to the "serendipity" element but with a ninth node added at the centroid of the element (see Figure 5). The primary unknowns are interpolated using the biquadratic, Lagrange functions

$$\mathbf{\Phi} = \mathbf{\Theta} = \mathbf{\Pi} = \left\{ \begin{array}{c} 1/4(1-s)(1-t)(-s-t-1) + 1/4(1-s^2)(1-t^2) \\ 1/4(1+s)(1-t)(s-t-1) + 1/4(1-s^2)(1-t^2) \\ 1/4(1+s)(1+t)(s+t-1) + 1/4(1-s^2)(1-t^2) \\ 1/4(1-s)(1+t)(-s+t-1) + 1/4(1-s^2)(1-t^2) \\ 1/2(1-s^2)(1-t) - 1/2(1-s^2)(1-t^2) \\ 1/2(1+s)(1-t^2) - 1/2(1-s^2)(1-t^2) \\ 1/2(1-s^2)(1+t) - 1/2(1-s^2)(1-t^2) \\ 1/2(1-s)(1-t^2) - 1/2(1-s^2)(1-t^2) \\ (1-s^2)(1-t^2) \end{array} \right\} \tag{69}$$

which are expressed in terms of the local element coordinate system.

The pressure approximations for the Lagrange element are the same as those given for the eight-node element. Thus, the pressure function can be continuous and described by

$$\mathbf{\Psi} = \left\{ \begin{array}{c} 1/4(1-s)(1-t) \\ 1/4(1+s)(1-t) \\ 1/4(1+s)(1+t) \\ 1/4(1-s)(1+t) \end{array} \right\} \tag{70}$$

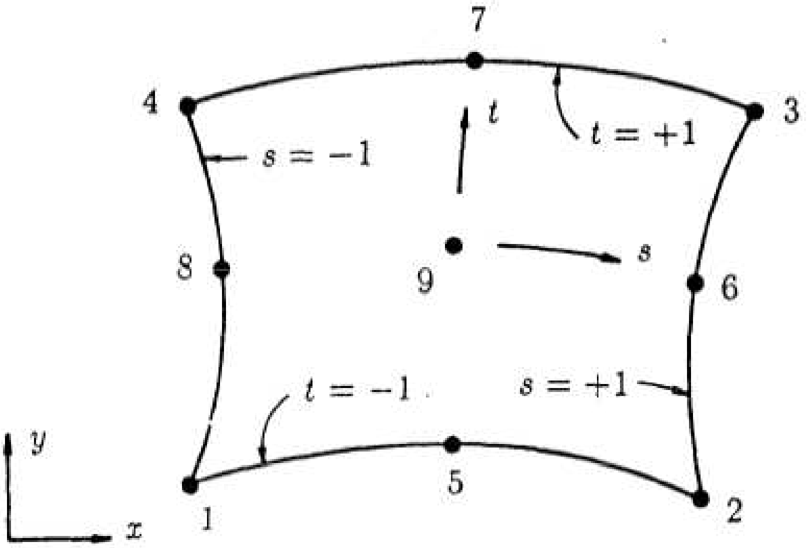

Figure 5: Nine-node quadrilateral element.

or discontinuous and given by

$$\boldsymbol{\Psi} = \left\{ \begin{array}{c} 1 \\ x \\ y \end{array} \right\} . \tag{71}$$

The parametric mapping concept described for the triangular element is also available for use with the quadrilaterals. Therefore, to relate the global coordinates x, y (or r, z) to the local s, t system, let

$$x = \boldsymbol{\Upsilon}^T \mathbf{x} \quad ; \quad y = \boldsymbol{\Upsilon}^T \mathbf{y}. \tag{72}$$

For an isoparametric quadrilateral then $\boldsymbol{\Upsilon} = \boldsymbol{\Phi}$ (equations (66) or (69)) ; the subparametric element requires $\boldsymbol{\Upsilon} = \boldsymbol{\Psi}$ (equations (67) or (70)).

4.3 Spatial Derivatives and Integrals

The construction of the various finite element coefficient matrices requires the integration of combinations of the interpolation functions and their spatial derivatives over the volume (area) of the element. The integration process is most easily carried out in the normalized or natural coordinate system for an element since the limits of integration are simple and independent of the global coordinates. The shape functions presented in the previous two sections were expressed in the natural coordinate system for the element. There remains the task of expressing spatial derivatives of the shape functions in the same normalized coordinates. The following relations, based on the chain rule and the parametric mapping ideas, are needed

$$\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial s}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial t}\end{matrix}\right\} = \begin{bmatrix}\frac{\partial x}{\partial s} & \frac{\partial y}{\partial s}\\ \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t}\end{bmatrix}\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial x}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial y}\end{matrix}\right\} = \begin{bmatrix}\frac{\partial\mathbf{\Upsilon}}{\partial s}^T\mathbf{x} & \frac{\partial\mathbf{\Upsilon}}{\partial s}^T\mathbf{y}\\ \\ \frac{\partial\mathbf{\Upsilon}}{\partial t}^T\mathbf{x} & \frac{\partial\mathbf{\Upsilon}}{\partial t}^T\mathbf{y}\end{bmatrix}\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial x}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial y}\end{matrix}\right\} = \mathbf{J}\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial x}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial y}\end{matrix}\right\} \tag{73}$$

where $\mathbf{\Lambda}$ represents any of the element interpolation functions (*e.g.*, $\mathbf{\Phi}, \mathbf{\Theta}, \mathbf{\Psi}$) and $\mathbf{J}$ is the Jacobian of the transformation from global coordinates x, y (or r, z) to the local element coordinates s, t. The parametric mapping scheme defined by (65) and (72) has been used to define the variation of x, y. Inverting (73) provides the required definition of the spatial derivatives of the shape functions in terms of the local element coordinates

$$\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial x}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial y}\end{matrix}\right\} = \mathbf{J^{-1}}\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial s}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial t}\end{matrix}\right\} = \frac{1}{|\mathbf{J}|}\begin{bmatrix}\frac{\partial\mathbf{\Upsilon}}{\partial t}^T\mathbf{y} & -\frac{\partial\mathbf{\Upsilon}}{\partial s}^T\mathbf{y}\\ \\ -\frac{\partial\mathbf{\Upsilon}}{\partial t}^T\mathbf{x} & \frac{\partial\mathbf{\Upsilon}}{\partial s}^T\mathbf{x}\end{bmatrix}\left\{\begin{matrix}\frac{\partial\mathbf{\Lambda}}{\partial s}\\ \\ \frac{\partial\mathbf{\Lambda}}{\partial t}\end{matrix}\right\} \tag{74}$$

where $|\mathbf{J}|$ indicates the determinant of the Jacobian matrix $\mathbf{J}$.

In performing integrations over the element volume it is necessary to transform the integration variables and limits from the global coordinates to the local element coordinates. The differential elemental volume (area in two dimensions) transforms according to

$$d\Omega = dx\,dy = |\mathbf{J}|\,ds\,dt \tag{75a}$$

for planar geometries and

$$d\Omega = r\,d\Theta\,dr\,dz = 2\pi\,r\,|\mathbf{J}|\,ds\,dt \tag{75b}$$

for axisymmetric geometries, where the circumferential dependence has been explicitly evaluated to produce the 2π factor. In the axisymmetric case the radius r would be interpolated by $r = \mathbf{\Upsilon}^T\mathbf{r}$. The integration limits for the integrals transform to the limits on the local coordinates s, t, *i.e.*, -1 to $+1$.

In the above equations the s, t variables for a quadrilateral have been used for the purpose of explanation. Similar relations for a triangular element can be derived by replacing s, t with L_1 and L_2. The L_3 variable does not enter the formulas due to the relation $L_1 + L_2 + L_3 = 1$.

4.4 Matrix Evaluation

With the definitions of the previous three sections it is now possible to derive a computational form for the matrix coefficients involved in the finite element equations of Section

3. For purposes of discussion, only a representative term from the matrix system will be considered in detail; the evaluation of the remaining terms follows in a similar manner.

Consider the (shear) component of the diffusion matrix given by equation (A7)

$$\mathbf{K_{12}} = \mathbf{K_{xy}} = \int_{\Omega} \boldsymbol{\Psi}^T \hat{\mu} \frac{\partial \boldsymbol{\Phi}}{\partial x} \frac{\partial \boldsymbol{\Phi}^T}{\partial y}\, d\Omega \tag{76}$$

which will be evaluated for a planar, eight-node, quadrilateral element. From the previous definitions in (74) and (75), equation (76) can be written as

$$\begin{aligned} \mathbf{K_{xy}} \quad &= \quad \int_{-1}^{+1}\int_{-1}^{+1} \boldsymbol{\Psi}^T \hat{\mu} \underbrace{\frac{1}{|\mathbf{J}|}\left[\frac{\partial \boldsymbol{\Upsilon}^T}{\partial t}\,\mathbf{y}\,\frac{\partial \boldsymbol{\Phi}}{\partial s} - \frac{\partial \boldsymbol{\Upsilon}^T}{\partial s}\,\mathbf{y}\,\frac{\partial \boldsymbol{\Phi}}{\partial t}\right]}_{\frac{\partial \boldsymbol{\Phi}}{\partial x}} \\ &\quad \cdot \underbrace{\left[-\frac{\partial \boldsymbol{\Upsilon}^T}{\partial t}\,\mathbf{x}\,\frac{\partial \boldsymbol{\Phi}^T}{\partial s} + \frac{\partial \boldsymbol{\Upsilon}^T}{\partial s}\,\mathbf{x}\,\frac{\partial \boldsymbol{\Phi}^T}{\partial t}\right]\frac{1}{|\mathbf{J}|}}_{\frac{\partial \boldsymbol{\Phi}^T}{\partial y}} |\mathbf{J}|\, dsdt \end{aligned}$$

with

$$|\mathbf{J}| = \frac{\partial \boldsymbol{\Upsilon}^T}{\partial t}\,\mathbf{y}\,\frac{\partial \boldsymbol{\Upsilon}^T}{\partial s}\,\mathbf{x} - \frac{\partial \boldsymbol{\Upsilon}^T}{\partial t}\,\mathbf{x}\,\frac{\partial \boldsymbol{\Upsilon}^T}{\partial s}\,\mathbf{y}. \tag{77}$$

For an isoparametric (curve-sided) element $\boldsymbol{\Upsilon} = \boldsymbol{\Phi}$, while a subparametric (straight-sided) element requires $\boldsymbol{\Upsilon} = \boldsymbol{\Psi}$. Thus, for the isoparametric case

$$\mathbf{K_{xy}} = \int_{-1}^{+1}\int_{-1}^{+1} \boldsymbol{\Psi}^T \hat{\mu}\left[\frac{\partial \boldsymbol{\Phi}^T}{\partial t}\,\mathbf{y}\,\frac{\partial \boldsymbol{\Phi}}{\partial s} - \frac{\partial \boldsymbol{\Phi}^T}{\partial s}\,\mathbf{y}\,\frac{\partial \boldsymbol{\Phi}}{\partial t}\right]\left[-\frac{\partial \boldsymbol{\Phi}^T}{\partial t}\,\mathbf{x}\,\frac{\partial \boldsymbol{\Phi}^T}{\partial s} + \frac{\partial \boldsymbol{\Phi}^T}{\partial s}\,\mathbf{x}\,\frac{\partial \boldsymbol{\Phi}^T}{\partial t}\right]\frac{dsdt}{|\mathbf{J}|}. \tag{78}$$

The above integral is of the general form

$$I = \int_{-1}^{+1}\int_{-1}^{+1} f(s,t)\, dsdt \tag{79}$$

where $f(s,t)$ is a rational function of the normalized coordinates. All of the element matrices are of this form and can be conveniently evaluated using a numerical quadrature procedure. That is, the integral in (79) can be evaluated by the formula

$$I = \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j f(s_i, t_j) \tag{80}$$

where W_i are weighting coefficients, s_i, t_j are quadrature points in the integration interval and n is the number of quadrature points in the formula. For quadrilateral elements NACHOS II employs a product Gauss quadrature rule as shown in (80) with $n = 3$. Triangular elements are evaluated using a seven point integration rule due to Hammer, *et al* [29].

Application of the quadrature formula in (80) to the integral in equation (78) produces the element coefficient matrix $\mathbf{K_{xy}}$. Since the integrand in (78) contains three separate interpolation functions ($\mathbf{\Psi}$, $\frac{\partial \mathbf{\Phi}}{\partial x}$, $\frac{\partial \mathbf{\Phi}}{\partial y}$) the numerically evaluated coefficients form a three-dimensional array or hypermatrix with dimensions $l \times m \times m$ where l and m are the dimensions (length) of the $\mathbf{\Psi}$ and $\mathbf{\Phi}$ vectors, respectively. When values of the l nodal point viscosities are known, a hypermatrix, vector product on the l index yields the standard, two-dimensional element matrix. A similar procedure is used for other variable coefficient terms in the element matrix equations, *e.g.*, advection terms. Terms without variable coefficients are constructed directly as two-dimensional arrays.

4.5 Penalty Matrix Evaluation

The penalty function formulation of the incompressibility constraint was considered in Section 3.2. In order to implement this approximation an additional matrix of the form

$$\mathbf{K_p} = \frac{1}{\epsilon}\mathbf{Q}\mathbf{M_p}^{-1}\mathbf{Q}^T \tag{81}$$

must be constructed, where the matrices $\mathbf{Q}$ and $\mathbf{M_p}$ are defined in Appendix A.

