A Note on Finite Element Formulation for Mantle Convection

Shijie Zhong

Department of Physics

University of Colorado at Boulder

Boulder, Colorado 80309

U.S.A.

Tel: 303-735-5095
Fax: 303-492-7935
E-mail: szhong@anquetil.colorado.edu

In preparation for Treatise in Geophysics

1. Introduction

The governing equations for mantle convection are derived from conservation laws of mass, momentum and energy. The temperature- and stress-dependent mantle viscosity in the momentum equation and nonlinear coupling between flow velocity and temperature in the energy equation require that numerical methods be used to solve these governing equations.

In this chapter, we will present several commonly used numerical methods in studies of mantle convection. We will first present governing equations, boundary and initial conditions for a mantle convection problem, and discuss the general strategy to numerically solve the convection problem (section 2). We will focus on a finite element method (section 3), while also discussing basic principles of finite difference, finite volume, and spectral methods (section 4). Our discussions are mostly for thermal convection in homogeneous, incompressible fluids with the Boussinesq approximation. However, we will also describe methods for more complicated and realistic mantle conditions including non-Newtonian rheology, solid-state phase transitions, and thermochemical (i.e., multi-component) convection (section 5). Following discussions on the methods, we will present some simple examples of thermal convection modeling with emphases on comparison with the boundary layer theory and structural formation dynamics from thermal convection and (section 6). At the end, we will discuss the new developments in computational sciences that may impact our future studies of mantle convection modeling.

2. Governing Equations and Initial and Boundary Conditions

The simplest mathematical formulation for mantle convection assumes incompressibility and the Boussinesq approximation. Under this formulation, the nondimensional conservation equations of the mass, momentum, and energy are

$$u_{i,i} = 0\,, \tag{1}$$

$$\sigma_{ij,j} + RaT\delta_{iz} = 0\,, \tag{2}$$

$$\frac{\partial T}{\partial t} + u_i T_{,i} = (\kappa T_{,i})_{,i} + \gamma\,, \tag{3}$$

where u_i, σ_{ij}, T, and γ are the velocity, stress tensor, temperature, and heat production rate, respectively; Ra is a Rayleigh number, δ_{iz} is a Kronecker delta function. Repeated indices denote summation, and $\mathrm{u}_{,\mathrm{i}}$ represents partial derivative of variable u with respect to coordinate x_i. These equations were obtained by using the following characteristic scales: length D, time D^2/κ; and temperature ΔT, where D is often the thickness of the mantle or a fluid, κ is thermal diffusivity, and ΔT is the temperature difference across the fluid layer. The stress tensor can be related to strain rate via the following constitutive equation

$$\sigma_{ij} = -P\delta_{ij} + 2\eta\varepsilon_{ij} = -P\delta_{ij} + \eta(u_{i,j} + u_{j,i})\,, \tag{4}$$

where P is the dynamic pressure and η is the viscosity.

Substituting equations (4) into (2) reveals three primary unknown variables: pressure, velocity, and temperature. The three governing equations (1)-(3) are sufficient to solve for these unknowns with adequate boundary and initial conditions. Initial condition is only needed for temperature due to its first order derivative with respect to time in the energy equation. Boundary conditions are in general a combination of prescribed stress and velocity for the momentum equation and of prescribed heat flux and temperature for the energy equation. The initial and boundary conditions can be expressed as:

$$T(r_i, t=0) = T_{init}(r_i), \tag{5}$$

$$u_i = g_i \text{ on } \Gamma_{g_i}\,, \quad \sigma_{ij} n_j = h_i \text{ on } \Gamma_{h_i}\,, \tag{6}$$

$$T = T_{bd} \text{ on } \Gamma_{T_{bd}}\,, \quad (T,_i)_n = q \text{ on } \Gamma_q\,. \tag{7}$$

Often free slip (i.e., zero shear stresses and normal velocities) and isothermal conditions are applied to the surface and bottom boundaries in studies of mantle dynamics, although in some studies surface velocities may be given in consistent with surface plate motions. When steady-state or statistically steady state solutions are to be sought, as they often are in mantle dynamics, the choice of initial condition can be rather arbitrary with affecting the final solutions.

Although full time dependent dynamics of thermal convection involves all three governing equations, an important subset of mantle dynamics problems, often termed as instantaneous Stokes flow problem, only require solutions of equations (1) and (2). For the Stokes flow problem, one may consider the dynamic effects of a given buoyancy field (e.g., derived from seismic structure) or prescribed surface plate motion on gravity anomalies, deformation and stress at the surface and the interior of the mantle [Hager and O'Connell, 1981; Hager and Richards, 1984, Ricard et al., 1984].

These governing equations require numerical solution procedures for three reasons. i) The advection of temperature in equation (2), $u_i T_{,i}$, represents a nonlinear coupling between velocity and temperature. ii) The constitutive law or equation (4) often contains nonlinearity in that stress and strain rate follow a power law relation. That is, the viscosity η in equation (4) can only be considered as effective viscosity that depends on stress, strain rate or flow velocity. iii) Even for the Stokes flow problem with a linear rheology, spatial variability in viscosity can make any analytic solution method difficult and impractical.

Irrespective of numerical methods, the general strategy to solve the coupled governing equations consists of the following two steps. i) Solve equations 1 and 2 (i.e., the instantaneous Stokes flow problem) for flow velocity for a given buoyancy or temperature. ii) Update the temperature to next time step from equation 3, using the velocity.

3. A Finite Element Method

Finite element (FE) methods are effective in solving differential equations with complicated geometry and material properties. A FE method was first employed in studying the effects of non-Newtonian rheology on mantle convection [Parmentier et al., 1976] and have been since widely used in the studies of mantle dynamics [Christensen, 1984; Baumgardner, 1985; King et al., 1990; van Keken et al., 1993; Moresi and Gurnis, 1996; Bunge et al., 1997; Zhong et al., 2000]. This section will go through some of the basic steps in using finite element methods in solving governing equations for thermal convection.

