CHAPTER ELEVEN

# Nonlinear Problems/Convection-Dominated Flows

For fluid dynamics associated with nonlinearity and discontinuity, there have been significant developments in the last two decades both in finite difference methods (FDM) and finite element methods (FEM). Concurrent with upwind schemes in space and Taylor series expansion of variables in time for FDM formulations with various orders of accuracy, numerous achievements have been made in FEM applications since the publication of an earlier text [Chung, 1978]. These new developments include generalized Galerkin methods (GGM), Taylor-Galerkin methods (TGM) [Donea, 1984], and the streamline upwind Petrov-Galerkin (SUPG) methods [Heinrich et al., 1977; Hughes and Brooks, 1982], alternatively referred to as the streamline diffusion method (SDM) [Johnson, 1987], and Galerkin/least squares (GLS) methods [Hughes and his co-workers, 1988–1998]. In the sections that follow, it will be shown that computational strategies such as SUPG or SDM and other similar methods can be grouped under the heading of generalized Petrov-Galerkin (GPG) methods. Recent developments include unstructured adaptive methods [Oden et al., 1986; Löhner, Morgan, and Zienkiewicz, 1985], characteristic Galerkin methods (CGM) [Zienkiewicz and his co-workers, 1994–1998], discontinuous Galerkin methods (DGM) [Oden and his co-workers, 1996–1998], and flowfield-dependent variation (FDV) methods [Chung and his coworkers, 1995–1999], among others. On the other hand, the concepts of FDM and FEM have been utilized in developing finite volume methods in conjunction with unstructured grids [Jameson, Baker, and Weatherill, 1986]. It appears that FDM and FEM continue to co-exist and develop into a mature technology, mutually benefitting from each other.

We begin in this chapter with the general discussion of boundary conditions for the nonlinear momentum equations, followed by Taylor-Galerkin methods (TGM) and generalized Petrov-Galerkin (GPG) methods as applied to Burgers' equations. Some special topics such as Newton-Raphson methods and artificial viscosity are also discussed in this chapter. Applications to the Navier-Stokes system of equations characterizing incompressible and compressible flows are presented in Chapters 12 and 13, respectively.

## 11.1 BOUNDARY AND INITIAL CONDITIONS

Detailed treatments of boundary conditions with reference to FDM were presented in Section 6.7. In FEM formulations, Neumann boundary conditions arise from the partial integration of the inner product governing equations. This is an important aspect unique and advantageous in FEM, not available in FDM.

In general, precise definitions and implementations of boundary and initial conditions play decisive roles in obtaining acceptable and accurate solutions in fluid mechanics and heat transfer. As seen in Chapters 1 and 2, Neumann boundary conditions are derived from the inner product of the partial differential equation with test functions and by means of partial integrations of this inner product down to the *m*th order from the 2*m*th order derivatives of the governing partial differential equations. Neumann boundary conditions arise "naturally" in this process with derivatives of order $2m-1, 2m-2, \ldots m$ (weak derivatives). Derivatives of order below $m$ ($m-1$, $m-2, \ldots 0$) are referred to as Dirichlet boundary conditions. These definitions as given in Chapters 1 and 2 for linear problems are applied to the nonlinear convective flows in this section.

Specification of boundary conditions depends on the types of partial differential equations (elliptic, parabolic, or hyperbolic) and types of flows (incompressible, compressible, vortical, irrotational, laminar, turbulent, chemically reacting, thermal radiation, surface tension, etc.). We shall limit our discussions of boundary and initial conditions to simpler and general topics of incompressible and compressible flows in this section. More complicated and specific subjects will be treated in their respective chapters and sections, Part Five, Applications.

### 11.1.1 INCOMPRESSIBLE FLOWS

For simplicity, let us first examine the steady-state incompressible flow governed by the conservation of mass and momentum. In order to obtain the correct forms for the boundary conditions, the governing equations must be written in conservation form. This is because the conservation form allows the partial integration to be carried out correctly. Thus, we write

*Continuity*

$$\mathrm{v}_{i,i} = 0 \tag{11.1.1a}$$

*Momentum*

$$\frac{\partial}{\partial x_i}(\rho \mathrm{v}_i \mathrm{v}_j - \sigma_{ij}) - \rho F_j = 0 \tag{11.1.1b}$$

where $\sigma_{ij}$ is the total stress tensor,

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} = -p\delta_{ij} + \mu(\mathrm{v}_{i,j} + \mathrm{v}_{j,i})$$

To determine the existence of Neumann (natural) boundary conditions, we construct an inner product of the residual of the governing partial differential equation with an appropriate variable which leads to a weak form. Since the primary variable is the velocity for the momentum equation, we write the energy due to the momentum as

$$J = (\mathrm{v}_j, R_j) = \int_\Omega \mathrm{v}_j \left[\frac{\partial}{\partial x_i}(\rho \mathrm{v}_i \mathrm{v}_j + p\delta_{ij} - \tau_{ij}) - \rho F_j\right] d\Omega \tag{11.1.2a}$$

Integrating (11.1.2a) by parts, we obtain

$$J = \int_\Gamma \mathrm{v}_j(\rho \mathrm{v}_i \mathrm{v}_j + p\delta_{ij} - \tau_{ij})\, n_i d\Gamma - \int_\Omega [\rho \mathrm{v}_{j,i}\mathrm{v}_i \mathrm{v}_j + \mathrm{v}_{j,j} p \\ - \mu(\mathrm{v}_{j,i}\mathrm{v}_{i,j} + \mathrm{v}_{j,i}\mathrm{v}_{j,i}) + \rho F_j \mathrm{v}_j] d\Omega \tag{11.1.2b}$$

Mathematically, the boundary integral in (11.1.2b) denotes the Neumann boundary conditions. Physically, it represents the energy required on the boundaries to be in balance with that available in the domain. Here, the boundary forces per unit normal surface area are identified as

$$\rho \mathrm{v}_i \mathrm{v}_j n_i + p\delta_{ij} n_i - \tau_{ij} n_i = S_j^{(1)} - S_j^{(2)} \quad \text{on } \Gamma_N \tag{11.1.3}$$

where $S_j^{(1)}$ and $S_j^{(2)}$ indicate the normal surface convective stress and normal surface viscous stress (traction), respectively.

$$S_j^{(1)} = \rho \mathrm{v}_i \mathrm{v}_j n_i + p\delta_{ij} n_i = \begin{cases} \rho u(u n_1 + \mathrm{v} n_2) + p n_1 \\ \rho \mathrm{v}(u n_1 + \mathrm{v} n_2) + p n_2 \end{cases} \quad (i, j = 1, 2) \tag{11.1.4}$$

$$S_j^{(2)} = \tau_{ij} n_i = \mu(\mathrm{v}_{i,j} + \mathrm{v}_{j,i}) n_i = \begin{cases} \mu\left(2\dfrac{\partial u}{\partial x} n_1 + \left(\dfrac{\partial u}{\partial y} + \dfrac{\partial \mathrm{v}}{\partial x}\right) n_2\right) \\ \mu\left(\left(\dfrac{\partial \mathrm{v}}{\partial x} + \dfrac{\partial u}{\partial y}\right) n_1 + 2\dfrac{\partial \mathrm{v}}{\partial y} n_2\right) \end{cases} \quad (i, j = 1, 2) \tag{11.1.5}$$

with $\mathrm{v}_1 = u$, $\mathrm{v}_2 = \mathrm{v}$. A glance at (11.1.4) indicates that $S_j^{(1)} = p n_j$ for a solid wall ($\mathrm{v}_i n_i = 0$), as defined in Figure 11.1.1 where the pressure is to be specified as a Neumann boundary condition. The normal surface traction $S_j^{(2)}$ is contributed by viscous stress normal to the surface. For example, consider the vertical surface where $\theta = 0°$ on the right side (outlet) and $\theta = 180°$ on the left side (inlet). Let us examine the outlet face where $n_1 = \cos(0°) = 1$ and $n_2 = \sin(0°) = 0$. For horizontal ($x$-axis) and vertical

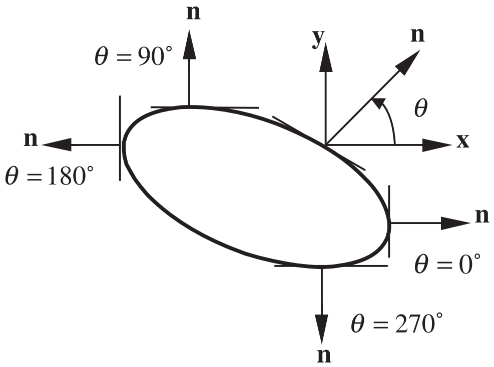


**Figure 11.1.1** Definitions of direct cosines in two-dimensional flows.

($y$-axis) directions, we have, respectively,

$$S_j^{(2)} = \tau_{ij} n_i = \begin{cases} 2\mu\left(\dfrac{\partial u}{\partial x}\right) \\ \mu\left(\dfrac{\partial \mathrm{v}}{\partial x} + \dfrac{\partial u}{\partial y}\right) \end{cases} \quad (i, j = 1, 2) \tag{11.1.6}$$

Alternatively, along the left side (inlet), $n_1 = \cos(180^\circ) = -1$ and $n_2 = \sin(180^\circ) = 0$,

$$S_j^{(2)} = \begin{cases} -2\mu\left(\dfrac{\partial u}{\partial x}\right) \\ -\mu\left(\dfrac{\partial \mathrm{v}}{\partial x} + \dfrac{\partial u}{\partial y}\right) \end{cases} \quad (i, j = 1, 2) \tag{11.1.7}$$

Similarly, for the top and bottom horizontal surfaces, respectively, with $\theta = 90^\circ$ and $\theta = 270^\circ$

$$S_j^{(2)} = \begin{cases} \mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial \mathrm{v}}{\partial x}\right) \\ 2\mu\left(\dfrac{\partial \mathrm{v}}{\partial y}\right) \end{cases} \quad (i, j = 1, 2) \tag{11.1.8}$$

and

$$S_j^{(2)} = \begin{cases} -\mu\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial \mathrm{v}}{\partial x}\right) \\ -2\mu\left(\dfrac{\partial \mathrm{v}}{\partial y}\right) \end{cases} \quad (i, j = 1, 2) \tag{11.1.9}$$

This completes the discussion of Neumann boundary conditions for the momentum equation. The continuity equation (11.1.1a) is a constraint condition for incompressibility or conservation of mass and is incapable of producing the Neumann boundary conditions. The Dirichlet (essential) boundary conditions arise from further integration by parts of the domain integral terms of (11.1.2b). Intuitively, we identify them as

$$\mathrm{v}_i = \overline{\mathrm{v}}_i \quad \text{on } \Gamma_D \tag{11.1.10}$$

Dirichlet boundary conditions may be implemented wherever available in addition to commonly assumed no-slip conditions along the solid walls. In principle, either Dirichlet or Neumann boundary conditions, not both, must be specified everywhere along the boundary surfaces for elliptic equations.

It is important to realize that the surface pressure is identified as a part of the Neumann boundary conditions in (11.1.4). For inclined surfaces, $n_1 \neq 0$, $n_2 \neq 0$, both components $S_1$ and $S_2$ contain the nonzero surface pressure and velocity gradients in both directions. Since no further integration by parts can be performed on the second term of the domain integral in (11.1.2b), the Dirichlet boundary condition does not arise. The reason for this is that we have $m = \frac{1}{2}$ for $p_{,i}$, 0th order ($2m - 1 = 0$) for the Neumann boundary condition and $-(\frac{1}{2})$th order ($m - 1 = -\frac{1}{2}$) for the Dirichlet boundary condition, implying that the pressure may be specified either as Neumann boundary conditions or as Dirichlet boundary conditions.

In view of these basic rules, any deviation arbitrarily chosen by practitioners may lead to incorrect solutions. Moreover, it is cautioned that any boundary nodes without specification of either Dirichlet or Neumann data are automatically construed as having enforced $S_i^{(1)} = S_i^{(2)} = 0$, because the finite element analog of the Neumann boundary vector in (11.1.2b) vanishes if either Dirichlet or Neumann data are not provided.

The numerical analysis involved in incompressible flows often requires the solution of Poisson equation for pressure in order to maintain the mass conservation and obtain accurate solutions of momentum equations. The pressure Poisson equation is obtained by constructing the divergence of the momentum equation. For incompressible flows, this operation leads to

$$p_{,ii} + (\rho \mathrm{v}_{i,j}\mathrm{v}_j)_{,i} = 0 \tag{11.1.11}$$

The inner product of (11.1.11) with $p$ becomes

$$J = \int_\Omega p[p_{,ii} + (\rho \mathrm{v}_{i,j}\mathrm{v}_j)_{,i}]d\Omega = 0$$

or

$$J = \int_\Gamma p\,(p_{,i}n_i + \rho \mathrm{v}_{i,j}\mathrm{v}_j n_i)\,d\Gamma - \int_\Omega (p_{,i}p_{,i} + \rho p_{,i}\mathrm{v}_{i,j}\mathrm{v}_j)\,d\Omega \tag{11.1.12}$$

It follows that Neumann boundary conditions are

$$S^{(1)} = p_{,i}n_i = \frac{\partial p}{\partial x}\cos\theta + \frac{\partial p}{\partial y}\sin\theta \tag{11.1.13a}$$

$$S^{(2)} = \rho(\mathrm{v}_i n_i)_{,j}\mathrm{v}_j = \rho\left(\frac{\partial u}{\partial x}\cos\theta + \frac{\partial \mathrm{v}}{\partial x}\sin\theta\right)u + \rho\left(\frac{\partial u}{\partial y}\cos\theta + \frac{\partial \mathrm{v}}{\partial y}\sin\theta\right)\mathrm{v} \tag{11.1.13b}$$

Here $S^{(1)}$ represents the normal surface pressure gradients. These data should be provided along the boundaries wherever the Dirichlet boundary conditions are not available. Notice that $S^{(2)}$ vanishes if $\mathrm{v}_i n_i = 0$ along the boundary nodes. In this case, of course, the pressure must be specified as Dirichlet boundary conditions alone, contrary to the case in the momentum equation, where pressure is treated as Neumann boundary conditions.

For transient problems, the momentum equation is written as

$$\rho\frac{\partial \mathrm{v}_j}{\partial t} + \frac{\partial}{\partial x_i}(\rho \mathrm{v}_i \mathrm{v}_j - \sigma_{ij}) - \rho F_j = 0 \tag{11.1.14}$$

In this case, the initial conditions consist of the initial data at $t = 0$ along the boundaries and the domain. For the velocity-pressure solutions of (11.1.1), the required initial conditions are

$$\mathrm{v}_i(x_i, 0) = \mathrm{v}_i^0 \quad \text{in } \overline{\Omega} = \Omega \cup \Gamma \tag{11.1.15a}$$

$$\mathrm{v}_i n_i(x_i, 0) = \mathrm{v}_i^0 n_i \quad \text{on } \Gamma \tag{11.1.15b}$$

In addition to these initial data, the Neumann boundary conditions of (11.1.4) and (11.1.5) at $t = 0$ should also be satisfied. Incompressibility conditions, $\mathrm{v}_{i,i}^0(x_i, 0) = 0$ in

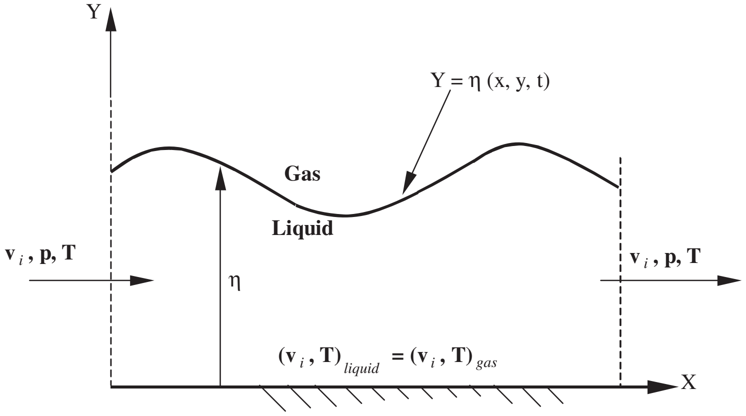


**Figure 11.1.2** Free surface flow boundary conditions with variable temperatures as well as velocity and pressure between liquid and gas.

$\Omega$ are difficult to enforce at $t = 0$. We require that the mass conservation be satisfied for $t > 0$ through an adequate numerical control such as the pressure Poisson equation to be discussed in Chapter 12.

The momentum equation (11.1.14), which is characterized as an elliptic-parabolic equation, tends toward parabolic if time dependency dominates and toward elliptic if spatial dependency dominates. Boundary conditions must be specified everywhere along the boundaries for elliptic equations, but not specified at the outlet surface for parabolic equations, as dicussed in Section 2.4.

Boundary conditions involved in two-phase flows or free surfaces of variable densities require additional information. As shown in Figure 11.1.2, the free surface is the boundary between liquid and gas,

$$Y = \eta(x, y, t)$$

$$p_{(liquid)} = p_{(gas)} - \sigma\left(R_x^{-1} + R_y^{-1}\right)$$

$$\mathrm{v}_{(liquid)} = \mathrm{v}_{(gas)} = \frac{D\eta}{Dt} = \frac{\partial\eta}{\partial t} + u\frac{\partial\eta}{\partial x} + \mathrm{v}\frac{\partial\eta}{\partial y}$$

$$\sigma_{ij(liquid)} = \sigma_{ij(gas)}$$

$$q_{y(liquid)} = q_{y(gas)}, \quad q_y = -k\frac{\partial T}{\partial y}$$

where $\sigma$ is the surface tension, $R_x$ and $R_y$ are the curvatures,

$$R_x^{-1} = \frac{\partial}{\partial x}\left\{\frac{\partial\eta}{\partial x}\left[1 + \left(\frac{\partial\eta}{\partial x}\right)^2 + \left(\frac{\partial\eta}{\partial y}\right)^2\right]^{-\frac{1}{2}}\right\}$$

$$R_y^{-1} = \frac{\partial}{\partial y}\left\{\frac{\partial\eta}{\partial y}\left[1 + \left(\frac{\partial\eta}{\partial x}\right)^2 + \left(\frac{\partial\eta}{\partial y}\right)^2\right]^{-\frac{1}{2}}\right\}$$

For simplified free-surface conditions between liquid and air, we may assume that

$$
\begin{aligned}
&p_{(liquid)} \cong p_{(gas)} - \sigma\left(\frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2}\right)\\
&\mathrm{v}_{(liquid)} \cong \frac{\partial\eta}{\partial t}\\
&\frac{\partial \mathrm{v}}{\partial y}_{(liquid)} \cong 0, \qquad \frac{\partial T}{\partial y}_{(liquid)} \cong 0\\
&p_{(liquid)} \cong p_{(atm)}
\end{aligned}
$$

In addition, we specify the velocity, pressure, and temperature at the inlet and outlet as well as the no-slip condition ($\mathrm{v} = 0$) at the wall. More detailed treatments of boundary conditions associated with surface tension will be given in Chapter 25, Multiphase Flows.

### 11.1.2 COMPRESSIBLE FLOWS

Compressible flows are characterized by additional terms for dilatation in the stress tensor and temporal and spatial variations of density.

$$\frac{\partial}{\partial t}(\rho \mathrm{v}_j) + \frac{\partial}{\partial x_i}(\rho \mathrm{v}_i \mathrm{v}_j + p\delta_{ij} - \tau_{ij}) - \rho F_j = 0 \tag{11.1.16a}$$

$$\frac{\partial\rho}{\partial t} + (\rho \mathrm{v}_i)_{,i} = 0 \tag{11.1.16b}$$

with

$$\tau_{ij} = \mu(\mathrm{v}_{i,j} + \mathrm{v}_{j,i}) - \frac{2}{3}\mu \mathrm{v}_{k,k}\delta_{ij}$$

For compressible flows, the normal surface convective stress, $S_j^{(1)}$, remains the same as in (11.1.4), but the normal surface traction, $S_j^{(2)}$, is modified as

$$S_j^{(2)} = \begin{cases} \mu\left[\frac{\partial u}{\partial x}n_1 + \frac{\partial u}{\partial y}n_2 + \frac{\partial u}{\partial x}n_1 + \frac{\partial \mathrm{v}}{\partial x}n_2 - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial \mathrm{v}}{\partial y}\right)n_1\right] \\ \mu\left[\frac{\partial \mathrm{v}}{\partial x}n_1 + \frac{\partial \mathrm{v}}{\partial y}n_2 + \frac{\partial u}{\partial y}n_1 + \frac{\partial \mathrm{v}}{\partial y}n_2 - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial \mathrm{v}}{\partial y}\right)n_2\right] \end{cases} \quad (j = 1, 2) \tag{11.1.17}$$

Thus, equations (11.1.6)–(11.1.9) are written as follows:
For $\theta = 0^\circ$

$$S_j^{(2)} = \begin{cases} \mu\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial \mathrm{v}}{\partial y}\right) \\ \mu\left(\frac{\partial \mathrm{v}}{\partial x} + \frac{\partial u}{\partial y}\right) \end{cases} \quad (j = 1, 2) \tag{11.1.18}$$

For $\theta = 180^\circ$

$$S_j^{(2)} = \begin{cases} -\mu\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial \mathrm{v}}{\partial y}\right) \\ -\mu\left(\frac{\partial \mathrm{v}}{\partial x} + \frac{\partial u}{\partial y}\right) \end{cases} \quad (j = 1, 2) \tag{11.1.19}$$

For $\theta = 90^\circ$

$$S_j^{(2)} = \begin{cases} \mu\left(\frac{\partial u}{\partial y} + \frac{\partial \mathrm{v}}{\partial x}\right) \\ \mu\left(\frac{4}{3}\frac{\partial \mathrm{v}}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}\right) \end{cases} \quad (j = 1, 2) \tag{11.1.20}$$

For $\theta = 270^\circ$

$$S_j^{(2)} = \begin{cases} -\mu\left(\frac{\partial u}{\partial y} + \frac{\partial \mathrm{v}}{\partial x}\right) \\ -\mu\left(\frac{4}{3}\frac{\partial \mathrm{v}}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}\right) \end{cases} \quad (j = 1, 2) \tag{11.1.21}$$

For compressible flows, combined solutions of the pressure Poisson equation are not required as the enforcement of the incompressibility condition is not necessary. Thus, the pressure will not be used as Dirichlet boundary conditions. It is still a part of the Neumann boundary conditions as specified in (11.1.4).

Dirichlet boundary conditions and initial conditions for compressible flows are the same as the incompressible flows. Enforcement of incompressibility conditions as initial conditions, however, is no longer necessary.

The elliptic-parabolic nature of (11.1.14) tends toward a hyperbolic type in high-speed flows if the viscosity effect is negligible, resulting in the Euler equation. In this case, the outflow boundary conditions are not to be specified but, rather, should be determined by the calculated upstream flows since the downstream effect toward upstream is not allowed. Details were discussed in Section 6.7 and will be covered also in Section 13.6.6 for compressible flows.

## CONCLUDING REMARKS

In identifying the Neumann boundary conditions, the conservation form of the momentum equations is used, in general, where convective terms as well as diffusion terms are integrated by parts. If the convective terms are not written in conservation form, however, no integration by parts is performed for the convective terms. In this case, the Neumann boundary conditions do not arise from the convective terms. This is the case for incompressible flows. In contrast, the conservation form is more convenient for compressible flows, and integration by parts for the convective term is carried out, resulting in the Neumann boundary conditions for compressible flows. This rule does not apply if a special test function (i.e., numerical diffusion test function) is used to induce artificial dissipation for the convective term as discussed in Section 11.3.

Specification of boundary conditions required for the Navier-Stokes system of equations is considerably more complicated, and will be discussed in Chapter 13.

## 11.2 GENERALIZED GALERKIN METHODS AND TAYLOR-GALERKIN METHODS

### 11.2.1 LINEARIZED BURGERS' EQUATIONS

To demonstrate the basic concept of generalized Galerkin methods (GGM), we consider the linearized Burgers' equations in the form,

$$R_i = \frac{\partial \mathrm{v}_i}{\partial t} + \overline{\mathrm{v}}_j \mathrm{v}_{i,j} - \nu \mathrm{v}_{i,jj} - f_i = 0 \quad (i = 1, 2, 3) \tag{11.2.1}$$

where $\overline{\mathrm{v}}_j$ is temporarily held constant in the time-marching steps and/or iteration cycles but updated in the following steps and/or iteration cycles. The standard finite element formulation of (11.2.1) with DST approximations was introduced as the GGM in Section 10.2. This requires the successive inner products of the form

$$(\hat{W}(\xi), E_i) = (\hat{W}(\xi), [W_\alpha(x), R_i]) = \int_\xi \hat{W}(\xi) \left[ \int_\Omega W_\alpha(x) R_i \, d\Omega \right] d\xi = 0 \tag{11.2.2}$$

in which $W_\alpha(x)$ and $\hat{W}(\xi)$ denote the spatial and temporal test functions, respectively. Furthermore, the trial functions for nodal values of variables are related as follows:

$$\mathrm{v}_i = \Phi_\alpha(x_i) \mathrm{v}_{\alpha i} \tag{11.2.3}$$

$$\mathrm{v}_{\alpha i} = \hat{\Phi}_m(\xi) \mathrm{v}_{\alpha i}^m \tag{11.2.4}$$

where $\Phi_\alpha(x)$ and $\hat{\Phi}_m(\xi)$ denote spatial and temporal trial functions, respectively, $\xi = t/\Delta t$, $\alpha =$ global spatial nodes, and $m =$ local temporal station $(n+1, n, n-1, \text{etc.})$.