For efficient numerical evaluation of the element level $\mathbf{K_p}$ matrix, it is important that the terms $\mathbf{Q}$ and $\mathbf{M_p}^{-1}$ be easily constructed. The $\mathbf{Q}$ matrix presents no difficulty and is easily evaluated by standard procedures. However, for $\mathbf{M_p}^{-1}$ to be evaluated requires that $\mathbf{M_p}$ be invertible at the element level. By definition

$$\mathbf{M_p} = \int_\Omega \mathbf{\Psi}\mathbf{\Psi}^T \, d\Omega \tag{82}$$

where $\mathbf{\Psi}$ is the interpolation function for the pressure. To be invertible at the element level the function $\mathbf{\Psi}$ should be defined only within the element, *i.e.*, be discontinuous between elements. One possible choice for such a function is the linear expansion $P = a + bx + cy$ as given in Sections 4.1 and 4.2 where

$$\mathbf{\Psi} = \left\{ \begin{array}{c} 1 \\ x \\ y \end{array} \right\}. \tag{83}$$

Using the definition in (83), the integral in (82) can be evaluated by a numerical quadrature procedure; the resulting (symmetric) 3 × 3 matrix is easily inverted by standard techniques to produce $\mathbf{M_p}^{-1}$.

The $\mathbf{\Psi}$ function given in (83) is not the only possible discontinuous pressure approximation. A bilinear interpolation defined at the 2 × 2 Gauss points in a quadrilateral element is also possible. However, this latter type of representation can lead to certain pathologies in the pressure solution (*i.e.*, checkerboard pattern) which requires an inconvenient filtering operation to ultimately obtain useful results. Implementation of the penalty method within NACHOS II is therefore limited to the use of the linear pressure approximation as given in (83).

4.6 Element Boundary Conditions and Source Terms

In this section the construction of boundary conditions and volumetric source terms for the element matrix equations are considered. Though the required force vectors are numerically evaluated in the same manner as the coefficient matrices, a number of additional assumptions and details are necessary that require further comment. Once again, the discussion will be focused on the momentum and energy equations since the treatment of forcing functions for the porous flow and auxiliary transport equations is identical.

4.6.1 Volumetric Forces

The force vectors for both the momentum and energy equations consist of two parts: a part due to volumetric forces (sources) and a part due to surface forces (fluxes). Consider first the volumetric terms, of which the heat source is typical

$$\mathbf{G_s} = \int_{\Omega} \mathbf{\Theta} Q \, d\Omega. \tag{84}$$

As given previously, the elemental volume (area in two-dimensions) can be expressed in terms of the normalized coordinates. To complete the specification of the integrand, the variation of the heat source within the element is required. Following the previous treatment of variable coefficients, Q may be interpolated such that

$$Q = \mathbf{\Psi}^T \mathbf{Q} \tag{85}$$

where $\mathbf{Q}$ are nodal point values of the volumetric heating. For most applications a bilinear variation of the source is sufficient and thus the choice of $\mathbf{\Psi}$ as the interpolating function. Combining these assumptions into (84) produces

$$\mathbf{G_s} = \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{\Theta} \mathbf{\Psi}^T |\mathbf{J}| \, dsdt \, \mathbf{Q} \tag{86}$$

for a planar quadrilateral. The integral in (86) can be evaluated with a standard numerical quadrature rule to produce a coefficient matrix, $\mathbf{H}$, with

$$\mathbf{G_s} = \mathbf{HQ}. \tag{87}$$

For known values of the nodal point heat source the matrix product in (87) produces the needed components of the element force vector. The above formulation permits an arbitrary variation of the heat source with temperature, spatial location, time, *etc.*

Similar procedures are invoked for the evaluation of the viscous dissipation term in the energy equation and the body force terms in the momentum equation. The viscous dissipation is assumed to be constant over an element which results in a force vector of the form

$$\mathbf{G_v} = \int_{-1}^{+1}\int_{-1}^{+1} \mathbf{\Theta} 4\mu I_2 |\mathbf{J}| \, dsdt = 4\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{\Theta} |\mathbf{J}| \, dsdt \, \mu I_2 \tag{88}$$

where μ and I_2 (see equation(27)) are evaluated at the element centroid.

In the body force terms of the momentum equation g_i is assumed constant over the element but the volumetric expansion coefficients are approximated by a bilinear interpolation function. Thus, for example, the thermal buoyancy term (equation (A7)) is given by

$$\mathbf{F_b} = \int_{-1}^{+1}\int_{-1}^{+1} \rho_0 g_i \mathbf{\Phi}\mathbf{\Psi}^T \hat{\beta} \mathbf{\Theta}^T |\mathbf{J}| \, dsdt \, (\mathbf{T} - T_0) \tag{89}$$

where $\hat{\beta}$ is a vector of nodal point values of the volumetric expansion coefficient. The presence of three interpolation functions in the integrand again results in a hypermatrix form for the coefficients; for known values of $\hat{\beta}$ and $\mathbf{T}$ the appropriate matrix products produce the needed body force vector.

4.6.2 Surface Forces

The remaining force vectors in the momentum and energy equations arise from surface forces and fluxes distributed along element boundaries. These terms need only be considered for those element sides coinciding with the "exterior" boundaries of the problem domain; contributions from interior element boundaries are cancelled by adjoining elements. As an example of a boundary term consider the heat flux term

$$\mathbf{G_f} = \int_{\Gamma} \mathbf{\Theta} q_i n_i \, d\Gamma \tag{90}$$

where Γ is the boundary of the element and $q_i n_i$ is the heat flux normal to the boundary. The boundary segment $d\Gamma$ for an element can be expressed in parametric form as

$$d\Gamma = \left[\left(\frac{\partial x}{\partial s} \right)^2 + \left(\frac{\partial y}{\partial s} \right)^2 \right]^{\frac{1}{2}} ds \tag{91}$$

where s is the normalized coordinate along the boundary as shown in Figure 6. Invoking the parametric mapping concept such that

$$x = \hat{\mathbf{\Upsilon}}^T \mathbf{x} \quad ; \quad y = \hat{\mathbf{\Upsilon}}^T \mathbf{y}$$

allows the boundary increment to be rewritten as

$$d\Gamma = \left[\left(\frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{x} \right)^2 + \left(\frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{y} \right)^2 \right]^{\frac{1}{2}} ds = \Delta \, ds. \tag{92}$$

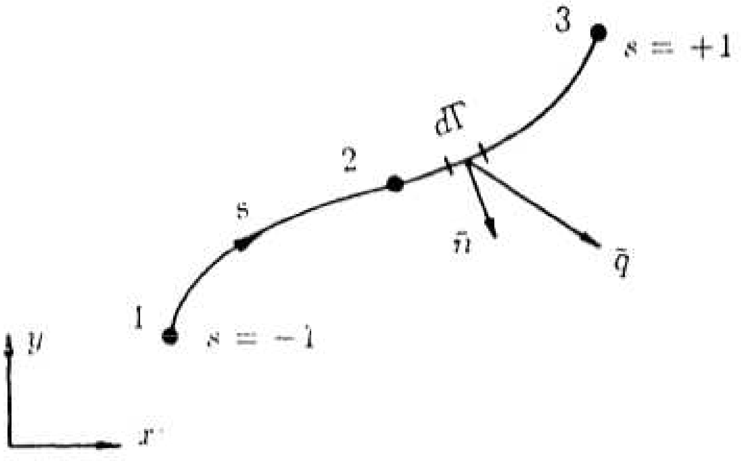

Figure 6: Nomenclature for element boundary geometry.

The ˆ notation indicates the restriction of the interpolation function to the element boundary, *i.e.*, the use of a one-dimensional or edge function. The function $\hat{\mathbf{\Upsilon}}$ may be either a linear or quadratic function, depending on the type of mapping used to describe the element geometry. For the case of a linear mapping

$$\hat{\mathbf{\Upsilon}} = \hat{\mathbf{\Psi}} = \left\{ \begin{array}{c} 1/2(1-s) \\ 1/2(1+s) \end{array} \right\} \tag{93}$$

and for the quadratic case

$$\hat{\mathbf{\Upsilon}} = \hat{\mathbf{\Phi}} = \hat{\mathbf{\Theta}} = \hat{\mathbf{\Pi}} = \left\{ \begin{array}{c} 1/2(s-1)s \\ (1-s^2) \\ 1/2(s+1)s \end{array} \right\}. \tag{94}$$

To complete the specification of the integrand in (90) the variation of $q_i n_i$ with s is required. From the boundary condition definitions in equations (20)-(21) the normal heat flux consists of three components

$$q_i n_i = f^q - q_c - q_r = f^q - h_c(T - T_c) - h_r(T - T_r) \tag{95}$$

In the present version of NACHOS II the applied flux f^q is assumed constant along an element boundary; the convective heat transfer coefficient and sink temperature are also assumed constant. For the radiative case the effective heat transfer coefficient h_r is interpolated quadratically along the element edge while the corresponding sink temperature is assumed constant. Under these assumptions the heat flux vector can be written as

$$\mathbf{G_f} = \int_{-1}^{+1} \hat{\mathbf{\Theta}} f^q \Delta\, ds - \int_{-1}^{+1} \hat{\mathbf{\Theta}} h_c \hat{\mathbf{\Theta}}^T \Delta\, ds\, (\mathbf{T} - T_c) - \int_{-1}^{+1} \hat{\mathbf{\Theta}} \hat{\mathbf{\Theta}}^T \mathbf{h_r} \hat{\mathbf{\Theta}}^T \Delta\, ds\, (\mathbf{T} - T_r) \tag{96}$$

or

$$\mathbf{G_f} = \mathbf{G_q} - \mathbf{G_c T} + \mathbf{G_c} T_c - \mathbf{G_r T} + \mathbf{G_r} T_r. \tag{97}$$

The integrals in (96) are evaluated using a three point Gauss quadrature rule. Note that some of the terms in $\mathbf{G_f}$ contain unknown element temperatures ($\mathbf{G_c T}$ and $\mathbf{G_r T}$); for solution purposes these terms are moved from the force vector to the left-hand-side of the matrix equation in (61).

A computation similar to the above may be carried out for the boundary integrals in the momentum equation. From equation (A7) the applied surface forces are given by

$$\mathbf{F_i} = \int_\Gamma \mathbf{\Phi} \mathcal{T}_i \, d\Gamma = \int_\Gamma \mathbf{\Phi} \tau_{ij} n_j \, d\Gamma$$

which for a planar, two-dimensional problem become

$$\mathbf{F_1} = \mathbf{F_x} = \int_\Gamma \mathbf{\Phi} \mathcal{T}_x \, d\Gamma = \int_\Gamma \mathbf{\Phi} \tau_{xx} n_x \, d\Gamma + \int_\Gamma \mathbf{\Phi} \tau_{xy} n_y \, d\Gamma$$

$$\mathbf{F_2} = \mathbf{F_y} = \int_\Gamma \mathbf{\Phi} \mathcal{T}_y \, d\Gamma = \int_\Gamma \mathbf{\Phi} \tau_{yx} n_x \, d\Gamma + \int_\Gamma \mathbf{\Phi} \tau_{yy} n_y \, d\Gamma. \tag{98}$$

In general, the force vector for each equation contains contributions from both an applied normal stress and a shear stress. Boundary conditions could be specified directly in terms of the quantities shown in equation (98), *i.e.*, the x and y components of the traction or the three independent components of the stress tensor. However, for boundaries with a complex shape, such forces are most conveniently described in a local coordinate system that is oriented normal and tangent to the element boundary. In this case the components of the traction vector are

$$\mathcal{T}_n = \tau_{nn} \mathbf{n} \qquad ; \qquad \mathcal{T}_t = \tau_{tn} \mathbf{n} \tag{99}$$

with

$$\mathbf{n} = n_x \mathbf{e_x} + n_y \mathbf{e_y}$$

where $\mathbf{n}$ is the normal to the element boundary and $\mathbf{e_i}$ are unit vectors in the coordinate directions (see Figure 6. The parametric description of the element boundary allows the components of the normal to be written as

$$n_x = \frac{1}{\Delta} \frac{\partial y}{\partial s} \qquad ; \qquad n_y = -\frac{1}{\Delta} \frac{\partial x}{\partial s}$$

where Δ is defined in equation (92). The x and y components of the traction can be expressed in terms of (99) as

$$\mathcal{T}_x = \tau_{nn} n_x + \tau_{tn} n_x$$

$$\mathcal{T}_y = \tau_{tn} n_y + \tau_{nn} n_y.$$

Combining all of the above definitions into the appropriate boundary integrals in (98) results in

$$\mathbf{F_x} = \int_{-1}^{+1} \mathbf{\Phi} \tau_{nn} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{y} \, ds - \int_{-1}^{+1} \mathbf{\Phi} \tau_{tn} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{x} \, ds$$

$$\mathbf{F_y} = \int_{-1}^{+1} \mathbf{\Phi} \tau_{tn} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{y}\, ds - \int_{-1}^{+1} \mathbf{\Phi} \tau_{nn} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{x}\, ds. \tag{100}$$

In writing (100) the parametric mapping concept was used to express the x and y coordinates in terms of the element edge functions $\hat{\mathbf{\Upsilon}}$; the edge functions can be either linear or quadratic depending on the shape of the element.