The FE formulation for the Stokes flow that is described by equations 1 and 2 is independent from that for the energy equation. Hughes [1987] gave detailed description on a Galerkin weak form FE formulation for the Stokes flow. Brooks [1980] developed a Streamline Upwind Petrov-Galerkin formulation (SUPG) for the energy equation involving advection and diffusion. These two formulations remain popular for solving these types of problems [Hughes, 2000] and are employed in mantle convection codes ConMan [King, 1990] and Citcom/CitcomS [Moresi and Gurnis, 1996; Zhong et al., 2000]. The descriptions presented here are tailored from Brooks [1981], Hughes [1987], and Ramage and Wathen [1994] specifically for thermal convection in an incompressible media, and they are also closely related to codes ConMan and Citcom.

3.1. The Stokes flow: A Weak Formulation, its FE Implementation and Solution

3.1.1. A Weak Formulation

The Galerkin weak formulation for the Stokes flow can be stated as: find flow velocity $u_i = v_i + g_i$ and pressure P, where g_i is the prescribed boundary velocity in equation 6 and $v_i \in \mathcal{V}$, and $P \in \mathcal{P}$, where $\mathcal{V}$ is a set of functions in which each function,

w_i, is equal to zero on Γ_{g_i}, and $\mathcal{P}$ is a set of functions *q*, and such that for all $w_i \in \mathcal{V}$ and $q \in \mathcal{P}$,

$$\int_\Omega w_{i,j}\sigma_{ij}d\Omega - \int_\Omega qu_{i,i}d\Omega = \int_\Omega w_i f_i d\Omega + \sum_{i=1}^{n_{sd}} \int_{\Gamma_{h_i}} w_i h_i d\Gamma \,. \tag{8}$$

w_i and q are also called weighting functions. Equation (8) is equivalent to equations (1) and (2) and boundary conditions equation (6), provided that $f_i = RaT\delta_{iz}$ [Hughes, 2000]. Equation (8) can be written as

$$\begin{aligned} &\int_\Omega w_{i,j}c_{ijkl}v_{k,l}d\Omega - \int_\Omega qv_{i,i}d\Omega - \int_\Omega w_{i,i}Pd\Omega \\ &= \int_\Omega w_i f_i d\Omega + \sum_{i=1}^{n_{sd}} \int_{\Gamma_{h_i}} w_i h_i d\Gamma - \int_\Omega w_{i,j}c_{ijkl}g_{k,l}d\Omega \,, \end{aligned} \tag{9}$$

where

$$c_{ijkl} = \eta(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \,, \tag{10}$$

is derived from constitutive equation (4).

It is often convenient to rewrite

$$w_{i,j}c_{ijkl}v_{k,l} = \boldsymbol{\varepsilon}(\vec{w})^T D\boldsymbol{\varepsilon}(\vec{v}) \,, \tag{11}$$

where for two-dimensional plane strain problems

$$\boldsymbol{\varepsilon}(\vec{v}) = \begin{Bmatrix} v_{1,1} \\ v_{2,2} \\ v_{1,2} + v_{2,1} \end{Bmatrix}, \qquad D = \begin{bmatrix} 2\eta & 0 & 0 \\ 0 & 2\eta & 0 \\ 0 & 0 & \eta \end{bmatrix}, \tag{12}$$

and it is straightforward to write the expressions for other type of geometries including three-dimensional, axisymmetric [Hughes, 2000], or spherical geometry [Zhong et al., 2000].

Domain Ω can be discretized with a set of grid points (Figure 1) and the velocity and pressure and their weighting functions can be expressed in terms of their nodal values and the shape functions:

$$\vec{v} = v_i\vec{e}_i = \sum_{A\in\Omega^v-\Gamma^v_{g_i}} N_A v_{iA}\vec{e}_i, \quad \vec{w} = w_i\vec{e}_i = \sum_{A\in\Omega^v-\Gamma^v_{g_i}} N_A w_{iA}\vec{e}_i\ , \quad \vec{g} = \sum_{A\in\Gamma^v_{g_i}} N_A g_{iA}\vec{e}_i\,, \qquad (13)$$

$$P = \sum_{B\in\Omega^p} M_B P_B\,, \quad q = \sum_{B\in\Omega^p} M_B q_B\,, \qquad (14)$$

where N_A is the shape functions for velocity at node A, M_B is the shape functions for pressure at node B, Ω^{v} is the set of velocity nodes, Ω^{p} is the set of pressure nodes, and $\Gamma^v_{g_i}$ is the set of velocity nodes along boundary Γ_{g_i} . Note that the velocity shape functions and velocity nodes can be different from those for pressure (Figure 1).

Substituting (13) into (11) leads to the following equation:

$$\begin{aligned}\varepsilon(\vec{w})^T D\varepsilon(\vec{v}) &= \varepsilon(\sum_{A\in\Omega^v-\Gamma^v_{g_i}} N_A w_{iA}\vec{e}_i)^T D\varepsilon(\sum_{B\in\Omega^v-\Gamma^v_{g_j}} N_B v_{jB}\vec{e}_j) \\ &= [\sum_{A\in\Omega^v-\Gamma^v_{g_i}} \varepsilon(N_A\vec{e}_i)^T w_{iA}]D[\sum_{B\in\Omega^v-\Gamma^v_{g_j}} \varepsilon(N_B\vec{e}_j)v_{jB}] \\ &= \sum_{A\in\Omega^v-\Gamma^v_{g_i}} w_{iA}[\sum_{B\in\Omega^v-\Gamma^v_{g_j}} \vec{e}_i{}^T B_A^T D B_B \vec{e}_j v_{jB}], \end{aligned} \qquad (15)$$

where for two-dimensional plane strain problems

$$B_A = \begin{bmatrix} N_{A,1} & 0 \\ 0 & N_{A,2} \\ N_{A,2} & N_{A,1} \end{bmatrix}. \qquad (16)$$

