APPENDIX D

Multigrid Algorithm

The full-approximation storage (FAS) multigrid algorithm is used to accelerate convergence to steady state (or to accelerate convergence of sub-iterations during a time-accurate computation). A sequence of grids $G_0, G_1, \ldots, G_N$ is defined, where G_N denotes the finest grid, and coarser grids are formed by successively deleting every other grid line in all three coordinate directions. The fine grid serves to damp the high-frequency errors; the coarser grids damp the low-frequency errors. The coarse grids are solved with a forcing function on the right-hand side, arising from restricting the residual from the finer meshes. The forcing function is the relative truncation error between the grids, such that the solution on the coarser meshes are driven by the fine grid residual. The resulting scheme on mesh G_i is given as

$$N_i \Delta \mathbf{q}_i^c = -[L_i(\mathbf{q}_i^c) - \tau_i] \equiv -R_i \tag{D-1}$$

where $\mathbf{q}_i^c$ is the current approximation to the solution on mesh G_i, N_i is the spatially-factored implicit matrix, and τ_i is the relative truncation error (where τ_N is defined to be zero). The relative truncation error is calculated as

$$\tau_i = L_i(I_{i+1}^i \mathbf{q}_{i+1}^c) - \hat{I}_{i+1}^i R_{i+1} \tag{D-2}$$

The operator I_{i+1}^i is a volume-weighted restriction operator that transfers values on the finer grid to the coarser grid

$$I_{i+1}^i \mathbf{q}_{i+1} = \frac{\sum \frac{\mathbf{q}_{i+1}}{J_{i+1}}}{\sum \frac{1}{J_{i+1}}} \tag{D-3}$$

where the summation is taken over the eight finer grid cells that make up the coarser grid cell. The restriction operator $\hat{I}_{i+1}^i$ represents a summation over the finer grid cells which make up the coarser cell

$$\hat{I}_{i+1}^i R_{i+1} = \sum R_{i+1} \tag{D-4}$$

The corrections V_i on the coarser meshes are used to update the finer mesh

$$\begin{aligned} V_i &= \mathbf{q}_i^c - I_{i+1}^i \mathbf{q}_{i+1}^c \\ \mathbf{q}_i^c &\leftarrow \mathbf{q}_i^c + I_{i-1}^i V_{i-1} \end{aligned} \tag{D-5}$$

where the prolongation operator corresponds to trilinear interpolation.

When correction smoothing (**issc** = 1) is employed, the corrections V_i are smoothed on G_i, before prolongation, with a Laplacian-type operator factored into three sweeps. Equation (D-5) is replaced with

$$\begin{aligned} V_i &= \mathbf{q}_i^c - I_{i+1}^i \mathbf{q}_{i+1}^c \\ (I - \varepsilon_\xi^c \delta_{\xi\xi})(I - \varepsilon_\eta^c \delta_{\eta\eta})(I - \varepsilon_\zeta^c \delta_{\zeta\zeta}) \tilde{V}_i &= V_i \\ \mathbf{q}_i^c &\leftarrow \mathbf{q}_i^c + I_{i-1}^i \tilde{V}_{i-1} \end{aligned} \tag{D-6}$$

The ε_ξ^c, ε_η^c, and ε_ζ^c coefficients are user-input values. When used, typical values are 0.3. The correction smoothing overcomes an odd-even decoupling sometimes encountered with the FAS algorithm on highly-stretched grids.

When residual smoothing (**issr** = 1) is employed, the same Laplacian-type operator is used to smooth the $\Delta\mathbf{q}_i^c$ values, just prior to updating the primitive variables on a given grid level.

$$(I - \varepsilon_\xi^r \delta_{\xi\xi})(I - \varepsilon_\eta^r \delta_{\eta\eta})(I - \varepsilon_\zeta^r \delta_{\zeta\zeta}) \tilde{\mathbf{q}}_i^c = \Delta\mathbf{q}_i^c \tag{D-7}$$

The ε_ξ^r, ε_η^r, and ε_ζ^r coefficients are again user-input values. When used, typical values are 0.3.