To complete the specification of the boundary forces, an assumption is made that the applied normal and shear stresses are constant along the element boundary. Using a form of the boundary condition given in (18) permits (100) to be rewritten as follows

$$\mathbf{F_x} = \int_{-1}^{+1} \mathbf{\Phi} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{y}\, ds\, f^{nn} - \int_{-1}^{+1} \mathbf{\Phi} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{x}\, ds\, f^{tn}$$

$$\mathbf{F_y} = \int_{-1}^{+1} \mathbf{\Phi} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{y}\, ds\, f^{tn} - \int_{-1}^{+1} \mathbf{\Phi} \frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{x}\, ds\, f^{nn}. \tag{101}$$

The indicated integrals are evaluated using a standard Gauss quadrature rule. The normal and shear components of the stress are specified separately. It should be emphasized that the above boundary condition is written in terms of the *total* normal stress, which is in general different from the pressure. From the constitutive relation (6) the normal stress is

$$\tau_{nn} = -\hat{P} + 2\mu \left(\frac{\partial u_n}{\partial n} \right).$$

In many practical cases, the viscous part of the normal stress is negligibly small and the normal stress is essentially equal to the pressure. When the viscous part is not negligible, the application of a normal stress boundary condition does not distinguish between contributions from the pressure and viscous parts but simply reflects their net effect.

It should be noted that by assuming constant values for the applied stress components along element boundaries, some types of flow boundaries are effectively eliminated from consideration. Free surface boundaries, in which surface tension is important, cannot generally be modeled since the radius of curvature varies along the domain boundary. Likewise, surface tension driven flows (Marangoni convection) could not be treated since the temperature dependent surface tension is a function of position on the boundary. Frictional type boundary conditions could be simulated provided the friction model is sufficiently simple. However, a slip boundary described by Navier's equation is not admissable due to its dependence on the local boundary velocity which is not constant on the boundary. Since these types of boundary conditions are somewhat specialized, the constant stress assumption is not a major restriction in most simulations. All of these conditions can be implemented in NACHOS II with some minor code modification.

4.6.3 Specified Dependent Variables

In addition to the ("natural") boundary conditions specified by the boundary integrals presented above, "essential" boundary conditions specifying particular values of the dependent variables must also be considered. Application of a specified velocity component, pressure or temperature results in the field equation for that particular nodal point quantity being replaced by a constraint equation that enforces the proper boundary value. When using a continuous pressure approximation it is important that the number of imposed pressure conditions be kept to a minimum. For each specified nodal point pressure an incompressibility constraint is removed from the system with an attendant loss of control on the conservation of mass. In many cases only a single value of the pressure needs to be specified to set the overall pressure level for the problem.

5 Solution Procedures

The major computational effort in any finite element procedure occurs in the solution of the assembled matrix equations that describe the discretized problem. This is especially true in the case of highly nonlinear equations or problems with coupled physical phenomena, both of which are generally found in the present case. In addition to computational efficiency these characteristics also introduce questions regarding the ability to achieve a solution, *i.e.*, convergence for a given set of data. The choice of a solution algorithm is therefore a critical element in the overall utility, robustness and efficiency of a computer code such as NACHOS II.

Before embarking on a description of the algorithms used in NACHOS II it is appropriate to review some of the features of the matrix problem to be solved. In its most general form the equation of interest can be written as

$$\overline{\mathbf{M}}\dot{\mathbf{V}} + \overline{\mathbf{K}}(\mathbf{U}, \mathbf{T}, \mathbf{c}_\alpha)\mathbf{V} = \overline{\mathbf{F}}(\mathbf{U}, \mathbf{T}, \mathbf{c}_\alpha)$$

or

$$\overline{\mathbf{M}}\dot{\mathbf{V}} + \overline{\mathbf{K}}(\mathbf{V})\mathbf{V} = \overline{\mathbf{F}}(\mathbf{V}). \tag{102}$$

This is a symbolic form for equation (61) in which the energy and auxiliary transport equations are combined with the momentum and continuity equations. The matrix $\overline{\mathbf{M}}$ represents a generalized mass or capacitance matrix, $\overline{\mathbf{K}}$ contains the advective and diffusive terms as well as the incompressibility constraint; $\overline{\mathbf{F}}$ provides the boundary and volumetric forcing functions. The vector $\mathbf{V}$ contains the unknown velocity components ($\mathbf{U}$), pressure ($\mathbf{P}$), temperature ($\mathbf{T}$) and auxiliary variables ($\mathbf{c}_\alpha$). The actual form of the $\overline{\mathbf{M}}$, $\overline{\mathbf{K}}$ and $\overline{\mathbf{F}}$ arrays depends on the use or omission of the porous flow models, penalty method and auxiliary transport equations. In all cases the matrices are large, sparse and banded in their structure.

Equation (102) represents a rather large class of problems ranging in complexity from isothermal flows to strongly coupled, multiprocess, buoyancy-driven flows. Many possible solution strategies could be envisioned for the various types of problems embedded in (102), though most fall into one of two catagories. In the previous version of NACHOS the general strategy was to decompose (102) into component equations and cyclically solve individual physical processes according to the degree of coupling between equations. This approach leads to great efficiency in the solution of any particular equation set (*e.g.*, continuity and momentum or energy) but generally provides reduced efficiency and robustness in the solution of strongly coupled systems. Also, code complexity is increased due to the tailoring of the cyclic processing to the type of problem. The present version of NACHOS II relies on the simpler procedure of always treating (102) as a single, complete system. Though not optimal for some types of problems (*e.g.*, weakly coupled forced convection) this approach has proved very cost-effective in the solution of strongly coupled flows.

In the following sections the solution algorithms for treating the steady and time-dependent forms of (102) are outlined. The solution procedures are defined in general terms and are meant to be applied to all of the classes and subclasses of flows defined by equation (102). When specific details of an algorithm are required, the case of non-isothermal, viscous flow will be used as an example; the treatment of other types of problems can be inferred by appropriate inclusion or omission of terms or equations.

5.1 Steady-State Algorithms

The time-independent form of (102) is

$$\overline{\mathbf{K}}(\mathbf{V})\mathbf{V} = \overline{\mathbf{F}}(\mathbf{V}) \tag{103}$$

which is recognized as a system of nonlinear, algebraic equations. The choice of an iterative solution method for a general purpose code is governed by several considerations. First, the chosen algorithm should be convergent for a wide range of problems with minimal sensitivity to variations in boundary conditions, material parameters and initial data. The rate of convergence should be reasonably high for computational economy. No one iterative method satisfies all these requirements for all problems and therefore NACHOS II contains several methods that can be used separately or in combination. All of the methods used are fixed point schemes that have their basis in the contraction mapping principle. A good discussion of this topic is available in [7].

5.1.1 Successive Substitution Method

A particularly simple iterative method with a large radius of convergence is the successive substitution (Picard, functional iteration) method described by

$$\overline{\mathbf{K}}(\mathbf{V}^n)\mathbf{V}^{n+1} = \overline{\mathbf{F}}(\mathbf{V}^n) \tag{104}$$

where the superscript indicates the iteration level. For strongly nonlinear problems the rate of convergence of (104) is generally fairly slow since it is only a first-order method. An improvement in convergence rate can sometimes be realized by use of a relaxation formula where

$$\overline{\mathbf{K}}(\mathbf{V}^n)\mathbf{V}^\star = \overline{\mathbf{F}}(\mathbf{V}^n) \tag{105}$$

and

$$\mathbf{V}^{n+1} = \alpha\mathbf{V}^n + (1-\alpha)\mathbf{V}^\star \qquad 0 \le \alpha < 1.$$

5.1.2 Newton's Method

In order to improve significantly on the rate of convergence, a second-order method, such as Newton's method, must be considered. Rewriting (103) as

$$\mathbf{R}(\mathbf{V}) = \overline{\mathbf{K}}(\mathbf{V})\mathbf{V} - \overline{\mathbf{F}}(\mathbf{V}) = \mathbf{0} \tag{106}$$

then Newton's method can be expressed as

$$\mathbf{R}(\mathbf{V}^n) = -\frac{\partial \mathbf{R}}{\partial \mathbf{V}}\bigg|_{\mathbf{V}^n}(\mathbf{V}^{n+1} - \mathbf{V}^n) = -\mathbf{J}(\mathbf{V}^n)(\mathbf{V}^{n+1} - \mathbf{V}^n)$$

which can be solved for $\mathbf{V}^{n+1}$ as

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{J}^{-1}(\mathbf{V}^n)\mathbf{R}(\mathbf{V}^n) \tag{107}$$

where $\mathbf{J}$ is the Jacobian matrix. The precise form of $\mathbf{J}$ is shown in Appendix B for the nonisothermal, viscous flow case. The Newton scheme in (107) has a superior rate of convergence compared to the simple algorithm in (104). However, the Newton method also has a somewhat smaller radius of convergence (*i.e.*, is more sensitive to the initial solution vector $\mathbf{V}^0$) and therefore the inclusion of both algorithms in a particular code is warranted. In many cases a sequential application of (104) and (107) provides the best method of solution. The Newton method can at times be improved by the use of a relaxation procedure similar to the one shown in (105).

5.1.3 Modified, Quasi-Newton and Continuation Methods

The major drawback to both the successive substitution and Newton algorithms is the computational expense involved in the solution of a large matrix problem at each iteration. One method for avoiding this expense in the Newton scheme is to not recompute the Jacobian at each iteration but instead work with a fixed iteration operator. This procedure, termed a modified Newton method, can be expressed as a variant of (107)

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{J}^{-1}(\mathbf{V}^0)\mathbf{R}(\mathbf{V}^n).$$

After the first iteration cycle the method is very inexpensive since it only requires the reduction of the force vector and a back-substitution for each cycle. Unfortunately, the method has a poor rate of convergence and is therefore not recommended.

A method that combines the general efficiency of a modified Newton method and the approximate convergence rate of the full Newton method is the quasi-Newton procedure. Rather than work with a constant or fully updated Jacobian, the quasi-Newton methods seek to approximately update the Jacobian using a simple recursive procedure. The efficiency of the method comes from the ability to update the inverse of the Jacobian, thus saving both the assembly and factorization time of the full method. A number of variants of the quasi-Newton procedure exist. The scheme used in NACHOS II is based on the Broyden update [30] as implemented by Engelman, *et al* [31]. The details of the method as used in NACHOS II are given in Appendix C.

A common failure of all of the iterative methods described previously is the lack of a sufficiently large radius of convergence. Given an initial estimate of the solution vector, the iterative method may not be able to reach a converged solution, since the intial guess was in some sense "too far away" from the required result. One method for circumventing such a difficulty is to incrementally approach the final solution through a series of intermediate solutions. These intermediate results may be of physical interest or simply be a means to obtain the required solution. The formal algorithms used to implement such a procedure are termed continuation (imbedding, incremental loading) methods and can be used in conjunction with any of the previously described iterative procedures. The continuation methods used in NACHOS II are described in Appendix D.

5.1.4 Convergence Criteria

The use of an iterative solution method necessitates the definition of a convergence and stopping criteria to terminate the iteration process. The usual measure of convergence is a norm on the change in the solution vector between successive iterations. NACHOS II employs a series of discrete norms defined by

$$\begin{aligned} d^U_{n+1} &= \frac{1}{U_{max}}\left[\sum_{i=1}^{N}\left(U_i^{n+1} - U_i^n\right)^2\right]^{\frac{1}{2}} \\ d^T_{n+1} &= \frac{1}{T_{max}}\left[\sum_{i=1}^{N}\left(T_i^{n+1} - T_i^n\right)^2\right]^{\frac{1}{2}} \qquad (108) \\ d^c_{n+1} &= \frac{1}{c_{max}}\left[\sum_{i=1}^{N}\left(c_i^{n+1} - c_i^n\right)^2\right]^{\frac{1}{2}}. \end{aligned}$$

In the definitions in (108) the subscript *max* indicates the maximum value of the variable found in the solution at iteration cycle $n+1$ and N is the total number of nodal points in the problem. The variable U indicates a velocity magnitude.

The criteria for terminating the iteration process is based on user supplied tolerances for each primary variable in the problem. The iterative algorithm is terminated when all of the appropriate inequalities listed below are satisfied.

$$\begin{aligned} d^U_{n+1} &\le \epsilon^U \\ d^T_{n+1} &\le \epsilon^T \qquad (109) \end{aligned}$$

$$d^c_{n+1} \leq \epsilon^c$$

The ϵ parameters are tolerances that are set by the user and have a default value of 0.001. The iterative process may also be terminated after a fixed number of iterations. This option acts as a backup criteria to prevent very slowly convergent or divergent problems from wasting computation time.

5.2 Transient Algorithms

The choice of a transient analysis procedure is basically governed by the same criteria as used in the selction of a steady-state algorithm. However, the inclusion of the temporal term in the matrix system requires that several additional criteria be considered. Equation (102) represents a discrete space, continuous time approximation to the original system of partial differential equations. A direct time integration procedure replaces the continuous time derivative with an approximation for the history of the dependent variables over a small portion of the problem time scale. The result is an incremental procedure that advances the solution by discrete steps in time. In constructing such a procedure, questions of numerical stablity and accuracy must be considered.