Substituting (13) and (14) into (9) leads to the following equation:

$$\sum_{A\in\Omega^v-\Gamma^v_{g_i}} w_{iA}[\sum_{B\in\Omega^v-\Gamma^v_{g_j}} (\vec{e}_i{}^T \int_\Omega B_A{}^T D B_B d\Omega \vec{e}_j v_{jB}) - \sum_{B\in\Omega^p} (\vec{e}_i \int_\Omega N_{A,i} M_B d\Omega P_B)]$$

$$-\sum_{A\in\Omega^p}[q_A \sum_{B\in\Omega^v-\Gamma^v_{g_j}}(\int_\Omega M_A N_{B,j}d\Omega\vec{e}_j v_{jB})]=$$

$$\sum_{A\in\Omega^v-\Gamma^v_{g_i}} w_{iA}[\int_\Omega N_A\vec{e}_i f_i d\Omega+\sum_{i=1}^{n_{sd}}\int_{\Gamma_{h_i}} N_A\vec{e}_i h_i d\Gamma-\sum_{B\in\Gamma^v_{g_j}}(\vec{e}_i{}^T\int_\Omega B_A{}^T DB_B d\Omega\vec{e}_j g_{jB})]. \quad (17)$$

Because equation (17) holds for any weighting functions w_{iA} and q_A, it implies the following two equations.

$$\sum_{B\in\Omega^v-\Gamma^v_{g_j}}(\vec{e}_i{}^T\int_\Omega B_A{}^T DB_B d\Omega\vec{e}_j v_{jB})-\sum_{B\in\Omega^p}(\vec{e}_i\int_\Omega N_{A,i}M_B d\Omega P_B)$$

$$=\int_\Omega N_A\vec{e}_i f_i d\Omega+\sum_{i=1}^{n_{sd}}\int_{\Gamma_{h_i}} N_A\vec{e}_i h_i d\Gamma-\sum_{B\in\Gamma^v_{g_j}}(\vec{e}_i{}^T\int_\Omega B_A{}^T DB_B d\Omega\vec{e}_j g_{jB}), \quad (18)$$

$$\sum_{B\in\Omega^v-\Gamma^v_{g_j}}(\int_\Omega M_A N_{B,j}d\Omega\vec{e}_j v_{jB})=0. \quad (19)$$

Combining (18) and (19) into a matrix form leads to:

$$\begin{bmatrix} K & G \\ G^T & 0 \end{bmatrix}\begin{Bmatrix} V \\ P \end{Bmatrix}=\begin{Bmatrix} F \\ 0 \end{Bmatrix}, \quad (20)$$

where the vector V contains the velocity at all the nodal points, the vector P is the pressure at all the pressure nodes, the vector F is the total force term resulting from the three terms on the right hand side of (18) or (9), the matrices K, G, and G^T are the stiffness matrix, discrete gradient operator, and discrete divergence operator, respectively, which are derived from the first and second terms of (18) and (19), respectively. Specifically, the stiffness matrix is given by

$$K_{lm}=\vec{e}_i{}^T\int_\Omega B_A{}^T DB_B d\Omega\vec{e}_j, \quad (21)$$

where subscripts A and B are the global velocity node numbers as in (13), i and j are the degree of freedom numbers ranging from 1 to n_{sd}, and l and m are the global equation numbers for the velocity ranging from 1 to $n_v n_{sd}$ where n_v is the number of velocity nodes.

3.1.2. A FE Implementation

We now present a FE implementation of the Galerkin weak formulation for the Stokes flow and the resulting expressions of different terms in (20). We first introduce the elements and shape functions. A key feature of finite element method is that a local basis function or shape function is used such that the value of a variable within an element depends only on that at nodal points of the element. For simplicity, we consider a two-dimensional domain with quadrilateral elements. We employ the mixed elements in which there are four velocity nodes per element each of which occupies a corner of the element, while the only pressure node is at the center of the element (Figure 1). For these quadrilateral elements, the velocity interpolation in each element uses bilinear shape functions, while the pressure is constant for each element.

As a general remark on FE modeling of deformation/flow of incompressible media, it is important to keep interpolation functions (i.e., shape functions) for velocities at least one order higher than those for pressure, as we did for our quadrilateral elements [Hughes, 2000]. The spurious flow field may arise if this condition is satisfied. The well known example is the “mesh locking” that arises from linear triangle elements with constant pressure per element for which the incompressibility (i.e., a fixed elemental area) constraint per element demands zero deformation/flow everywhere in the domain [Hughes, 2000].

For any given element *e*, velocity and pressure within this element can be expressed through the following interpolation,

$$\vec{v} = v_i \vec{e}_i = \sum_{a=1}^{n_{en}} N_a v_{ia} \vec{e}_i \ , \quad \vec{w} = w_i \vec{e}_i = \sum_{b=1}^{n_{en}} N_b w_{ib} \vec{e}_i \ , \quad \vec{g} = g_i \vec{e}_i = \sum_{a=1}^{n_{en}} N_a g_{ia} \vec{e}_i \ , \qquad (22)$$

$$P = \sum_{a=1}^{n_{ep}} M_a P_a \ , \quad q = \sum_{a=1}^{n_{ep}} M_a q_a \ , \qquad (23)$$

where n_{en} and n_{ep} are the numbers of velocity and pressure nodes per element, respectively, and n_{en}=4 and n_{ep}=1 for our quadrilateral elements. The shape function N_a for a=1, …, n_{en}, depends on coordinates, and N_a is 1 at node a and linearly decreases to zero at other nodes of the element. The localness of the shape functions greatly simplifies implementation and computational aspects of the Galerkin weak formulation. For example, the integrals in (18) and (19) may be decomposed into sum of integrals from each element and the matrix equation (20) may be decomposed into sums of elemental contributions. Specifically, we may introduce elemental stiffness matrix, discrete gradient and divergence operators, and force term.

$$k^e = [k^e_{lm}],\ \ g^e=[g^e_{ln}],\ f^e = \{f^e_l\}, \tag{24}$$

where $1 \le l,m \le n_{en}n_{sd}$, $1 \le n \le n_{ep}$ (note that for quadrilateral elements, n_{en}=4, n_{ep}=1, and n_{sd}=2), k^e is a square matrix of $n_{en}n_{sd}$ by $n_{en}n_{sd}$, and g^e is a matrix of $n_{en}n_{sd}$ by n_{ep},

$$k^e_{lm} = \vec{e}_i^{\,T} \int_{\Omega^e} B_a{}^T D B_b d\Omega \vec{e}_j\ , \tag{25}$$

where l=n_{sd}(a-1)+i, m=n_{sd}(b-1)+j, a,b=1,…,n_{en}, and i,j=1,…,n_{sd},

$$g^e_{ln} = -\vec{e}_i \int_{\Omega^e} N_{a,i} M_n d\Omega\,, \tag{26}$$

where n=1,…,n_{ep}, and the rest symbols have the same definitions as before,

$$f^e_l = \int_{\Omega^e} N_a f_i d\Omega + \int_{\Gamma^e_{h_i}} N_a h_i d\Gamma - \sum_{m=1}^{n_{sd}n_{en}} k^e_{lm} g^e_m\ . \tag{27}$$