Setting the spatial test function $W_\alpha$ equal to the spatial trial function $\Phi_\alpha$ and integrating (11.2.2) by parts in the spatial domain, we obtain

$$\int_\xi \hat{W}(\xi) [A_{\alpha\beta} \dot{\mathrm{v}}_{\beta i} + (B_{\alpha\beta} + K_{\alpha\beta}) \mathrm{v}_{\beta i} - F_{\alpha i} - G_{\alpha i}] d\xi = 0 \tag{11.2.5}$$

with

$$A_{\alpha\beta} = \int_\Omega \Phi_\alpha \Phi_\beta d\Omega, \qquad B_{\alpha\beta} = \int_\Omega \Phi_\alpha \Phi_{\beta,j} \overline{\mathrm{v}}_j \, d\Omega$$

$$K_{\alpha\beta} = \int_\Omega \nu \Phi_{\alpha,j} \Phi_{\beta,j} \, d\Omega \qquad G_{\alpha i} = \int_\Gamma \nu \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta d\Gamma g_{\beta i} \qquad F_{\alpha i} = \int_\Omega \Phi_\alpha \Phi_\beta d\Omega f_{\beta i}$$

Notice here that all matrices are the same as in Chapter 10 except for $B_{\alpha\beta}$, which is called the convection matrix. Choosing a linear variation of a variable in the temporal domain

$$\mathrm{v}_{\alpha i} = (1 - \xi) \mathrm{v}_{\alpha i}^n + \xi \mathrm{v}_{\alpha i}^{n+1}$$

we obtain from (11.2.5)

$$[A_{\alpha\beta} + \eta \Delta t (B_{\alpha\beta} + K_{\alpha\beta})] \mathrm{v}_{\beta i}^{n+1} = [A_{\alpha\beta} - \Delta t (1 - \eta) (B_{\alpha\beta} + K_{\alpha\beta})] \mathrm{v}_{\beta i}^n + \Delta t (F_{\alpha i} + G_{\alpha i}) \tag{11.2.6}$$

where the temporal parameter $\eta$ is defined as

$$\eta = \frac{\int_0^1 \hat{W}(\xi)\,\xi\,d\xi}{\int_0^1 \hat{W}(\xi)\,d\xi}$$

For $\hat{W}(\xi) = \delta(\xi - 1/2)$ or $\hat{W}(\xi) = 1$ with $0 \leq \xi \leq 1$, the temporal parameter becomes $\eta = 1/2$. Thus,

$$\left[A_{\alpha\beta} + \frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})\right]\mathrm{v}_{\beta i}^{n+1} = \left[A_{\alpha\beta} - \frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})\right]\mathrm{v}_{\beta i}^{n} + \Delta t(F_{\alpha i} + G_{\alpha i}) \tag{11.2.7}$$

We may rearrange (11.2.7) in the form

$$\left[A_{\alpha\beta} + \frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})\right]\frac{(\mathrm{v}_{\beta i}^{n+1} - \mathrm{v}_{\beta i}^{n})}{\Delta t} = -(B_{\alpha\beta} + K_{\alpha\beta})\,\mathrm{v}_{\beta i}^{n} + F_{\alpha i} + G_{\alpha i} \tag{11.2.8}$$

This is identical to the special case of the Taylor-Galerkin Methods (TGM) reported by Donea [1984]. If $\bar{\mathrm{v}}_j$ in (11.2.1) is no longer held constant, then the temporal trial functions $\hat{\Phi}_\alpha(\xi)$ or temporal test functions $\hat{W}(\xi)$, or both, may be chosen as higher order polynomials, which would introduce additional temporal stations as shown in Section 10.2. Note that the scheme as given by (11.2.8) is implicit and resembles the Crank-Nicholson scheme. In contrast to (11.2.7) in which $\eta = 1/2$ is fixed, we may choose $0 \leq \eta \leq 1$. Such choice is general and the expression given by (11.2.6) is known as the generalized Galerkin method (GGM) for the linearized convection-diffusion equation.

To prove that (11.2.8) is the same as the TGM of Donea [1984], we proceed as follows: Expanding $\mathrm{v}_i^{n+1}$ in Taylor series about $\mathrm{v}_i^n$, we write

$$\mathrm{v}_i^{n+1} = \mathrm{v}_i^n + \Delta t\frac{\partial \mathrm{v}_i^n}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 \mathrm{v}_i^n}{\partial t^2} + \frac{\Delta t^3}{6}\frac{\partial^3 \mathrm{v}_i^n}{\partial t^3} + \mathrm{O}(\Delta t^4) \tag{11.2.9}$$

Taking a time derivative of (11.2.1) for the time step $n$ and substituting the result into the above leads to

$$\frac{\mathrm{v}_i^{n+1} - \mathrm{v}_i^n}{\Delta t} = \left(-\bar{\mathrm{v}}_j\frac{\partial}{\partial x_j} + \nu\frac{\partial^2}{\partial x_j \partial x_j}\right)\mathrm{v}_i^n + \frac{\Delta t}{2}\left(-\bar{\mathrm{v}}_j\frac{\partial}{\partial x_j} + \nu\frac{\partial^2}{\partial x_j \partial x_j}\right)\frac{\partial \mathrm{v}_i^n}{\partial t}$$
$$+ \frac{\Delta t^2}{6}\left(\bar{\mathrm{v}}_j\bar{\mathrm{v}}_k\frac{\partial^2}{\partial x_j \partial x_k} - 2\nu\bar{\mathrm{v}}_j\frac{\partial^3}{\partial x_j \partial x_k \partial x_k} + \nu^2\frac{\partial^4}{\partial x_j \partial x_j \partial x_k \partial x_k}\right)\frac{\partial \mathrm{v}_i^n}{\partial \mathrm{t}} + f_i \tag{11.2.10a}$$

with

$$\frac{\partial \mathrm{v}_i^n}{\partial t} = \frac{\mathrm{v}_i^{n+1} - \mathrm{v}_i^n}{\Delta t}$$

Although the third order time derivative in (11.2.9) may be useful for the convection dominated flows without the viscous terms, we shall choose to neglect it for our purpose here to establish the analogy between GGM and TGM. Rearranging (11.2.10a) leads to

$$\left[1-\frac{\Delta t}{2}\left(-\bar{\mathrm{v}}_j\frac{\partial}{\partial x_j}+\nu\frac{\partial^2}{\partial x_j\partial x_j}\right)\right]\frac{\mathrm{v}_i^{n+1}-\mathrm{v}_i^n}{\Delta t}=\left(-\bar{\mathrm{v}}_j\frac{\partial}{\partial x_j}+\nu\frac{\partial^2}{\partial x_j\partial x_j}\right)\mathrm{v}_i^n+f_i \tag{11.2.10b}$$

The Galerkin finite element analog for (11.2.10b) yields

$$\int_\Omega \Phi_\alpha\left\{\left[1-\frac{\Delta t}{2}\left(-\bar{\mathrm{v}}_j\frac{\partial}{\partial x_j}+\nu\frac{\partial^2}{\partial x_j\partial x_j}\right)\right]\frac{\Phi_\beta\left(\mathrm{v}_{\beta i}^{n+1}-\mathrm{v}_{\beta i}^n\right)}{\Delta t}\right.$$
$$\left.+\left(\bar{\mathrm{v}}_j\frac{\partial}{\partial x_j}-\nu\frac{\partial^2}{\partial x_j\partial x_j}\right)\Phi_\beta \mathrm{v}_{\beta i}^n-\Phi_\beta f_{\beta i}\right\}d\Omega=0 \tag{11.2.10c}$$

Integrating the above equation by parts, we obtain the result identical to (11.2.8):

$$\left[A_{\alpha\beta}+\frac{\Delta t}{2}(B_{\alpha\beta}+K_{\alpha\beta})\right]\frac{\left(\mathrm{v}_{\beta i}^{n+1}-\mathrm{v}_{\beta i}^n\right)}{\Delta t}=-(B_{\alpha\beta}+K_{\alpha\beta})\,\mathrm{v}_{\beta i}^n+F_{\alpha i}+G_{\alpha i} \tag{11.2.11a}$$

which can then be rearranged in the form shown in (11.2.7),

$$\left[A_{\alpha\beta}+\frac{\Delta t}{2}(B_{\alpha\beta}+K_{\alpha\beta})\right]\mathrm{v}_{\beta i}^{n+1}=\left[A_{\alpha\beta}-\frac{\Delta t}{2}(B_{\alpha\beta}+K_{\alpha\beta})\right]\mathrm{v}_{\beta i}^n+\Delta t(F_{\alpha i}+G_{\alpha i}) \tag{11.2.11b}$$

It has been shown that the GGM approach with the temporal test function given by $\hat{W}(\xi)=\delta(\xi-1/2)$ or $\hat{W}(\xi)=1$ is identical to TGM proposed by Donea [1984] without the effect of the third order time derivative in the Taylor series expansion. This analogy of GGM to TGM does not hold true for the nonlinear Burgers' equations ($\mathrm{v}_j\neq\bar{\mathrm{v}}_j$) as will be demonstrated in Section 11.2.5 in which an explicit numerical diffusion arises in TGM, contributing to both stability and accuracy for the solution of nonlinear equations in general. The presence of the third order time derivative in the Taylor series expansion as originally proposed by Donea [1984] will be discussed in Section 11.2.3 in relation with the Euler method, leap-frog method, and Crank-Nicolson method.

### Numerical Diffusion

In general, for convection dominated flows, numerical diffusion is required to stabilize the solution process. To see whether the algorithm of GGM or TGM as given by (11.2.8) or (11.2.11a) does provide such a numerical diffusion, we may trace from (11.2.11b) back to (11.2.10a) with $\Delta t^2$ terms neglected.

$$\int_\Omega \Phi_\alpha\left(\frac{\partial \mathrm{v}_i}{\partial t}+\bar{\mathrm{v}}_j\mathrm{v}_{i,j}-\nu\mathrm{v}_{i,jj}-f_i\right)d\Omega=-\int_\Omega\frac{\Delta t}{2}\bar{\mathrm{v}}_j\Phi_{\alpha,j}(\bar{\mathrm{v}}_k\mathrm{v}_{i,k}-\nu\mathrm{v}_{i,kk}-f_i)d\Omega$$

in which the difference equation has been converted to the differential equation, with boundary integrals neglected upon integration by parts in the right-hand side. Note also that integration by parts was performed only for the convective terms. The viscous terms and body forces on the right-hand side may be neglected. The GGM formulation can then be applied to the left-hand side. It is clear that the first term on the right-hand side,

$$C_{\alpha\beta} = \int_\Omega \frac{\Delta t}{2}\overline{\mathrm{v}}_k\overline{\mathrm{v}}_j\Phi_{\alpha,k}\Phi_{\beta,j}d\Omega = \int_\Omega \overline{\nu}_{kj}\Phi_{\alpha,k}\Phi_{\beta,j}d\Omega \tag{11.2.12a}$$

represents the numerical diffusion matrix with $\overline{\nu}_{kj} = \frac{\Delta t}{2}\overline{\mathrm{v}}_k\overline{\mathrm{v}}_j$ being the artificial viscosity for convection. The numerical diffusion matrix $C_{\alpha\beta}$ should be added to the convection matrix $B_{\alpha\beta}$ in (11.2.8) particularly for high-speed convection-dominated flows.

$$B_{\alpha\beta} = \int_\Omega \Phi_\alpha\Phi_{\beta,j}\overline{\mathrm{v}}_j d\Omega + \int_\Omega \overline{\nu}_{kj}\Phi_{\alpha,k}\Phi_{\beta,j}d\Omega \tag{11.2.12b}$$

We shall further discuss this issue for the nonlinear Burgers' equations in Section 11.2.5. Note that a variety of approximations in GGM for the temporal test function $\hat{W}(\xi)$ and the temporal trial functions in (11.2.4) may lead to different forms of numerical diffusion. Similar consequences arise for TGM if the third order time derivative in the Taylor series expansion in (11.2.9) is retained.

***Remarks:*** In general, we may consider TGM to be a special case of GGM with $\eta = 1/2$ being chosen in (11.2.6). This is not true in some special cases of TGM as derived by Donea [1984].

### 11.2.2 TWO-STEP EXPLICIT SCHEME

Nonlinear problems can be solved explicitly by splitting the equation into two parts within a time step. Equation (11.2.7) or (11.2.8) may be rewritten in the form

***Step 1***

$$A_{\alpha\beta}X_{\beta i}^{(1)} = -(B_{\alpha\beta} + K_{\alpha\beta})\mathrm{v}_{\beta i}^n + F_{\alpha i} + G_{\alpha i} \tag{11.2.13a}$$

***Step 2***

$$A_{\alpha\beta}X_{\beta i}^{(2)} = -\frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})X_{\beta i}^{(1)} \tag{11.2.13b}$$

where

$$X_{\beta i}^{(1)} = \frac{\Delta\mathrm{v}_{\beta i}^{(1)}}{\Delta t}, \qquad X_{\beta i}^{(2)} = \frac{\Delta\mathrm{v}_{\beta i}^{(2)} - \Delta\mathrm{v}_{\beta i}^{(1)}}{\Delta t} \tag{11.2.14a,b}$$

Note that substitution of (11.2.14) into (11.2.13b) recovers (11.2.11) if the following assumption is made upon convergence:

$$\Delta\mathrm{v}_{\beta i}^{(2)} - \Delta\mathrm{v}_{\beta i}^{(1)} = \mathrm{v}_{\beta i}^{n+1} - \mathrm{v}_{\beta i}^n \tag{11.2.15}$$

A glance at (11.2.13a) and (11.2.13b) suggests that the solution of (11.2.13a) for $X_{\beta i}^{(1)}$ (Step 1) can be substituted into the right-hand side of (11.2.13b) to determine $X_{\beta i}^{(2)}$ (Step 2). At convergence, it is seen that

$$\frac{\Delta\mathrm{v}_{\beta i}^{(2)}}{\Delta t} \to \frac{\Delta\mathrm{v}_{\beta i}^{(1)}}{\Delta t} \to \frac{\Delta\mathrm{v}_{\beta i}^{n+1}}{\Delta t} = \frac{\mathrm{v}_{\beta i}^{n+1} - \mathrm{v}_{\beta i}^n}{\Delta t} \cong 0$$

and that (11.2.11b) arises by combining (11.2.13a) with (11.2.13b). This process is known as the two-step scheme, similar to the Lax-Wendroff scheme, contributing to an increase in accuracy and/or convergence.

It follows from (11.2.14) and (11.2.15) that the unknowns $\mathrm{v}_{\beta i}^{n+1}$ can be computed from

$$\mathrm{v}_{\beta i}^{n+1} = \mathrm{v}_{\beta i}^{n} + \Delta t\left(X_{\beta i}^{(1)} + X_{\beta i}^{(2)}\right) \tag{11.2.16}$$

which will then be substituted back into Step 1 (11.2.13a) for the next time step, thus continuously marching in time until steady-state is reached.

In (11.2.13a) and (11.2.13b) the inverse of the mass matrix $A_{\alpha\beta}$ would be simple if we chose to use the so-called lumped mass matrix as follows: Let $A_{\alpha\beta}^{(L)}$ be the lumped mass matrix, $A_{\alpha\beta}^{(C)}$ the consistent mass matrix as defined by $A_{\alpha\beta}$ in (11.2.13).

The lumped mass matrix is diagonal with entries from the tributary areas (sum of the row contributions). For example, the lumped mass matrix, $A_{NM}^{(L)}$, for a triangular element may be obtained from the consistent mass matrix, $A_{NM}^{(C)}$, as follows:

$$A_{NM}^{(C)} = \frac{A}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$A_{NM}^{(L)} = \sum_{p=1}^{3} A_{(N)p}^{(C)}\delta_{NM} = A_{(NN)}^{(L)} = \begin{bmatrix} A_{(11)}^{(L)} & 0 & 0 \\ 0 & A_{(22)}^{(L)} & 0 \\ 0 & 0 & A_{(33)}^{(L)} \end{bmatrix} \tag{11.2.17}$$

with

$$A_{(11)}^{(L)} = A_{(11)}^{(C)} + A_{(12)}^{(C)} + A_{(13)}^{(C)} = \frac{4A}{12}$$

$$A_{(22)}^{(L)} = A_{(21)}^{(C)} + A_{(22)}^{(C)} + A_{(23)}^{(C)} = \frac{4A}{12}$$

$$A_{(33)}^{(L)} = A_{(31)}^{(C)} + A_{(32)}^{(C)} + A_{(33)}^{(C)} = \frac{4A}{12}$$

Notice here that the index within the parentheses is not associated with summing. Thus we obtain

$$A_{(NM)}^{(L)} = \frac{A}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Write (11.2.13a) or (11.2.13b) in the form

$$A_{\alpha\beta}^{(C)} Y_{\beta i} = W_{\alpha i}$$

or

$$\left(A_{\alpha\beta}^{(C)} + A_{\alpha\beta}^{(L)} - A_{\alpha\beta}^{(L)}\right) Y_{\beta i} = W_{\alpha i}$$

which may be rewritten as

$$A^{(L)}_{\alpha\beta}Y_{\beta i} = W_{\alpha i} - A^{(C)}_{\alpha\beta}Y_{\beta i} + A^{(L)}_{\alpha\beta}Y_{\beta i}$$

Let the left-hand side and the right-hand side be the $r+1$ iterative cycle and the $r$ iterative cycle, respectively:

$$A^{(L)}_{\alpha\beta}\Delta Y^{r+1}_{\beta i} = W^{r}_{\alpha i} - A^{(C)}_{\alpha\beta}Y^{r}_{\beta i} \tag{11.2.18}$$

where

$$\Delta Y^{r+1}_{\beta i} = Y^{r+1}_{\beta i} - Y^{r}_{\beta i}$$

The iterations implied by (11.2.18) may be applied to Step 1 (11.2.13a) and then to Step 2 (11.2.13b) until each step acquires a satisfactory convergence. It has been shown that, in many instances, the lumped mass approach often leads to excellent results.

For two-dimensional problems, the $A^{(e)}_{NM}$ matrix must be expanded so that both $x$- and $y$-direction components of $\mathrm{v}_i$ can be accommodated. As noted earlier, this may be achieved by means of the Kronecker delta. This will expand (11.2.18) into a 6 × 6 matrix for triangular elements and an 8 × 8 matrix for quadrilateral elements when coupled with $A_{\alpha\beta}$.

To transform the generalized finite element equations given by (11.2.7) to the two-step solution scheme, we may establish the following procedure. Consider the matrix form of (11.2.7) written as

$$D\mathrm{v}^{n+1} = E\mathrm{v}^{n} + \Delta t H \tag{11.2.19}$$

where

$$D = A + B + C, \qquad E = A - B - C \tag{11.2.20}$$

(a) Rearrange (11.2.19) in the form

$$D\frac{\mathrm{v}^{n+1}-\mathrm{v}^{n}}{\Delta t} = F\frac{\mathrm{v}^{n}}{\Delta t} + H \tag{11.2.21}$$

with $F = E - D$

(b) Define

$$\Delta\mathrm{v}^{(2)} - \Delta\mathrm{v}^{(1)} = \mathrm{v}^{n+1} - \mathrm{v}^{n} \tag{11.2.22}$$

$$X^{(1)} = \frac{\Delta\mathrm{v}^{(1)}}{\Delta t} \tag{11.2.23a}$$

$$X^{(2)} = \frac{\Delta\mathrm{v}^{(2)} - \Delta\mathrm{v}^{(1)}}{\Delta t} \tag{11.2.23b}$$

(c) Write Step 1

$$AX^{(1)} = F\frac{\mathrm{v}^{n}}{\Delta t} + H \tag{11.2.24}$$

(d) Write Step 2

$$AX^{(2)} = (A - D)X^{(1)} \tag{11.2.25}$$

It can be shown that substitution of (11.2.24) into (11.2.25) together with (11.2.22) and (11.2.23) recovers (11.2.21) and subsequently (11.2.19).

If quadratic approximations are used for the temporal domain, then we write

$$D\mathrm{v}^{n+1} = E\mathrm{v}^n + G\mathrm{v}^{n-1} + \Delta t H \tag{11.2.26}$$

The two-step scheme becomes

***Step 1***

$$AX^{(1)} = F\frac{\mathrm{v}^n}{\Delta t} + \frac{G\mathrm{v}^{n-1}}{\Delta t} + H \tag{11.2.27}$$

***Step 2***

$$AX^{(2)} = (A - D)X^{(1)} \tag{11.2.28}$$

The data for $G\mathrm{v}^{n-1}$ are saved from the previous time station and used as additional source terms. A similar approach can be used for all higher approximations which will contain the terms of $\mathrm{v}^{n-2}$, $\mathrm{v}^{n-3}$, etc.

If $f_i$ is time dependent, and if $\overline{\mathrm{v}}_j$ in (11.2.1) is treated as a variable, and not held constant even during the discrete time step, then the second derivative in the Taylor series expansion would carry additional terms. In this case, $\overline{\mathrm{v}}_j$ on the left-hand side of (11.2.10b) becomes $\mathrm{v}_j^n$, and $\overline{\mathrm{v}}_j$ on the right-hand side of (11.2.10b) takes the form with a fractional step (i.e., $n + 1/2$),

$$\overline{\mathrm{v}}_j - \mathrm{v}_j^{n+\frac{1}{2}} = \mathrm{v}_j^n + \frac{\Delta t}{2}\frac{\partial \mathrm{v}_j}{\partial t} \tag{11.2.29}$$

and

$$f_j - f_j^{n+\frac{1}{2}} = f_j^n + \frac{\Delta t}{2}\frac{\partial f_j}{\partial t} \tag{11.2.30}$$

which would require the three-step solution scheme.

***Step 1***

$$A_{\alpha\beta}X_{\beta i}^{(0)} = -\frac{1}{2}(B_{\alpha\beta} + K_{\alpha\beta})\mathrm{v}_{\beta i}^n + F_{\alpha i} + G_{\alpha i} \tag{11.2.31}$$

with

$$X_{\beta i}^{(0)} = \frac{\mathrm{v}_{\beta i}^{n+\frac{1}{2}} - \mathrm{v}_{\beta i}^n}{\Delta t}$$

***Step 2***

$$A_{\alpha\beta}X_{\beta i}^{(1)} = -B_{\alpha\beta}\mathrm{v}_{\beta i}^{n+\frac{1}{2}} - K_{\alpha\beta}\mathrm{v}_{\beta i}^n + F_{\alpha i}^{n+\frac{1}{2}} + G_{\alpha i} \tag{11.2.32}$$

***Step3***

$$A_{\alpha\beta}X_{\beta i}^{(2)} = -\frac{1}{2}(B_{\alpha\beta} + K_{\alpha\beta})\Delta t X_{\beta i}^{(1)} \tag{11.2.33}$$

The GGM analog for the three-step scheme requires the use of quadratic functions in the temporal trial functions $\hat{\Phi}_m$, which will involve $\Delta t^2$ and three time steps, including a fractional time step.

### 11.2.3 RELATIONSHIP BETWEEN FEM AND FDM

It is interesting to note that the GGM formulations lead to finite difference results such as Euler Method, Leapfrog Method, Crank-Nicolson Method, etc. We will examine these results below.

**Euler Method**

Consider the convection equation

$$\frac{\partial \mathrm{v}_i}{\partial t} + \bar{\mathrm{v}}_j \mathrm{v}_{i,j} = 0 \tag{11.2.34}$$

Taking a time derivative of (11.2.34) gives

$$\frac{\partial^2 \mathrm{v}_i}{\partial t^2} + \bar{\mathrm{v}}_j \left(\frac{\partial \mathrm{v}_i}{\partial t}\right)_{,j} \Rightarrow \frac{\partial^2 \mathrm{v}_i}{\partial t^2} - \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \mathrm{v}_{i,kj} = 0 \tag{11.2.35}$$

A further differentiation of (11.2.35) yields

$$\frac{\partial^3 \mathrm{v}_i}{\partial t^3} - \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \left(\frac{\partial \mathrm{v}_i}{\partial t}\right)_{,kj} = 0 \tag{11.2.36}$$

Expanding $\mathrm{v}_i^{n+1}$ in Taylor series about $\mathrm{v}_i^n$ to the third order derivative, we obtain

$$\mathrm{v}_i^{n+1} = \mathrm{v}_i^n + \Delta t \frac{\partial \mathrm{v}_i^n}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 \mathrm{v}_i^n}{\partial t^2} + \frac{\Delta t^3}{3!} \frac{\partial^3 \mathrm{v}_i^n}{\partial t^3} \tag{11.2.37}$$

Rearranging (11.2.37) to determine the first derivative of $\mathrm{v}_i^n$ gives

$$\frac{\mathrm{v}_i^{n+1} - \mathrm{v}_i^n}{\Delta t} = \frac{\partial \mathrm{v}_i^n}{\partial t} + \frac{\Delta t}{2} \frac{\partial^2 \mathrm{v}_i^n}{\partial t^2} + \frac{\Delta t^2}{6} \frac{\partial^3 \mathrm{v}_i^n}{\partial t^3} \tag{11.2.38}$$

Substituting (11.2.34) through (11.2.36) into (11.2.38) leads to

$$\frac{\mathrm{v}_i^{n+1} - \mathrm{v}_i^n}{\Delta t} = -\bar{\mathrm{v}}_j \mathrm{v}_{i,j}^n + \frac{\Delta t}{2} \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \mathrm{v}_{i,kj}^n + \frac{\Delta t^2}{6} \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \left(\frac{\partial \mathrm{v}_i^n}{\partial t}\right)_{,kj} \tag{11.2.39}$$

with

$$\frac{\partial \mathrm{v}_i^n}{\partial t} = \frac{\mathrm{v}_i^{n+1} - \mathrm{v}_i^n}{\Delta t}$$

Equation (11.2.39) may be written as

$$\left(1 - \frac{\Delta t^2}{6} \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \frac{\partial^2}{\partial x_j \partial x_k}\right) \frac{\Delta \mathrm{v}_i^{n+1}}{\Delta t} = -\bar{\mathrm{v}}_j \mathrm{v}_{i,j}^n + \frac{\Delta t}{2} \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \mathrm{v}_{i,kj}^n \tag{11.2.40}$$

where $\Delta \mathrm{v}_i^{n+1} = \mathrm{v}_i^{n+1} - \mathrm{v}_i^n$.