Though explicit integration methods have been used [32] in the solution of (102), preference is usually given to implicit procedures for this class of problems. The explicit methods are plagued by several difficulties, including: a) the natural implicitness of the pressure in an incompressible flow, b) severe time step restrictions needed to maintain stablility of the integration process, c) the problems of diagonalizing and inverting $\overline{\mathbf{M}}$ in a cost-effective manner for a variety of element types and d) reductions in accuracy due to the diagonalization of $\overline{\mathbf{M}}$. Implicit integration methods, though computationally expensive, are desirable due to their increased stability and the consistent treatment of the pressure. Two types of implicit integrators are available in NACHOS II. Both methods make use of a predictor/corrector cycle to improve efficiency and accuracy; both may be used with either a fixed time step or a dynamic time step selection algorithm. A solution of the resulting nonlinear, algebraic system for each timeplane is obtained by Newton's method. These procedures were originally developed by Gresho, *et al* [33] and are used in NACHOS II with only minor modifications to their original form.

5.2.1 Forward/Backward Euler Integration

The first-order integration method used in NACHOS II employs a forward Euler scheme as a predictor with the backward Euler method functioning as the corrector step. Omitting the details of the derivation, the application of the explicit, forward Euler formula to equation (102) produces

$$\overline{\mathbf{M}}\mathbf{V}^p_{n+1} = \overline{\mathbf{M}}\mathbf{V}_n + \Delta t_n \left[\overline{\mathbf{F}}(\mathbf{V}_n) - \overline{\mathbf{K}}(\mathbf{V}_n)\mathbf{V}_n\right]. \tag{110}$$

This can be written in a form that is more suitable for computation by replacing the bracketed term with a rearranged form of (102) to produce

$$\mathbf{V}^p_{n+1} = \mathbf{V}_n + \Delta t_n \dot{\mathbf{V}}_n. \tag{111}$$

In equations (110) and (111) the subscript indicates the timeplane, the superscript p denotes a predicted value and $\Delta t_n = t_{n+1} - t_n$. By using the form shown in (111) a matrix inversion of $\overline{\mathbf{M}}$ is avoided; the "acceleration" vector $\dot{\mathbf{V}}_n$ is computed from a form of the corrector formula as shown below. Note that since the forward Euler scheme is explicit, it is applied only to the velocity, temperature and auxiliary variables contained in the $\mathbf{V}$ vector. The pressure, being implicit, is not predicted with this formula.

The corrector step of the first-order scheme is provided by the backward Euler (or fully implicit) method. When applied to equation (102) this implicit method yields

$$\overline{\mathbf{M}}\mathbf{V}_{n+1} = \overline{\mathbf{M}}\mathbf{V}_n + \Delta t_n \left[\overline{\mathbf{F}}(\mathbf{V}_{n+1}) - \overline{\mathbf{K}}(\mathbf{V}_{n+1})\mathbf{V}_{n+1}\right] \tag{112}$$

or in a form more suitable for computation

$$\left[\frac{1}{\Delta t_n}\overline{\mathbf{M}} + \overline{\mathbf{K}}(\mathbf{V}_{n+1})\right]\mathbf{V}_{n+1} = \frac{1}{\Delta t_n}\overline{\mathbf{M}}\mathbf{V}_n + \overline{\mathbf{F}}(\mathbf{V}_{n+1}). \tag{113}$$

The implicit nature of this method is evident from the form of (113), since it is in effect, a nonlinear, algebraic system for the variables in $\mathbf{V}$ at timeplane $n+1$. The pressure is included in the vector $\mathbf{V}$ in (113).

The solution to (113) at timeplane $n+1$ can be achieved by an iteration procedure such as Newton's method. The rate of convergence of Newton's method is greatly increased if the initial solution estimate is "close" to the true solution. The solution predicted from (111) provides this initial guess for the iterative procedure in a cost-effective manner. Appendix B provides the details of implementing Newton's method for this procedure.

5.2.2 Adams-Bashforth/Trapezoid Rule Integration

An integration method that is second-order in time can be developed along the same lines as described above. A second-order equivalent to the forward Euler method is the variable step, Adams-Bashforth predictor given by

$$\mathbf{V}^p_{n+1} = \mathbf{V}_n + \frac{\Delta t_n}{2}\left[\left(2 + \frac{\Delta t_n}{\Delta t_{n-1}}\dot{\mathbf{V}}_n\right) - \left(\frac{\Delta t_n}{\Delta t_{n-1}}\dot{\mathbf{V}}_{n-1}\right)\right] \tag{114}$$

where $\Delta t_n = t_{n+1} - t_n$ and $\Delta t_{n-1} = t_n - t_{n-1}$. This formula can be used to predict the solution vector (excluding the pressure) given two "acceleration" vectors from previous timeplanes; no matrix solution is required.

A compatible corrector formula for use with (114) is available in the form of the trapezoid rule. When applied to equation (102) the trapezoid rule produces

$$\left[\frac{2}{\Delta t_n}\overline{\mathbf{M}} + \overline{\mathbf{K}}(\mathbf{V}_{n+1})\right]\mathbf{V}_{n+1} = \frac{2}{\Delta t_n}\overline{\mathbf{M}}\mathbf{V}_n + \overline{\mathbf{M}}\dot{\mathbf{V}}_n + \overline{\mathbf{F}}(\mathbf{V}_{n+1}). \tag{115}$$

Equation (115) is observed to be a nonlinear, algebraic system for the vector $\mathbf{V}_{n+1}$ and can again be solved using an iterative procedure such as Newton's method. Appendix B provides the details of implementing Newton's method for this procedure.

5.2.3 Integration Procedures

The integration formulas outlined above form the basis for the solution of time-dependent problems in NACHOS II. The similarity of the first- and second-order methods makes it possible to include both procedures in a single, overall algorithm. The major steps in the time integration procedure are outlined here.

At the beginning of each time step it is assumed that all of the required solution and "acceleration" vectors are known and the time increment for the next step has been selected. To advance the solution from time t_n to time t_{n+1} then requires the following steps:

1) A tentative solution vector, $\mathbf{V}^p_{n+1}$, is computed using the predictor equations (111) or (114). The pressure variables are not included in this prediction.

2) The corrector equations (113) or (115) are solved for the "true" solution, $\mathbf{V}_{n+1}$. This involves the iterative solution of (113) or (115) via Newton's method. The predicted values $\mathbf{V}^p_{n+1}$ are used to initialize the Jacobian for the Newton iteration.

3) The "acceleration" vectors are updated using the new solution $\mathbf{V}_{n+1}$ and the "inverted" forms of the corrector formulas. For the first-order method the acceleration is computed from the backward Euler definition

$$\dot{\mathbf{V}}_{n+1} = \frac{1}{\Delta t_n}(\mathbf{V}_{n+1} - \mathbf{V}_n)$$

while the second-order accelerations are derived from the trapezoid rule

$$\dot{\mathbf{V}}_{n+1} = \frac{2}{\Delta t_n}(\mathbf{V}_{n+1} - \mathbf{V}_n) - \dot{\mathbf{V}}_n.$$

4) A new integration time step is computed. The time step selection process is based on an analysis of the time truncation errors in the predictor and corrector formulas as described in Section 5.2.4. If a constant time step is being used, this step is omitted.

5) Return to step 1 for next time increment.

In actual implementation the Newton iteration process in step 2 is not carried to absolute convergence. Rather, a one-step Newton correction is employed as advocated in [33]. This procedure is quite efficient and can be very accurate provided the time step is suitably controlled.

5.2.4 Time Step Control

Both of the time integration procedures available in NACHOS II can be used with a fixed, user specified time step or a time step that changes only at certain points during the integration interval. However, the *a priori* selection and modification of a reasonable integration time step can be a difficult task, especially for a complex flow. One of the benefits of using the predictor/corrector algorithms described here is that it provides a rational basis for dynamically selecting the time step.

The detailed derivation of the time step selection formula is omitted here. The reader interested in further details is referred to [33]. The general ideas for the time step selection process come from the well-established procedures for solving ordinary differential equations. By comparing the time truncation errors for two time integration methods of comparable order, a formula can be developed to predict the next time step based on a user specified error tolerance. In the present case, the time truncation errors for the explicit predictor and implicit corrector steps are analyzed and provide the required formulas.

The time step estimation formula is given by [33] as

$$\Delta t_{n+1} = \Delta t_n \left(b \cdot \frac{\epsilon^t}{d_{n+1}} \right)^m \tag{116}$$

where $m = 1/2$, $b = 2$ for the first-order method and $m = 1/3$, $b = 3(1 + \Delta t_{n-1}/\Delta t_n)$ for the second-order scheme. The user specified error tolerance for the integration process is ϵ^t, which has a default value of 0.001. The quantity d_{n+1} is an appropriate norm on the integration error, which is defined as the difference between the predicted solution and the corrected value. In NACHOS II the following norms are used

$$d^U_{n+1} = \frac{1}{U_{max}} \left[\sum_{i=1}^{N} \left(U_{i(n+1)} - U^p_{i(n+1)} \right)^2 \right]^{\frac{1}{2}}$$

$$d^T_{n+1} = \frac{1}{T_{max}} \left[\sum_{i=1}^{N} \left(T_{i(n+1)} - T^p_{i(n+1)} \right)^2 \right]^{\frac{1}{2}} \tag{117}$$

$$d_{n+1}^{c} = \frac{1}{c_{max}} \left[\sum_{i=1}^{N} \left(c_{i(n+1)} - c_{i(n+1)}^{p} \right)^2 \right]^{\frac{1}{2}}.$$

During the integration process a series of new time steps are computed using each of the norms defined in (117). The resulting time steps are compared and the smallest is used for the next integration step. This procedure allows the dominant physical process to control the integration algorithm.

Unlike the procedure described in [33], NACHOS II always uses the newly computed time step derived from (116). If $\Delta t_{n+1} \leq 0.5 \Delta t_n$ a warning message is given to indicate a large reduction in the time step has occurred. However, the previous time step is not rejected nor recomputed.

5.2.5 Initialization

The predictor equations (111) and (114) require that one or more accleration vectors be available at each timeplane in order to estimate a new solution vector. At the beginning of a transient solution these vectors are not generally available and thus a special starting procedure must be used. The approach taken in NACHOS II is to use the dissipative backward Euler method for the first few steps and then switch to either of the standard predictor/corrector methods. This procedure has the advantage that any nonphysical features of the numerical model (*e.g.*, boundary conditions that do not initially satisify the continuity equation) are quickly damped by the backward Euler scheme.

For the first time step, the implicit, backward Euler scheme is used alone; the second step uses a forward Euler predictor and backward Euler corrector. Both of these steps use a fixed, user supplied time step. At the third step, the usual predictor/corrector integration procedure begins and automatic time step selection is started, if this option has been requested. The initial time step supplied by the user to start the problem should be very conservative to prevent large time step reductions when the automatic selection procedure takes control.

5.3 Matrix Solution Procedures

When the algorithms of the previous sections are applied at a given iteration or time step, the result is a matrix equation of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{118}$$

In the problems considered here the matrix **A** is large, sparse, banded and unsymmetric. A solution to (118) can be achieved by either an iterative (*e.g.*, Gauss-Seidel) or direct method. The most prevalent solution techniques in finite element applications are the direct methods and in particular, the various forms of Gauss elimination.

The solution method used in NACHOS II is a special form of Gauss elimination, called the frontal solution method. The basic premise of the frontal method is that the process of assembling the system matrix, **A**, from the individual element matrices and the reduction of **A** by standard Gauss elimination may be efficiently intertwined. In processing each finite element in the discretized problem, the frontal procedure passes through the following basic steps:

a) assembly of element equations into global matrix **A**,

b) check each equation in the assembled system to determine if all contributions to that equation have been made,

c) condense from the system by Gauss elimination the equations for all degrees of freedom that have been completely assembled,

d) return to step a) for the next element.

By combining the assembly and reduction process computer storage is effectively minimized since only the currently "active" (incompletely assembled) equations are retained in main memory. Following reduction of the matrix **A** to an upper triangular form, a back-substitution algorithm completes the solution process for the vector **x**.

The frontal solution package used in NACHOS II was developed by Benner [34] and represents a highly modified version of the program originally published by Hood [35]. Both of these packages were explicitly developed for the solution of unsymmetric systems and both employ a pivoting strategy to handle the occurrence of zero coefficients on the diagonal of the **A** matrix (the zeros occur due to the form of the incompressibility constraint). Benner's version of the frontal procedure has been highly vectorized for maximum efficiency on current computer hardware.

6 Pre- and Post-Processing

The NACHOS II program was designed to be a self-contained analysis package with the necessary options to set up a problem, solve for the required dependent variables and analyze the resultant solution in terms of derived quantitites and graphical output. The present section documents some of the numerical procedures used in the pre-solution and data analysis sections of the program. The inclusion of mesh generation and plotting options within NACHOS II does not preclude the use of other special purpose programs designed for these tasks. As described in the User's Manual [1], NACHOS II will read and output the files needed to communicate with a mesh generator and/or a graphics package. However, some of the built-in capabilities for data analysis will not generally be available in such stand-alone programs.