Determinations of these elemental matrices and force term require evaluations of integrals over each element with integrands that involve the shape functions and their derivatives. It is often convenient to use isoparameteric elements for which the coordinates and velocities in an element have the interpolation schemes [Hughes, 2000]. For example, for 2-D quadrilateral elements that we discussed earlier, the velocity shape functions for node a of an element in a parent domain with coordinates $\xi \in (-1,1)$ and $\eta \in (-1,1)$ is given as

$$N_a(\xi, \eta) = 1/4(1 + \xi_a \xi)(1 + \eta_a \eta), \tag{28}$$

where (ξ_a, η_a) is (-1,-1), (1,-1), (1,1) and (-1,1) for a=1, 2, 3 and 4, respectively. The pressure shape functions for M_a is 1, as there is only one pressure node per element. Although the integrations in (25), (26), and (27) are in the physical domain (i.e., x_1 and x_2 coordinates) rather than the parent domain, they can be expressed in the parent domain through coordinate transformation. These integrals can be evaluated using numerical integration schemes including the Gaussian quadrature. Numerical integration of these integrals is discussed in details in Hughes [2000] and will not be discussed here.

With elemental k^e, g^e, and f^e determined, it is straightforward to assemble them into global matrix equation (20). If an iterative solution method is used to solve (20), one may carry out calculations of the left hand side of (20) element by element without assembling elemental matrices and force terms into the global matrix equation form (20).

3.1.3. Solution Methods for the Matrix Equation

We now discuss solution methods for matrix equation (20). We will focus on iterative solution methods, because they require significantly less memory and computation than direct solution approaches. Iterative solution approaches are the only feasible and practical approaches for 3-D problems. Later we will briefly discuss a penalty formulation for the incompressible Stokes flow that requires a direct solution approach and is only effective for 2-D problems.

The matrix on the left hand side of (20), although symmetric, is not positive-definite. However, the stiffness matrix K is symmetric positive-definite. Efficient solution approaches should take advantage of these special properties. An efficient method is the Uzawa algorithm which is implemented in Citcom code [Moresi and Solomatov, 1995].

In the Uzawa algorithm, matrix equation (20) is broken into two coupled systems of equations [Atanga and Silvester, 1992; Ramage and Wathen, 1994]:

$$KV + GP = F\,, \tag{29}$$

$$G^T V = 0\,. \tag{30}$$

Combining these two equations and eliminating V lead to the discrete Poisson equation for pressure [Hughes, 1987]

$$(G^T K^{-1} G)P = G^T K^{-1} F\,. \tag{31}$$

Notice that matrix $\hat{K} = G^T K^{-1} G$ is symmetric positive definite. Although in practice equation (31) cannot be directly used to solve for P due to difficulties in obtaining K^{-1}, we may use it to build a pressure correction approach by using a conjugate gradient algorithm which does not require construction of matrix $\hat{K}$ [Ramage and Wathen, 1994]. The procedure is presented and discussed in details as follows.

With the conjugate gradient algorithm, for symmetric positive definite $\hat{K}$, the solution to a linear system of equations $\hat{K}P = H$ can be obtained with the operations in the left column of Figure 2 [Golub and van Loan, 1989, page 523].

For equations (29) and (31) for both velocities and pressure, with initial guess pressure P_0=0, the initial velocity V_0 can be obtained from

$$KV_0 = F\,, \quad \text{or} \quad V_0 = K^{-1}F\,, \tag{32}$$

and the initial residual for pressure equation (31), $r_{0,}$ and search direction, s_1, are $r_0 = s_1 = H = G^T K^{-1} F = G^T V_0$ (see the left column of Figure 2). To determine the search step α_k in the conjugate gradient algorithm, we need to compute the product of search direction s_k and $\hat{K}$, $s_k^T \hat{K} s_k$ (Figure 2). This product can be evaluated without explicitly constructing $\hat{K}$ for the following reasons.

The product can be written as

$$s_k^T \hat{K} s_k = s_k^T G^T K^{-1} G s_k = (G s_k)^T K^{-1} G s_k . \tag{33}$$

If we define u_k, such that

$$K u_k = G s_k , \quad \text{or} \quad u_k = K^{-1} G s_k , \tag{34}$$

then we have

$$s_k^T \hat{K} s_k = (G s_k)^T K^{-1} G s_k = (G s_k)^T u_k . \tag{35}$$

This indicates that if we solve (34) for u_k with Gs_k as the force term, the product $s_k^T \hat{K} s_k$ can be obtained without actually forming $\hat{K}$. Similarly, $\hat{K} s_k$ in updating the residual r_k (the left column of Figure 2) can be obtained without forming $\hat{K}$, because $\hat{K} s_k = G^T K^{-1} G s_k = G^T u_k$.

As the pressure P is updated via $P_k = P_{k-1} + \alpha_k s_k$ from the conjugate gradient algorithm (Figure 2), the velocity field can also be updated accordingly via

$$V_k = V_{k-1} - \alpha_k u_k . \tag{36}$$

This can be seen from the following derivation. At iteration step k-1, the pressure and velocity are P_{k-1} and V_{k-1}, respectively, and they satisfy equation (29),

$$K V_{k-1} + G P_{k-1} = F . \tag{37}$$

At iteration step k, the updated pressure is P_k, and the velocity $V_k = V_{k-1} + v$, where v is the unknown increment to be determined. Substituting P_k and V_k into (29) and considering $P_k = P_{k-1} + \alpha_k s_k$, $V_k = V_{k-1} + v$, and (37) lead to

$$K v + \alpha_k G s_k = 0 , \quad \text{or} \quad v = -\alpha_k K^{-1} G s_k . \tag{38}$$

From (34), it is clear that the velocity increment $v = -\alpha_k u_k$, and consequently equation (36) updates the velocity.