We construct the Galerkin finite element integral for (11.2.40) in the form

$$\left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \frac{\Delta \mathrm{v}_{\beta i}^{n+1}}{\Delta t} = -\left(B_{\alpha\beta} + \frac{\Delta t}{2} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n} + \frac{\Delta t}{2} G_{\alpha i} \tag{11.2.41}$$

where

$$A_{\alpha\beta} = \int_\Omega \Phi_\alpha \Phi_\beta \, d\Omega, \qquad B_{\alpha\beta} = \int_\Omega \Phi_\alpha \Phi_{\beta,j} \bar{\mathrm{v}}_j d\Omega,$$

$$K_{\alpha\beta} = \int_\Omega \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \Phi_{\alpha,j} \Phi_{\beta,k} \, d\Omega,$$

$$G_{\alpha i} = \int_\Gamma \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta \, d\Gamma g_{\beta i}, \quad g_{\beta i} = (\bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \mathrm{v}_{i,k} n_j)_\beta$$

It should be noted that (11.2.41) is equivalent to the Generalized Galerkin finite element equations,

$$\left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n+1} = \left(A_{\alpha\beta} - \Delta t \, B_{\alpha\beta} - \frac{\Delta t^2}{3} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n} + \frac{\Delta t^2}{2} G_{\alpha i} \tag{11.2.42}$$

The two-step solution scheme for (11.2.41) becomes

$$A_{\alpha\beta} X_{\beta i}^{(1)} = -\left(B_{\alpha\beta} + \frac{\Delta t^2}{2} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n} + \frac{\Delta t}{2} G_{\alpha i} \tag{11.2.43}$$

$$A_{\alpha\beta} X_{\beta i}^{(2)} = -\frac{\Delta t^2}{6} K_{\alpha\beta} X_{\beta i}^{(1)} \tag{11.2.44}$$

with $X_{\beta i}^{(1)}$ and $X_{\beta i}^{(2)}$ defined as in (11.2.14). Notice that, in dealing with the advection equation with diffusion, we have included the third order time derivative [see (11.2.37)] which resulted in the numerical (artificial) diffusion characterized by the second order spatial derivative in (11.2.40) or the matrix $K_{\alpha\beta}$ in (11.2.41). The presence of these terms is responsible for the stability of numerical solution.

### Leapfrog Method

The leapfrog method is obtained by writing the Taylor series of $\mathrm{v}_i^{n-1}$ about $\mathrm{v}_i^n$ to the third order,

$$\mathrm{v}_i^{n-1} = \mathrm{v}_i^n - \Delta t \frac{\partial \mathrm{v}_i^n}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 \mathrm{v}_i^n}{\partial t^2} - \frac{\Delta t^3}{3!} \frac{\partial^3 \mathrm{v}_i^n}{\partial t^3} \tag{11.2.45}$$

Subtracting (11.2.45) from (11.2.37) and rearranging, we obtain

$$\left(1 - \frac{\Delta t^2}{6} \bar{\mathrm{v}}_j \bar{\mathrm{v}}_k \frac{\partial^2}{\partial x_j \partial x_k}\right) \frac{\Delta \mathrm{v}_i^{n+1}}{2\Delta t} = -\bar{\mathrm{v}}_j \mathrm{v}_{i,j}^n \tag{11.2.46}$$

with $\Delta \mathrm{v}_i^{n+1} = \mathrm{v}_i^{n+1} - \mathrm{v}_i^{n-1}$. The finite element analog of (11.2.46) becomes

$$\left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \frac{\Delta \mathrm{v}_{\beta i}^{n+1}}{2\Delta t} = -B_{\alpha\beta} \mathrm{v}_{\beta i}^n \tag{11.2.47}$$

The corresponding Generalized Galerkin finite element equations, neglecting the

Neumann boundary conditions, are given by

$$\left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n+1} = -2\Delta t\, B_{\alpha\beta} \mathrm{v}_{\beta i}^{n} + \left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n-1} \tag{11.2.48}$$

The two-step solution scheme consists of

$$\frac{1}{2} A_{\alpha\beta} X_{\beta i}^{(1)} = -\Delta t\, B_{\alpha\beta} \mathrm{v}_{\beta i}^{n} \tag{11.2.49}$$

$$A_{\alpha\beta} X_{\beta i}^{(2)} = -\frac{\Delta t^2}{6} K_{\alpha\beta} X_{\beta i}^{(1)} \tag{11.2.50}$$

By definition for the leapfrog method, the variables $\mathrm{v}_{\beta i}^{n+1}$ are calculated as

$$\mathrm{v}_{\beta i}^{n+1} = \mathrm{v}_{\beta i}^{n-1} + 2\Delta t\left(X_{\beta i}^{(1)} + X_{\beta i}^{(2)}\right) \tag{11.2.51}$$

Thus, initially both $\mathrm{v}_{\alpha i}^{n}$ and $\mathrm{v}_{\alpha i}^{n-1}$ are assumed to be known and, for the next time step, $\mathrm{v}_{\alpha i}^{n}$ becomes $\mathrm{v}_{\alpha i}^{n-1}$.

The leapfrog scheme may be revised to involve $\mathrm{v}_{\alpha i}^{n}$ instead of $\mathrm{v}_{\alpha i}^{n-1}$ (11.2.51) in the incremental form. This will alter the process as follows:

$$\left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \frac{\Delta \mathrm{v}_{\beta i}^{n+1}}{\Delta t} = \frac{1}{\Delta t}\left[\left(-2\Delta t\, B_{\alpha\beta} - A_{\alpha\beta} - \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n} + \left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n-1}\right] \tag{11.2.52}$$

The two-step solution scheme is now in the form

$$A_{\alpha\beta} X_{\beta i}^{(1)} = \frac{1}{\Delta t}\left[\left(-2\Delta t\, B_{\alpha\beta} - A_{\alpha\beta} - \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n} + \left(A_{\alpha\beta} + \frac{\Delta t^2}{6} K_{\alpha\beta}\right) \mathrm{v}_{\beta i}^{n-1}\right] \tag{11.2.53}$$

$$A_{\alpha\beta} X_{\beta i}^{(2)} = -\frac{\Delta t^2}{6} K_{\alpha\beta} X_{\beta i}^{(1)} \tag{11.2.54}$$

This will then allow the variables $\mathrm{v}_{\alpha i}^{n+1}$ to be calculated as

$$\mathrm{v}_{\alpha i}^{n+1} = \mathrm{v}_{\alpha i}^{n} + \Delta t\left(X_{\alpha i}^{(1)} + X_{\alpha i}^{(2)}\right) \tag{11.2.55}$$

### Crank-Nicolson Method

The Crank-Nicolson method is obtained by writing the Taylor series of $\mathrm{v}_i^n$ about $\mathrm{v}_i^{n+1}$ to the third order:

$$\mathrm{v}_i^n = \mathrm{v}_i^{n+1} - \Delta t \frac{\partial \mathrm{v}_i^{n+1}}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 \mathrm{v}_i^{n+1}}{\partial t^2} - \frac{\Delta t^3}{3!} \frac{\partial^3 \mathrm{v}_i^{n+1}}{\partial t^3} \tag{11.2.56}$$

Making use of the relation

$$\frac{1}{2}\left(\frac{\partial \mathrm{v}_i^{n+1}}{\partial t} + \frac{\partial \mathrm{v}_i^{n}}{\partial t}\right) = \frac{\mathrm{v}_i^{n+1} - \mathrm{v}_i^{n}}{\Delta t} \tag{11.2.57}$$

and in view of (11.2.35) and (11.2.36), and subtracting (11.2.56) from (11.2.37), we arrive at

$$\left(1-\frac{\Delta t^2}{6}\bar{\mathrm{v}}_j\bar{\mathrm{v}}_k\frac{\partial^2}{\partial x_j\partial x_k}\right)\frac{\Delta \mathrm{v}_i^{n+1}}{\Delta t}=-\frac{\bar{\mathrm{v}}_j}{2}\left(\frac{\partial \mathrm{v}_i^n}{\partial x_j}+\frac{\partial \mathrm{v}_i^{n+1}}{\partial x_j}\right)$$
$$+\frac{\Delta t}{4}\bar{\mathrm{v}}_j\bar{\mathrm{v}}_k\left(\frac{\partial^2 \mathrm{v}_i^n}{\partial x_j\partial x_k}-\frac{\partial^2 \mathrm{v}_i^{n+1}}{\partial x_j\partial x_k}\right) \qquad (11.2.58)$$

$$\left(A_{\alpha\beta}-\frac{\Delta t^2}{12}K_{\alpha\beta}+\frac{\Delta t}{2}B_{\alpha\beta}\right)\mathrm{v}_{\beta i}^{n+1}=\left(A_{\alpha\beta}+\frac{\Delta t^2}{12}K_{\alpha\beta}-\frac{\Delta t}{2}B_{\alpha\beta}\right)\mathrm{v}_{\beta i}^{n} \qquad (11.2.59)$$

This is the implicit Crank-Nicolson scheme. However, we may convert (11.2.59) into a two-step explicit scheme as follows:

(a) Rewrite the finite element equation in the time-step difference form

$$\left(A_{\alpha\beta}-\frac{\Delta t^2}{12}K_{\alpha\beta}+\frac{\Delta t}{2}B_{\alpha\beta}\right)\frac{\Delta \mathrm{v}_{\beta i}^{n+1}}{\Delta t}=-B_{\alpha\beta}\mathrm{v}_{\beta i}^{n} \qquad (11.2.60)$$

(b) The two-step explicit form is written using the procedure described earlier,

$$A_{\alpha\beta}X_{\beta i}^{(1)}=-B_{\alpha\beta}\mathrm{v}_{\beta i}^{n} \qquad (11.2.61)$$

$$A_{\alpha\beta}X_{\beta i}^{(2)}=\left(\frac{\Delta t^2}{12}K_{\alpha\beta}-\frac{\Delta t}{2}B_{\alpha\beta}\right)X_{\beta i}^{(1)} \qquad (11.2.62)$$

***Remarks:*** Appropriate choices of the finite element test functions for $W_\alpha$, $\Phi_\alpha$, $\hat{\Phi}_m$, and $W(\xi)$ enable the finite element analogs of Euler (11.2.42), leapfrog (11.2.48), and Crank-Nicolson (11.2.59) to be generated without the Taylor series expansion. Other forms of finite difference schemes may be generated by adding discontinuous functions to $W_\alpha$, which we shall elaborate in Section 11.3.

## 11.2.4 CONVERSION OF IMPLICIT SCHEME INTO EXPLICIT SCHEME

It follows from the approaches discussed in previous sections for the explicit schemes that it is possible to convert all implicit schemes into explicit schemes. Consider the generalized temporal-spatial finite element equations written in matrix form.

$$(A+B)\mathrm{v}^{n+1}=(A+C)\mathrm{v}^n+(A+D)\mathrm{v}^{n-1}+(A+E)\mathrm{v}^{n-2}+\cdots-\Delta t H \qquad (11.2.63)$$

where $B=B_1+B_2+\cdots$, $C=C_1+C_2+\cdots$, $D=D_1+D_2+\cdots$, $E=E_1+E_2+\cdots$, etc. Note that various forms of (11.2.63) result from unlimited choices of functions in $\Phi_\alpha$, $\hat{\Phi}_m$, and $\hat{W}(\xi)$ in Section 11.2.

The conversion process consists of the following steps:

(a) Write (11.2.63) in an incremental form,

$$(A+B)\frac{\Delta \mathrm{v}^{n+1}}{\Delta t}=[(A+C)-(A+B)]\frac{\mathrm{v}^n}{\Delta t}+(A+D)\frac{\mathrm{v}^{n-1}}{\Delta t}$$
$$+(A+E)\frac{\mathrm{v}^{n-2}}{\Delta t}+\cdots-H \qquad (11.2.64)$$

where

$$\Delta \mathrm{v}^{n+1} = \mathrm{v}^{n+1} - \mathrm{v}^{n} \tag{11.2.65}$$

(b) Step 1 is constructed by rewriting (11.2.64) with all terms other than the mass matrix $A$ removed from the left-hand side of (11.2.64) and designating $\Delta \mathrm{v}^{n+1}$ as $\Delta \mathrm{v}^{(1)}$, called the first increment,

$$AX^{(1)} = [(A+C) - (A+B)]\frac{\mathrm{v}^{n}}{\Delta t} + (A+D)\frac{\mathrm{v}^{n-1}}{\Delta t} + \cdots - H \tag{11.2.66}$$

where

$$X^{(1)} = \frac{\Delta \mathrm{v}^{(1)}}{\Delta t}$$

(c) Step 2 is constructed by setting the product of the mass matrix and the second increment $X^{(2)}$, which is equated to the variant of the first increment,

$$AX^{(2)} = [A - (A+B)]X^{(1)} \tag{11.2.67}$$

where

$$X^{(2)} = \frac{\Delta \mathrm{v}^{n+1} - \Delta \mathrm{v}^{(1)}}{\Delta t} \tag{11.2.68}$$

(d) The variable $\mathrm{v}^{n+1}$ is given by

$$\mathrm{v}^{n+1} = \mathrm{v}^{n} + \Delta t\left(X^{(1)} + X^{(2)}\right) \tag{11.2.69}$$

A glance at (11.2.69) reveals that, for a steady-state condition, $t \to \infty$, and $\mathrm{v} = \mathrm{v}^{n+1} = \mathrm{v}^{n} = \mathrm{v}^{n-1} = \mathrm{v}^{n-2} = \cdots$, we obtain

$$(B + C + D + E + \cdots)\mathrm{v} = H \tag{11.2.70}$$

Thus, it is expected that a steady-state solution would result as recursive calculations are carried out consecutively.

### 11.2.5 TAYLOR-GALERKIN METHODS FOR NONLINEAR BURGERS' EQUATIONS

Let us consider the nonlinear Burgers' equations of the form

$$\frac{\partial \mathrm{v}_i}{\partial t} + \mathrm{v}_j \mathrm{v}_{i,j} - \nu \mathrm{v}_{i,jj} = f_i \tag{11.2.71}$$

The Taylor series expansion of (11.2.71) as given in (11.2.9) without the third order derivative term becomes

$$\begin{aligned}\Delta \mathrm{v}_i^{n+1} = -\Delta t(\mathrm{v}_j \mathrm{v}_{i,j} - \nu \mathrm{v}_{i,jj} - f_i)^n + \frac{\Delta t^2}{2}\Bigg[\mathrm{v}_k \frac{\partial}{\partial x_k}(\mathrm{v}_j \mathrm{v}_{i,j} - \nu \mathrm{v}_{i,jj} - f_i) \\ + \mathrm{v}_{i,j}(\mathrm{v}_k \mathrm{v}_{j,k} - \nu \mathrm{v}_{j,kk} - f_j) - \nu \frac{\partial^2}{\partial x_j \partial x_j}(\mathrm{v}_k \mathrm{v}_{i,k} - \nu \mathrm{v}_{i,kk} - f_i) + \frac{\partial f_i}{\partial t}\Bigg]^n\end{aligned} \tag{11.2.72}$$

from which the original differential equation can be recovered in the form,

$$\frac{\partial \mathrm{v}_i}{\partial t} + \mathrm{v}_j \mathrm{v}_{i,j} - \nu \mathrm{v}_{i,jj} - f_i = S_i \tag{11.2.73}$$

where

$$S_i = \frac{\Delta t}{2}\left[\mathrm{v}_k \frac{\partial}{\partial \mathrm{x}_k}(\mathrm{v}_j \mathrm{v}_{i,j} - \nu \mathrm{v}_{i,jj} - f_i)\right] \tag{11.2.74}$$

with higher order derivative terms and products of the gradients in (11.2.72) being neglected. It is clear that the right-hand side of (11.2.74) appears as numerical diffusion.

Applying the Galerkin integral to the right-hand side of (11.2.74) and integrating by parts, we obtain

$$\int_\Omega \Phi_\alpha S_i \, d\Omega = -\frac{\Delta t}{2}\int_\Omega \mathrm{v}_\mathrm{k} \mathrm{v}_j \Phi_{\alpha,k} \Phi_{\beta,j} d\Omega \mathrm{v}_{\beta i} \tag{11.2.75}$$

where all terms other than the convective terms are negligible in practical applications. Thus, the numerical diffusion matrix is identified as

$$C_{\alpha\beta} = \int_\Omega \overline{\nu}_{kj} \Phi_{\alpha,k} \Phi_{\beta,j} d\Omega \tag{11.2.76}$$

with the numerical viscosity,

$$\overline{\nu}_{kj} = \frac{\Delta t}{2} \mathrm{v}_k \mathrm{v}_j \tag{11.2.77}$$

It is interesting to note that, using an entirely different approach, the numerical diffusion similar to (11.2.76) and (11.2.77) arises in the generalized Petrov-Galerkin (GPG) methods to be presented in Sections 11.3 and 11.4. More general treatments of TGM will be covered in Section 13.2.

## 11.3 NUMERICAL DIFFUSION TEST FUNCTIONS

In GGM described in Section 11.2, various degrees of polynomials (linear, quadratic, cubic, etc.) may be adopted for desired accuracy of solution. However, in convection-dominated problems, an adequate amount of numerical diffusion or artificial viscosity is required for numerical stability. To this end, the so-called streamline-upwind Petrov-Galerkin (SUPG) method [Heinrich et al., 1977; Hughes and Brooks, 1982] has been successfully used. In this case, the local finite element test functions consist of standard Galerkin test functions plus numerical diffusion test functions. There are many forms of numerical diffusion test functions as reported by Hughes and his co-workers during the 1980s. A similar approach is referred to as the streamline diffusion method (SDM) by Johnson [1987].

Computational stability is provided effectively through various forms of SUPG, SDM, or other similar strategies. All of these approaches are nonstandard Galerkin methods and, for simplicity, they may be combined into a single name "Generalized Petrov-Galerkin (GPG) methods. The concept of GPG for the one-dimensional Burgers' equation will be introduced first in order to identify a one-dimensional numerical diffusion test function which provides the numerical stability, followed by multidimensional numerical diffusion test functions representing the streamline diffusion and discontinuity-capturing schemes.

### 11.3.1 DERIVATION OF NUMERICAL DIFFUSION TEST FUNCTIONS

The concept of streamline diffusion began with the backward (often called upwinding) finite difference scheme for the convection-diffusion equation first given by Spalding [1972]. The idea is to introduce the numerical diffusion in the direction of flow or along the streamline parallel to the velocity in order to obtain stable solutions. In the following, we use the convection-diffusion equation to demonstrate the concept of streamline upwinding or streamline diffusion. Our objective here is to prove that numerical stability can be achieved by test functions written in the form,

$$W_N^{(e)} = \Phi_N^{(e)} + \Psi_N^{(e)} \tag{11.3.1}$$

where $W_N^{(e)}$ represents the generalized Petrov-Galerkin test functions which are the sum of the standard Galerkin test function $\Phi_N^{(e)}$ and the numerical diffusion test function $\Psi_N^{(e)}$. The numerical diffusion test function $\Psi_N^{(e)}$ in (11.3.1) is intended for adding numerical diffusion practiced in the finite difference literature. However, in the sequel, it will be shown that the derivation of numerical diffusion test functions involves significant physical aspects of convection-dominated flows.

To elucidate the argument involved in this approach, we look at the unsteady convection equation of the form

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{11.3.2}$$

Substituting (11.3.2) into Taylor series of the type (11.2.9), we obtain

$$u^{n+1} = u^n + \Delta t\left(-a\frac{\partial u^n}{\partial x}\right) + \frac{\Delta t^2}{2}\left(a^2\frac{\partial^2 u^n}{\partial x^2}\right) \tag{11.3.3}$$

If $u^{n+1} = u^n$ at steady-state, we may set $a\Delta t = C\Delta x$, where $C$ is the nondimensional artificial viscosity (equal to Courant number for $a = u$, or $C = u\Delta t/\Delta x$), and rewrite (11.3.3) in the form

$$a\left(\frac{\partial u}{\partial x} - \frac{C\Delta x}{2}\frac{\partial^2 u}{\partial x^2}\right) = 0 \tag{11.3.4}$$

in which the second term of the left-hand side of (11.3.4) represents the numerical diffusion, equivalent to the artificial viscosity. Denoting $\alpha = C/2$ and $h = \Delta x$ as the nondimensional numerical diffusion parameter and the mesh parameter, respectively, we may construct the following inner product:

$$\int \Phi_N^{(e)} a\left(\frac{\partial u}{\partial x} - \alpha h\frac{\partial^2 u}{\partial x^2}\right) dx = 0 \tag{11.3.5}$$

Integrating (11.3.5) by parts, we obtain

$$\int \left(\Phi_N^{(e)} + \alpha h\frac{\partial \Phi_N^{(e)}}{\partial x}\right) a\frac{\partial u}{\partial x} dx = \overset{*}{\Phi}_N a\alpha h\frac{\partial u}{\partial x}$$

where the integral on the left-hand side is known as the Petrov-Galerkin integral. For

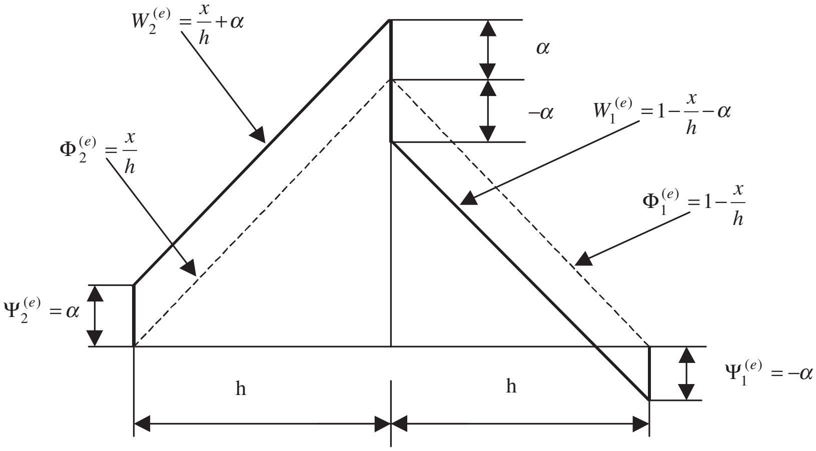

$W_2^{(e)} = \frac{x}{h} + \alpha$
$\alpha$
$-\alpha$
$W_1^{(e)} = 1 - \frac{x}{h} - \alpha$
$\Phi_2^{(e)} = \frac{x}{h}$
$\Phi_1^{(e)} = 1 - \frac{x}{h}$
$\Psi_2^{(e)} = \alpha$
$\Psi_1^{(e)} = -\alpha$
h
h

**Figure 11.3.1** Linear generalized test functions for one-dimensional element $W_N^{(e)}$ with $\Psi_N^{(e)} = \alpha h(\partial \Phi_N^{(e)}/\partial x)$ constant within the element, discontinuous at boundaries.

Dirichlet problems $\overset{*}{\Phi}_N = 0$, we have

$$\int W_N^{(e)} \left( a \frac{\partial u}{\partial x} \right) dx = 0 \tag{11.3.6}$$

with

$$W_N^{(e)} = \Phi_N^{(e)} + \Psi_N^{(e)} \tag{11.3.7}$$

$$\Phi_N^{(e)} = \left( 1 - \frac{x}{h}, \frac{x}{h} \right) \tag{11.3.8}$$

$$\Psi_N^{(e)} = \alpha h \frac{\partial \Phi_N^{(e)}}{\partial x} \tag{11.3.9}$$

The foregoing process indicates that the numerical diffusion may be applied to the convective term in (11.3.2) through (11.3.6), with the explicit form given by (11.3.9) representing the numerical diffusion test function. This is a variational weak form as constructed by the inner product of the numerical diffusion test function and the convection term.

Substituting (11.3.8) into (11.3.9), we arrive at

$$\Psi_N^{(e)} = [-\alpha, \quad \alpha] \tag{11.3.10}$$

which indicates that the numerical test function is constant within an element, equal to one-half of the Courant number, but discontinuous at boundaries, as shown in Figure 11.3.1. It will be shown that one-dimensional numerical diffusion test functions given in (11.3.9) arise from the two-dimensional numerical diffusion test functions to be discussed in Section 11.3.2.