6.1 Mesh Generation

For most analysis situations, the major demand on the user of a finite element program arises from the preparation of the input data and in particular the construction of a suitable element mesh. The mesh generation scheme built into NACHOS II provides a simple method for discretizing geometries of moderate complexity with a minimum of user input.

The mesh generation scheme is based on the isoparametric mapping techniques proposed by Zienkiewicz and Phillips [36]. For purposes of grid construction, the problem domain is assumed to be made up of a collection of quadrilateral or triangular shaped parts or subregions, the distribution of which is determined by the user. Within each part an isoparametric mapping is used to approximate the region boundary. By specifying a number of x, y (or r, z) coordinates on the boundary of each part, the limits of the region and the type of interpolation functions used to define the boundary shape are determined. Linear, quadratic and cubic descriptions of the boundary shape for each side of a region are available.

The mesh points (nodes) within a region are generated automatically once the number of nodes along a boundary is specified. Local nodal point spacing is easily adjusted by use of a gradient parameter. Nodal points within a region are identified by an I,J numbering system; there is no requirement that the I,J numbering be continuous between regions. Options for copying, reflecting and filling regions are also available. Full details on the operation and capabilities of the mesh generation section of NACHOS II are described in [1].

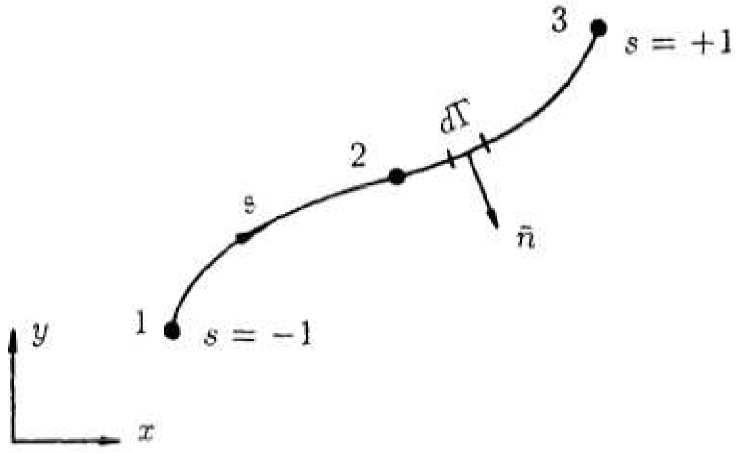

Figure 7: Definition of element boundary for stream function computation.

6.2 Stream Function Computation

A quantity that is often useful in the graphic display of computed flow fields is the stream function. For two-dimensional incompressible flows, the stream function is the remaining nonzero component of a vector potential which satisfies the conservation of mass equation identically. By definition,

$$u_1 = u_x = \frac{\partial \psi}{\partial y} \quad ; \quad u_2 = u_y = -\frac{\partial \psi}{\partial x} \tag{119}$$

and since the change in the stream function, $\delta\psi$, is an exact differential, then

$$\delta\psi = \int_A^B \mathbf{u} \cdot \mathbf{n} \, d\Gamma \tag{120}$$

with

$$\mathbf{u} = u_x \mathbf{e_x} + u_y \mathbf{e_y}$$
$$\mathbf{n} = n_x \mathbf{e_x} + n_y \mathbf{e_y}$$

where $\mathbf{n}$ is the normal to the integration path $d\Gamma$, $\mathbf{u}$ is the velocity vector along the path and $\mathbf{e_i}$ are unit vectors in the coordinate directions.

The calculation of the change in the stream function within a finite element can be carried out using (120) once a suitable integration path AB is identified. In NACHOS II the integration path is taken along the element boundaries. Consider the typical element boundary shown in Figure 7 with the following definitions

$$u_x = \hat{\mathbf{\Phi}}^T \mathbf{u_x} \quad ; \quad u_y = \hat{\mathbf{\Phi}}^T \mathbf{u_y}$$

$$x = \hat{\boldsymbol{\Upsilon}}^T \mathbf{x} \quad ; \quad y = \hat{\boldsymbol{\Upsilon}}^T \mathbf{y} \tag{121}$$

where $\hat{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{\Upsilon}}$ are interpolation (edge) functions and $\mathbf{u_x}$, $\mathbf{u_y}$, $\mathbf{x}$, $\mathbf{y}$ are vectors of nodal point velocities and coordinates. The normal vector is given by

$$\mathbf{n} = \frac{1}{\Delta}\frac{\partial y}{\partial s}\mathbf{e_x} - \frac{1}{\Delta}\frac{\partial x}{\partial s}\mathbf{e_y} \tag{122}$$

with $d\Gamma$ defined by

$$d\Gamma = \left[\left(\frac{\partial x}{\partial s} \right)^2 + \left(\frac{\partial y}{\partial s} \right)^2 \right]^{\frac{1}{2}} ds$$

or using the definitions of (121)

$$d\Gamma = \left[\left(\frac{\partial \hat{\boldsymbol{\Upsilon}}}{\partial s}^T \mathbf{x} \right)^2 + \left(\frac{\partial \hat{\boldsymbol{\Upsilon}}}{\partial s}^T \mathbf{y} \right)^2 \right]^{\frac{1}{2}} ds = \Delta\, ds.$$

Combining these relations with the definition for $\delta\psi$ produces

$$\delta\psi = \int_{-1}^{+1} \left(\frac{\partial \hat{\boldsymbol{\Upsilon}}}{\partial s}^T \mathbf{y}\ \hat{\boldsymbol{\Phi}}^T \mathbf{u_x} - \frac{\partial \hat{\boldsymbol{\Upsilon}}}{\partial s}^T \mathbf{x}\ \hat{\boldsymbol{\Phi}}^T \mathbf{u_y} \right) ds. \tag{123}$$

The interpolation function definitions were given previously in equations (93) and (94); the function $\hat{\boldsymbol{\Upsilon}}$ can be either linear or quadratic depending on the shape of the element boundary. The change in the stream function along any element boundary can be computed from (123) once the element geometry and velocity fields are specified. Computation of the stream function field for an entire finite element mesh is generated by applying (123) along successive element boundaries, starting at a node for which a base value of ψ has been specified.

The calculation of the Stokes stream function for axisymmetric flows follows a similar procedure with the appropriate definition for ψ,

$$u_1 = u_r = \frac{1}{r}\frac{\partial \psi}{\partial z} \quad ; \quad u_2 = u_z = -\frac{1}{r}\frac{\partial \psi}{\partial r} \tag{124}$$

and

$$\mathbf{u} = u_r r\ \mathbf{e_r} + u_z r\ \mathbf{e_z}$$
$$\mathbf{n} = n_r \mathbf{e_r} + n_z \mathbf{e_z}.$$

6.3 Fluid Stress Computation

The computation of the stress fields for the fluid flow follow directly from the definition of those quantities. For a planar geometry the components of the stress tensor are

$$\tau_{xx} = -P + 2\mu \frac{\partial u_x}{\partial x}$$

$$\begin{aligned}\tau_{yy} &= -P + 2\mu\frac{\partial u_y}{\partial y} \qquad (125)\\ \tau_{xy} &= \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right).\end{aligned}$$

Substitution of the finite element approximations for u_i, P and μ into (125) produces the following

$$\begin{aligned}\tau_{xx} &= -\boldsymbol{\Psi}^T\mathbf{P} + 2\boldsymbol{\Psi}^T\hat{\mu}\frac{\partial\boldsymbol{\Phi}}{\partial x}^T\mathbf{u_x}\\ \tau_{yy} &= -\boldsymbol{\Psi}^T\mathbf{P} + 2\boldsymbol{\Psi}^T\hat{\mu}\frac{\partial\boldsymbol{\Phi}}{\partial y}^T\mathbf{u_y} \qquad (126)\\ \tau_{xy} &= \boldsymbol{\Psi}^T\hat{\mu}\left(\frac{\partial\boldsymbol{\Phi}}{\partial y}^T\mathbf{u_x} + \frac{\partial\boldsymbol{\Phi}}{\partial x}^T\mathbf{u_y}\right).\end{aligned}$$

The spatial derivatives of the interpolation functions in (126) can be converted to derivatives involving the local element coordinates through use of the coordinate transformation defined in Section 4.3. When this transformation is invoked the stress components become

$$\begin{aligned}\tau_{xx} &= -\boldsymbol{\Psi}^T\mathbf{P} + \boldsymbol{\Psi}^T\hat{\mu}\frac{1}{|\mathbf{J}|}\left[\frac{\partial\boldsymbol{\Upsilon}}{\partial t}^T\mathbf{y}\,\frac{\partial\boldsymbol{\Phi}}{\partial s}^T\mathbf{u_x} - \frac{\partial\boldsymbol{\Upsilon}}{\partial s}^T\mathbf{y}\,\frac{\partial\boldsymbol{\Phi}}{\partial t}^T\mathbf{u_x}\right]\\ \tau_{yy} &= -\boldsymbol{\Psi}^T\mathbf{P} + \boldsymbol{\Psi}^T\hat{\mu}\frac{1}{|\mathbf{J}|}\left[-\frac{\partial\boldsymbol{\Upsilon}}{\partial t}^T\mathbf{x}\,\frac{\partial\boldsymbol{\Phi}}{\partial s}^T\mathbf{u_y} + \frac{\partial\boldsymbol{\Upsilon}}{\partial s}^T\mathbf{x}\,\frac{\partial\boldsymbol{\Phi}}{\partial t}^T\mathbf{u_y}\right] \qquad (127)\\ \tau_{xy} &= \boldsymbol{\Psi}^T\hat{\mu}\frac{1}{|\mathbf{J}|}\left[-\frac{\partial\boldsymbol{\Upsilon}}{\partial t}^T\mathbf{x}\,\frac{\partial\boldsymbol{\Phi}}{\partial s}^T\mathbf{u_x} + \frac{\partial\boldsymbol{\Upsilon}}{\partial s}^T\mathbf{x}\,\frac{\partial\boldsymbol{\Phi}}{\partial t}^T\mathbf{u_x} + \frac{\partial\boldsymbol{\Upsilon}}{\partial t}^T\mathbf{y}\,\frac{\partial\boldsymbol{\Phi}}{\partial s}^T\mathbf{u_y} - \frac{\partial\boldsymbol{\Upsilon}}{\partial s}^T\mathbf{y}\,\frac{\partial\boldsymbol{\Phi}}{\partial t}^T\mathbf{u_y}\right]\end{aligned}$$

where the Jacobian $\mathbf{J}$ was defined in equation (77). The above definitions are written for a quadrilateral element; the corresponding form for a triangular element can be obtained by replacing the s, t coordinates with the appropriate L_i coordinates. Also, the function $\boldsymbol{\Upsilon}$ is defined as either a quadratic or linear interpolation function depending on whether the particular element is isoparametric or subparametric.

The expressions in (127) allow the stress components to be evaluated at any point s_0, t_0 within an element for a known element geometry and solution field. Note that since the stresses depend on velocity gradients they are discontinuous between elements. The NACHOS II program permits stresses to be computed on element boundaries, midway between adjacent nodes, or at the 2×2 Gauss points within the element. When the Gauss point option is used the stresses may also be extrapolated to the corner nodes and averaged between adjacent elements to produce a continuous stress field.

The stress fields for an axisymmetric flow are computed in the same manner as shown above but with the following definitions for the stress components

$$\tau_{rr} = -P + 2\mu\frac{\partial u_r}{\partial r}$$

$$\tau_{zz} = -P + 2\mu \frac{\partial u_z}{\partial z} \tag{128}$$

$$\tau_{\theta\theta} = -P + 2\mu \frac{u_r}{r}$$

$$\tau_{rz} = \mu \left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z} \right).$$

For some applications the vorticity field is of interest and this quantity is computed in the same manner as the fluid stresses. By definition, the vorticity for a two-dimensional flow is

$$\omega = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}. \tag{129}$$

By using the same methods as outlined for the stress components the vorticity can be evaluated at any point within an element. The points of evaluation in NACHOS II are the same as those allowed for the stresses; extrapolation and averaging of the discontinuous, element based vorticity is also available as an option.