The final algorithm is given in the right column of Figure 2 [Ramage and Wathen, 1994]. The efficiency of this algorithm depends on how efficiently equation (34) is solved. The stiffness matrix K is symmetric positive definite, and this allows for numerous possible solution approaches. For example, multi-grid solvers have been used for solving (34) [Moresi and Solomatov, 1995] as in Citcom code. Newer versions of Citcom including CitcomS/CitcomCU employ full multigrid solvers with a consistent projection scheme for (34) and are even more efficient [Zhong et al., 2000]. Pre-conditioning for the pressure equation can be of great help in improving the convergence for the pressure field, as discussed in Moresi and Solomatov [1995].

3.2. The Stokes flow: A Penalty Formulation

For 2-D problems, an efficient method to solve the incompressible Stokes flow is a penalty formulation with a reduced and selective integration. This method has been widely used in 2-D thermal convection problems, for example, in ConMan [King et al., 1990]. We now briefly discuss this penalty formulation, and detailed descriptions can be found in Hughes [2000].

The key feature in the penalty formulation is to allow for slight compressibility or $u_{k,k} \approx 0$. Here it is helpful to make an analogy to isotropic elasticity. The constitutive equations for both compressible and incompressible isotropic elasticity are given by the following two equations,

$$\sigma_{ij} = -P\delta_{ij} + \eta(u_{i,j} + u_{j,i}), \tag{39}$$

$$u_{k,k} + P/\lambda = 0, \tag{40}$$

where λ is the Lame constant which is finite for compressible media but infinite for incompressibility media (i.e., to satisfy $u_{k,k} = 0$ for finite P). To allow for slight compressibility, λ is taken finite but significantly larger than η, such that the error associated with the slight compressibility is negligibly small. Using words of 64 bit long (i.e., double precision), $\lambda/\eta \sim 10^7$ is effective. For finite λ, the constitutive equation becomes

$$\sigma_{ij} = \lambda u_{k,k}\delta_{ij} + \eta(u_{i,j} + u_{j,i}), \tag{41}$$

which replaces equation (4).

An interesting consequence of this new constitutive equation is that the pressure is no longer needed in the momentum equation, and this simplifies the finite element analysis. The weak form of the resulting Stokes flow problem is

$$\int_\Omega w_{i,j}c_{ijkl}v_{k,l}d\Omega = \int_\Omega w_i f_i d\Omega + \sum_{i=1}^{n_{sd}} \int_{\Gamma_{h_i}} w_i h_i d\Gamma - \int_\Omega w_{i,j}c_{ijkl}g_{k,l}d\Omega, \tag{42}$$

where

$$c_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \eta(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \tag{43}$$

The FE implementation of equation (42) is similar to that in section 3.1. With the pressure excluded as a primary variable, the matrix equation is simply

$$[K]\{V\} = \{F\}. \tag{44}$$

While the elemental force vector is defined the same as that in (27), the elemental stiffness needs some modification in comparison with that in (25).

$$k_{lm}^e = \vec{e}_i^{\,T}\left(\int_{\Omega^e} B_a{}^T D B_b d\Omega + \int_{\Omega^e} B_a{}^T \overline{D} B_b d\Omega\right)\vec{e}_j, \tag{45}$$

where the first integral is the same as in (25) but the second integral is a new addition with

$$\overline{D} = \begin{bmatrix} \lambda & \lambda & 0 \\ \lambda & \lambda & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{46}$$

The matrix equation (44) only yields correct solution for velocities if a reduced and selective integration scheme is used to evaluate the elemental stiffness matrix [e.g., Hughes, 2000]. Specifically, the numerical quadrature scheme for the second integral of () needs to be one order lower than that used for the first integral. For example, if for a 2-D problem a 2x2 Gaussian quadrature scheme is used to evaluate the first integral, then a one-point Gaussian quadrature scheme is needed for the second integral. Hughes [2000] discussed the equivalence theorem for the mixed elements and the penalty formulation with the reduced and selective integration. Moresi et al. [1996] showed that these two formulations yield essentially identical results for the Stokes flow problems by comparing solutions from ConMan code employing a penalty formulation and Citcom which uses a mixed formulation.

Finally, we make two remarks about this penalty formulation.

First, although the pressure is not directly solved from the matrix equation, the pressure can be obtained through post-processing via $P = -\lambda u_{k,k}$ for each element. Such obtained pressure fields often display a checkerboard pattern. However, a pressure smoothing scheme [Hughes, 2000] seems to work well. The pressure field is important in many geophysical applications including computing dynamic topography and melt migration.

Second, with λ/η ~10^7, the stiffness matrix is not well conditioned and is not suited for any iterative solvers. A direct solver is required for this type of equations, as done in ConMan. This implies that this formulation may not be applicable to 3-D problems because of the memory and computation requirements associated with direct solvers. Reducing λ/η improves the condition for the stiffness matrix, however, this is not recommended as it results in large error associated with compressibility.

3.3. The SUPG Formulation for the Energy Equation

This section introduces a SUPG (streamline upwind Petrov-Galerkin) formulation and a predictor-multicorrector explicit algorithm for time dependent energy equation (i.e., equation 3). This method was developed by Hughes [1987] and Brooks [1981] some twenty years ago and remains an effective method in finite element solutions of the equations with advection and diffusion such as our energy equation. Finite element mantle convection codes Citcom and ConMan both employ this method for solving the energy equation.

A weak form formulation for the energy equation (3) and boundary conditions (7) is [Brooks, 1981],

$$\int_{\Omega} w(\dot{T} + u_i T_{,i}) d\Omega + \int_{\Omega} w_{,i} (\kappa T_{,i}) d\Omega + \sum_{e} \int_{\Omega_e} \overline{w}[\dot{T} + u_i T_{,i} - (\kappa T_{,i})_{,i} - \gamma] d\Omega$$

$$= \int_{\Omega} w\gamma d\Omega + \int_{\Gamma_q} wq d\Gamma - \int_{\Omega} w_{,i} \kappa g_{,i} d\Omega, \qquad (47)$$

where w is the regular weighting functions and is zero on Γ_q, $\dot{T}$ is the time derivative of temperature, and $\overline{w}$ is the streamline upwind contribution to the weighting functions.