## 11.3.2 STABILITY AND ACCURACY OF NUMERICAL DIFFUSION TEST FUNCTIONS

Let us examine the convection-diffusion equation,

$$\hat{R} \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = 0 \tag{11.3.11}$$

with $\hat{R}$ being the Reynolds number (per unit length) $\hat{R} = \rho u/\mu = u/d$, where $d$ is the kinematic viscosity (but will be referred to as diffusivity in the following). Notice that $\hat{R} = \rho c_p u/k$ is regarded as the Peclet number if $u$ is taken as temperature. Then $d = k/\rho c_p$ becomes the thermal diffusivity. We write the local element Petrov-Galerkin integral for (11.3.11) as

$$\int_0^h W_N^{(e)}\left(\hat{R}\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2}\right)dx = \int_0^h \left(\Phi_N^{(e)} + \Psi_N^{(e)}\right)\left(\hat{R}\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2}\right)dx = 0 \qquad (11.3.12)$$

Apply integration by parts only to the product with the standard Galerkin test function $\Phi_N^{(e)}$ which will then produce a boundary term, whereas the integration of the product term with the numerical diffusion test function $\Psi_N^{(e)}$ is to be performed only over the interior domain, not involving the boundaries.

$$\left\{\int_0^h \left[\hat{R}\left(\Phi_N^{(e)}\frac{\partial \Phi_M^{(e)}}{\partial x} + \alpha h \frac{\partial \Phi_N^{(e)}}{\partial x}\frac{\partial \Phi_M^{(e)}}{\partial x}\right) + \frac{\partial \Phi_N^{(e)}}{\partial x}\frac{\partial \Phi_M^{(e)}}{\partial x} - \alpha h \frac{\partial \Phi_N^{(e)}}{\partial x}\frac{\partial^2 \Phi_M^{(e)}}{\partial x^2}\right]dx\right\} u_M^{(e)}$$

$$= \overset{*}{\Phi}{}_N^{(e)} \frac{\partial u}{\partial x}\bigg|_0^h \qquad (11.3.13a)$$

If linear trial functions are used, then the second derivative term vanishes, so that we have

$$\left(B_{NM}^{(e)} + C_{NM}^{(e)}\right)u_M^{(e)} + K_{NM}^{(e)} u_M^{(e)} = G_N^{(e)} \qquad (11.3.13b)$$

where

$$B_{NM}^{(e)} = \int_0^h \hat{R}\Phi_N^{(e)}\frac{\partial \Phi_M^{(e)}}{\partial x}dx$$

is the standard convection matrix and

$$C_{NM}^{(e)} = \int_0^h \hat{R}\alpha h \frac{\partial \Phi_N^{(e)}}{\partial x}\frac{\partial \Phi_M^{(e)}}{\partial x}dx$$

represents the numerical diffusion matrix implying the numerical diffusion arising from the convection term. The last integral term $K_{NM}^{(e)}$ is identified as the physical diffusion matrix.

$$K_{NM}^{(e)} = \int_0^h \frac{\partial \Phi_N^{(e)}}{\partial x}\frac{\partial \Phi_M^{(e)}}{\partial x}dx$$

Evaluating these integrals, we obtain

$$B_{NM}^{(e)} + C_{NM}^{(e)} = \frac{\hat{R}}{2}\begin{bmatrix} -1+2\alpha & 1-2\alpha \\ -1-2\alpha & 1+2\alpha \end{bmatrix}$$

$$K_{NM}^{(e)} = \frac{1}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Consider a two-element system with nodes at $i-1$, $i$, and $i+1$ and the global form of (11.3.13). Expanding the global equation corresponding to the node at $i$ and assuming that the Neumann boundary conditions are unspecified ($\overset{*}{\Phi}{}_N^{(e)} = 0$), we obtain

$$\left[1 + \frac{R}{2}(2\alpha + 1)\right] u_{i-1} - (2 + 2R\alpha)\, u_i + \left[1 + \frac{R}{2}(2\alpha - 1)\right] u_{i+1} = 0 \tag{11.3.14}$$

where $R$ is the local Reynolds number, $R = \hat{R}h$. Equation (11.3.14) represents the forward, central, and backward finite difference equations for $\alpha = -1/2$, $\alpha = 0$, and $\alpha = 1/2$, respectively. The backward difference form ($\alpha = 1/2$) given by

$$\hat{R}\frac{u_i - u_{i-1}}{h} - \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0 \tag{11.3.15a}$$

can be modified by transforming the convection term into the central difference form to identify the numerical diffusion with the coefficient $\hat{R}h/2$,

$$\hat{R}\left(\frac{u_{i+1} - u_{i-1}}{2h}\right) - \left(\frac{\hat{R}h}{2} + 1\right)\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right) = 0 \tag{11.3.15b}$$

This is equivalent to the differential equation

$$\hat{R}\frac{\partial u}{\partial x} - \hat{\alpha}\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial x^2} = 0 \tag{11.3.16}$$

with $\hat{\alpha} = \hat{R}h/2$ being the coefficient of numerical viscosity and $\hat{\alpha}(\partial^2 u/\partial x^2)$ representing the effect of numerical diffusion. We say that the effect of numerical diffusion is built into this equation if the backward difference is used. We may consider $\hat{\alpha}$ as being equivalent to the artificial viscosity.

To obtain the condition for stability (11.3.14), we proceed as follows: Let $G = 1 + R\alpha$ and $H = R/2$. Rewrite (11.3.14) in the form

$$(G - H)u_{i+1} - 2Gu_i + (G + H)u_{i-1} = 0 \tag{11.3.17}$$

where we assume the relations at the nodes $i + 1$, $i$, and $i - 1$ as

$$u_i = c\phi^i, \qquad u_{i+1} = c\phi^{i+1}, \qquad u_{i-1} = c\phi^{i-1} \tag{11.3.18a,b,c}$$

Substituting the above into (11.3.17) yields

$$(G - H)\,\phi^{i+1} - 2G\phi^i + (G + H)\,\phi^{i-1} = 0$$

For $i = 1$, we obtain the quadratic equation

$$(G - H)\phi^2 - 2G\phi + (G + H) = 0$$

Solving for $\phi$, we arrive at two values of $\phi$

$$\phi = \begin{cases} 1 \\ \dfrac{G + H}{G - H} \end{cases}$$

These results call for two constants in (11.3.18).

Now we revise the relation in (11.3.18a) in the form

$$u_i = c_1 + c_2\left(\frac{G + H}{G - H}\right)^i = c_1 + c_2\left[\frac{1 + \dfrac{R}{2}(2\alpha + 1)}{1 + \dfrac{R}{2}(2\alpha - 1)}\right]^i \tag{11.3.19}$$

For stability, the denominator of the $c_2$ term must be positive,

$$G - H > 0$$

or

$$1 + \frac{R}{2}(2\alpha - 1) > 0 \tag{11.3.20a}$$

which provides the stability criteria

$$\begin{cases} \alpha = 0 & \text{if } R < 2 \\ \alpha \geq \dfrac{1}{2} - \dfrac{1}{R} & \text{if } R \geq 2 \end{cases} \tag{11.3.20b,c}$$

It is clear that the forward difference with $\alpha = -1/2$ (11.3.20a) becomes unconditionally unstable for $R > 1$, whereas the central difference ($\alpha = 0$) is conditionally stable and the backward difference ($\alpha = 1/2$) provides an unconditional stability. For accuracy, we set the exact solution as

$$u = c_1 + c_2 e^{\hat{R}x}$$

which, for $x = hi$, becomes

$$u_i = c_1 + c_2 e^{Ri} \tag{11.3.21}$$

Setting (11.3.21) equal to (11.3.19), we obtain the relationship

$$\left[\frac{1 + \dfrac{R}{2}(2\alpha + 1)}{1 + \dfrac{R}{2}(2\alpha - 1)}\right]^i = e^{Ri}$$

Taking a natural logarithm of the above leads to

$$\mathrm{l}n\left(\frac{G + H}{G - H}\right) = 2\,\mathrm{cot}h^{-1}\left(\frac{G}{H}\right) = 2\,\mathrm{cot}h^{-1}\left(\frac{1 + R\alpha}{R/2}\right) = R$$

from which we obtain

$$2\alpha = \mathrm{cot}h\left(\frac{R}{2}\right) - \frac{2}{R} \tag{11.3.22}$$

with

$$\alpha = \frac{1}{2}C = \frac{1}{2}\overline{\alpha} \tag{11.3.23}$$

This is the criterion for accuracy. Here, the one-dimensional numerical diffusion parameter $\alpha$, which assures the accuracy, is found to be a function of the local Reynolds number. It should be noted that the value of $\alpha$ is one-half of that in Heinrich et al. [1977], and $\overline{\alpha} = C$, called the effective numerical diffusion parameter, is indeed the Courant number.

Substituting (11.3.23) into (11.3.22) leads to

$$\overline{\alpha} = \mathrm{cot}h\,H - \frac{1}{H} \tag{11.3.24}$$

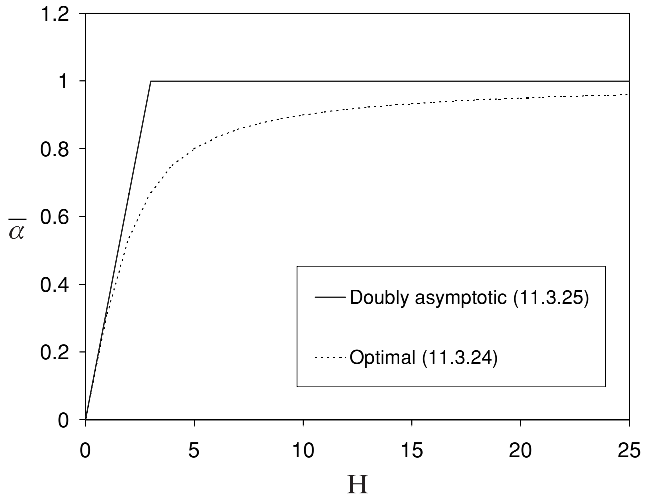


**Figure 11.3.2** Effective numerical diffusivity $\overline{\alpha}$.

It can be shown that, expanding coth$H$ in infinite series and retaining terms of fourth order accuracy in $H$ (doubly asymptotic approximation) results in

$$\overline{\alpha} = H/3, \quad \text{if } -3 \le H \le 3 \tag{11.3.25a}$$

$$\overline{\alpha} = \operatorname{sgn} H, \quad \text{if } |H| > 3 \tag{11.3.25b}$$

The values of $\overline{\alpha}$ determined by (11.3.20), (11.3.24), and (11.3.25) are referred to as the critical value, optimal value, and higher order value, respectively (Figure 11.3.2) [Heinrich et al., 1977; Brooks and Hughes, 1982]. It is seen that the doubly asymptotic approximation (11.3.25) is the simpler and practical approach.

It follows from these observations that, for two-dimensional isoparametric elements, the numerical diffusion parameters $\alpha_\xi$ and $\alpha_\eta$ are defined as (Figure 11.3.3)

$$\alpha_\xi = \frac{1}{2}\overline{\alpha}_\xi \tag{11.3.26a}$$

$$\alpha_\eta = \frac{1}{2}\overline{\alpha}_\eta \tag{11.3.26b}$$

with the two-dimensional effective numerical diffusion parameters, $\overline{\alpha}_\xi$ and $\overline{\alpha}_\eta$, defined as

$$\overline{\alpha}_\xi = \coth\left(\frac{R_\xi}{2}\right) - \frac{2}{R_\xi} \tag{11.3.27a}$$

$$\overline{\alpha}_\eta = \coth\left(\frac{R_\eta}{2}\right) - \frac{2}{R_\eta} \tag{11.3.27b}$$

where the local Reynolds numbers in the $\xi$ and $\eta$ directions are of the form

$$R_\xi = \frac{\mathrm{v}_\xi h_\xi}{d}, \qquad R_\eta = \frac{\mathrm{v}_\eta h_\eta}{d}$$

For multidimensional convection-dominated problems, the directional properties of velocity are expected to play a key role. The numerical diffusion must be provided in the direction of flow or along the streamlines parallel to the velocity in both steady and

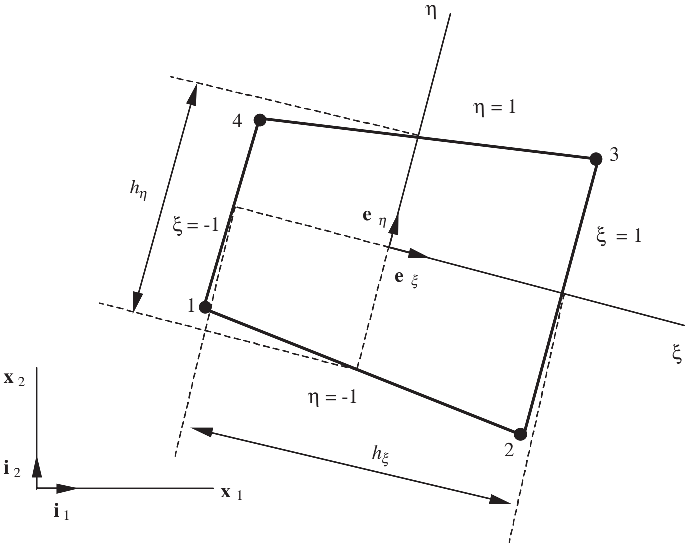


**Figure 11.3.3** Four node isoparametric element; $\mathbf{e}_\xi$ and $\mathbf{e}_\eta$ are the unit vectors along the ξ and η axes, respectively.

time-dependent problems as well. First of all, the gradient of the standard test function in (11.3.9) is of a vector quantity. This will require that the dot product of the gradient of the standard test function $\Phi_N^{(e)}$ with another vector quantity such as the velocity vector along the streamlines be formed such that a scalar function arises at each local node $N$,

$$\mathbf{v}\cdot\nabla\Phi_N^{(e)} = \mathrm{v}_i\Phi_{N,i}^{(e)} \quad (i = 1, 2) \tag{11.3.28}$$

Thus, the quantity $\alpha h$ in (11.3.9) for the one-dimensional case must be altered to accommodate the appropriate dimensional properties in (11.3.28) which call for a scalar, say $\tau$, such that

$$\Psi_N^{(e)} = \tau\mathrm{v}_i\Phi_{N,i}^{(e)} \tag{11.3.29}$$

with $\tau$ being the numerical diffusion factor having dimensions of time (often called the intrinsic time scale),

$$\tau = \frac{1}{2}(\alpha_\xi h_\xi \mathrm{v}_\xi + \alpha_\eta h_\eta \mathrm{v}_\eta)/S = \frac{1}{4}(\overline{\alpha}_\xi h_\xi \mathrm{v}_\xi + \overline{\alpha}_\eta h_\eta \mathrm{v}_\eta)/S = \frac{1}{\sqrt{16}}(\overline{\alpha}_\xi h_\xi \mathrm{v}_\xi + \overline{\alpha}_\eta h_\eta \mathrm{v}_\eta)/S \tag{11.3.30}$$

where $S = \mathbf{v}\cdot\mathbf{v}$, and $\overline{\alpha}_\xi = \coth(R_\xi/2) - 2/R_\xi$, etc. The coefficient $1/\sqrt{16}$ for the numerical diffusion factor $\tau$ in (11.3.30) disagrees with an arbitrary value of $1/\sqrt{15}$ adopted by Raymond and Garder [1976], and subsequently Brooks and Hughes [1982] as determined from the numerical experimentation for unsteady flows. The derivation demonstrated here, however, is based on the definition of Courant number and the criterion for accuracy which leads to $1/\sqrt{16}$ instead of $1/\sqrt{15}$. For the purpose of generality, the solution schemes employing the numerical diffusion test function given by (11.3.29) are termed "generalized Petrov-Galerkin (GPG)" instead of SUPG. The unfortunate choice of the term "SUPG" for various reasons was discussed in Hughes [1987]. The SUPG methods as referred to today imply far beyond the classical upwind methods [Spalding, 1972] or classical Petrov-Galerkin methods [Mikhlin, 1964] so that more general identification appears to be in order. Thus, it is suggested that the term "generalized Petrov-Galerkin (GPG)" may be a reasonable compromise.

For two-dimensional elements with isoparametric coordinates (Figure 11.3.3), we express the velocity components as

$$\mathrm{v}_\xi = \mathbf{v}\cdot\mathbf{e}_\xi, \qquad \mathrm{v}_\eta = \mathbf{v}\cdot\mathbf{e}_\eta$$

where the isoparametric unit vectors $\mathbf{e}_\xi$ and $\mathbf{e}_\eta$ are given by

$$\mathbf{e}_\xi = \frac{1}{\sqrt{J}_\xi}\frac{\partial x_i}{\partial \xi}\mathbf{i}_i, \qquad \mathbf{e}_\eta = \frac{1}{\sqrt{J}_\eta}\frac{\partial x_i}{\partial \eta}\mathbf{i}_i, \qquad J_\xi = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2, \; J_\eta = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2$$

It follows from (11.3.27) that the two-dimensional numerical diffusion test function reduces to that of one dimension given by (11.3.9):

$$\Psi_N^{(e)} = \tau \mathrm{v}_1 \Phi_{N,1}^{(e)} = \left(\frac{\alpha h u}{u^2}\right) u \frac{\partial \Phi_N^{(e)}}{\partial x} = \alpha h \frac{\partial \Phi_N^{(e)}}{\partial x} \tag{11.3.31}$$

which establishes the complete link between the one- and two-dimensional aspects of the numerical diffusion test functions. It is interesting to note that, in due course of derivation of the one-dimensional numerical diffusion test function (11.3.9), the notion of time scale for the numerical diffusion factor $\tau$ did not arise, but is now taken into account as the numerical diffusion must be applied in the direction of flow with velocity specified in multidimensional cases.

Due to the fact that the gradient $\nabla \Phi_N^{(e)}$ is included in $\Psi_N^{(e)}$, it is clear that the use of the generalized test functions (11.3.1) brings the numerical diffusion automatically into the formulation. This is equivalent to the retention of artificial viscosity terms in FDM.

Using the similar procedure, the test functions for 3-D problems (with isoparametric coordinates $\xi$, $\eta$, and $\zeta$) can be obtained. The three-dimensional test function may still be written in the general form (11.3.29).

$$\Psi_N^{(e)} = \tau \mathrm{v}_i \Phi_{N,i}^{(e)}, \quad (i = 1, 2, 3) \tag{11.3.32}$$

where

$$\tau = \frac{1}{6}(\overline{\alpha}_\xi h_\xi \mathrm{v}_\xi + \overline{\alpha}_\eta h_\eta \mathrm{v}_\eta + \overline{\alpha}_\zeta h_\zeta \mathrm{v}_\zeta)/S \tag{11.3.33}$$

$$\overline{\alpha}_\zeta = \coth\left(\frac{R_\zeta}{2}\right) - \frac{2}{R_\zeta}, \qquad R_\zeta = \frac{\mathrm{v}_\zeta h_\zeta}{d}, \qquad S = u^2 + \mathrm{v}^2 + w^2$$

Thus

$$\Psi_N^{(e)} = \tau\left(u\frac{\partial \Phi_N^{(e)}}{\partial x} + \mathrm{v}\frac{\partial \Phi_N^{(e)}}{\partial y} + w\frac{\partial \Phi_N^{(e)}}{\partial z}\right)$$

Once again, it should be emphasized that the numerical diffusion is activated along the stream line direction, which provides numerical stability. However, it has been observed that, as the convection domination becomes significant, it is not possible to eliminate entirely some numerical oscillations. We require additional measures in order to resolve numerical stability, known as the discontinuity-capturing scheme, which is discussed next.

### 11.3.3 DISCONTINUITY-CAPTURING SCHEME

In the presence of very high gradients of a variable such as in shock waves, one may apply numerical diffusion parallel to the direction of velocity gradients in addition to the streamline direction. Hughes et al. [1986] and Johnson [1987] investigated the so-called "discontinuity-capturing scheme" (DCS) and demonstrated improvements over the case of a numerical diffusion function applied only along the streamline direction. To be consistent with the present notations adopted in this book, a slightly modified version of DCS is presented below.

The basic idea is that the numerical diffusion is applied not only in the direction of velocity along the streamline, $\mathrm{v}^i = \mathrm{v}_i^{(a)}$, but also along the direction, $\mathrm{v}_i = \mathrm{v}_i^{(b)}$, parallel to the velocity gradients directed toward acceleration as shown in Figure 11.3.4. Note that $\mathrm{v}_i^{(b)}$ is the projection of $\mathrm{v}_i^{(a)}$ and thus the effect of the discontinuity-capturing will be significant if the angle θ becomes small, which represents very sharp surface discontinuities such as in shock waves.

To implement this scheme, we consider that the numerical diffusion test functions consist of the sum of the streamline numerical diffusion test function, $\Psi_N^{(a)}$, and the gradient numerical diffusion test function, $\Psi_N^{(b)}$,

$$\Psi_N^{(e)} = \Psi_N^{(a)} + \Psi_N^{(b)} \tag{11.3.34}$$

with

$$\Psi_N^{(a)} = \tau \mathrm{v}_i^{(a)} \Phi_{N,i}^{(e)} = \tau \mathrm{v}_i \Phi_{N,i}^{(e)} \quad \text{for streamline diffusion} \tag{11.3.35a}$$

$$\Psi_N^{(b)} = \tau \mathrm{v}_i^{(b)} \Phi_{N,i}^{(e)} = \tau^{(b)} \mathrm{v}_i \Phi_{N,i}^{(e)} \quad \text{for discontinuity-capturing} \tag{11.3.35b}$$

where $\mathrm{v}_i^{(a)} = \mathrm{v}_i$, and $\mathrm{v}_i^{(b)}$ is the projection of $\mathrm{v}_i^{(a)}$ parallel to velocity gradients directed toward acceleration per unit mass, $A_j$,

$$A_j = \mathrm{v}_{j,k}\mathrm{v}_k \tag{11.3.36}$$

$$\mathrm{v}_i^{(b)} = \mathrm{v}_i^{(a)} \cos\theta = \mathrm{v}_\mathrm{i}\mathrm{v}_j A_j/\gamma \tag{11.3.37}$$

$$\tau^{(b)} = \tau \mathrm{v}_j A_j/\gamma, \quad \text{discontinuity-capturing factor} \tag{11.3.38}$$

with $\gamma = |\mathrm{v}_j||A_j|$. Thus, it is seen that the discontinuity-capturing scheme is simply to add an extra numerical diffusion test function applied parallel to the velocity gradients directed toward acceleration. For distorted elements, we may encounter $\tau^{(b)} - \tau$ to be

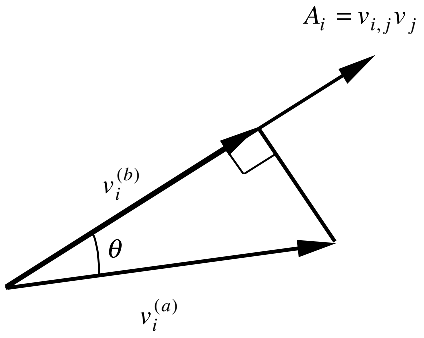


**Figure 11.3.4** Discontinuity capturing scheme, $v_i^{(a)} = v_i$, $v_i^{(b)} = v_i^{(a)} \cos\theta = v_i v_j A_j/|v_j||A_j|$

negative. In this case we choose

$$\tau^{(b)} - \tau = \max\left(0, \quad \tau^{(b)} - \tau\right) \tag{11.3.39}$$

so that $\tau^{(b)} - \tau$ always remains positive. Further details are found in Hughes et al. [1986].

## 11.4 GENERALIZED PETROV-GALERKIN (GPG) METHODS

### 11.4.1 GENERALIZED PETROV-GALERKIN METHODS FOR UNSTEADY PROBLEMS

For illustration, let us consider the Burgers' equation in the form,

$$R_i = \frac{\partial \mathrm{v}_i}{\partial t} + \mathrm{v}_{i,j}\mathrm{v}_j - \nu \mathrm{v}_{i,jj} - f_i = 0$$

The finite element formulation of the generalized Petrov-Galerkin (GPG) methods using the numerical diffusion test functions projected on the discontinuous temporal test function or DST as given in (8.2.41) or (10.2.5) is written in the form.

$$\int_\xi \hat{W}(\xi) \int_\Omega W_\alpha R_i \, d\Omega d\xi = 0 \tag{11.4.1}$$

Here, the temporal test functions $\hat{W}(\xi)$ were discussed in Section 10.2.1, whereas the Petrov-Galerkin test functions $W_\alpha$ are the global form of the local test functions as the sum of the standard Galerkin test functions and the numerical diffusion test function for streamline diffusion.

$$W_\alpha = \Phi_\alpha + \Psi_\alpha^{(a)} \tag{11.4.2}$$

If the discontinuity-capturing scheme is desired, this can be added to (11.4.1) by constructing the product of $\Psi_\alpha^{(b)}$ and the convection term of the residual, leading to the GPG equations of the form,

$$\int_\xi \hat{W}(\xi) \int_\Omega \left[ \left(\Phi_\alpha + \Psi_\alpha^{(a)}\right)\left(\frac{\partial \mathrm{v}_i}{\partial t} + \mathrm{v}_{i,j}\mathrm{v}_j - \nu \mathrm{v}_{i,jj} - f_i\right) + \Psi_\alpha^{(b)} \mathrm{v}_j \mathrm{v}_{i,j}\right] d\Omega d\xi = 0 \tag{11.4.3}$$

Note that the integration by parts is to be performed only with respect to the Galerkin test functions, which will lead to the Neumann boundary conditions, whereas those terms of the residual associated with numerical diffusion test functions will not be integrated by parts since they should be contained within the elements as a measure of numerical diffusion. Thus, the GPG integral takes the form, known as the *variational equation*,

$$\begin{aligned}
&\int_\xi \hat{W}(\xi)\left[\int_\Omega \left(\Phi_\alpha \Phi_\beta \frac{\partial \mathrm{v}_{\beta i}}{\partial t} + \mathrm{v}_j \Phi_\alpha \Phi_{\beta,j} \mathrm{v}_{\beta i} + \nu \Phi_{\alpha,j}\Phi_{\beta,j}\mathrm{v}_{\beta i} - \Phi_\alpha f_i\right) d\Omega\right. \\
&\quad \left. - \int_\Gamma \Phi_\alpha^* \nu \mathrm{v}_{i,j} n_j d\Gamma\right] d\xi + \int_\xi \hat{W}(\xi) \int_\Omega \tau \mathrm{v}_k \Phi_{\alpha,k}\left(\frac{\partial \Phi_\beta \mathrm{v}_{\beta i}}{\partial t} + \mathrm{v}_j \Phi_{\beta,j}\mathrm{v}_{\beta i}\right. \\
&\quad \left. - \nu \Phi_{\beta,jj}\mathrm{v}_{\beta i} - f_i\right) d\Omega d\xi + \int_\xi \hat{W}(\xi) \int_\Omega \tau^{(b)} \mathrm{v}_k \mathrm{v}_j \Phi_{\alpha,k}\Phi_{\beta,j}\mathrm{v}_{\beta i} \, d\Omega d\xi = 0
\end{aligned} \tag{11.4.4}$$

The first integral indicates the Galerkin integral, with the second representing the streamline diffusion, and the third integral indicates the discontinuity-capturing.