6.4 Flux Computation

The diffusion fluxes associated with the energy equation and the auxiliary transport equations can be computed in NACHOS II on an element-by-element basis. In the following only the heat fluxes will be specifically considered since the fluxes for the auxiliary equations are completely analogous. Fourier's law provides the definition of the conductive heat flux as

$$q_x = -k \frac{\partial T}{\partial x}$$

$$q_y = -k \frac{\partial T}{\partial y} \tag{130}$$

with the following definitions also required

$$q_n = \mathbf{q} \cdot \mathbf{n}$$

$$\mathbf{q} = q_x \mathbf{e_x} + q_y \mathbf{e_y}$$

$$\mathbf{n} = n_x \mathbf{e_x} + n_y \mathbf{e_y}.$$

Reference to Figure 7 and the definition in equation (122) allows the normal heat flux to be written as

$$q_n = q_x \frac{1}{\Delta} \frac{\partial y}{\partial s} - q_y \frac{1}{\Delta} \frac{\partial x}{\partial s} \tag{131}$$

and Δ is defined as before

$$\Delta = \left[\left(\frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{x} \right)^2 + \left(\frac{\partial \hat{\mathbf{\Upsilon}}^T}{\partial s} \mathbf{y} \right)^2 \right]^{\frac{1}{2}}.$$

The components of the heat flux vector are computed from (130) and the finite element approximation for T and k. Employing the coordinate transformation from Section 4.3 the fluxes are

$$q_x = -\mathbf{\Psi}^T \hat{\mathbf{k}} \frac{1}{|\mathbf{J}|} \left[\frac{\partial \mathbf{\Upsilon}}{\partial t}^T \mathbf{y} \frac{\partial \mathbf{\Theta}}{\partial s}^T \mathbf{T} - \frac{\partial \mathbf{\Upsilon}}{\partial s}^T \mathbf{y} \frac{\partial \mathbf{\Theta}}{\partial t}^T \mathbf{T} \right]$$

$$q_y = -\mathbf{\Psi}^T \hat{\mathbf{k}} \frac{1}{|\mathbf{J}|} \left[-\frac{\partial \mathbf{\Upsilon}}{\partial t}^T \mathbf{x} \frac{\partial \mathbf{\Theta}}{\partial s}^T \mathbf{T} + \frac{\partial \mathbf{\Upsilon}}{\partial s}^T \mathbf{x} \frac{\partial \mathbf{\Theta}}{\partial t}^T \mathbf{T} \right] . \tag{132}$$

The definitions in (132) are sufficient to define the flux components at any point s_0, t_0 in the element. NACHOS II evaluates the flux components on the element boundary, midway between adjacent nodes or at the 2 × 2 Gauss points. Flux components can also be combined with the definition in (131) to generate the normal flux on the element boundary; fluxes normal to the element boundary are integrated along the boundary to define the total energy transfer to or from the element.

6.5 Graphical Output

The batch plotting package in NACHOS II permits a full range of plots to be constructed at the option of the user. Plots of the finite element mesh can be generated as well as contours, time histories, profiles and vector representations of all the relevant dependent variables in an analysis. In addition to the built-in plotting, NACHOS II writes an output file that can be easily used with a stand-alone graphics package. These facilities are fully described in [1].

7 References

1. D. K. Gartling, "NACHOS II - A Finite Element Computer Program for Incompressible Flow Problems, Part II - User's Manual," SAND86-1817, Sandia National Laboratories, Albuquerque, N.M. (1986)

2. D. K. Gartling, "NACHOS II - A Finite Element Computer Program for Incompressible Flow Problems, Part III - Example Analyses," SAND86-1818, Sandia National Laboratories, Albuquerque, N.M. (1986)

3. E. B. Becker, G. F. Carey and J. T. Oden, "*Finite Elements, An Introduction, Volume I*," Prentice-Hall, New Jersey, 1981

4. O. C. Zienkiewicz, "*The Finite Element Method*," McGraw-Hill, London, 1977

5. K. H. Huebner and E. A. Thornton, "*The Finite Element Method for Engineers*," John Wiley, New York, 1982

6. J. N. Reddy, "*An Introduction to the Finite Element Method*," McGraw-Hill,New York, 1984

7. J. T. Oden, "*Finite Elements of Nonlinear Continua*," McGraw-Hill, New York, 1972

8. D. D. Gray and A. Giorgini, "The Validity of the Boussinesq Approximation for Liquids and Gases," *Int. J. Heat Mass Transfer*, **19**, 545-551 (1976)

9. D. K. Gartling and C. E. Hickox, "A Numerical Study of the Applicability of the Boussinesq Approximation for a Fluid-Saturated Porous Medium," *Int. J. Num. Meth. Fluids*,**5**, 995-1013 (1985)

10. D. A. Nield, "Non-Darcy Effects in Convection in a Saturated Porous Medium," *Convective Flows in Porous Media*, R. A. Wooding and I. White, Eds., DSIR Science Information Publishing Center, Wellington, New Zealand, 129-139 (1985)

11. D. A. Nield and D. D. Joseph, "Effects of Quadratic Drag on Convection in a Staurated Porous Medium," *Phys. Fluids*, **28**, 995-997 (1985)

12. J. Bear, *Dynamics of Fluids in Porous Media*, American Elsevier, New York, 1972

13. R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*, McGraw-Hill, New York, 1972

14. D. K. Gartling, "Finite Element Analysis of Convective Heat Transfer Problems With Change of Phase," *Computer Methods in Fluids*, K. Morgan, C. Taylor and C. A. Brebbia, Eds., Pentech Press, London, 257-284 (1980)

15. K. Morgan, "A Numerical Analysis of Freezing and Melting With Convection," *Comp. Meth. Applied Mech. Engr.*, **28**, 275-284 (1981)

16. W. Prager, *Introduction to Mechanics of Continua*, Ginn and Co., New York, 1961

17. R. B. Bird, R. C. Armstrong and O. Hassager, *Dynamics of Polymeric Liquids, Volume 1, Fluid Mechanics*, John Wiley, New York, 1977

18. J. G. Oldroyd, "A Rational Formulation of the Equations of Plastic Flow for a Bingham Solid," *Proc.Cambridge Phil. Soc.*, **43**, 100-105 (1947)

19. D. K. Gartling and N. Phan-Thien, "A Numerical Simulation of a Plastic Fluid in a Parallel-Plate Plastometer," *J. Non-Newtonian Fluid Mech.*, **14**, 347-360 (1984)

20. E. J. O'Donovan and R. I. Tanner, "Numerical Study of the Bingham Squeeze Film Problem," *J. Non-Newtonian Fluid Mech.*, **15**, 75-83 (1984)

21. H. Tennekes and J. L. Lumley, *A First Course in Turbulence*, MIT Press, Massachusetts, 1972

22. W. C. Reynolds, "Computation of Turbulent Flows," *Annual Review of Fluid Mechanics*, **8**, 183-208 (1976)

23. B. A. Finlayson *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972

24. G. F. Carey and J. T. Oden, *Finite Elements, A Second Course, Volume II*," Prentice-Hall, New Jersey, 1983

25. M. Engelman, R. Sani, P. M. Gresho and M. Bercovier, "Consistent vs. Reduced Integration Penalty Methods for Incompressible Media Using Several Old and New Elements," *Int. J. Num. Meth. Fluids*, **2**, 25-42 (1982)

26. C. Taylor and P. Hood, "A Numerical Solution of the Navier-Stokes Equations Using FEM Techniques," *Computers and Fluids*, **1**, 73-100 (1973)

27. R. L. Sani, P. M. Gresho, R. L. Lee and D. F. Griffiths, "The Cause and Cure (?) of the Spurious Pressures Generated by Certain FEM Solutions of the Incompressible Navier-Stokes Equations," *Int. J. Num. Meth. Fluids*, **1**,17-43 (1981)

28. I. Ergatoudis, B. M. Irons and O. C. Zienkiewicz, "Curved, Isoparametric, 'Quadrilateral', Elements for Finite Element Analysis," *Int. J. Solids Structures*, **4**, 31-42 (1968)

29. P. C. Hammer, O. P. Marlowe and A. H. Stroud, "Numerical Integration Over Simplexes and Cones," *Math. Tables Aids Comp.*, **10**, 130-137 (1956)

30. C. G. Broyden, "A Class of Methods for Solving Nonlinear Simultaneous Equations," *Math. Comp.*, **19**, 577-593 (1965)

31. M. S. Engelman, G. Strang and K. J. Bathe, "The Application of Quasi-Newton Methods in Fluid Mechanics," *Int. J. Num. Meth. Engr.*, **17**, 707-718 (1981)

32. P. M. Gresho, S. T. Chan, R. L. Lee and C. D. Upson, “A Modified Finite Element Method for Solving the Time-Dependent, Incompressible Navier-Stokes Equations. Part 1: Theory,” *Int. J. Num. Meth. Fluids*, **4**, 557-598 (1984)

33. P. M. Gresho, R. L. Lee and R. L. Sani,“On the Time Dependent Solution of the Incompressible Navier-Stokes Equations in Two and Three Dimensions,” *Recent Advances in Numerical Methods in Fluids, Volume 1*, Pineridge Press, Swansea, U. K., 27-81 (1980)

34. R. E. Benner, “Vectorized Frontal and Skyline Methods in Finite Element Analysis,” *Comm. Appl. Num. Meth.*, submitted (1986)

35. P. Hood, “Frontal Solution Program for Unsymmetric Matrices,” *Int. J. Num. Meth. Engr.*, **10**, 379-399 (1976)

36. O. C. Zienkiewicz and D. V. Phillips, “An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by “Isoparametric” Coordinates,” *Int. J. Num. Meth. Engr.*, **3**, 519-528 (1971)

Appendix A - Finite Element Equations for Convective Heat Transfer

The manipulations required to transform the basic field equations into a discrete system are outlined here for the case of nonisothermal, viscous flow. Using the definition of the Galerkin procedure, equation (41), and the finite element approximations in (40) allows the following integral relations to be written for the balance laws in (34), (35) and (38)

Momentum:

$$\left[\int_\Omega \rho_0 \boldsymbol{\Phi}\boldsymbol{\Phi}^T \, d\Omega\right]\dot{\mathbf{u}}_\mathbf{i} + \left[\int_\Omega \rho_0 \boldsymbol{\Phi}\boldsymbol{\Phi}^T \mathbf{u_j} \frac{\partial \boldsymbol{\Phi}}{\partial x_j}^T \, d\Omega\right]\mathbf{u_i} + \left[\int_\Omega \mu \frac{\partial \boldsymbol{\Phi}}{\partial x_j}\frac{\partial \boldsymbol{\Phi}}{\partial x_j}^T \, d\Omega\right]\mathbf{u_i}$$

$$+\left[\int_\Omega \mu \frac{\partial \boldsymbol{\Phi}}{\partial x_j}\frac{\partial \boldsymbol{\Phi}}{\partial x_i}^T \, d\Omega\right]\mathbf{u_j} - \left[\int_\Omega \frac{\partial \boldsymbol{\Phi}}{\partial x_i}\boldsymbol{\Psi}^T \, d\Omega\right]\mathbf{P} = \qquad (A1)$$

$$-\left[\int_\Omega \rho_0 g_i \beta \boldsymbol{\Phi}\boldsymbol{\Theta}^T \, d\Omega\right](\mathbf{T} - T_0) + \left[\int_\Gamma \boldsymbol{\Phi}\tau_{ij} n_j \, d\Gamma\right]$$

Continuity:

$$-\left[\int_\Omega \boldsymbol{\Psi}\frac{\partial \boldsymbol{\Phi}}{\partial x_i}^T \, d\Omega\right]\mathbf{u_i} = \mathbf{0} \qquad (A2)$$

Energy:

$$\left[\int_\Omega \rho_0 C \boldsymbol{\Theta}\boldsymbol{\Theta}^T \, d\Omega\right]\dot{\mathbf{T}} + \left[\int_\Omega \rho_0 C \boldsymbol{\Theta}\boldsymbol{\Phi}^T \mathbf{u_j}\frac{\partial \boldsymbol{\Theta}}{\partial x_j}^T \, d\Omega\right]\mathbf{T} + \left[\int_\Omega k \frac{\partial \boldsymbol{\Theta}}{\partial x_i}\frac{\partial \boldsymbol{\Theta}}{\partial x_i}^T \, d\Omega\right]\mathbf{T} =$$

$$\left[\int_\Omega \boldsymbol{\Theta} Q \, d\Omega\right] + \left[\int_\Omega \boldsymbol{\Theta} 2\mu D_{ij} D_{ij} \, d\Omega\right] + \left[\int_\Gamma \boldsymbol{\Theta} q_i n_i \, d\Gamma\right] \qquad (A3)$$

In arriving at the above equations, the Green-Gauss or divergence theorem has been used to reduce the second-order diffusion terms in (A1) and (A2) to first-order terms plus a boundary integral. The appearance of the boundary integral corresponds to the “natural” boundary conditions for the problem. Note that for nonzero values of the natural boundary conditions, the conditions in (18), (20) and (21) can be used to define the fluxes.

Several terms in equations (A1)–(A3) contain material properties that are often functions of temperature or some other dependent variable. It is convenient to allow these properties to have an explicit spatial variation within an element. Thus, let μ, k and β be approximated within an element by

$$\mu(x_i, t) = \boldsymbol{\Psi}^T(x_i)\hat{\mu}(t)$$

$$
\begin{aligned}
k(x_i,t) &= \mathbf{\Psi}^T(x_i)\hat{k}(t) && (A4)\\
\beta(x_i,t) &= \mathbf{\Psi}^T(x_i)\hat{\beta}(t)
\end{aligned}
$$

where $\mathbf{\Psi}$ is an interpolation function and the $\hat{}$ quantitites are vectors of nodal point material properties. In most applications a bilinear variation of the material property within the element is adequate and thus the use of the $\mathbf{\Psi}$ function. Since the dependent variables are known at the nodal points the above form provides a straightforward method for evaluating variable property values.