The finite element implementation of (47) is similar to what was discussed for the Stokes flow in section 3.1.2. While the weighting function w is similar to what was defined in (22) except it is now a scalar, the streamline upwind part $\overline{w}$ is defined through artificial diffusivity $\tilde{\kappa}$ as

$$\overline{w} = \tilde{\kappa} \hat{u}_j w_{,j} / |u|, \qquad (48)$$

where $|u|$ is the magnitude of flow velocity, $\hat{u}_j = u_j / |u|$ represents the directions of flow velocity, and $\tilde{\kappa}$ is defined as

$$\tilde{\kappa} = (\sum_{i=1}^{n_{sd}} \tilde{\xi}_i u_i h_i) / 2, \qquad (49)$$

$$\tilde{\xi}_i = \begin{cases} -1-1/\alpha_i, & \alpha_i < -1 \\ 0, & -1 \le \alpha_i \le 1 \\ 1-1/\alpha_i, & \alpha_i > 1 \end{cases}, \qquad \text{for } \alpha_i = \frac{u_i h_i}{2\kappa}, \tag{50}$$

where u_i and h_i are flow velocity and element lengths in certain directions. It should be pointed out that (49) and (50) are empirical and other forms are possible. Such defined streamline upward weighting function $\overline{w}$ can be thought as adding artificial diffusion to the actual diffusion term to lead to total diffusivity,

$$\kappa + \tilde{\kappa}\hat{u}_i\hat{u}_j \,. \tag{51}$$

$\overline{w}$ is discontinuous across elemental boundaries, different from *w*. This is why the integral in the third term of (47) is for each element. $\tilde{w} = w + \overline{w}$ is also sometimes called the Petrov-Galerkin weighting functions.

A reasonable assumption is the weighted diffusion for an element in the third term of (47), $\overline{w}(\kappa T_{,i})_{,i}$, is negligibly small. Therefore, (47) can be written as

$$\int\limits_{\Omega} w_{,i}(\kappa T_{,i})d\Omega + \sum_{e} \int\limits_{\Omega_e} \tilde{w}(\dot{T} + u_i T_{,i} - \gamma)d\Omega = \int\limits_{\Gamma_q} wq d\Gamma - \int\limits_{\Omega} w_{,i}\kappa g_{,i} d\Omega \,. \tag{52}$$

We now present relevant matrices at an element level. The $\dot{T}$ term in (47) implies that a mass matrix is needed and it is given as

$$m_{ab}^{e} = \int\limits_{\Omega^e} N_a N_b d\Omega \,, \tag{53}$$

where *a*,*b*=1,…,n_{en}.

Elemental stiffness k^e is,

$$k_{ab}^{e} = \int\limits_{\Omega^e} B_a{}^T \kappa B_b d\Omega \,, \tag{54}$$

where for 2-D problems

$$B_a{}^T = \begin{pmatrix} N_{a,1} & N_{a,2} \end{pmatrix}. \tag{55}$$

Elemental force vector f^e is given as,

$$f_a^e = \int_{\Omega^e} \tilde{N}_a \gamma d\Omega + \int_{\Gamma_q^e} N_a q d\Gamma - \sum_{b=1}^{n_{en}} k_{ab}^e g_b^e \ , \qquad (56)$$

where $\tilde{N}_a$ is the Petrov-Galerkin shape function.

Elemental advection matrix c^e is given as,

$$c_{ab}^e = \int_{\Omega^e} \tilde{N}_a u_i N_{b,i} d\Omega \ , \qquad (57)$$

The combined matrix equation may be written as

$$M\dot{\Phi} + (K + C)\Phi = F \ , \qquad (58)$$

where Φ is the unknown temperature, and *M*, *K*, *C*, and *F* are the total mass, stiffness, advection matrices and force vector assembled from all the elements.

Equation (58) can be solved a predictor-corrector algorithm [Hughes, 2000] with some initial condition for temperature (e.g., equation 5). Suppose that temperature and its time derivative at time step n are given, Φ_n and $\dot{\Phi}_n$, the solutions at time step n+1 with time increment Δt can be obtained with the following algorithm:

1) Predictor: $\Phi_{n+1}^0 = \Phi_n + \Delta t(1-\alpha)\dot{\Phi}_n$, $\dot{\Phi}_{n+1}^0 = 0$, iteration step *i*=0, (59)

2) Solving: $M^* \Delta\dot{\Phi}_{n+1}^i = \prod_e (f_{n+1}^e - m^e \dot{\Phi}_{n+1}^i - (k^e + c^e)\Phi_{n+1}^i)$, (60)

3) Corrector: $\Phi_{n+1}^{i+1} = \Phi_{n+1}^i + \Delta t \alpha \Delta\dot{\Phi}_{n+1}^i$, $\dot{\Phi}_{n+1}^{i+1} = \dot{\Phi}_{n+1}^i + \Delta\dot{\Phi}_{n+1}^i$, (61)

4) If needed, set iteration step *i*= *i* +1 and go back step 2.

We make four remarks for this algorithm. First, this method is 2nd order accurate if α=0.5 [Hughes, 2000]. Second, typically two iterations are sufficient. Third, in (60), $\prod$

represents the operation of assembling elemental matrix into global matrix, and M^* is the lumped mass matrix which essentially makes this scheme an explicit scheme. Fourth, time increment Δt needs to satisfy Courant time stepping constraints to make the scheme stable.

4. Incorporating More Realistic Physics

In section 2, we presented the governing equations for thermal convection in a homogeneous incompressible fluid with a Newtonian (linear) rheology and the Boussinesq approximation. However, the Earth's mantle is likely much more complicated with heterogeneous composition and non-Newtonian rheology. In addition, non-Boussinesq effects such as solid-solid phase transitions may play an important role in affecting the dynamics of the mantle. In this section, we will discuss the methods that help incorporate these more realistic physics in studies of mantle convection. We will focus on modeling thermochemical convection, solid-state phase transitions, and non-Newtonian rheology.

4.1. Thermochemical Convection

Thermal convection for a compositionally heterogeneous mantle has gained a lot of interests in recent years [Lenardic and Kaula, 1993; Tackley, 1998; Davaille, 1999; Kellogg et al., 1999], with focus on the roles of mantle compositional anomalies and crustal structure in mantle dynamics. This is also called thermochemical convection. Different from purely thermal convection for which the fluid has the same composition, thermochemical convection involves fluids with different compositions. Here we will present governing equations and numerical methods for solving these equations.