Assume that the trial function is linear, independent of time, with the numerical diffusion due to the source term being negligible. Furthermore, if the temporal test function, $W(\xi) = \delta(\xi - 1/2)$ or $W(\xi) = 1$ is used and the variation of nodal values of the variables $\mathrm{v}_i$ is linear, then we obtain [see (10.2.13) or (11.2.6)]

$$[A_{\alpha\beta} + \eta\Delta t(B_{\alpha\beta} + C_{\alpha\beta} + K_{\alpha\beta})]\,\mathrm{v}_{\beta i}^{n+1} = [A_{\alpha\beta} - (1-\eta)\Delta t(B_{\alpha\beta} + C_{\alpha\beta} + K_{\alpha\beta})]\,\mathrm{v}_{\beta i}^{n} + \Delta t(F_{\alpha i} + G_{\alpha i}) \tag{11.4.5}$$

where the definitions of all terms are shown in Section 11.2 except that various forms of the numerical diffusion matrix, $C_{\alpha\beta}$, are given below.

$$C_{\alpha\beta} = \int_\Omega \tau \mathrm{v}_k \mathrm{v}_j \Phi_{\alpha,k}\Phi_{\beta,j}\,d\Omega \tag{11.4.6a}$$

for streamline diffusion, and

$$C_{\alpha\beta} = \int_\Omega \left(\tau + \tau^{(b)}\right) \mathrm{v}_k \mathrm{v}_j \Phi_{\alpha,k}\Phi_{\beta,j}\,d\Omega \tag{11.4.6b}$$

for combined streamline diffusion and discontinuity-capturing. It is seen that the numerical difffusion factor $\tau$ or $\tau + \tau^{(b)}$ in GPG corresponds to $\Delta t/2$ in (11.2.76) for TGM, but is much more complicated and actually flowfield-dependent. Note also that effects of numerical diffusion associated with terms other than convection are neglected in (11.4.5). The complexity of the numerical diffusion factor increases significantly for the case of the Navier-Stokes system of equations as discussed in Section 13.3.

Various options for temporal approximations or higher order accuracy may be selected as discussed in Section 10.2. For the case of streamline diffusion (11.4.6a) with the temporal parameter, $\eta = 1$, and linear trial and test functions of finite elements, the expression given by (11.4.5) is identical to equation 25 of Shakib and Hughes [1991] for the constant-in-time approximations of the space-time Galerkin/least squares (GLS) in one-dimensional problems. The GLS formulation will be described in the following section.

### 11.4.2 SPACE-TIME GALERKIN/LEAST SQUARES METHODS

The formal discussion of the least squares methods (LSM) of obtaining the FEM equations will be presented in the later chapters. However, in order to understand the Galerkin/least squares (GLS) methods reported by Hughes and his co-workers, we examine briefly a basic procedure for the least squares formulation. First, let us introduce the least squares variational function,

$$\Pi = \int_\Omega \frac{1}{2} R_j R_j\,d\Omega$$

which is then to be minimized with respect to the nodal variables $\mathrm{v}_{\alpha i}$. In this process, we multiply $\Pi$ by the numerical diffusion factor, $\tau$.

$$\delta\Pi = \frac{\partial \tau \Pi}{\partial \mathrm{v}_{\alpha i}}\delta \mathrm{v}_{\alpha i} = 0 \tag{11.4.7}$$

or

$$\frac{\partial \tau \Pi}{\partial \mathrm{v}_{\alpha i}} = \tau \int_\Omega \frac{\partial R_j}{\partial \mathrm{v}_{\alpha i}} R_j\,d\Omega = 0 \tag{11.4.8}$$

Performing the differentiation in (11.4.8) and applying the temporal approximations, we obtain

$$\int_\xi \hat{W}(\xi)\int_\Omega \tau\left(\frac{\partial}{\partial t}+\mathrm{v}_k\frac{\partial}{\partial x_k}-\nu\frac{\partial^2}{\partial x_k\partial x_k}\right)\Phi_\alpha\left(\frac{\partial \mathrm{v}_\mathrm{i}}{\partial t}+\mathrm{v}_j\mathrm{v}_{i,j}-\nu\mathrm{v}_{i,jj}-f_i\right)d\Omega d\xi=0 \tag{11.4.9}$$

which may be written as

$$\int_\xi \hat{W}(\xi)\int_\Omega \tau(L\Phi_\alpha)(L\mathrm{v}_i-f_i)\,d\Omega d\xi=0 \tag{11.4.10}$$

where $L$ is the differential operator,

$$L=\frac{\partial}{\partial t}+\mathrm{v}_k\frac{\partial}{\partial x_k}-\nu\frac{\partial^2}{\partial x_k\partial x_k} \tag{11.4.11}$$

At this point, we add the least squares integral (11.4.10) and the discontinuity-capturing term as developed in Section 11.3.3 to the standard Galerkin integral. If we choose only the convective term in (11.4.11), then, these steps lead to the form identical to the generalized Petrov-Galerkin scheme given by (11.4.4). The sum of the standard Galerkin integral, the discontinuity capturing term, and the least squares integral represented by (11.4.10) is referred to as the space-time Galerkin/least squares (GLS) methods [Hauke and Hughes, 1998]. Note that the contributions from additional terms other than the convective terms in (11.4.11) are negligible.

The space-time GLS formulation is another form of generalized Petrov-Galerkin (GPG) methods in which the only difference from the GPG methods of Section 11.4.1 is the numerical diffusion test functions for streamline diffusion,

$$\Psi_\alpha^{(a)}=\tau L\Phi_\alpha \tag{11.4.12}$$

where the numerical diffusion factor $\tau$ can be constructed by introducing the local curvilinear coordinate contravariant metric tensor [Shakib and Hughes, 1991],

$$g^{ij}=\left(\frac{\partial x_k}{\partial \xi_i}\frac{\partial x_k}{\partial \xi_j}\right)^{-1} \tag{11.4.13}$$

With some algebra, it can be shown that one possible option for $\tau$ is of the form

$$\tau=\left[\left(\frac{2}{\Delta t}\right)^2+\left(\frac{2|\mathrm{v}_i|}{|h_i|}\right)^2+9\left(\frac{4\nu}{|h_i|^2}\right)^2\right]^{-\frac{1}{2}} \tag{11.4.14}$$

where $h_i$ denotes the average element size in local coordinates. Note that if only the convective term is chosen in (11.4.9), then the GLS formulation becomes identical to the GPG formulation given by (11.4.4). The standard least squares methods will be discussed in Section 12.1.8 for incompressible flows and in Section 14.2 for compressible flows.

Applications of GPG to the Navier-Stokes system of equations require some modifications for the numerical diffusion test functions in which entropy variables can be employed to advantage. This subject will be discussed in Section 13.4.

***Remarks:*** The temporal integral with the temporal test function $\hat{W}(\xi)$ first introduced in (10.2.5) plays the role identical to the process referred to as the discontinuous space-time integral [Shakib and Hughes, 1991; Tezduyar, 1997]. Many possible options of this temporal test function can be chosen (Tables 10.2.1 and 10.2.2). Explicit forms of integrals (11.4.4) plus the least squares integrals (11.4.9) as applied to the Navier-Stokes system of equations are shown in (13.3.19).

## 11.5 SOLUTIONS OF NONLINEAR AND TIME-DEPENDENT EQUATIONS AND ELEMENT-BY-ELEMENT APPROACH

As was shown in Section 10.3.2, the global assembly of local stiffness matrices can be avoided via the element-by-element (EBE) scheme. In dealing with nonlinear and time-dependent equations, however, some modifications are required. We discuss in this section the Newton-Raphson methods of solving nonlinear time-dependent equations, followed by the generalized minimal residual (GMRES) equation solver and EBE scheme.

### 11.5.1 NEWTON-RAPHSON METHODS

Recall that in Section 11.2.1 we held $\overline{\mathrm{v}}_j$ constant in $\overline{\mathrm{v}}_j \mathrm{v}_{i,j}$, which was meant to be updated in each step of calculations. Otherwise, GGM or GPG, methods described in the previous sections, must be modified in order to solve nonlinear equations. For example, we may write (11.2.6) of the GGM formulation in the form where $\overline{\mathrm{v}}_j$ is no longer held constant.

$$E_{\alpha i} = A_{\alpha\beta}\mathrm{v}_{\beta i}^{n+1} + \eta\Delta t\left(B_{\alpha\beta j\gamma}\mathrm{v}_{\gamma j}^{n+1}\mathrm{v}_{\beta i}^{n+1} + K_{\alpha\beta}\mathrm{v}_{\beta i}^{n+1}\right) - A_{\alpha\beta}\mathrm{v}_{\beta i}^{n}$$
$$+ (1-\eta)\Delta t\left(B_{\alpha\beta j\gamma}\mathrm{v}_{\gamma j}^{n}\mathrm{v}_{\beta i}^{n} + K_{\alpha\beta}\mathrm{v}_{\beta i}^{n}\right) - \Delta t(F_{\alpha i} + G_{\alpha i}) = 0 \qquad (11.5.1)$$

with

$$B_{\alpha\beta j\gamma} = \int_\Omega \Phi_\alpha \Phi_{\beta,j}\Phi_\gamma d\Omega \qquad (11.5.2)$$

This form is based on the assumption that the squares and products of velocity components vary linearly within the time step as in (11.2.6),

$$\mathrm{v}_{\gamma j}^{n+1}\mathrm{v}_{\beta i}^{n+1} = (1-\xi)\mathrm{v}_{\gamma j}^{n}\mathrm{v}_{\beta i}^{n} + \xi\mathrm{v}_{\gamma j}^{n+1}\mathrm{v}_{\beta i}^{n+1} \qquad (11.5.3)$$

One of the most efficient approaches to solve nonlinear equations is the Newton-Raphson method developed from the Taylor series expansion of the residual of the type in (11.5.1).

$$E_{\alpha i}^{n+1,r+1} = E_{\alpha i}^{n+1,r} + \frac{\partial E_{\alpha i}^{n+1,r}}{\partial \mathrm{v}_{\beta j}^{n+1,r}}\Delta\mathrm{v}_{\beta j}^{n+1,r+1} + \cdots = 0 \qquad (11.5.4)$$

which implies that the residual at a given time station $n+1$ as incremented to the $r+1$ iteration cycle from the previous cycle $r$ should vanish if (11.5.1) is to be satisfied. Retaining only up to and including the first order term in (11.5.4), we obtain

$$J_{\alpha\beta ij}^{n+1,r}\Delta\mathrm{v}_{\beta j}^{n+1,r+1} = -E_{\alpha i}^{n+1,r} \qquad (11.5.5)$$

where

$$\Delta\mathrm{v}_{\beta j}^{n+1,r+1} = \mathrm{v}_{\beta j}^{n+1,r+1} - \mathrm{v}_{\beta j}^{n+1,r} \qquad (11.5.6)$$

and $J_{\alpha\beta ij}^{n+1,r}$ is the Jacobian,

$$J_{\alpha\beta ij}^{n+1,r} = \frac{\partial E_{\alpha i}^{n+1,r}}{\partial \mathrm{v}_{\beta j}^{n+1,r}}$$

or

$$\begin{aligned}
J_{\alpha\beta ij}^{n+1,r} &= A_{\alpha\eta}\frac{\partial \mathrm{v}_{\eta i}^{n+1,r}}{\partial \mathrm{v}_{\beta j}^{n+1,r}} + \eta\Delta t\left[B_{\alpha\eta k\gamma}\left(\frac{\partial \mathrm{v}_{\gamma k}^{n+1,r}}{\partial \mathrm{v}_{\beta j}^{n+1,r}}\mathrm{v}_{\eta i}^{n+1,r} + \mathrm{v}_{\gamma k}^{n+1,r}\frac{\partial \mathrm{v}_{\eta i}^{n+1,r}}{\partial \mathrm{v}_{\beta j}^{n+1,r}}\right) + K_{\alpha\eta}\frac{\partial \mathrm{v}_{\eta i}^{n+1,r}}{\partial \mathrm{v}_{\beta j}^{n+1,r}}\right] \\
&= A_{\alpha\eta}\delta_{\eta\beta}\delta_{ij} + \eta\Delta t\left[B_{\alpha\eta k\gamma}\left(\delta_{\gamma\beta}\delta_{kj}\mathrm{v}_{\eta i}^{n+1,r} + \mathrm{v}_{\gamma k}^{n+1,r}\delta_{\eta\beta}\delta_{ij}\right) + K_{\alpha\eta}\delta_{\eta\beta}\delta_{ij}\right] \\
&= A_{\alpha\beta}\delta_{ij} + \eta\Delta t\left[B_{\alpha\gamma j\beta}\mathrm{v}_{\gamma i}^{n+1,r} + B_{\alpha\beta k\gamma}\delta_{ij}\mathrm{v}_{\gamma k}^{n+1,r} + K_{\alpha\beta}\delta_{ij}\right] \qquad (11.5.7)
\end{aligned}$$

with

$$B_{\alpha\beta j\gamma} = \int_\Omega \Phi_\alpha\Phi_{\gamma,j}\Phi_\beta\, d\Omega, \quad B_{\alpha\beta k\gamma} = \int_\Omega \Phi_\alpha\Phi_{\beta,k}\Phi_\gamma\, d\Omega$$

The Newton-Raphson procedure described above may be simplified by revising the Jacobian matrix and the right-hand side residual as follows:

$$J_{\alpha\beta ij}^{n+1,r} = A_{\alpha\beta}\delta_{ij} + \frac{\Delta t}{2}(B_{\alpha\beta ij} + K_{\alpha\beta ij})$$

with

$$B_{\alpha\beta} = \int_\Omega \Phi_\alpha\Phi_{\beta,j}\bar{\mathrm{v}}_j d\Omega$$

and (11.5.1) being replaced by

$$\begin{aligned}
E_{\alpha i}^{n+1,r} &= A_{\alpha\beta}\mathrm{v}_{\beta i}^{n+1} + \frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})\mathrm{v}_{\beta i}^{n+1} - A_{\alpha\beta}\mathrm{v}_{\beta i}^{n} + \frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})\mathrm{v}_{\beta i}^{n} \\
&\quad - \Delta t(F_{\alpha i} + G_{\alpha i}) = A_{\alpha\beta}\Delta\mathrm{v}_{\beta i}^{n+1,r} + \frac{\Delta t}{2}(B_{\alpha\beta} + K_{\alpha\beta})\Delta\mathrm{v}_{\beta i}^{n+1,r} - \Delta t(F_{\alpha i} + G_{\alpha i})
\end{aligned}$$

The Newton-Raphson iterations are performed using (11.5.5) within each time step until convergence which requires that $\Delta\mathrm{v}_{\beta j}^{n+1,r+1} \cong 0$ in (11.5.5) before proceeding to the next time step in (11.5.7).

### 11.5.2 ELEMENT-BY-ELEMENT SOLUTION SCHEME FOR NONLINEAR TIME DEPENDENT FEM EQUATIONS

The linear and nonlinear simultaneous algebraic equations arising from the entire assembled global system of FEM formulations may be solved using direct or iterative methods. For a very large system, iterative methods are preferable to direct methods. Furthermore, it is often necessary to devise special techniques such as the frontal methods [Irons, 1970; Hood, 1976] or element-by-element (EBE) solution methods [Fox and Stanton, 1968; Irons, 1970]. In these methods, the standard assembly process of local stiffness matrices is not necessary. Instead, the product of a matrix by a vector can be obtained by assembling the product of local element matrices and the corresponding part of the vector, thus reducing the cost of computer time and storage. Initial contributions of the EBE concept to a large system of equations include Ortiz, Pinsky, and Taylor

[1983], Hughes, Frencz, and Hallquist [1987], Nour-Omid [1984], and Nour-Omid and Parlett [1985], among others.

Recall that we discussed the EBE algorithm for the linear equations in Section 10.3. For nonlinear stiffness matrices and time dependent problems, the procedure for EBE must be modified. These topics are elaborated below.

If the system of equations is nonlinear, then we may replace the preconditioner $D_{\alpha\beta}$ (see Section 10.3.2) by the Newton-Raphson Jacobian matrix. The global FEM nodal error can be written as

$$E_\alpha = K_{\alpha\beta} U_\beta - F_\alpha \tag{11.5.8}$$

Applying the Newton-Raphson scheme as shown in Section 11.5.1, we may rewrite (10.3.15) in the form

$$U_\alpha^{r+1} = U_\alpha^r - J_{\alpha\beta}^{-1} (\overline{F}_\beta - F_\beta)^r \tag{11.5.9}$$

where the EBE scheme is applied to the stiffness matrix as presented in Section 10.3.2 and the Jacobian matrix $J_{\alpha\beta}$ is given by

$$J_{\alpha\beta} = \frac{\partial E_\alpha}{\partial U_\beta} \tag{11.5.10}$$

which is considered as the preconditioning matrix. Here, as shown in (10.3.17), we may replace $J_{\alpha\beta}$ in (11.5.9) by the main diagonal of $J_{\alpha\beta}$ so that

$$U_\alpha^{r+1} = U_\alpha^r - J_{(\alpha\alpha)}^{-1} (\overline{F}_\alpha - F_\alpha)^r \tag{11.5.11}$$

The solution is obtained similarly as in (10.3.17) except that $J_{(\alpha\alpha)}$ and $\overline{F}_\alpha$ are nonlinear and must be updated at each iteration. Note that $\overline{F}_\alpha$ is converted from the EBE-based stiffness matrices.

In order to improve the solution accuracy, we may use the preconditioned conjugate gradient (PCG) method or the method known as the Lanczos/ORTHORES solver [Jea and Young, 1983]. In this method, begin with a starting value $U_\alpha^o$ and compute

$$U_\alpha^{r+1} = a^{r+1}\left(b^{r+1} D_\alpha^r + U_\alpha^r\right) + (1 - a^{r+1}) U_\alpha^r \tag{11.5.12}$$

with

$$\begin{aligned}
b^{r+1} &= \frac{\left(D_\alpha^r E_\alpha^r\right)}{\left(D_\delta^r K_{\delta\beta} D_\beta^r\right)} \qquad r = 0, 1, \ldots \\
a^{r+1} &= \begin{cases} 1 & r = 0 \\ \left[1 - \dfrac{b^{r+1}}{b^r} \dfrac{\left(D_\alpha^r E_\alpha^r\right)}{\left(D_\delta^{r-1}\, E_\delta^{r-1}\right)} \dfrac{1}{a^r}\right]^{-1} & r \geq 1 \end{cases} \\
E_\alpha^r &= \begin{cases} F_\alpha - K_{\alpha\beta} U_\beta^o & r = 0 \\ a^r\left(-b^r K_{\alpha\beta} D_\beta + E_\alpha^{r-1}\right) + (1 - a^r) E_\alpha^{r-2} & r = 1, 2, \ldots \end{cases} \\
D_\alpha^r &= Q_{\alpha\beta}^{-1} E_\beta^r \quad r \geq 0
\end{aligned} \tag{11.5.13}$$

where $Q_{\alpha\beta}$ is the Jacobi preconditioner,

$$Q_{\alpha\beta} = dia(K_{\alpha\beta}) \tag{11.5.14}$$

Thus, the inverse of $Q_{\alpha\beta}$ is the reciprocal of the diagonal of $K_{\alpha\beta}$ which can be partitioned for EBE computations.

The preconditioner may be constructed from the square root of the main diagonal of the stiffness matrix. To this end, we write (11.5.11) in the form

$$E_\alpha = \overline{F}_\alpha - \overline{K}_{\alpha\beta}\overline{U}_\beta \tag{11.5.15}$$

with

$$\overline{K}_{\alpha\beta} = W_{\alpha\gamma}^{-\frac{1}{2}} K_{\gamma\eta} W_{\eta\beta}^{-\frac{1}{2}} \qquad \overline{U}_\beta = W_{\beta\gamma}^{-\frac{1}{2}} U_\gamma$$

$$\overline{F}_\alpha = \bigcup_{e=1}^{E} \overline{F}_N^{(e)} \Delta_{N\alpha}^{(e)} \qquad \overline{F}_N^{(e)} = W_{NR}^{(e)\,\frac{1}{2}} F_R^{(e)}$$

$$W_{NR}^{(e)} = dia\left(K_{NR}^{(e)}\right)$$

For known initial solution vector $U_\beta^o$, compute

$$E_\alpha^o = \overline{F}_\alpha - \overline{K}_{\alpha\beta} U_\beta^o \tag{11.5.16}$$

Subsequent steps are the same as in (11.5.15). The final solution is obtained as

$$U_\alpha = W_{\alpha\beta}^{-\frac{1}{2}} \overline{U}_\beta = dia(K_{\alpha\beta})^{-\frac{1}{2}} \overline{U}_\beta \tag{11.5.17}$$

The Lanczos/ORTHOMIN solver [Jea and Young, 1983] may be used. In this scheme, the preconditioning processes (11.5.15) through (11.5.16) are used together with the following steps:

***Step 1***

$$E_\alpha^o = \overline{F}_\alpha - \overline{K}_{\alpha\beta} U_\beta^o$$

$$P_\alpha^o = E_\alpha^o$$

$$D_\alpha^o = \tilde{P}_\alpha^o = E_\alpha^o$$

$$b^o = \frac{\left(D_\alpha^o E_\alpha^o\right)}{\left(D_\delta^o \overline{K}_{\delta\beta} D_\beta^o\right)}$$

$$U_\alpha^1 = U_\alpha^o + b^o P_\alpha^o$$

***Step 2***

$$b^r = \frac{\left(D_\alpha^r E_\alpha^r\right)}{\left(D_\delta^r \overline{K}_{\delta\beta} D_\beta^r\right)}$$

$$P_\alpha^r = E_\alpha^r + \overline{b}^r P_\alpha^{r-1}$$

$$\tilde{P}_\alpha^r = D_\alpha^r + \overline{b}^r \tilde{P}_\alpha^{r-1}$$

$$\overline{b}^r = \frac{\left(D_\alpha^r E_\alpha^r\right)}{\left(D_\delta^{r-1} E_\delta^{r-1}\right)}$$

$$E_\alpha^{r+1} = E_\alpha^r - \overline{b}^r \overline{K}_{\alpha\beta} P_\beta^r$$

$$D_\alpha^{r+1} = D_\alpha^r - b^r \overline{K}_{\alpha\beta} \overline{P}_\beta^r$$

$$U_\alpha^{r+1} = U_\alpha^r + b^r P_\beta^r$$

Iterative solutions through the above steps lead to the final converged solution as

$$U_\alpha = W_{\alpha\beta}^{-\frac{1}{2}} \overline{U}_\beta = dia(K_{\alpha\beta})^{-\frac{1}{2}} \overline{U}_\beta \tag{11.5.18}$$

For time-dependent problems, we may consider the main diagonal of the mass matrix as the preconditioner. For example, the matrix equation

$$(M_{\alpha\beta} + \theta \Delta t K_{\alpha\beta}) U_\beta^{n+1} = [M_{\alpha\beta} + (1-\theta)\Delta t K_{\alpha\beta}] U_\beta^n + \Delta t F_\alpha \tag{11.5.19}$$

can be written as

$$\left(\delta_{\alpha\beta} + \theta \Delta t M_{\alpha\gamma}^{-\frac{1}{2}} K_{\gamma\eta} M_{\eta\beta}^{-\frac{1}{2}}\right) \overline{U}_\beta^{n+1} = \left[\delta_{\alpha\beta} - (1-\theta)\Delta t M_{\alpha\gamma}^{-\frac{1}{2}} K_{\gamma\eta} M_{\eta\beta}^{-\frac{1}{2}}\right] \overline{U}_\beta^n + \Delta t \overline{F}_\alpha \tag{11.5.20}$$

where $\overline{U}_\alpha^n = M_{\alpha\beta}^{\frac{1}{2}} U_\beta$ and $\overline{F}_\alpha = M_{\alpha\beta}^{-\frac{1}{2}} F_\beta$. Note that the eigenvalues of (11.5.22) are the same as those of (11.5.20) such that

$$|\delta_{\alpha\beta} + \theta \Delta t\, M_{\alpha\gamma} K_{\gamma\beta}| = \left| M_{\alpha\gamma}^{-\frac{1}{2}} \left(\delta_{\gamma\eta} + \theta \Delta t\, M_{\gamma\zeta}^{-\frac{1}{2}} K_{\zeta\delta} M_{\delta\eta}\right) M_{\eta\beta}^{-\frac{1}{2}} \right| \tag{11.5.21}$$

Rewriting (11.5.15) in the form

$$E_\alpha = A_{\alpha\beta} \overline{U}_\beta^{n+1} - B_{\alpha\beta} \overline{U}_\beta^n - \Delta t \overline{F}_\alpha \tag{11.5.22}$$

it is now possible to apply steps 1 and 2 of the steady-state case with initial and boundary conditions applied to (11.5.22).