Once the form of the interpolation functions $\mathbf{\Phi}$, $\mathbf{\Psi}$ and $\mathbf{\Theta}$ 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 (A1)–(A3) may be evaluated to produce the required coefficient matrices. Normally the integral evaluation is carried out via a numerical quadrature procedure. The discretized system is given by the following matrix equations for the fluid mechanics (it is common to write the momentum and continuity equations as a single system):

$$
\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\begin{Bmatrix} \dot{\mathbf{u}}_\mathbf{1} \\ \dot{\mathbf{u}}_\mathbf{2} \\ \dot{\mathbf{P}} \end{Bmatrix}
+
\begin{bmatrix} \mathbf{C_1(u_1)+C_2(u_2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_1(u_1)+C_2(u_2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\begin{Bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{P} \end{Bmatrix}
$$

$$
+\begin{bmatrix} \mathbf{2K_{11}+K_{22}} & \mathbf{K_{21}} & \mathbf{-Q_1} \\ \mathbf{K_{12}} & \mathbf{K_{11}+2K_{22}} & \mathbf{-Q_2} \\ \mathbf{-Q_1^T} & \mathbf{-Q_2^T} & \mathbf{0} \end{bmatrix}
\begin{Bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{P} \end{Bmatrix}
=
\begin{Bmatrix} \mathbf{F_1(T)} \\ \mathbf{F_2(T)} \\ \mathbf{0} \end{Bmatrix}
\qquad (A5)
$$

and the energy transport

$$
[\mathbf{N}]\left\{\dot{\mathbf{T}}\right\} + [\mathbf{D(u_1)+D(u_2)}]\{\mathbf{T}\} + [\mathbf{L_{11}+L_{22}}]\{\mathbf{T}\} = \{\mathbf{G(T)}\} \qquad (A6)
$$

The coefficient matrices shown in (A5)–(A6) are defined by

$$
\begin{aligned}
\mathbf{M} &= \int_\Omega \rho_0 \mathbf{\Phi}\mathbf{\Phi}^T\, d\Omega \\
\mathbf{C_i(u_i)} &= \int_\Omega \rho_0 \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{u_i} \frac{\partial \mathbf{\Phi}}{\partial x_i}^T d\Omega \\
\mathbf{K_{ij}} &= \int_\Omega \mathbf{\Psi}^T \hat{\mu} \frac{\partial \mathbf{\Phi}}{\partial x_i}\frac{\partial \mathbf{\Phi}}{\partial x_j}^T d\Omega \\
\mathbf{Q_i} &= \int_\Omega \frac{\partial \mathbf{\Phi}}{\partial x_i} \mathbf{\Psi}^T\, d\Omega \\
\mathbf{F_i(T)} &= \int_\Omega \rho_0 g_i \mathbf{\Psi}^T \hat{\beta} \mathbf{\Phi}\mathbf{\Theta}^T\, d\Omega(\mathbf{T}-T_0) + \int_\Gamma \mathbf{\Phi}\tau_{ij}n_j\, d\Gamma && (A7)\\
\mathbf{N} &= \int_\Omega \rho_0 C \mathbf{\Theta}\mathbf{\Theta}^T\, d\Omega \\
\mathbf{D_i(u_i)} &= \int_\Omega \rho_0 C \mathbf{\Theta}\mathbf{\Phi}^T \mathbf{u_i} \frac{\partial \mathbf{\Theta}}{\partial x_i}^T d\Omega
\end{aligned}
$$

$$\mathbf{L_{ii}} = \int_\Omega \mathbf{\Psi}^T \hat{k} \frac{\partial \mathbf{\Theta}}{\partial x_i} \frac{\partial \mathbf{\Theta}}{\partial x_i}^T \, d\Omega$$

$$\mathbf{G} = \int_\Omega \mathbf{\Theta} Q \, d\Omega + \int_\Omega \mathbf{\Theta} 2\mu D_{ij} D_{ij} \, d\Omega + \int_\Gamma \mathbf{\Theta} q_i n_i \, d\Gamma$$

where (A4) has been used to define the variable material properties. Several of the terms in (A7) contain three distinct interpolation functions. When these terms are evaluated by a numerical quadrature procedure the resultant coefficients are stored in a three-dimensional array or hypermatrix. The remaining terms are stored as standard two-dimensional arrays. See Section 4.6 for a discussion of the evaluation and storage of the force vectors **F** and **G**.

Though equations (A5) and (A6) are the working forms of the element matrix system, it is convenient to condense the system for purposes of solution algorithm development and discussion. Thus, rewriting (A5) and (A6)

$$\mathbf{M\dot{U}} + \mathbf{C(U)U} - \mathbf{QP} + \mathbf{K(U,T)U} + \mathbf{B(T)T} = \mathbf{F(T)} \qquad (A8)$$

$$-\mathbf{Q^T U} = \mathbf{0} \qquad (A9)$$

$$\mathbf{N\dot{T}} + \mathbf{D(U)T} + \mathbf{L(T)T} = \mathbf{G(T,U)} \qquad (A10)$$

with

$$\mathbf{U}^T = \{\mathbf{u_1}^T, \mathbf{u_2}^T\}$$

and where the explicit dependence of each term on material properties has been indicated. In writing (A8) the buoyancy term has been separated from the general force vector **F**. Note that the solid body conduction equation is obtained from (A10) by neglecting the transport term **D**.

The above formulation is modified slightly when the penalty function approach to the incompressibility constraint is employed. In this case the continuity equation (A2) is modified to

$$\frac{\partial u_i}{\partial x_i} = -\epsilon P \qquad (A11)$$

Application of a Galerkin procedure to (A11) in conjunction with the finite element approximation of equation (40) produces the following

$$\left[\int_\Omega \mathbf{\Psi} \frac{\partial \mathbf{\Phi}^T}{\partial x_i} d\Omega\right] \mathbf{u_i} = -\left[\int_\Omega \epsilon \mathbf{\Psi}\mathbf{\Psi}^T d\Omega\right] \mathbf{P} \qquad (A12)$$

or in terms of a matrix equation

$$\mathbf{Q}^T\mathbf{U} = -\epsilon \mathbf{M_p P} \qquad (A13)$$

where the components of **Q** are defined in (A7) and

$$\mathbf{M_p} = \int_\Omega \mathbf{\Psi}\mathbf{\Psi}^T d\Omega.$$

If $\mathbf{M_p}$ can be inverted (see Section 4.5) then (A13) can be written as

$$\mathbf{P} = -\frac{1}{\epsilon}\mathbf{M_p}^{-1}\mathbf{Q}^T\mathbf{U} \qquad (A14)$$

When the above definition for the pressure is substituted into the original momentum equation (A8) the matrix system becomes

$$\mathbf{M}\dot{\mathbf{U}} + \mathbf{C}(\mathbf{U})\mathbf{U} + \mathbf{K_p}\mathbf{U} + \mathbf{K}(\mathbf{U},\mathbf{T})\mathbf{U} + \mathbf{B}(\mathbf{T})\mathbf{T} = \mathbf{F}(\mathbf{T}) \qquad (A15)$$

$$\mathbf{N}\dot{\mathbf{T}} + \mathbf{D}(\mathbf{U})\mathbf{T} + \mathbf{L}(\mathbf{T})\mathbf{T} = \mathbf{G}(\mathbf{T},\mathbf{U}) \qquad (A16)$$

with

$$\mathbf{K_p} = \frac{1}{\epsilon}\mathbf{Q}\mathbf{M_p}^{-1}\mathbf{Q}^T.$$

Equations (A15) and (A16) represent the penalty form of the nonisothermal flow problem.

For completeness in the development of the basic equations, the finite element form of the porous flow equations must be considered. Following the usual procedure the balance laws in (36)-(38) are written in a weighted integral form as

Momentum (Porous Flow):

$$\left[\int_\Omega \frac{\rho_0}{\phi}\mathbf{\Phi}\mathbf{\Phi}^T\, d\Omega\right]\dot{\mathbf{u}}_\mathbf{i} + \left[\int_\Omega \frac{\rho_0 \hat{c}}{\sqrt{\kappa}}\mathbf{\Phi}\mathbf{\Phi}^T\|\vec{\mathbf{u}}\|\mathbf{\Phi}^T\, d\Omega\right]\mathbf{u_i} + \left[\int_\Omega \frac{\mu}{\kappa}\mathbf{\Phi}\mathbf{\Phi}^T\, d\Omega\right]\mathbf{u_i}$$

$$+\left[\int_\Omega \mu_e \frac{\partial\mathbf{\Phi}}{\partial x_j}\frac{\partial\mathbf{\Phi}}{\partial x_j}^T\, d\Omega\right]\mathbf{u_i} + \left[\int_\Omega \mu_e \frac{\partial\mathbf{\Phi}}{\partial x_j}\frac{\partial\mathbf{\Phi}}{\partial x_i}^T\, d\Omega\right]\mathbf{u_j} - \left[\int_\Omega \frac{\partial\mathbf{\Phi}}{\partial x_i}\mathbf{\Psi}^T\, d\Omega\right]\mathbf{P} = \qquad (A17)$$

$$\left[\int_\Omega \rho_0 g_i \beta \mathbf{\Phi}\mathbf{\Theta}^T\, d\Omega\right](\mathbf{T} - T_0) + \left[\int_\Gamma \mathbf{\Phi}\tau_{ij}n_j\, d\Gamma\right]$$

Continuity:

$$-\left[\int_\Omega \mathbf{\Psi}\frac{\partial\mathbf{\Phi}}{\partial x_i}^T\, d\Omega\right]\mathbf{u_i} = \mathbf{0} \qquad (A18)$$

Energy:

$$\left[\int_\Omega (\rho_0 C)_e \mathbf{\Theta}\mathbf{\Theta}^T\, d\Omega\right]\dot{\mathbf{T}} + \left[\int_\Omega \rho_0 C\mathbf{\Theta}\mathbf{\Phi}^T\mathbf{u_j}\frac{\partial\mathbf{\Theta}}{\partial x_j}^T\, d\Omega\right]\mathbf{T} + \left[\int_\Omega k_e \frac{\partial\mathbf{\Theta}}{\partial x_i}\frac{\partial\mathbf{\Theta}}{\partial x_i}^T\, d\Omega\right]\mathbf{T} =$$

$$\left[\int_\Omega \mathbf{\Theta}Q\, d\Omega\right] + \left[\int_\Gamma \mathbf{\Theta}q_i n_i\, d\Gamma\right] \qquad (A19)$$

As with the Navier-Stokes system the second-order diffusion terms in (A17) and (A19) have been transformed via the divergence theorem. Also, the viscous dissipation term has been omitted from the energy equation.

Equations (A17)-(A19) can be evaluated for a particular choice of interpolation functions to produce the following matrix system

$$\begin{bmatrix} \tilde{\mathbf{M}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{M}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}_\mathbf{1} \\ \dot{\mathbf{u}}_\mathbf{2} \\ \dot{\mathbf{P}} \end{Bmatrix} + \begin{bmatrix} \tilde{\mathbf{C}}_\mathbf{1}(\mathbf{u_1}) + \tilde{\mathbf{C}}_\mathbf{2}(\mathbf{u_2}) + \tilde{\mathbf{A}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}_\mathbf{1}(\mathbf{u_1}) + \tilde{\mathbf{C}}_\mathbf{2}(\mathbf{u_2}) + \tilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{P} \end{Bmatrix}$$

$$+ \begin{bmatrix} \mathbf{2}\tilde{\mathbf{K}}_\mathbf{11} + \tilde{\mathbf{K}}_\mathbf{22} & \tilde{\mathbf{K}}_\mathbf{21} & -\tilde{\mathbf{Q}}_\mathbf{1} \\ \tilde{\mathbf{K}}_\mathbf{12} & \tilde{\mathbf{K}}_\mathbf{11} + \mathbf{2}\tilde{\mathbf{K}}_\mathbf{22} & -\tilde{\mathbf{Q}}_\mathbf{2} \\ -\tilde{\mathbf{Q}}_\mathbf{1}^\mathbf{T} & -\tilde{\mathbf{Q}}_\mathbf{2}^\mathbf{T} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{P} \end{Bmatrix} = \begin{Bmatrix} \tilde{\mathbf{F}}_\mathbf{1}(\mathbf{T}) \\ \tilde{\mathbf{F}}_\mathbf{2}(\mathbf{T}) \\ \mathbf{0} \end{Bmatrix} \tag{$A20$}$$

and the energy transport

$$\left[\tilde{\mathbf{N}}\right]\left\{\dot{\mathbf{T}}\right\} + \left[\tilde{\mathbf{D}}(\mathbf{u_1}) + \tilde{\mathbf{D}}(\mathbf{u_2})\right]\{\mathbf{T}\} + \left[\tilde{\mathbf{L}}_\mathbf{11} + \tilde{\mathbf{L}}_\mathbf{22}\right]\{\mathbf{T}\} = \left\{\tilde{\mathbf{G}}(\mathbf{T})\right\} \tag{$A21$}$$

The coefficient matrices in (A20) and (A21) are defined below. In defining these matrices the interpolation of variable material properties has been considered.