4.1.1. Governing equations

Governing equations for thermochemical convection include a transport equation that describes the movement of compositions, in addition to the conservation laws of the mass, momentum and energy (i.e., equations 1-3). Suppose that *C* describes the compositional field, the transport equation is

$$\frac{\partial C}{\partial t} + u_i C_{,i} = 0 \,. \qquad (62)$$

This transport equation is similar to the energy equation (3) except that it does not contain diffusive and source terms. For a two-component system such as the crust-mantle system or depleted-primordial mantle system, *C* can be either 0 or 1, representing either component. If the fluids of different compositions have intrinsically different density, then the momentum equation (2) needs to be modified to take into account of the compositional effects on the buoyancy

$$\sigma_{ij,j} + Ra(T - \beta C)\delta_{iz} = 0 \,, \qquad (63)$$

where β is the buoyancy number [van Keken et al., 1997; Tackley and King, 2003] and is defined as

$$\beta = \Delta\rho / (\rho \Delta T \alpha) \,, \qquad (64)$$

where Δρ is the density difference between the two compositions, ρ and ΔT are the reference density and temperature, and α is the reference coefficient of thermal expansion.

A special class of thermochemical convection problems examine how the mantle compositional heterogeneity is stirred by mantle convection [e.g., Gurnis and Davies, 1986; Kellogg, 1992; van Keken and Zhong, 1999]. For these studies on the mixing of the mantle, we may assume that the fluids with different composition have identical density with β=0.

5.1.2. Solution approaches

Solving the conservation equations of the mass, momentum and energy for thermochemical convection is identical to what was introduced in section 3 for purely thermal convection. The additional compositional buoyancy term in the momentum equation (63) does not present any new difficulties numerically, provided that the composition *C* is given. The new challenge is to solve the transport equation (62) effectively.

A number of techniques have been developed or adapted in solving the transport equation in thermochemical convection studies. They include a field method with a filter [e.g., Lenardic and Kaula, 1993], a marker chain method [van Keken et al., 1997; Zhong and Hager, 2003], and a particle method [e.g., Schmeling, 1993; Tackley, 1998; Tackley and King, 2003]. As reviewed by van Keken et al. [1997], while these techniques work to some extent, they also have their limitations, particularly in treating entrainment and numerical diffusion of composition *C*. We will briefly discuss each of these methods with more emphasis on the particle method.

In the particle method, the transport equation for *C* (i.e., 62) is not solved directly. Composition *C* at a given time is represented by a set of particles. This representation requires a mapping from the distribution of particles to compositional field *C* which is often represented on a numerical mesh. With the mapping, to update *C*, all that is needed is to update the position of each particle to obtain a updated distribution of particles. This effectively solves the transport equation for *C*.

Two different particle methods have been used to map distribution of particles to *C*: absolute and ratio methods [Tackley and King, 2003]. In the absolute method, particles are only used to represent one type of composition (e.g., for dense component or with *C*=1). The population density of particles can be mapped to *C*. For example, *C* for an element/grid cell with volume Ω_e and particles N_e can be given as

$$C_e = AN_e / \Omega_e \,, \tag{65}$$

where the constant *A* is the reciprocal of initial density of particles for composition *C*=1 (i.e., total number of particles divided by the volume of composition *C*=1). Clearly, the absence of particles in an element/grid cell represents *C*=0. A physical unrealistic situation with *C*>1 may arise due to statistical fluctuations in particle distribution or particle settling. Therefore, for this method to work effectively, a large number of particles are required [Tackley and King, 2003].

In the ratio method, two different types of particles are used to represent the compositional field *C*, type 1 for *C*=0 and type 2 for *C*=1. *C* for an element/grid cell that includes type 1 particles N_1 and type 2 particles N_2 is given as

$$C_e = N_2 /(N_1 + N_2)\,. \tag{66}$$

In the ratio method, *C* can never be greater than 1. Tackley and King [2003] found that the ratio method is particularly effective in modeling thermochemical convection in which the two components occupy similar amount of volumes.

We now discuss briefly procedures to update the positions of particles. One commonly used method is a high order Runge-Kutta method [e.g., van Keken et al., 1997]. Here we present a predictor-corrector scheme for updating the particle positions [e.g., Zhong and Hager, 2003]. Suppose that at time $t = t_0$, flow velocity is $\vec{u}_0$ and compositional field is C_0 that is defined by a set of particles with coordinates, $\vec{x}_0^i$, for particle *i*. The algorithm for solving composition at the next time step $t = t_0 + dt = t_1$, C_1, can be summarized as follows:

(1) Using a forward Euler scheme, predict the new position for each particle *i* with $\vec{x}_{1p}^i = \vec{x}_0^i + \vec{u}_0 dt$ and mapping the particles to compositional field C_{1p} at t = t$_1$.

(2) Using the predicted C_{1p}, solve the Stokes equation for new velocity $\vec{u}_{1p}$.

(3) Using a modified Euler scheme with second order accuracy, compute the position for each particle *i* with $\vec{x}_1^i = \vec{x}_0^i + 0.5(\vec{u}_0 + \vec{u}_{1p})dt$ and compositional field C_1 at t = t$_1$.

The marker chain method is similar to the particle method in a number of ways. In the marker chain method, composition *C* is defined by an interfacial boundary that separates two different components. The interfacial boundary is a line for 2-D problems or a surface for 3-D. Using the flow velocity, one tracks the evolution of the interfacial boundary and hence composition *C*. Often the interfacial boundary is represented by particles or markers. Therefore, updating the interfacial boundary is essentially the same as updating the particles in the particle method. Composition *C* on a numerical grid which is desired for solving the momentum and energy equations (63) and (3) can be obtained by projection. As van Keken et al. [1997] indicated, the marker chain method is rather effective for compositional anomalies with relatively simple structure and geometry in 2-D.

The field method is probably the most straightforward. By setting diffusivity to be zero, we can employ the same solver for the energy equation (e.g., in section 3.3) to solve the transport equation for *C*. However, this often introduces numerical artifacts including numerical oscillations and numerical diffusion. Lenardic and Kaula [1993] introduced a filter scheme that removes the numerical oscillations while conserving the total mass of compositional field.