### 11.5.3 GENERALIZED MINIMAL RESIDUAL ALGORITHM

The conjugate gradient method discussed in Section 10.3.1 is accurate and efficient for linear symmetric matrix equations. However, for problems in CFD where nonsymmetric nonlinear, indefinite matrices are involved, the Generalized Minimal Residual (GMRES) algorithm has been proved to be efficient [Saad and Schultz, 1986; Saad, 1996]. This method is based on the property of minimizing the norm of the residual vector over a Krylov space. The Krylov space is a general concept based on the simple observation that in any sequence of iterates there will be a smallest set of consecutive iterates which are linearly dependent, and that the coefficients of a vanishing combination are the coefficients of a divisor to the characteristic polynomial. See Householder [1964] for a detailed discussion of the Krylov space.

For the purpose of our discussion, let us consider the global form of the finite element equations in the form,

$$K_{\alpha\beta} U_\beta = F_\alpha \tag{11.5.23}$$

in which preconditioning through the EBE scheme is to be implemented as in Section 11.5.2.

One of the most effective iteration methods for solving large sparse asymmetric linear and nonlinear systems of equations is a combination of the CGM with preconditions in minimizing the norm of residual vector over a Krylov space

$$K^{(r)} = span\left[U_0, KU_0, K^2 U_0 \ldots, K^{(r-1)} U_0\right] \tag{11.5.24}$$

This algorithm is a generalization of the MINRES [Paige and Saunders, 1975] for solving nonsymmetric linear systems and Arnoldi process [Arnoldi, 1951] which is an analogue of the Lanczos algorithm for nonsymmetric matrices [Lanczos, 1950]. In the GMRES scheme, we determine $U_\beta^{(o)} + \overline{U}_\beta$ where $U_\beta^{(o)}$ is the initial guess and $\overline{U}_\beta$ is a member of the Krylov space $K$ of dimension $r$ such that the $L_2$ norm error

$$\| E_\alpha \| = \left\| F_\alpha - K_{\alpha\beta}\left(U_\beta^{(o)} + \overline{U}_\beta\right) \right\| \tag{11.5.25}$$

is minimized. Here, we use a smaller value for $r$ and restarting the algorithm after every $r$ step; thereby, the amount of storage required can be minimized.

The step-by-step GMRES scheme is as follows:

First, let us define:

$E_\alpha^{(r)}$ = total error vector

$\overline{E}_\alpha^{(i)}$ = error coefficient vector

$\| E_\alpha^{(j)} \|$ = normed error

$\tilde{E}_\alpha^{(j)}$ = adjusted error

$\| \tilde{E}_\alpha^{(j)} \|$ = normed adjusted error

$a^{(i,j)}$ = normed error coefficient

$y^{(j)}$ = minimizer error vector

(1) Choose $U_\beta^{(o)}$ and compute

$$E_\alpha^{(o)} = F_\alpha - K_{\alpha\beta} U_\beta^{(0)} = F_\alpha - \overline{F}_\alpha^{(0)}, \quad \overline{F}_\alpha = \bigcup_{e=1}^{E} \overline{F}_N^{(0)(e)} \Delta_{N\alpha}^{(e)},$$

$$\overline{F}_N^{(0)(e)} = K_{NM}^{(e)} U_M^{(0)(e)}$$

$$\overline{E}_\alpha^{(1)} = E_\alpha^{(o)} / \| E_\alpha^{(o)} \| \quad \text{(Gram-Schmidt orthogonalization)}$$

(2) Iterate for $i = 1, 2, \ldots r$

$$a^{(i+1,j)} = \tilde{E}_\alpha^{(i+1)} \overline{E}_\alpha^{(j)} = K_{\alpha\beta} \overline{E}_\alpha^{(i)} \overline{E}_\beta^{(j)}, \qquad j = 1, 2, \ldots, i$$

$$\tilde{E}_\alpha^{(i)} = K_{\alpha\beta} \overline{E}_\beta^{(i)} - \sum_{j=1}^{i} a^{(i+1,j)} \overline{E}_\alpha^{(j)}$$

$$\overline{E}_\alpha^{(i+1)} = \tilde{E}_\alpha^{(i)} / \| \tilde{E}_\alpha^{(i)} \|$$

(3) Approximate solution:

Let us consider a matrix consisting of the columns of residuals in the form

$$B_{\beta\xi}^{(r)} = \begin{bmatrix} \overline{E}_1^{(1)} & \overline{E}_1^{(2)} & \cdots & \overline{E}_1^{(r)} \\ \overline{E}_2^{(1)} & \overline{E}_2^{(2)} & \cdots & \overline{E}_2^{(r)} \\ \vdots & \vdots & & \vdots \\ \overline{E}_n^{(1)} & \overline{E}_n^{(2)} & \cdots & \overline{E}_n^{(r)} \end{bmatrix} \tag{11.5.26}$$

Then, it can be shown that

$$K_{\alpha\beta} B_{\beta\xi}^{(r,r)} = B_{\alpha\eta}^{(r,r+1)} H_{\eta\xi}^{(r+1,r)} \tag{11.5.27}$$

where $H_{\eta\xi}^{(r+1,r)}$ is the upper Hessenberg matrix of the form

$$H_{\eta\xi}^{(r+1,1)} = \begin{bmatrix} a^{(1,1)} & a^{(2,1)} & \cdot & \cdot & a^{(r,1)} \\ \|\tilde{E}_\alpha^{(1)}\| & a^{(2,2)} & \cdot & \cdot & a^{(r,2)} \\ 0 & \|\tilde{E}_\alpha^{(2)}\| & \cdot & \cdot & a^{(r,3)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdot & \cdot & \|\tilde{E}^{(r)}\| \end{bmatrix} \tag{11.5.28}$$

Here, the idea is to find a vector $y_\xi$ which will minimize the residual error as follows:

$$\begin{aligned} \min\left\| F_\alpha - K_{\alpha\beta}\left(U_\beta^{(0)} + E_\beta\right)\right\| &= \min\left\| E_\alpha^{(0)} - K_{\alpha\beta} B_{\beta\xi}^{(r,r)} y_\xi \right\| \\ &= \left\| B_{\alpha\eta}^{(r+1)}\left(e_\eta - H_{\eta\xi}^{(r+1)} y_\xi\right)\right\| \\ &= \left\| \overline{e}_\alpha - \overline{H}_{\alpha\xi}^{(r+1)} y_\xi \right\| \cong 0 \end{aligned} \tag{11.5.29}$$

with

$$\overline{e}_\alpha = \left\{\left\| E_\alpha^{(1)} \right\|, 0, \ldots 0\right\}^T \tag{11.5.30}$$

$$y_\xi = \overline{H}_{\alpha\xi}^{-1} \overline{e}_\alpha \tag{11.5.31}$$

The minimization process above does not provide the approximate solution explicitly at each step. Thus, it is difficult to determine when to stop. This may be simplified using the so-called Q-R algorithm as suggested by Saad and Schultz [1986]. In this approach, we utilize the Givens-Householder rotation matrix, $R_{\alpha\eta}$, such that

$$\overline{H}_{\alpha\xi} = R_{\alpha\eta} H_{\eta\xi} \tag{11.5.32}$$

where

$$R_{\alpha\xi} = R_r R_{r-1} \ldots . R_1$$

$$R_{\alpha\eta} = \begin{bmatrix} 1 & & & & & & & & \\ & \cdot & & & & & & & \\ & & \cdot & & & & & & \\ & & & 1 & & & & & \\ & & & & c_r & s_r & & & \\ & & & & -s_r & c_r & & & \\ & & & & & & 1 & & \\ & & & & & & & \cdot & \\ & & & & & & & & \cdot \\ & & & & & & & & & 1 \end{bmatrix} \tag{11.5.33}$$

with $c_r^2 + s_r^2 = 1$ and the size of the matrix being $(m+1) \times (m \times 1)$ for $m$ steps of the GMRES iterations. The scalars $c_r$ and $s_r$ of the $r$th rotation $R_r$, being orthogonal, are defined as

$$c_r = \frac{H_{rr}}{\sqrt{(H_{rr})^2 + H^2_{r+1,r}}}, \qquad s_r = \frac{H_{r+1,r}}{\sqrt{((H_{rr})^2 + H^2_{r+1,r})}} \tag{11.5.34}$$

For example, let us assume $r$ steps of the GMRES iterations so that (11.5.28) is written as

$$\left\| e_\alpha - H^{r+1}_{\alpha\xi} y_\xi \right\| = \left\| R_{\alpha\xi}\left(\overline{e}_\alpha - \overline{H}^{r+1}_{\alpha\xi} y_\xi\right)\right\| = \left\| \overline{e}_\alpha - \overline{H}^{r+1}_{\alpha\xi} y_\xi \right\| \tag{11.5.35}$$

leading to the minimization,

$$\min\left\| \overline{e}_\alpha - \overline{H}^{r+1}_{\alpha\xi} y_\xi \right\| = \left|\overline{e}^{r+1}_\alpha\right| \tag{11.5.36}$$

and $y_\xi$ satisfies

$$\begin{bmatrix} \overline{H}_{1,1} & \cdot & \cdot & \overline{H}_{1,r-1} & \overline{H}_{1,r} \\ 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \overline{H}_{r-1,r-1} & \overline{H}_{r-1,r} \\ 0 & 0 & 0 & 0 & \overline{H}_{r,r} \end{bmatrix} \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_{r-1} \\ y_r \end{bmatrix} = \begin{bmatrix} \overline{e}_1 \\ \cdot \\ \cdot \\ \overline{e}_{r-1} \\ \overline{e}_r \end{bmatrix} \tag{11.5.37}$$

in which the back substitution provides the inverse required in (11.5.31).

To obtain the Hessenberg matrix in (11.5.37), we proceed as follows. If $m = 5$, then we have

$$\overline{H}_5 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \\ & h_{32} & h_{33} & h_{34} & h_{35} \\ & & h_{43} & h_{44} & h_{45} \\ & & & h_{54} & h_{55} \\ & & & & h_{65} \end{bmatrix} \tag{11.5.38}$$

$$h^{(1)} = \begin{bmatrix} h_{11} \\ h_{21} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a^{(1,1)} \\ \|\tilde{E}_\alpha\| \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{11.5.39}$$

$$r_1 = \left(h^2_{11} + h^2_{21}\right)^{1/2}, \qquad c_1 = h_{11}/r_1, \qquad s_1 = h_{21}/r_1$$

The first column of $\overline{H}_5$ becomes

$$\overline{h}^{(1)} = R_1 h^{(1)} = \begin{bmatrix} r_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \overline{h}^{(m)} = R_m R_{m-1} \cdots R_2 \overline{h}^{(1)}$$

Similarly,

$$\overline{e}^{(1)}_\alpha = R_1 \overline{e}^{(0)}_\alpha, \qquad \overline{e}^{(m)} = R_m R_{m-1} \cdots R_2 \overline{e}^{(0)}_\alpha$$

This process leads to the tridiagonalized form,

$$\overline{H}^{(5)} = \begin{bmatrix} h_{11}^{(5)} & h_{12}^{(5)} & h_{13}^{(5)} & h_{14}^{(5)} & h_{15}^{(5)} \\ & h_{22}^{(5)} & h_{23}^{(5)} & h_{24}^{(5)} & h_{25}^{(5)} \\ & & h_{33}^{(5)} & h_{34}^{(5)} & h_{35}^{(5)} \\ & & & h_{44}^{(5)} & h_{45}^{(5)} \\ & & & & h_{55}^{(5)} \\ & & & & 0 \end{bmatrix} \qquad (11.5.40)$$

which is then inserted in (11.5.37) to determine $y_\xi$, required in (11.5.31).

(4) Calculate the error residuals $\overline{U}_\alpha^{(r)}$,

$$E_\alpha^{(r)} = E_{\alpha r} - y_r$$

(5) The converged solution is obtained as

$$U_\alpha = U_\alpha^o + E_\alpha^{(r)}$$

### Example 11.5.1

Solve the following equations with an unsymmetric stiffness matrix using the GMRES algorithm. Compare with the exact solution: $U_1 = 1$, $U_2 = 2$, $U_3 = 3$.

$$\begin{bmatrix} 3 & 2 & -2 \\ -4 & -1 & 1 \\ 5 & -2 & -1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ -2 \end{bmatrix}$$

***Solution:***

Note that the EBE process is omitted here for simplicity. (The global matrix equation is used instead of the EBE column vector.) The EBE process must be used for a large system of equations. See Section 11.5.4 for EBE implementations.

1. Choose $U_\beta^{(0)} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$ (This is a deliberate choice to be much different from the exact solution.)
2. Compute

$$E_\alpha^{(0)} = F_\alpha - K_{\alpha\beta} U_\beta^{(0)} = \begin{bmatrix} -10 \\ 10 \\ -12 \end{bmatrix} \qquad \| E_\alpha^{(0)} \| = \sqrt{344} = 18.5472$$

$$\overline{E}_\alpha^{(1)} = \frac{E_\alpha^{(0)}}{\| E_\alpha^{(0)} \|} = \begin{bmatrix} -0.5392 \\ 0.5392 \\ -0.6470 \end{bmatrix}$$

3. Iterate for $i = 1, 2, \ldots, r$

(a) $i = 1$:

$$\tilde{E}_\alpha^{(1)} = K_{\alpha\beta} \overline{E}_\beta^{(1)} = \begin{bmatrix} 0.7543 \\ 0.9705 \\ -3.1272 \end{bmatrix} \qquad \text{For } j = 1, \ldots, i:$$

$$a^{(1,1)} = \tilde{E}_\alpha^{(1)} \overline{E}_\alpha^{(1)} = 2.1395$$

$$\tilde{E}_\alpha^{(1)} = \tilde{E}_\alpha^{(1)} - a^{(1,1)} \overline{E}_\alpha^{(1)} = \begin{bmatrix} 1.9084 \\ -0.1831 \\ -1.7429 \end{bmatrix}$$

$$\| \tilde{E}_\alpha^{(1)} \| = 2.5910$$

$$\overline{E}_\alpha^{(2)} = \frac{\tilde{E}_\alpha^{(1)}}{\| \tilde{E}_\alpha^{(1)} \|} = \begin{bmatrix} 0.7366 \\ -0.0707 \\ -0.6727 \end{bmatrix}$$

(b) $i = 2$:

$$\tilde{E}_\alpha^{(2)} = K_{\alpha\beta} \overline{E}_\beta^{(2)} = \begin{bmatrix} 3.4137 \\ -3.5482 \\ 4.4968 \end{bmatrix}$$

For $j = 1, 2$ Do

$j = 1$:

$$a^{(2,1)} = \tilde{E}_\alpha^{(2)} \overline{E}_\alpha^{(1)} = -6.6630$$

$$\tilde{E}_\alpha^{(2)} = \tilde{E}_\alpha^{(2)} - a^{(2,1)} \overline{E}_\alpha^{(1)} = \begin{bmatrix} -0.1788 \\ 0.0442 \\ 0.1858 \end{bmatrix}$$

$j = 2$:

$$a^{(2,2)} = \tilde{E}_\alpha^{(2)} \overline{E}_\alpha^{(2)} = -0.2598$$

$$\tilde{E}_\alpha^{(2)} = \tilde{E}_\alpha^{(2)} - a^{(2,2)} \overline{E}_\alpha^{(2)} = \begin{bmatrix} 0.0126 \\ 0.0259 \\ 0.0111 \end{bmatrix}$$

$$\| \tilde{E}_\alpha^{(2)} \| = 0.0308$$

$$\overline{E}_\alpha^{(3)} = \frac{\tilde{E}_\alpha^{(2)}}{\| \tilde{E}_\alpha^{(2)} \|} = \begin{bmatrix} 0.4084 \\ 0.8392 \\ 0.3590 \end{bmatrix}$$

(c) $i = 3$:

$$\tilde{E}_\alpha^{(3)} = K_{\alpha\beta} \overline{E}_\beta^{(3)} = \begin{bmatrix} 2.1856 \\ -2.1138 \\ 0.0045 \end{bmatrix}$$

For $j = 1, \ldots 3$ Do

$j = 1$:

$$a^{(3,1)} = \tilde{E}_\alpha^{(3)} \overline{E}_\alpha^{(1)} = -2.3209$$

$$\tilde{E}_\alpha^{(3)} = \tilde{E}_\alpha^{(3)} - a^{(3,1)} \overline{E}_\alpha^{(1)} = \begin{bmatrix} 0.9342 \\ -0.8624 \\ -1.4972 \end{bmatrix}$$

$j = 2$:

$$a^{(3,2)} = \tilde{E}_\alpha^{(3)} \overline{E}_\alpha^{(2)} = -1.7561$$

$$\tilde{E}_\alpha^{(3)} = \tilde{E}_\alpha^{(3)} - a^{(3,2)} \overline{E}_\alpha^{(2)} = \begin{bmatrix} -0.3593 \\ -0.7383 \\ -0.3159 \end{bmatrix}$$

$j = 3$:

$$a^{(3,3)} = \tilde{E}_\alpha^{(3)} \overline{E}_\alpha^{(3)} = -0.8798$$

$$\tilde{E}_\alpha^{(3)} = \tilde{E}_\alpha^{(3)} - a^{(3,3)} \overline{E}_\alpha^{(3)} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\| \tilde{E}_\alpha^{(3)} \| = 0$$

$$\overline{E}_\alpha^{(4)} = \frac{\tilde{E}_\alpha^{(3)}}{\| \tilde{E}_\alpha^{(3)} \|} = 0.$$

4. Construct Hessenberg matrix

$$\begin{bmatrix} a^{(1,1)} & a^{(2,1)} & a^{(3,1)} \\ \| \tilde{E}_\alpha^{(1)} \| & a^{(2,2)} & a^{3,2} \\ 0 & \| \tilde{E}_\alpha^{(2)} \| & a^{3,3} \\ 0 & 0 & \| \tilde{E}_\alpha^{(3)} \| \end{bmatrix} \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \end{bmatrix} = \begin{bmatrix} \| E_\alpha^{(0)} \| \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2.1395 & -6.6630 & -2.3210 \\ 2.5910 & -0.2598 & 1.7561 \\ 0 & 0.0308 & -0.8798 \end{bmatrix} \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \end{bmatrix} = \begin{bmatrix} 18.5472 \\ 0 \\ 0 \end{bmatrix}$$

5. Apply Givens rotation to reduce matrix for tridiagonalization.
   (a) First rotation:

$$c_j = \frac{h_{jj}}{r_j}, \qquad s_j = \frac{h_{j+1,j}}{r_j}, \qquad r_j = \sqrt{h_{jj}^2 + h_{j+1,j}^2}$$

$$c_1 = \frac{a^{(1,1)}}{\sqrt{\left(a^{(1,1)}\right)^2 + \left(\| \tilde{E}_\alpha^{(1)} \|\right)^2}} = 0.6367 \; s_1 = \frac{\tilde{E}_\alpha^{(1)}}{\sqrt{\left(a^{(1,1)}\right)^2 + \left(\| \tilde{E}_\alpha^{(1)} \|\right)^2}} = 0.7711$$

$$\begin{bmatrix} c & s & \\ -s & c & \\ & & 1 \end{bmatrix} \begin{bmatrix} a^{(1,1)} & a^{(2,1)} & a^{(3,1)} \\ \| \tilde{E}_\alpha^{(1)} \| & a^{(2,2)} & a^{(3,2)} \\ & \| \tilde{E}_\alpha^{(2)} \| & a^{(3,3)} \\ & & \| \tilde{E}_\alpha^{(3)} \| \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \| E_\alpha^{(0)} \| \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 3.3602 & -4.4429 & -0.1237 \\ 0 & 4.9723 & 2.9079 \\ 0 & 0.0308 & -0.8798 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 11.8097 \\ -14.3014 \\ 0 \end{bmatrix}$$

   (b) Second rotation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix} \begin{bmatrix} 3.3602 & -4.4429 & -0.1237 \\ 0 & 4.9723 & 2.9079 \\ 0 & 0.0308 & -0.8798 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix} \begin{bmatrix} 11.8097 \\ -14.3014 \\ 0 \end{bmatrix}$$

$c_2 = 0.9999 \qquad s_2 = 0.0062$

$$\begin{bmatrix} 3.3602 & -4.4429 & -0.1237 \\ 0 & 4.9724 & 2.9024 \\ 0 & 0 & -0.8798 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 11.8097 \\ -14.3012 \\ 0.0886 \end{bmatrix}$$

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \end{bmatrix} = \begin{bmatrix} -0.2157 \\ -2.8185 \\ -0.0988 \end{bmatrix}$$

6. Compute residual

$$\begin{bmatrix} \overline{E}_1^{(1)} & \overline{E}_1^{(2)} & \overline{E}_1^{(3)} \\ \overline{E}_2^{(1)} & \overline{E}_2^{(2)} & \overline{E}_2^{(3)} \\ \overline{E}_3^{(1)} & \overline{E}_3^{(2)} & \overline{E}_3^{(3)} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} E_1^{(r)} \\ E_2^{(r)} \\ E_3^{(r)} \end{bmatrix}$$

$$\begin{bmatrix} E_1^{(r)} \\ E_2^{(r)} \\ E_3^{(r)} \end{bmatrix} = \begin{bmatrix} -0.5392 & 0.7366 & 0.4084 \\ 0.5392 & -0.0707 & 0.8392 \\ -0.6470 & -0.6727 & 0.3590 \end{bmatrix} \begin{bmatrix} -0.2157 \\ -2.8185 \\ -0.0987 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 2 \end{bmatrix}$$

7. Update $U_\beta$

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} U_1^{(0)} \\ U_2^{(0)} \\ U_3^{(0)} \end{bmatrix} + \begin{bmatrix} E_1^{(r)} \\ E_2^{(r)} \\ E_3^{(r)} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Note that the exact solution has been obtained.

### 11.5.4 COMBINED GPG-EBE-GMRES PROCESS

We consider the solution by generalized Petrov-Galerkin (GPG) method using EBE-GMRES solver. The global GPG equation (11.4.5) may be written in a local form.

$$\left[A_{NM}^{(e)} + \eta\Delta t\left(B_{NM}^{(e)} + C_{NM}^{(e)} + K_{NM}^{(e)}\right)\right] v_{Mi}^{(e)n+1}$$
$$= \left[A_{NM}^{(e)} - (1-\eta)\Delta t\left(B_{NM}^{(e)} + C_{NM}^{(e)} + K_{NM}^{(e)}\right)\right] v_{Mi}^{(e)n} + \Delta t\left(F_{Mi}^{(e)n} + G_{Mi}^{(e)n}\right) \qquad (11.5.41)$$

or

$$R_{NM}^{(e)}\, v_{Mi}^{(e)n+1} = Q_{Ni}^{(e)n} \qquad (11.5.42)$$

For illustration, let us consider the global and local configurations as given in Figure 11.5.4.1.