$$\begin{aligned}
\tilde{\mathbf{M}} &= \int_\Omega \frac{\rho_0}{\phi} \boldsymbol{\Phi}\boldsymbol{\Phi}^T \, d\Omega \\
\tilde{\mathbf{C}}_\mathbf{i}(\mathbf{u_i}) &= \int_\Omega \frac{\rho_0 \hat{c}}{\sqrt{\kappa}} \boldsymbol{\Phi}\boldsymbol{\Phi}^T \mathbf{u_i} \|\vec{\mathbf{u}}\| \boldsymbol{\Phi}^T \, d\Omega \\
\tilde{\mathbf{A}} &= \int_\Omega \frac{\mu}{\kappa} \boldsymbol{\Phi}\boldsymbol{\Phi}^T \, d\Omega \\
\tilde{\mathbf{K}}_\mathbf{ij} &= \int_\Omega \boldsymbol{\Psi}^T \hat{\mu}_e \frac{\partial \boldsymbol{\Phi}}{\partial x_i} \frac{\partial \boldsymbol{\Phi}}{\partial x_j}^T \, d\Omega \\
\tilde{\mathbf{Q}}_\mathbf{i} &= \int_\Omega \frac{\partial \boldsymbol{\Phi}}{\partial x_i} \boldsymbol{\Psi}^T \, d\Omega \\
\tilde{\mathbf{F}}_\mathbf{i}(\mathbf{T}) &= \int_\Omega \rho_0 g_i \boldsymbol{\Psi}^T \hat{\beta} \boldsymbol{\Phi}\boldsymbol{\Theta}^T \, d\Omega (\mathbf{T} - T_0) + \int_\Gamma \boldsymbol{\Phi} \tau_{ij} n_j \, d\Gamma \qquad (A22) \\
\tilde{\mathbf{N}} &= \int_\Omega (\rho_0 C)_e \boldsymbol{\Theta}\boldsymbol{\Theta}^T \, d\Omega \\
\tilde{\mathbf{D}}_\mathbf{i}(\mathbf{u_i}) &= \int_\Omega \rho_0 C \boldsymbol{\Theta}\boldsymbol{\Phi}^T \mathbf{u_i} \frac{\partial \boldsymbol{\Theta}}{\partial x_i}^T \, d\Omega \\
\tilde{\mathbf{L}}_\mathbf{ii} &= \int_\Omega \boldsymbol{\Psi}^T \hat{k}_e \frac{\partial \boldsymbol{\Theta}}{\partial x_i} \frac{\partial \boldsymbol{\Theta}}{\partial x_i}^T \, d\Omega \\
\tilde{\mathbf{G}} &= \int_\Omega \boldsymbol{\Theta} Q \, d\Omega + \int_\Gamma \boldsymbol{\Theta} q_i n_i \, d\Gamma
\end{aligned}$$

The equations (A20)-(A22) are the porous media equivalents of the Navier-Stokes equations in (A5)-(A7). The penalty approximation of (A11) could be also be used with this formulation.

Appendix B - Implementation of Newton's Method

Newton's method for solving a system of nonlinear, algebraic equations is used in the solution algorithms for both steady-state and time-dependent problems. Details of the implementation of Newton's method are given here for the case of nonisothermal, vsicous flow. The auxiliary transport and porous flow equations are not explicitly considered in the present section since their treatment is completely analogous to the equations for the basic nonisothermal problem.

The most general case of interest stems from the time-dependent problem and the use of the trapezoid integration rule. From equation (115) the basic matrix system is

$$\left[\frac{2}{\Delta t_n}\overline{\mathbf{M}} + \overline{\mathbf{K}}(\mathbf{V}_{n+1})\right]\mathbf{V}_{n+1} = \frac{2}{\Delta t_n}\overline{\mathbf{M}}\mathbf{V}_n + \overline{\mathbf{M}}\dot{\mathbf{V}}_n + \overline{\mathbf{F}}(\mathbf{V}_{n+1}) \qquad (B1)$$

which is a nonlinear system for the vector $\mathbf{V}_{n+1}$. The vectors $\mathbf{V}_n$ and $\dot{\mathbf{V}}_n$ are assumed known. To apply Newton's method to (B1) the system is first rewritten as

$$\mathbf{R}(\mathbf{V}_{n+1}) = \frac{2}{\Delta t_n}\overline{\mathbf{M}}\mathbf{V}_{n+1} + \overline{\mathbf{K}}(\mathbf{V}_{n+1})\mathbf{V}_{n+1} - \overline{\mathbf{F}}_n - \overline{\mathbf{F}}(\mathbf{V}_{n+1}) = \mathbf{0} \qquad (B2)$$

where the vector $\overline{\mathbf{F}}_n$ contains the known terms involving $\mathbf{V}_n$ and $\dot{\mathbf{V}}_n$. Newton's method for the solution of (B2) can then be written as

$$\mathbf{R}(\mathbf{V}^k_{n+1}) = -\left.\frac{\partial\mathbf{R}}{\partial\mathbf{V}}\right|_{\mathbf{V}^k_{n+1}}(\mathbf{V}^{k+1}_{n+1} - \mathbf{V}^k_{n+1}) = -\mathbf{J}(\mathbf{V}^k_{n+1})(\mathbf{V}^{k+1}_{n+1} - \mathbf{V}^k_{n+1}) \qquad (B3)$$

or when solved for $\mathbf{V}^{k+1}_{n+1}$

$$\mathbf{V}^{k+1}_{n+1} = \mathbf{V}^k_{n+1} - \mathbf{J}^{-1}(\mathbf{V}^k_{n+1})\mathbf{R}(\mathbf{V}^k_{n+1}) \qquad (B4)$$

The superscript k indicates the Newton iteration number while the subscript continues to indicate a particular timeplane in the integration process. Note that in the predictor/corrector procedures outlined in Section 5.2 the $k = 1$ iteration would employ the value of $\mathbf{V}$ predicted by the explicit predictor formula, *i.e.*, $\mathbf{V}^1_{n+1} = \mathbf{V}^p_{n+1}$.

The key to Newton's method is the formation of the Jacobian, $\mathbf{J}$. To define the components of $\mathbf{J}$ it is first necessary to define the specific components of $\mathbf{R}$ and $\mathbf{V}$. For the nonisothermal, viscous flow case the components of $\mathbf{R}$ and $\mathbf{V}$ can be obtained from equation (45), the definitions in equations (A5) and (A6) and the formula for the trapezoid rule in (B2). Thus, the components of $\mathbf{R}$ are

$$\mathbf{R_{u_1}} = \frac{2}{\Delta t_n}\mathbf{Mu_1} + \mathbf{C_1(u_1)u_1} + \mathbf{C_2(u_2)u_1} + 2\mathbf{K_{11}u_1} + \mathbf{K_{22}u_1} + \mathbf{K_{21}u_2}$$
$$-\mathbf{Q_1P} + \mathbf{B_1T} - \mathbf{F_{n1}} - \mathbf{F_1} \qquad (B5)$$

$$\mathbf{R_{u_2}} = \frac{2}{\Delta t_n}\mathbf{M u_2} + \mathbf{C_1(u_1)u_2} + \mathbf{C_2(u_2)u_2} + \mathbf{K_{12}u_1} + \mathbf{K_{11}u_2} + 2\mathbf{K_{22}u_2}$$

$$-\mathbf{Q_2 P} + \mathbf{B_2 T} - \mathbf{F_{n2}} - \mathbf{F_2} \qquad (B6)$$

$$\mathbf{R_P} = -\mathbf{Q_1}^T\mathbf{u_1} - \mathbf{Q_2}^T\mathbf{u_2} \qquad (B7)$$

$$\mathbf{R_T} = \frac{2}{\Delta t_n}\mathbf{NT} + \mathbf{D_1(u_1)T} + \mathbf{D_2(u_2)T} + \mathbf{L_{11}T} + \mathbf{L_{22}T} - \mathbf{G} \qquad (B8)$$

and the components of $\mathbf{V}$ are $\mathbf{u_1}, \mathbf{u_2}, \mathbf{P}$ and $\mathbf{T}$. In writing equations (B5) - (B8) the sub and superscript notation has been omitted to avoid confusion.

From the definition of the Jacobian in (B3) then

$$\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{V}} = \begin{bmatrix} \frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{u_1}} & \frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{u_2}} & \frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{P}} & \frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{T}} \\ \frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{u_1}} & \frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{u_2}} & \frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{P}} & \frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{T}} \\ \frac{\partial \mathbf{R_P}}{\partial \mathbf{u_1}} & \frac{\partial \mathbf{R_P}}{\partial \mathbf{u_2}} & \frac{\partial \mathbf{R_P}}{\partial \mathbf{P}} & \frac{\partial \mathbf{R_P}}{\partial \mathbf{T}} \\ \frac{\partial \mathbf{R_T}}{\partial \mathbf{u_1}} & \frac{\partial \mathbf{R_T}}{\partial \mathbf{u_2}} & \frac{\partial \mathbf{R_T}}{\partial \mathbf{P}} & \frac{\partial \mathbf{R_T}}{\partial \mathbf{T}} \end{bmatrix} \qquad (B9)$$

The components of $\mathbf{J}$ in (B9) are computed directly from (B5)-(B8) and produce the following

$$\frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{u_1}} = \frac{2}{\Delta t_n}\mathbf{M} + \mathbf{C_1(u_1)} + \mathbf{C_2(u_2)} + \mathbf{C_1(1)u_1} + 2\mathbf{K_{11}} + \mathbf{K_{22}}$$

$$\frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{u_2}} = \mathbf{C_2(1)u_1} + \mathbf{K_{21}}$$

$$\frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{P}} = -\mathbf{Q_1}$$

$$\frac{\partial \mathbf{R_{u_1}}}{\partial \mathbf{T}} = \mathbf{B_1}$$

$$\frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{u_1}} = \mathbf{C_1(1)u_2} + \mathbf{K_{12}}$$

$$\frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{u_2}} = \frac{2}{\Delta t_n}\mathbf{M} + \mathbf{C_1(u_1)} + \mathbf{C_2(u_2)} + \mathbf{C_2(1)u_2} + \mathbf{K_{11}} + 2\mathbf{K_{22}}$$

$$\frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{P}} = -\mathbf{Q_2}$$

$$\frac{\partial \mathbf{R_{u_2}}}{\partial \mathbf{T}} = \mathbf{B_2}$$

$$\frac{\partial \mathbf{R_P}}{\partial \mathbf{u_1}} = -\mathbf{Q_1}^T$$

$$\frac{\partial \mathbf{R_P}}{\partial \mathbf{u_2}} = -\mathbf{Q_2}^T$$

$$
\begin{aligned}
\frac{\partial \mathbf{R_P}}{\partial \mathbf{P}} &= \mathbf{0} \\
\frac{\partial \mathbf{R_P}}{\partial \mathbf{T}} &= \mathbf{0} \\
\frac{\partial \mathbf{R_T}}{\partial \mathbf{u_1}} &= \mathbf{D_1}(\mathbf{1})\mathbf{T} \\
\frac{\partial \mathbf{R_T}}{\partial \mathbf{u_2}} &= \mathbf{D_2}(\mathbf{1})\mathbf{T} \\
\frac{\partial \mathbf{R_T}}{\partial \mathbf{P}} &= \mathbf{0} \\
\frac{\partial \mathbf{R_T}}{\partial \mathbf{T}} &= \frac{2}{\Delta t_n}\mathbf{N} + \mathbf{D_1}(\mathbf{u_1}) + \mathbf{D_2}(\mathbf{u_2}) + \mathbf{L_{11}} + \mathbf{L_{22}}
\end{aligned}
$$

The components of $\mathbf{J}$ are combinations of the basic element matrices found in the original equations plus a few new terms that arise due to differentiation of terms that are nonlinear in the dependent variables, *e.g.*, the advection matrices. By studying the form of the $\mathbf{C_i}$ and $\mathbf{D_i}$ matrices given by equation (A7), it can be seen that the derivative of these terms can be computed analytically. In addition, the hypermatrix storage scheme used for these nonlinear terms greatly simplifies the formation of the derivatives needed in the Jacobian.

All of the nonlinearities found in the components of (B5) - (B8) are not explicitly accounted for in the derivatives defining the Jacobian. For example, when a non-Newtonian viscosity is used the $\mathbf{K}$ terms are functions of $\mathbf{u_1}, \mathbf{u_2}$ and should properly be differentiated with respect to these components. The complexity of treating all possible material property and boundary condition variations in this strict manner is prohibitive. In addition, these normally mild nonlinearities do not significantly affect the convergence of the Newton algorithm when treated in a "first-order" manner using a Picard or successive substitution procedure. Therefore, NACHOS II uses a strict Jacobian formulation only for the highly nonlinear advection terms and employs a Picard updating procedure for all other nonlinearities in the equations.

The previous derivation can be employed with the backward Euler integration formula by redefining the coefficients on $\mathbf{M}$ and $\mathbf{N}$ and altering the definition of $\overline{\mathbf{F}}_{\mathbf{n}}$. The above form of the Jacobian can be also be applied to the steady-state solution procedures by simply omitting those terms associated with the mass and capacitance matrices, $\mathbf{M}$ and $\mathbf{N}$. Forms similar to (B5) - (B8) can be defined for the porous flow equations and a Jacobian constructed in a completely analogous manner. The addition of the auxiliary transport equations simply increases the number of terms in the Jacobian; the definitions of the additional components of $\mathbf{J}$ are similar to those derived for the energy equation.

Appendix C - Implementation of a Quasi-Newton Method

The quasi-Newton procedure is not implemented in Version 2.00 of the NACHOS II program. A future release of the program will contain the quasi-Newton algorithm as described in Section 5.1.3.

Appendix D - Implementation of Continuation Methods

The continuation procedures are not implemented in Version 2.00 of the NACHOS II program. A future release of the program will contain a continuation algorithm as described in Section 5.1.3.