4.2. Non-Newtonian Rheology

Laboratory studies suggest that the deformation of olivine, the main component in the upper mantle, follows a power-law rheology [e.g., Karato and Wu, 1993]:

$$\varepsilon = A\tau^{n}, \tag{67}$$

where ε is the strain rate, τ is the deviatoric stress, the pre-exponent constant *A* represents other effects such as grain size and water content, and the exponent *n* is ~3. The nonlinearity in the rheology arises from $n \neq 1$.

The effects of non-Newtonian rheology on mantle convection were first investigated by Parmentier et al. [1978] and Christensen [1984]. More recent efforts have been focused on how non-Newtonian rheology including visco-plastic rheology may lead to dynamic generation of plate tectonics [King and Hager, 1991; Bercovici, 1994; Moresi and Solomatov, 1998, Zhong et al., 1998; Tackley, 1998].

Solutions of non-linear problems in general require an iterative approach. The power-law rheolgy may be written as an expression for effective viscosity

$$\sigma_{ij} = -P\delta_{ij} + 2\eta_{eff}\varepsilon_{ij} , \tag{68}$$

$$\eta_{eff} = \tau / \varepsilon = \frac{1}{A}\varepsilon^{\frac{1-n}{n}} , \tag{69}$$

where ε in (69) is the second invariant of strain rate tensor,

$$\varepsilon = (\frac{1}{2}\varepsilon_{ij}\varepsilon_{ij})^{1/2} . \tag{70}$$

It is clear that the effective viscosity depends on strain rate which in turn depends on flow velocity. Therefore, a general strategy for this problem is: 1) starting with some guessed effective viscosity, solve the Stokes flow problem for flow velocities; 2) update the effective viscosity with the newly determined strain rate, and solve the Stokes flow again; 3) keep this iterative process until flow velocities are convergent.

Implementation of this iterative scheme is straightforward. The convergence for this iterative process depends on the exponent *n*. For regular power-law rheology with *n*~3, convergence is usually not a problem. However, for large *n* (e.g., in case of visco-plastic rheology), the iteration may converge very slowly or may diverge. Often some forms of damping may help improve convergence significantly.

Reference

Atanga, J., D. Silvester, Iterative methods for stabilized mixed velocity-pressure finite elements, *Int. J. Numer. Methods in Fluids*, **14**, 71-81, 1992.

Christensen, U. R., and Yuen, D. A., Layered convection induced by phase changes, *J. Geophys. Res.*, **90**, 10,291– 10,300, 1985.

Hager, B. H., and M. A. Richards, Long-wavelength variations in Earth’s geoid: Physical models and dynamical implications, *Phil. Trans. R. Soc. Lond.*, A **328**, 309-327, 1989.

McNamara, A.K. and Zhong, S., Thermochemical structures within a spherical mantle: Superplumes or piles? *J. Geophys. Res*., **109**, B07402, doi:10.1029/2003JB002847, 2004.

Moresi, L., and M. Gurnis, Constraints on the lateral strength of slabs from three-dimensional dynamic flow models, *Earth Planet. Sci. Lett*., **138**, 15-28, 1996

Ramage, A., and A.J. Wathen, Iterative solution techniques for the Stokes and Navier-Stokes equations, *Int. J. Numer. Methods in Fluids,* **19**, 67-83, 1994.

Tackley, P.J., S. D. King, Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations, *Geochem. Geophys. Geosys.*, **4**, doi:10.1029/2001GC000214, 2003.

Tackley, P.J., Three-dimensional simulations of mantle convection with a thermochemical CMB boundary layer: D"?, in The Core-Mantle Boundary Region, edited by Gurnis.et. al., pp. 231-253, American Geophysical Union, 1998.

Van Keken, P.E., King, S.D., Schmeling, H., Christensen, U.R., Neumeister, D. and Doin, M.-P., A comparison of methods for the modeling of thermochemical convection, *J. Geophys. Res.*, **102**, 22477-22496, 1997.

Verfurth, R., A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem, *IMA J. Numer. Anal*., **4**, 441-455, 1984.

Zhong, S., Dynamics of thermal plumes in 3D isoviscous thermal convection, *Geophys. J. Int.*, **154**, 162, 289-300, 2005.

Zhong, S. and B. H. Hager, Entrainment of a dense layer by thermal plumes, *Geophys. J. Int.*, **154**, 666-676, 2003.

Zhong, S., Zuber, M. T., Moresi, L. N. and Gurnis, M., Role of temperature dependent viscosity and surface plates in spherical shell models of mantle convection, *J. Geophys. Res.*, **105**, 11063-11082, 2000.

```
$k$=0; $P_0$=0; $r_0$=$H$
while |$r_k$|>a given tolerance, ε
    $k$=$k$+1
    if $k$=1
        $s_1 = r_0$
    else
        $\beta_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2}$
        $s_k = r_{k-1} + \beta_k s_{k-1}$
    end
    $\alpha_k = r_{k-1}^T r_{k-1} / s_k^T \hat{K} s_k$
    $P_k = P_{k-1} + \alpha_k s_k$
    $r_k = r_{k-1} - \alpha_k \hat{K} s_k$
end
$P$=$P_k$
```

```
$k$=0; $P_0$=0
Solve $KV_0$=$F$
$r_0$=$H$=$G^T V_0$
while |$r_k$|>a given tolerance, ε
    $k$=$k$+1
    if $k$=1
        $s_1 = r_0$
    else
        $\beta_k = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2}$
        $s_k = r_{k-1} + \beta_k s_{k-1}$
    end
    Solve $Ku_k$=$Gs_k$
    $\alpha_k = r_{k-1}^T r_{k-1} /(G s_k)^T u_k$
    $P_k = P_{k-1} + \alpha_k s_k$
    $V_k = V_{k-1} - \alpha_k u_k$
    $r_k = r_{k-1} - \alpha_k G^T u_k$
end
$P$=$P_k$ , $V$=$V_k$
```

Figure 2. The left column is the original conjugate gradient algorithm, and the right column is the modified algorithm for solving (29) and (31).