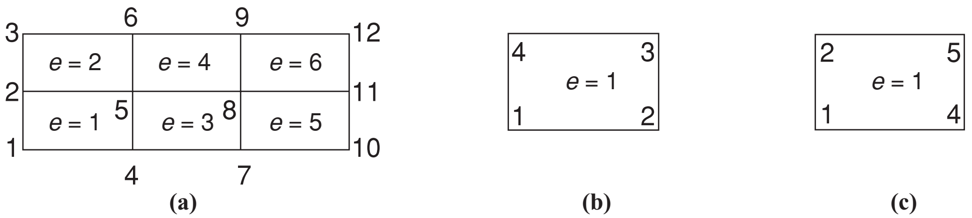


**Figure 11.5.4.1** Global and local configurations. (**a**) Global system. (**b**) Local. (**c**) Global.

Using the four-node isoparametric element on the left-hand side of (11.5.42) for $e = 1$, we have

$$D_{Ni}^{(1)(n+1)} = R_{NM}^{(1)}\, v_{Mi}^{(1)(n+1)} \tag{11.5.43}$$

or

$$\begin{bmatrix} D_{11}^{(1)} \\ D_{12}^{(1)} \\ D_{41}^{(1)} \\ D_{42}^{(1)} \\ D_{51}^{(1)} \\ D_{52}^{(1)} \\ D_{21}^{(1)} \\ D_{22}^{(1)} \end{bmatrix}^{(n+1)} = \begin{bmatrix} R_{11}^{(1)} & 0 & R_{14}^{(1)} & 0 & R_{15}^{(1)} & 0 & R_{12}^{(1)} & 0 \\ 0 & R_{11}^{(1)} & 0 & R_{11}^{(1)} & 0 & R_{11}^{(1)} & 0 & R_{11}^{(1)} \\ R_{41}^{(1)} & 0 & R_{44}^{(1)} & 0 & R_{45}^{(1)} & 0 & R_{42}^{(1)} & 0 \\ 0 & R_{41}^{(1)} & 0 & R_{44}^{(1)} & 0 & R_{45}^{(1)} & 0 & R_{42}^{(1)} \\ R_{51}^{(1)} & 0 & R_{54}^{(1)} & 0 & R_{55}^{(1)} & 0 & R_{52}^{(1)} & 0 \\ 0 & R_{51}^{(1)} & 0 & R_{54}^{(1)} & 0 & R_{55}^{(1)} & 0 & R_{52}^{(1)} \\ R_{21}^{(1)} & 0 & R_{24}^{(1)} & 0 & R_{25}^{(1)} & 0 & R_{22}^{(1)} & 0 \\ 0 & R_{21}^{(1)} & 0 & R_{24}^{(1)} & 0 & R_{25}^{(1)} & 0 & R_{22}^{(1)} \end{bmatrix} \begin{bmatrix} v_{11}^{(1)} \\ v_{12}^{(1)} \\ v_{41}^{(1)} \\ v_{42}^{(1)} \\ v_{51}^{(1)} \\ v_{52}^{(1)} \\ v_{21}^{(1)} \\ v_{22}^{(1)} \end{bmatrix}^{(n+1)}$$

with the local element node numbers being replaced by the global node numbers for global assembly.

The assembled column vector $D_{\alpha i}$ takes the form

$$D_{\alpha i} = \bigcup_{e=1}^{E} D_{Ni}^{(e)} \Delta_{N\alpha}^{(e)} = \bigcup_{e=1}^{E} R_{NM}^{(e)}\, v_{Mi}^{(e)}\, \Delta_{N\alpha}^{(e)} \tag{11.5.44}$$

This operation is identical to the summing process, as shown in Table 11.5.1. with

$$D_{11}^{(1)} = R_{11}^{(1)}\, v_{11}^{(1)} + R_{14}^{(1)}\, v_{41}^{(1)} + R_{15}^{(1)}\, v_{51}^{(1)} + R_{12}^{(1)}\, v_{21}^{(1)}$$
$$D_{12}^{(1)} = R_{11}^{(1)}\, v_{12}^{(1)} + R_{14}^{(1)}\, v_{42}^{(1)} + R_{15}^{(1)}\, v_{52}^{(1)} + R_{12}^{(1)}\, v_{22}^{(1)}$$

etc.

For illustration let us consider the geometry given in Figure 11.5.4.1c. It represents 189 × 2 = 378 equations given by the column vector $D_{\alpha i}$, which is assembled from 8 × 8 local stiffness matrices multiplied by the 8 × 1 local variable unknown column vectors.

**Table 11.5.1** Global Summing Procedure

| Node | $e = 1$ | $e = 2$ | $e = 3$ | $e = 4$ | $e = 5$ | $e = 6$ | $D_{\alpha i}$ (sum) |
|---|---|---|---|---|---|---|---|
| 1 | $D_{11}^{(1)}$<br>$D_{12}^{(1)}$ | | | | | | $D_{11} = D_{11}^{(1)}$<br>$D_{12} = D_{12}^{(1)}$ |
| 2 | $D_{21}^{(1)}$<br>$D_{22}^{(1)}$ | $D_{21}^{(2)}$<br>$D_{22}^{(2)}$ | | | | | $D_{21} = D_{21}^{(1)} + D_{21}^{(2)}$<br>$D_{22} = D_{22}^{(1)} + D_{21}^{(2)}$ |
| 3 | | $D_{31}^{(2)}$<br>$D_{32}^{(2)}$ | | | | | $D_{31} = D_{31}^{(2)}$<br>$D_{32} = D_{32}^{(2)}$ |
| 4 | $D_{41}^{(1)}$<br>$D_{42}^{(1)}$ | | $D_{41}^{(3)}$<br>$D_{42}^{(3)}$ | | | | $D_{41} = D_{41}^{(1)} + D_{41}^{(2)}$<br>$D_{42} = D_{42}^{(1)} + D_{42}^{(2)}$ |
| 5 | $D_{51}^{(1)}$<br>$D_{52}^{(1)}$ | $D_{51}^{(2)}$<br>$D_{52}^{(2)}$ | $D_{51}^{(3)}$<br>$D_{52}^{(3)}$ | $D_{51}^{(4)}$<br>$D_{52}^{(4)}$ | | | $D_{51} = D_{51}^{(1)} + D_{51}^{(2)} + D_{51}^{(3)} + D_{51}^{(4)}$<br>$D_{52} = D_{52}^{(1)} + D_{52}^{(2)} + D_{52}^{(3)} + D_{52}^{(4)}$ |
| 6 | | $D_{61}^{(2)}$<br>$D_{62}^{(2)}$ | | $D_{61}^{(4)}$<br>$D_{62}^{(4)}$ | | | $D_{61} = D_{61}^{(2)} + D_{61}^{(4)}$<br>$D_{62} = D_{62}^{(2)} + D_{62}^{(4)}$ |
| 7 | | | $D_{71}^{(3)}$<br>$D_{72}^{(3)}$ | | $D_{71}^{(5)}$<br>$D_{72}^{(5)}$ | | $D_{71} = D_{71}^{(3)} + D_{71}^{(5)}$<br>$D_{72} = D_{72}^{(3)} + D_{72}^{(5)}$ |
| 8 | | | $D_{81}^{(3)}$<br>$D_{82}^{(3)}$ | $D_{81}^{(4)}$<br>$D_{82}^{(4)}$ | $D_{81}^{(5)}$<br>$D_{82}^{(5)}$ | $D_{81}^{(6)}$<br>$D_{82}^{(6)}$ | $D_{81} = D_{81}^{(3)} + D_{81}^{(4)} + D_{81}^{(5)} + D_{81}^{(6)}$<br>$D_{82} = D_{82}^{(3)} + D_{82}^{(4)} + D_{82}^{(5)} + D_{82}^{(6)}$ |
| 9 | | | | $D_{91}^{(4)}$<br>$D_{92}^{(4)}$ | | $D_{91}^{(6)}$<br>$D_{92}^{(6)}$ | $D_{91} = D_{91}^{(4)} + D_{91}^{(6)}$<br>$D_{92} = D_{92}^{(4)} + D_{92}^{(6)}$ |
| 10 | | | | | $D_{10,1}^{(5)}$<br>$D_{10,2}^{(5)}$ | | $D_{10,1} = D_{10,1}^{(5)}$<br>$D_{10,2} = D_{10,2}^{(5)}$ |
| 11 | | | | | $D_{11,1}^{(5)}$<br>$D_{11,2}^{(5)}$ | $D_{11,1}^{(6)}$<br>$D_{11,2}^{(6)}$ | $D_{11,1} = D_{11,1}^{(5)} + D_{11,1}^{(6)}$<br>$D_{11,2} = D_{11,2}^{(5)} + D_{11,2}^{(6)}$ |
| 12 | | | | | | $D_{12,1}^{(6)}$<br>$D_{12,2}^{(6)}$ | $D_{12,1} = D_{12,1}^{(6)}$<br>$D_{12,2} = D_{12,2}^{(6)}$ |

We follow the procedure similar to the one given in Example 11.5.1 except that we use the EBE process here. Thus, instead of global matrix $K_{\alpha\beta}$ (378 × 378) we now have a column vector $D_{\alpha i}(378 \times 1)$.

1. Specify initial and boundary conditions on all boundary nodes and assume values for all interior nodes ($v_{Mi}^{(e)} = 0$, for example)
2. Compute the error coefficient vector $\overline{E}_{\alpha i}^{(1)}$

$$E_{\alpha i}^{(0)} = Q_{\alpha i} - D_{\alpha i},$$

with $Q_{\alpha i} = \cup_{e=1}^{E} Q_{Ni}^{(e)} \Delta_{N\alpha}^{(e)}$and $D_{\alpha i}$ as determined from (11.5.44).

$$\overline{E}_{\alpha i}^{(1)} = \frac{E_{\alpha i}^{(0)}}{\| E_{\alpha i}^{(0)} \|} \qquad \text{(Gram-Schmidt process)}$$

3. Iterate for $i = 1, 2, 3, \ldots r$, say $r = 4$

For this example calculate the adjusted error vector $\tilde{E}^{(1)}_{\alpha i}$, the normed error coefficient $a^{(1,1)}$, and a new error coefficient vector $\overline{E}^{(2)}_{\alpha i}$.

(a) $i = 1$:

$$\tilde{E}^{(1)}_{\alpha i} = \bigcup_{e=1}^{E} E^{(e)(1)}_{Ni}\Delta^{(e)}_{N\alpha} = \bigcup_{e=1}^{E} R^{(e)}_{NM}\overline{E}^{(e)}_{Mi}\Delta^{(e)}_{N\alpha}$$

$j = 1$:

$$a^{(1,1)} = \tilde{E}^{(1)}_{\alpha i}\overline{E}^{(1)}_{\alpha i}$$

$$\tilde{E}^{(1)}_{\alpha i} = \tilde{E}^{(1)}_{\alpha i} - a^{(1,1)}\overline{E}^{(1)}_{\alpha i}$$

$$\overline{E}^{(2)}_{\alpha i} = \frac{\tilde{E}^{(1)}_{\alpha i}}{\| \tilde{E}^{(1)}_{\alpha i} \|}$$

(b) $i = 2$: (Calculate, similarly, new adjusted error vector, normed error coefficients, and error coefficient vector.)

$$\tilde{E}^{(2)}_{\alpha i} = \bigcup_{e=1}^{E} E^{(e)(2)}_{Ni}\Delta^{(e)}_{N\alpha} = \bigcup_{e=1}^{E} R^{(e)}_{NM}\overline{E}^{(e)(2)}_{Mi}\Delta^{(e)}_{N\alpha}$$

$j = 1$:

$$a^{(2,1)} = \tilde{E}^{(2)}_{\alpha i}\overline{E}^{(1)}_{\alpha i}$$

$$\tilde{E}^{(2)}_{\alpha i} = \tilde{E}^{(2)}_{\alpha i} - a^{(2,1)}\overline{E}^{(1)}_{\alpha i}$$

$j = 2$:

$$a^{(2,2)} = \tilde{E}^{(2)}_{\alpha i}\overline{E}^{(2)}_{\alpha i}$$

$$\tilde{E}^{(2)}_{\alpha i} = \tilde{E}^{(2)}_{\alpha i} - a^{(2,2)}\overline{E}^{(2)}_{\alpha i}$$

$$\overline{E}^{(3)}_{\alpha i} = \frac{\tilde{E}^{(2)}_{\alpha i}}{\| \tilde{E}^{(2)}_{\alpha i} \|}$$

(c) $i = 3$, similarly,

$$\tilde{E}^{(3)}_{\alpha i} = \bigcup_{e=1}^{E} E^{(e)(3)}_{Ni}\Delta^{(e)}_{N\alpha} = \bigcup_{e=1}^{E} R^{(e)}_{NM}\overline{E}^{(e)(3)}_{Mi}\Delta^{(e)}_{N\alpha}$$

$j = 1$:

$$a^{(3,1)} = \tilde{E}^{(3)}_{\alpha i}\overline{E}^{(1)}_{\alpha i}$$

$$\tilde{E}^{(3)}_{\alpha i} = \tilde{E}^{(3)}_{\alpha i} - a^{(3,1)}\overline{E}^{(1)}_{\alpha i}$$

$j = 2$:

$$a^{(3,2)} = \tilde{E}^{(3)}_{\alpha i}\overline{E}^{(2)}_{\alpha i}$$

$$\tilde{E}^{(3)}_{\alpha i} = \tilde{E}^{(3)}_{\alpha i} - a^{(3,2)}\overline{E}^{(2)}_{\alpha i}$$

$j = 3$:

$$a^{(3,3)} = \tilde{E}^{(3)}_{\alpha i}\overline{E}^{(3)}_{\alpha i}$$

$$\tilde{E}^{(3)}_{\alpha i} = \tilde{E}^{(3)}_{\alpha i} - a^{(3,3)}\overline{E}^{(3)}_{\alpha i}$$

$$\overline{E}^{(4)}_{\alpha i} = \frac{\tilde{E}^{(3)}_{\alpha i}}{\| \tilde{E}^{(3)}_{\alpha i} \|}$$

(d) $i = 4$: Again similarly,

$$\tilde{E}^{(4)}_{\alpha i} = \bigcup_{e=1}^{E} E^{(e)(4)}_{Ni}\Delta^{(e)}_{N\alpha} = \bigcup_{e=1}^{E} R^{(e)}_{NM}\overline{E}^{(e)(4)}_{Mi}\Delta^{(e)}_{N\alpha}$$

$j = 1$:

$$a^{(4,1)} = \tilde{E}^{(4)}_{\alpha i}\overline{E}^{(1)}_{\alpha i}$$

$$\tilde{E}^{(4)}_{\alpha i} = \tilde{E}^{(4)}_{\alpha i} - a^{(4,1)}\overline{E}^{(1)}_{\alpha i}$$

$j = 2$:

$$a^{(4,2)} = \tilde{E}^{(4)}_{\alpha i}\overline{E}^{(2)}_{\alpha i}$$

$$\tilde{E}^{(4)}_{\alpha i} = \tilde{E}^{(4)}_{\alpha i} - a^{(4,2)}\overline{E}^{(2)}_{\alpha i}$$

$j = 3$:

$$a^{(4,3)} = \tilde{E}^{(4)}_{\alpha i}\overline{E}^{(3)}_{\alpha i}$$

$$\tilde{E}^{(4)}_{\alpha i} = \tilde{E}^{(4)}_{\alpha i} - a^{(4,3)}\overline{E}^{(3)}_{\alpha i}$$

$j = 4$:

$$a^{(4,4)} = \tilde{E}^{(4)}_{\alpha i}\overline{E}^{(4)}_{\alpha i}$$

$$\tilde{E}^{(4)}_{\alpha i} = \tilde{E}^{(4)}_{\alpha i} - a^{(4,4)}\overline{E}^{(3)}_{\alpha i} \approx 0$$

$$\overline{E}^{(5)}_{\alpha i} = \frac{\tilde{E}^{(4)}_{\alpha i}}{\|\tilde{E}^{(4)}_{\alpha i}\|} \approx 0$$

4. Construct Hessenberg matrix to calculate the minimizer vector $y_r$ ($r = 4$ in this case)

$$\begin{bmatrix} a^{(1,1)} & a^{(2,1)} & a^{(3,1)} & a^{(4,1)} \\ \|\tilde{E}^{(1)}_{\alpha i}\| & a^{(2,2)} & a^{(3,2)} & a^{(4,2)} \\ & \|\tilde{E}^{(2)}_{\alpha i}\| & a^{(3,3)} & a^{(4,3)} \\ & & \|\tilde{E}^{(3)}_{\alpha i}\| & a^{(4,4)} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} \|E^{(0)}_{\alpha i}\| \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\|\tilde{E}^{(4)}_{\alpha i}\| \cong 0$ is assumed.

5. Apply Givens rotations to reduce Hessenberg matrix to an upper triangular form in order to find the minimizer error vector $y$, as shown in step 5 of Example 11.5.1

6. Compute residuals (for the case of Figure 11.6.3.1a)

$$\underset{(378\times 4)}{\begin{bmatrix} \overline{E}^{(1)}_1 & \overline{E}^{(2)}_1 & \overline{E}^{(3)}_1 & \overline{E}^{(4)}_1 \\ \overline{E}^{(1)}_2 & \overline{E}^{(2)}_2 & \overline{E}^{(3)}_2 & \overline{E}^{(4)}_2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \overline{E}^{(1)}_{378} & \overline{E}^{(2)}_{378} & \overline{E}^{(3)}_{378} & \overline{E}^{(4)}_{378} \end{bmatrix}}\underset{(4\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}} = \underset{(378\times 1)}{\begin{bmatrix} E^{(r)}_1 \\ E^{(r)}_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E^{(r)}_{378} \end{bmatrix}}$$

7. Update $U_{\beta i}$

$$\begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v_{378} \end{bmatrix} = \begin{bmatrix} v_1^{(0)} \\ v_2^{(0)} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v_{378}^{(0)} \end{bmatrix} + \begin{bmatrix} E_1^{(r)} \\ E_2^{(r)} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E_{378}^{(r)} \end{bmatrix}$$

If the adjusted error vector $\tilde{E}_{\alpha i}^{(4)}$ and the error coefficient vector $\overline{E}_{\alpha i}^{(5)}$ are not approximately zero, then further iterations will be required.

### 11.5.5 PRECONDITIONING FOR EBE-GMRES

Although Krylov subspace methods such as the GMRES method are well founded theoretically, they are likely to suffer from slow convergence for fluid dynamics applications, especially in the problems involving high Mach numbers and high Reynolds numbers. Preconditioning is a key ingredient in the success of Krylov subspace methods in these applications. In creating a preconditioner for the EBE equations, the first step is to normalize each element matrix using a scaling transformation that can be viewed as an initial level of preconditioning, often called “pre-preconditioning” [Saad, 1996; Shakib et al. 1991]. Typically, a diagonal, or a block diagonal, scaling is first applied to the element matrices to obtain scaled element matrices.

***Step 1: Pre-preconditioning***

Consider the local finite element equations given by

$$R_{NMrs}^{(e)}\,\Delta U_{Ms}^{(e)} = Q_{Nr}^{(e)} \tag{11.5.45}$$

The left-hand side may be written as

$$C_{Nr}^{(e)} = R_{NMrs}^{(e)}\,\Delta U_{Ms}^{(e)} \tag{11.5.46}$$

The EBE process provides

$$C_{\alpha r}^{n+1} = \bigcup_{e=1}^{E} C_{Nr}^{(e)}\,\Delta_{N\alpha}^{(e)} \tag{11.5.47}$$

with

$$Q_{\alpha r}^{n+1} = \bigcup_{e=1}^{E} Q_{Nr}^{(e)}\,\Delta_{N\alpha}^{(e)} \tag{11.5.48}$$

Construct the diagonal scaling matrix $D_{\alpha\beta rs}$ in the form

$$D_{\alpha\beta rs} = \bigcup_{e=1}^{E} R_{\alpha prs}^{(e)}\,\delta_{pM}\Delta_{M\beta}^{(e)}$$

Note that since the off-diagonal terms of $D_{\alpha\beta rs}$ are zero, $D_{\alpha\beta rs}$ can be stored as a vector.

Performing the preconditioning operations on the unassembled element equations requires three steps:

(1) Gather, or localize, the components of the global diagonal vector into local element vectors. Let $D^{(e)}_{NMrs}$ denote the local diagonal matrix for element $(e)$.

(2) Perform the preconditioning operations on the element level. Equation (11.5.48) is transformed into

$$\tilde{R}^{(e)}_{NMrs}\,\Delta\tilde{U}^{(e)}_{Ms} = \tilde{Q}^{(e)}_{Nr} \tag{11.5.49}$$

where

$$\begin{aligned}
\tilde{R}^{(e)}_{NMrs} &= (\tilde{D}^{(e)}_{Np})^{-\frac{1}{2}}\,R_{pqrs}(\tilde{D}^{(e)}_{qM})^{-\frac{1}{2}}\\
\Delta\tilde{U}^{(e)}_{Mr} &= (D^{(e)}_{Mp})^{-\frac{1}{2}}\,\Delta U^{(e)}_{pr}\\
Q^{(e)}_{Nr} &= (D^{(e)}_{Np})^{-\frac{1}{2}}\,Q^{(e)}_{pr}
\end{aligned}$$

with

$$\tilde{C}^{(e)}_{Nr} = \tilde{R}^{(e)}_{NMrs}\Delta U^{(e)}_{Ms}$$

(3) Scatter, or globalize, the components of the local element vectors into the global vectors as follows:

$$\tilde{C}^{(e)}_{\alpha r} = \bigcup_{e=1}^{E}\tilde{C}^{(e)}_{Nr}\,\Delta^{(e)}_{N\alpha},\qquad \tilde{Q}^{(e)}_{\alpha r} = \bigcup_{e=1}^{E}\tilde{Q}^{(e)}_{Nr}\,\Delta^{(e)}_{N\alpha} \tag{11.5.50}$$

***Step 2: Main preconditioning by upper and lower triangular matrices***

The second step in defining an EBE preconditioner is to regularize the transformed element matrices from step 1. Using Winget regularization, the diagonal of each coefficient matrix is forced to be the identity matrix. In other words, the regularized matrix is defined as

$$\bar{R}^{(e)}_{NMrs} = \tilde{R}^{(e)}_{NMrs} - diag(\tilde{R}^{(e)}_{NMrs}) + I_{NMrs} \tag{11.5.51}$$

Finally, the factorization must be chosen for the preconditioning matrix. We choose the LU factorization for the regularized matrix $\bar{R}^{(e)}_{NMrs}$ to produce the preconditioning matrix $G^{(e)}_{NMrs}$ of the form

$$G^{(e)}_{NMrs} = L^{(e)}_{Nprt}\,U^{(e)}_{pMts} \tag{11.5.52}$$

where $L^{(e)}_{Nprt}$and $U^{(e)}_{pMts}$ are obtained by factoring the regularized matrix $\bar{R}^{(e)}_{NMrs}$ into a unit lower and an upper triangular matrix. In other words,

$$G^{(e)} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ L_{21} & 1 & 0 & 0 & \cdots & \vdots \\ L_{31} & L_{32} & 1 & 0 & \cdots & \vdots \\ L_{41} & L_{42} & \cdots & \ddots & 0 & \\ \vdots & \cdots & \cdots & \cdots & \ddots & 0 \\ L_{M1} & L_{M2} & \cdots & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} & \cdots & \cdots & U_{1M} \\ 0 & U_{22} & U_{23} & U_{24} & \cdots & U_{2M} \\ 0 & 0 & U_{33} & U_{34} & \cdots & U_{3M} \\ 0 & 0 & 0 & U_{44} & \cdots & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & U_{MM} \end{bmatrix}$$

where the indices $rt$ and $ts$ are omitted for simplicity.

Notice that in practice, $L^{(e)}_{Nprt}$and $U^{(e)}_{PMts}$ can be stored together.

We premultiply the left-hand and right-hand sides of (11.5.49) by the inverse of the preconditioned local element matrices as follows:

$$G^{(e)^{-1}}_{pNrt}\, R^{(e)}_{NMts}\, \Delta U^{(e)}_{Ms} = G^{(e)^{-1}}_{pNrs}\, Q_{Ns} \tag{11.5.53}$$

However, in practice we do not actually calculate the inverse of the preconditioning matrix. Instead, consider writing the right-hand side of (11.5.53) as

$$\hat{Q}^{(e)}_{Nr} = L^{(e)^{-1}}_{NMrt} U^{(e)^{-1}}_{Mpts} \tilde{Q}^{(e)}_{ps}, \quad \text{or} \quad L^{(e)}_{NMrt} U^{(e)}_{Mpts} \hat{Q}^{(e)}_{ps} = \tilde{Q}^{(e)}_{Nr} \tag{11.5.54}$$

Consider rewriting (11.5.54) as

$$L^{(e)}_{NMrs} Z^{(e)}_{Ms} = \tilde{Q}^{(e)}_{Nr} \tag{11.5.55}$$

where $Z^{(e)}_{Mr} = U^{(e)}_{MNrs} \tilde{Q}^{(e)}_{Ns}$. Since $L^{(e)}_{NMrs}$ is lower triangular, Equation (11.5.55) can be solved for $Z^{(e)}_{Mr}$ using forward reduction. Then, the equation $U^{(e)}_{MNrs} \hat{Q}^{(e)}_{Ns} = Z^{(e)}_{Mr}$ can be solved for $\hat{Q}^{(e)}_{Ns}$, which is the right-hand side of (11.5.53), by back substitution. A similar operation is performed to evaluate the left-hand side of (11.5.53). The element values are then mapped to the global column vector as shown below.

$$\hat{C}^{(e)}_{Nr} = G^{(e)^{-1}}_{pNrt}\, R^{(e)}_{pMts}\, \Delta \tilde{U}^{(e)}_{Ms}, \qquad \hat{C}_{\alpha r} = \bigcup_{e=1}^{E} \hat{C}^{(e)}_{Nr}\, \Delta^{(e)}_{N\alpha}$$

$$\hat{Q}^{(e)}_{Nr} = G^{(e)^{-1}}_{NMrs}\, \tilde{Q}^{(e)}_{Ms}, \qquad \hat{Q}_{\alpha r} = \bigcup_{e=1}^{E} Q^{(e)}_{Nr}\, \Delta^{(e)}_{N\alpha}$$

The pre-conditioned GMRES process begins with

$$E^{(0)}_{\alpha r} = \hat{Q}^{(0)}_{\alpha r} - \hat{C}^{(0)}_{\alpha r}$$

and

$$\overline{E}^{(1)}_{\alpha i} = \frac{E^{(0)}_{\alpha i}}{\left\| E^{(0)}_{\alpha i} \right\|}$$

Step 2 of the GMRES procedure described in Section 11.5.3 is rewritten as follows:
GMRES iteration: For $i = 1, 2, 3, \ldots, r$ Do

$$\overline{E}^{(i+1)}_{\alpha r} = G^{-1}_{\alpha\gamma rt} R_{\gamma\beta ts} \overline{E}^{(i)}_{\beta s} = \bigcup_{e=1}^{E} G^{(e)^{-1}}_{NMrt} R^{(e)}_{Mpts} \overline{E}^{(e)}_{ps} \Delta^{(e)}_{N\alpha}$$

The rest follows identically as in step 2 through step 6.

## 11.6 EXAMPLE PROBLEMS

### 11.6.1 NONLINEAR WAVE EQUATION (CONVECTION EQUATION)

Consider the first order nonlinear wave equation of the form used in Section 4.7.5.

$$\begin{aligned}&\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \quad 0 \le x \le 4\\ &u(x,0) = 1 \quad 0 \le x \le 2\\ &u(x,0) = 0 \quad 2 \le x \le 4\end{aligned}$$

***Required:*** Solve with GPG using the numerical diffusion given by (11.3.32).

***Solution:*** The GPG formulation begins with

$$\int_0^L W(\xi)\left[\int \Phi_\alpha\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x}\right)dx + \int \Psi_\alpha u\frac{\partial u}{\partial x}dx\right]d\xi = 0$$

with

$$\Psi_N^{(e)} = \tau u \frac{\partial \Phi_N^{(e)}}{\partial x}$$

where $\tau$ is the numerical diffusion factor (intrinsic time scale),

$$\tau = \frac{C}{2}\frac{h}{u}$$

with $C$ being the CFL number,

$$C = \overline{\alpha} = \coth H - \frac{1}{H}$$

which is characterized by the numerical diffusion as shown in Figure 11.3.2 defining the accuracy and stability for the solution of the nonlinear convection equation.

As a result, it is seen that dispersion or dissipation errors decrease with mesh refinements, as shown in Figure 11.6.1. Accuracies deteriorate significantly with inadequate numerical diffusivity constants outside the stability and accuracy criteria.

### 11.6.2 PURE CONVECTION IN TWO DIMENSIONS

The two-dimensional pure convection equation for a concentration cone placed in a rotating velocity field, as shown in Figure 11.6.2a is given by

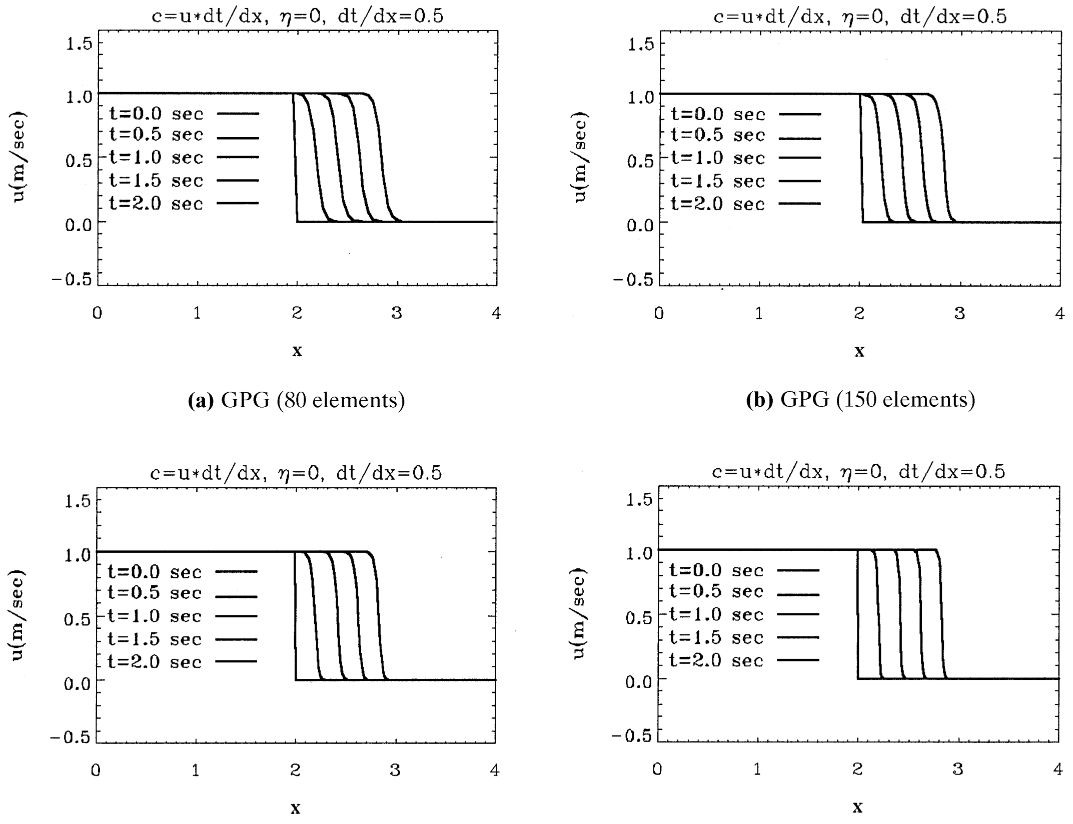
$$\frac{\partial u}{\partial t} + A_i\frac{\partial u}{\partial x_i} = 0$$

where

$$A_i = (a\cos\theta, a\sin\theta) \quad \text{with } a = 1/2$$

***Initial Data:***

$$u_0 = \begin{cases}\frac{1}{2}(1+\cos 4\rho\pi) & \rho \le \frac{1}{4}\\ 0 & \text{otherwise}\end{cases}$$



**(a)** GPG (80 elements)

**(b)** GPG (150 elements)

**(c)** GPG (200 elements)

**(d)** GPG (400 elements)

**Figure 11.6.1** GPG solutions for nonlinear convection shock wave propagation (lumped mass matrix).

where

$$\rho^2 = (x - 0)^2 + (y + 0.5)^2$$

***Required:*** Solve using the GTG and GPG methods with lumped and consistent mass matrices. Carry out until 1 revolution is reached.

***Solution:*** For the computation, a 32 × 32 grid mesh in a 2.0 × 2.0 domain is chosen, and initial cosine hill with unit magnitude is centered at (0.0, −0.5) whose base radius spans eight elements in Figure 11.6.2b. Use a constant time step, $\Delta t = 2\pi/400$. The total number of nodes is 1089, and all boundary conditions are Dirichlet type, $u = 0$, a complete rotation is accomplished in 400 time steps. The Courant number at the peak of the cone is approximately 1/4.

For the GTG method with the lumped mass, the solution with one iteration (Figure 11.6.2c) has wiggles and reduced cone height more than those with three iterations (Figure 11.6.2d); an improved solution is obtained for the case of consistent mass (Figure 11.6.2e) for $t = \pi/4$ as compared with that for lumped mass. The results of the GPG method at $t = \pi/4$ are shown in (Figure 11.6.2f) (1), (2), (3), and (4) corresponding to the numerical diffusivity of $\overline{\alpha} = 10^{-4}, 10^{-2}, 1,$ and $10^2$, respectively. In Figure 11.6.2g,

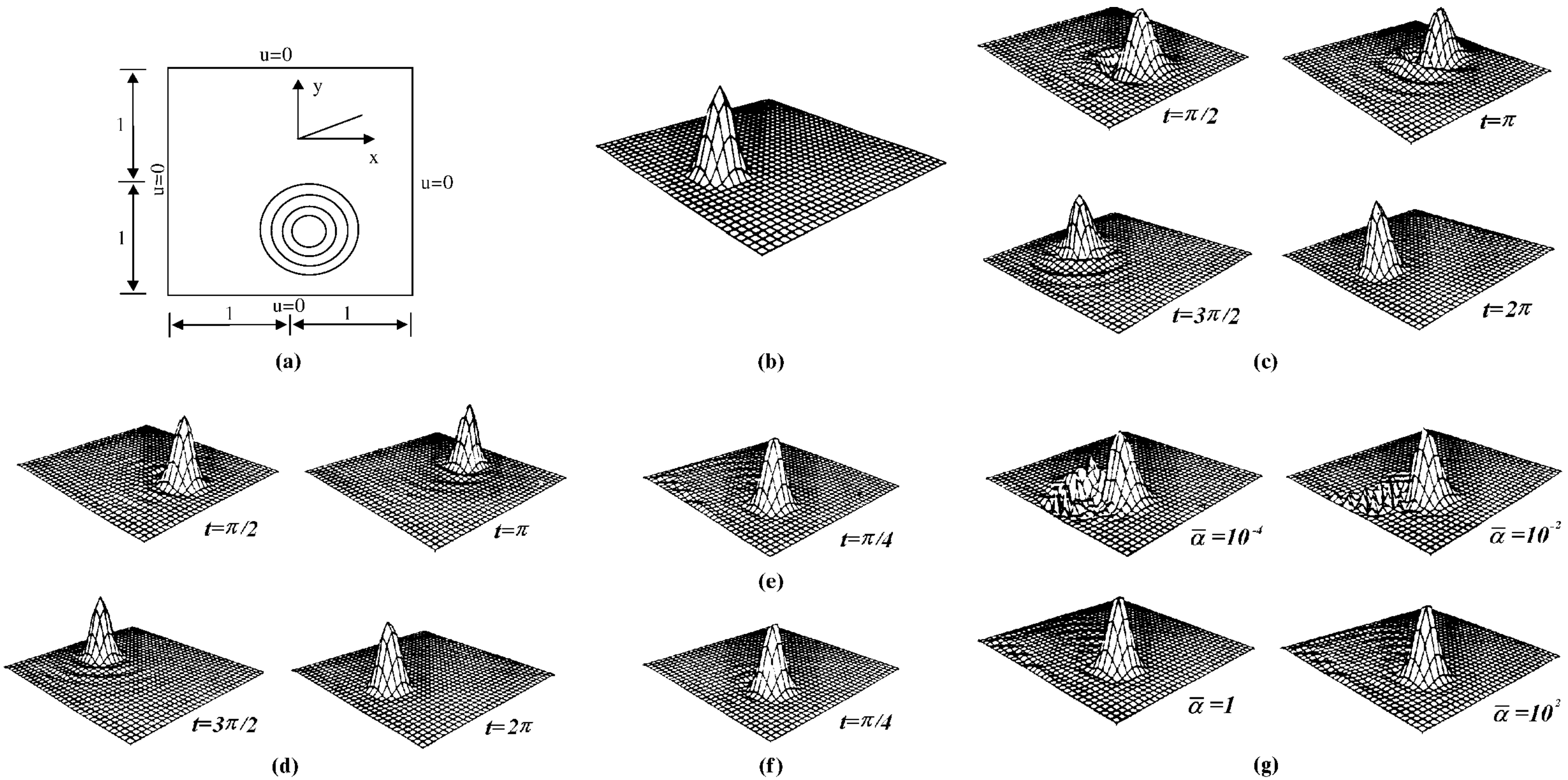


**Figure 11.6.2** Rotating cone with cosine hill. **(a)** Geometry, rotating cone. **(b)** Unit initial cosine hill at $x = 0$, $y = -0.5$, $t = 0$. **(c)** Lumped mass, one iteration. **(d)** Lumped mass, three iterations. **(e)** Consistent mass, GTG. **(f)** Lumped mass, GTG. **(g)** Consistent mass, GPG.

(1) and (2), the GPG methods show oscillatory behavior at $\overline{\alpha} = 10^{-4}$ and $10^{-2}$, which disappears at $\overline{\alpha} = 1$ and $10^2$ in Figure 11.6.2g, (3) and (4). Although the GPG methods provide numerical diffusion in the direction of the flow for stability, the methods may be restricted within the low Reynolds numbers unlike the GTG methods.

### 11.6.3 SOLUTION OF 2-D BURGERS' EQUATION

The purpose of this section is to show the effectiveness of GPG for the solution of the Burgers' equations with convection terms and its solution convergence as a function of the grid refinements. We use the geometry as shown in Figure 11.6.3.1, the same geometry as in Section 10.4.2.

***Given:*** The Burgers' equations with the nonlinear convection terms are given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \nu\frac{\partial u}{\partial y} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - f_x = 0$$

$$\frac{\partial \nu}{\partial t} + u\frac{\partial \nu}{\partial x} + \mathrm{v}\frac{\partial \mathrm{v}}{\partial y} - \nu\left(\frac{\partial^2 \mathrm{v}}{\partial x^2} + \frac{\partial^2 \mathrm{v}}{\partial y^2}\right) - f_y = 0$$

with

$$f_x = -\frac{1}{(1+t)^2} + \frac{x^2 + 2xy}{(1+t)} + 3x^3y^2 - 2\nu y$$

$$f_y = -\frac{1}{(1+t)^2} + \frac{y^2 + 2xy}{(1+t)} + 3y^3x^2 - 2\nu x$$

***Exact Solution:***

$$u = \frac{1}{1+t} + x^2 y$$

$$\nu = \frac{1}{1+t} + xy^2$$

***Required:*** Solve the Burgers' equations using GPG for the coarse, intermediate, and fine meshes as shown in Figure 11.6.3.1. Neumann boundary conditions are to be specified at nodes marked by $N$ and all other boundary nodes are Dirichlet. They are computed by the exact solution as given above. Use bilinear isoparametric elements with $\nu = 1$, $\Delta t = 10^{-4}$, and $\eta = 1/2$. Begin with the initial conditions $u = 0$ and $\mathrm{v} = 0$ specified everywhere.

***Solution:*** Shown in Figure 11.6.3.2 are the solutions at $x = 2$ and $y = 1$ for the coarse, intermediate, and fine meshes. It is seen that, although the initial conditions as given are $u = 0$ and $\mathrm{v} = 0$, they quickly rise toward the exact solution. For the coarse grid, however, the solution overshoots considerably. The convergence to the exact solution is evident for the intermediate grid and significantly for the fine grid.

## 11.7 SUMMARY

The generalized Galerkin methods (GGM) introduced in Chapter 10 have been extended to the Taylor Galerkin methods (TGM) and to the generalized Petrov-Galerkin

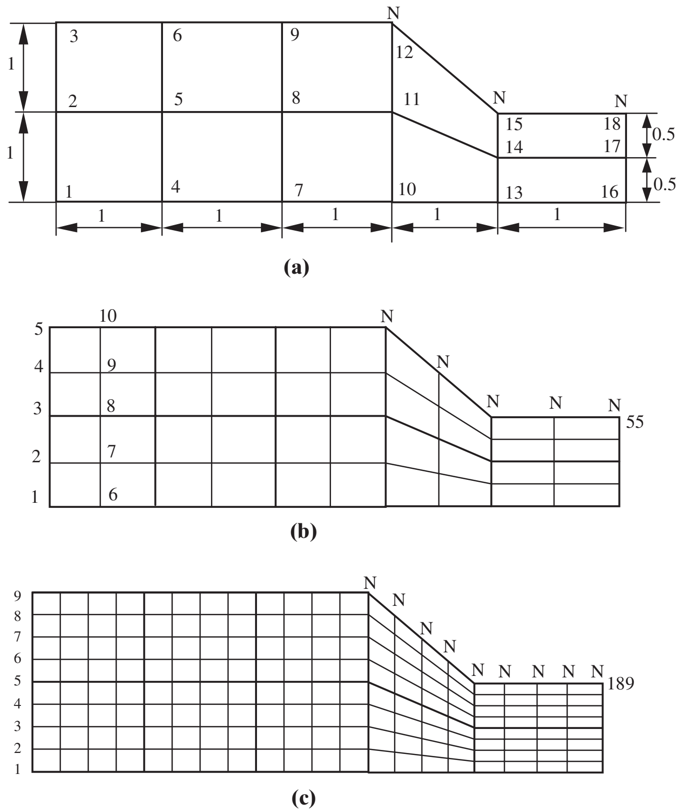


**Figure 11.6.3.1** Geometries for Section 11.6.3, N representing Neumann boundary conditions, Dirichlet and Neumann boundary conditions are prescribed from the exact solution. **(a)** Coarse grid. **(b)** Intermediate grid (halved from the coarse grid). **(c)** Fine grid (halved from the intermediate grid).

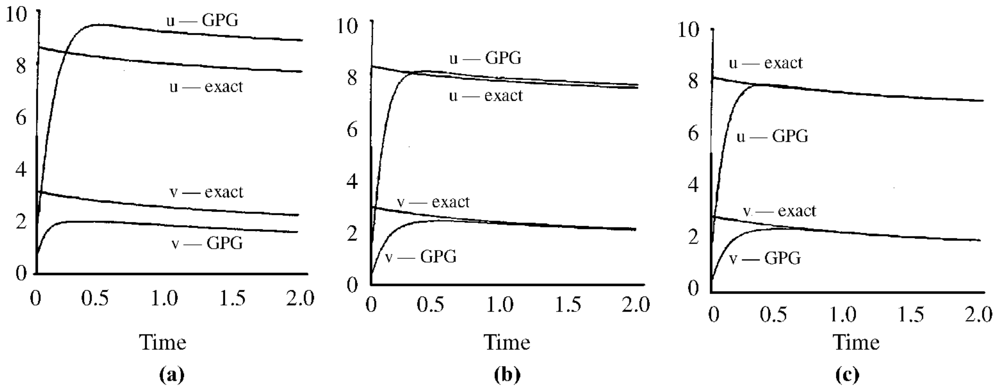


**Figure 11.6.3.2** Solution of 2-D Burgers' equations, $x = 2$, $y = 1$ $(\nu = 1)$. **(a)** Coarse grid. **(b)** Intermediate grid. **(c)** Fine grid.

(GPG) methods in order to cope with convection-dominated flows. It was shown that the basic idea of TGM is to provide numerical diffusivity. In GPG, more rigorous approaches to treat convection-dominated flows are employed through SUPG, discontinuity-capturing scheme, and space-time Galerkin/least squares. The significant features available in GPG are to explicitly provide numerical diffusion in the direction of streamline and toward velocity gradients or acceleration. Furthermore, the concept of least squares is applied to reinforce the numerical diffusivity.

In this chapter, we also examined numerical solution of nonlinear equations using the Newton-Raphson methods. The element-by-element methods in which the assembly of total stiffness matrices is replaced by the element-by-element vector operation introduced in Section 10.3.2 are extended to the nonlinear equations. Furthermore, we reviewed GMRES which is regarded as the most rigorous equation solver for nonlinear, nonsymmetric matrices.

Major applications in CFD are the solutions of the Navier-Stokes system of equations for incompressible and compressible flows. These are the topics to be discussed in the next two chapters.

## REFERENCES

Arnoldi, W. A. [1951]. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9, 17–29.

Brooks, A. and Hughes, T. J. R. [1982]. Streamline upwind Petrov/Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 32, 181.

Christie, I., Griffiths, D. F., Mitchel, A. R., and Zienkiewicz, O. C. [1976]. *Int. J. Num. Eng.*, 10, 1389–96.

Chung, T. J. [1978]. *Finite Element Analysis in Fluid Dynamics*. New York: McGraw-Hill.

———. [1999]. Transitions and interactions of inviscid/viscous, compressible/incompressible and laminar/turbulent flows. *Int. J. Num. Meth. Fl.*, 31, 223–46.

Donea, J. [1984]. A Taylor-Galerkin method for convective transport problems. *Int. J. Num. Meth. Eng.*, 20, 101–19.

Fox, R. L. and Stanton, E. L. [1968]. Developments in structural analysis by direct energy minimization. *AIAA J.*, 6, 1036–42.

Heinrich, J. C., Huyakorn, P. S., Zienkiewicz, O. C., and Mitchell, A. R. [1977]. An upwind finite element scheme for two-dimensional convective transport equation. *Int. J. Num. Meth. Eng.*, 11, no. 1, 131–44.

Hauke, G. and Hughes. T. J. R. [1998]. A comparative study of different sets of variables for solving compressible and incompressible flows. *Comp. Meth. Appl. Mech. Eng.*, 153, 1–44.

Hood, P. [1976]. Frontal solution program for unsymmetric matrices. *Int. J. Num. Meth.*, 10, 379–99.

Hood, P. and Taylor, C. [1974]. Navier-Stokes equations using mixed interpolation. In Oden et al. (eds.), *Finite Element Methods in Flow Problems*, Huntsville: University of Alabama Press.

Householder, A. S. [1964]. *Theory of Matrices in Numerical Analysis*. Johnson, CO: Blaisdell.

Hughes, T. J. R. [1987]. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. *Int. J. Num. Meth. Fl.*, 7, 1261–75.

Hughes, T. J. R. and Brooks, A. N. [1982]. A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions: application to the streamline upwind procedure. In R. H. Gallagher et al. (eds.), *Finite Elements in Fluids*, London: Wiley.

Hughes, T. J. R, Franca L. P., and Hulbert, G. M. [1986]. A new finite element formulation for computational fluid dynainics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. *Comp. Meth. Appl. Mech. Eng.*, 58, 329–36.

Hughes, T. J. R., Frencz, R. M., and Hallquist, J. O. [1987]. Large scale vectorized implicit calculations in solid mehanics on a Cray–MP/48 utilizing EBE preconditioned conjugate gradients. *Comp. Meth. Appl. Mech. Eng.*, 61, 215–48.

Hughes, T. J. R., Levit, I., and Winget, J. [1983]. An element-by-element implicit algorithm for heat conduction, *ASCE J. Eng. Mech. Div.*, 109, 576–85.

Hughes, T. J. R. and Mallet, M. [1986]. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multi-dimensional advective-diffusive systems. *Comp. Meth. Appl. Mech. Eng.*, 58, 305–28.

Hughes, T., Mallet, M., and Mizukami, A. [1986]. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Comp. Meth. Appl. Mech. Eng.*, 54, 341–55.

Hughes, T. J. R. and Tezduyar, T. E. [1984]. Finite element methods for first order hyperbolic systems with particular emphasis on the compressible Euler equations. *Comp. Meth. Appl. Mech. Eng.*, 45, 217–84.

Irons, B. M. [1970]. A frontal solution program for finite element analysis. *Int. J. Num. Meth. Eng.*, 2, 5–32.

Jameson, A., Baker, T. J., and Weatherill, N. P. [1986]. Calculation of inviscid transonic flow over a complete aircraft. AIAA-86-0103.

Jea, K. C. and Young, D. M. [1983]. On the simplification of generalized conjugate-gradient methods for nonsymmetrizable linear systems. *Linear Algebra Appl.*, 52, 399–417.

Johnson, C. [1987]. Numerical Solution of Partial Differential Equations on the Element Method Student litteratur, Lund, Sweden.

Lanczos, C. [1950]. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Stand.*, 45, 255–82.

Löhner, R., Morgan, K., and Zienkiewicz, O. C. [1985]. An adaptive finite element procedure for compressible high speed flows. *Comp. Meth. Appl. Mech. Eng.*, 51, 441–65.

Mikhlin, S. G. [1964]. *Variational Methods in Mathematical Physics*. Oxford, UK: Pergamon Press.

Nour-Omid, B. [1984]. A preconditioned conjugate gradient method for finite element equations. In W. K. Liu et al. (eds.), *Innovative Methods for Nonlinear Problems*, England: Swansea.

Nour-Omid, B. and Parlett, B. N. [1985]. Element preconditioning using splitting techniques. *SIAM J. Sci. Comp.*, 6, 761–70.

Oden, J. T., Babuska, I., and Baumann, C. E. [1998]. A discontinuous *hp* finite element methods for diffusion problems. *J. Comp. Phy.*, 146, 491–519.

Oden, J. T. and Demkowicz, L. [1991]. h-p adaptive finite element methods in computational fluid dynamics. *Comp. Meth. Appl. Mech. Eng.*, 89, (1–3): 1140.

Oden, J. T., Demkowicz, L., Strouboulis, T., and Devloo, P. [1986]. Adaptive finite element methods for the analysis of inviscid compressible flow: I. Fast refinement/unrefinement and moving mesh methods for unstructured meshes. *Comp. Meth. Appl. Mech. Eng.*, 59, 327–62.

Ortiz, M., Pinsky, P. M., Taylor, R. L. [1983]. Unconditionally stable element-by-element algorithm for synamic problems. *Comp. Meth. Appl. Mech. Eng.*, 36, 223–39.

Paige, C. C. and Saunders, M. A. [1975]. Solution of sparse indefinite systems of linear equations. *SIAM J. Num. Anal.*, 12, 617–24.

Raymond, W. H. and Garder, A. [1976]. Selective damping in a Galerkin method for solving wave problems with variable grids. *Mon. Weather Rev.* 104, 1583–90.

Saad, Y. [1996]. *Iterative Methods for Sparse Linear System*. Boston: PWS Publishing.

Saad, Y. and Schultz, M. H. [1986]. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7, 856–69.

Shakib, F. and Hughes, T. J. R. [1991]. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least squares algorithms. *Comp. Meth. Appl. Mech. Eng.*, 87, 35–58.

Spalding, D. B. [1972]. A novel finite-difference formulation for differential expressions involving both first and second derivatives. *Int. J. Num. Meth. Eng.*, 4, 551–59.

Tezduyar, T. [1997]. Advanced Flow Simulation and Modeling. AHPCRC 97-050, Minneapolis: University of Minnesota.

Zienkiewicz, O. C. and Codina, R. [1995]. A general algorithm for compressible and incompressible flow – Part I. Characteristic-based scheme. *Int. J. Num. Meth. Fl.*, 20, 869–85.