Chapter 20

Compressible Navier-Stokes and Euler Equations

The steady-state Euler equations of inviscid flows will be treated here as a special, limit case of the full steady-state Compressible Navier-Stokes (CNS) equations. Some terms can be dropped in the inviscid limit, but there is no essential difference between the numerical solution of inviscid flows and that of slightly viscous flows, because $O(h)$ *artificial* viscosity should anyway be used in relaxation, and it should closely resemble the *physical* viscosity to ensure that only physical discontinuities are admitted at $h \to 0$. The double-discretization scheme (§10.2) will be used; in the present context this simply means that the artificial viscosity, when needed, is employed only in the relaxation operators, not in the difference operators by which residuals are calculated in the fine-to-coarse transfers. The latter can generally use simple central differencing, but both types of operators should respect, as much as possible, the flow discontinuities. The multigrid processes will first be described in terms of the simpler quasi-linear form of the equations, discretized on a staggered grid, and then their modification to the conservation form and to non-staggered discretization will be discussed.

20.1 The differential equations

20.1.1 Conservation laws and simplification

The time-dependent Compressible Navier-Stokes (CNS) equations in d dimensions may be written in in conservation form as

$$\frac{\partial W}{\partial t} + \sum_{j=1}^{d} \partial_j F_j = \underline{f}, \tag{20.1}$$

where $\partial_j = \partial/\partial x_j$ and

$$W = \begin{pmatrix} \rho u_1 \\ \vdots \\ \rho u_d \\ \rho \\ e \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j u_1 + \tau_{j1} \\ \vdots \\ \rho u_j u_d + \tau_{jd} \\ \rho u_j \\ e u_j + \sum_i u_i \tau_{ij} - \kappa \partial_j \varepsilon \end{pmatrix}, \quad \underline{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_d \\ f_\rho \\ f_\varepsilon \end{pmatrix},$$

$$\tau_{ij} = \tau_{ji} = -\mu(\partial_i u_j + \partial_j u_i) - \lambda \delta_{ij} \sum_{k=1}^{d} \partial_k u_k + p\delta_{ij}, \tag{20.2}$$

ρ being the fluid density, $\underline{u} = (u_1, \ldots, u_d)$ the velocity vector, e the total energy per unit volume, ε the specific internal energy, p the pressure, λ and μ the viscosity coefficients, and κ the coefficient of heat conductivity. (20.1) is a system of $d+2$ equations: the first d equations are the d momentum equations, next is the continuity (mass conservation) equation, and the last is the energy equation. The $d+2$ basic unknowns may be considered to be $\underline{u}$, ρ and ε, in terms of which e is given by

$$e = \rho\left(\varepsilon + \frac{q^2}{2}\right), \qquad q^2 = \sum_{i=1}^{d} u_i^2, \tag{20.3}$$

and p by the equation of state

$$p = \overline{p}(\varepsilon, \rho). \tag{20.4}$$

For a perfect gas, for example, the equation of state is $p = (\gamma - 1)\varepsilon\rho$, where γ is the ratio of specific heats. Generally, $p_\varepsilon = \partial\overline{p}/\partial\varepsilon$ and $p_\rho = \partial\overline{p}/\partial\rho$ are positive. The coefficients λ, μ and κ are given functions, usually, of ε (or functions of the fluid temperature, which in turn is a function of ε). We will treat these coefficients as constants, since they change slowly. The whole discussion below will remain precisely valid as long as any change in any of these coefficients over a meshsize is small compared with the coefficient itself.

In most aerodynamical applications the right-hand side $\underline{f}$ vanishes, but there are other applications where the external body force $(f_1, \ldots, f_d)$, or the mass source f_ρ, or the energy source f_ε do not vanish. A general $\underline{f}$ is assumed here, mainly because many of the numerical experiments are set to have known specified solutions by pre-arranging $\underline{f}$ accordingly.

The steady-state CNS equations are given by

$$\sum_{j=1}^{d} \partial_j F_j = \underline{f}, \tag{20.5}$$

together with (20.3) and (20.4). It will be convenient below to substitute (20.3) into equations (20.5), but to treat (20.4) as an additional equation

and p as an additional unknown. Allowing (20.4) not to be held exactly satisfied until convergence will substantially simplify the solution process, and more than justify the additional storage needed for p. Thus we will have a system of $n = d + 3$ differential equations in n unknown functions. The n-vector $U = (u_1, \ldots, u_d, \rho, \varepsilon, p)$ will serve as our vector of unknowns.

Simplified equations, but not in conservation form, are obtained as follows. The *i-th* momentum equation is simplified by subtracting from it u_i times the continuity equation. Then the energy equation is simplified by subtracting from it u_i times the *i-th* (simplified) momentum equation for $i = 1, \ldots, d$, and ε/ρ times the continuity equation. The resulting system is

$$\rho \sum_j u_j \partial_j u_i + \sum_j \partial_j \tau_{ij} = f_i \qquad (i = 1, \ldots, d) \tag{20.6a}$$

$$\sum_j \partial_j (\rho u_j) = f_\rho \tag{20.6b}$$

$$\rho \sum_j u_j \partial_j \varepsilon - \sum_j \partial_j (\kappa \partial_j \varepsilon) + \sum_{i,j} \tau_{ij} \partial_i u_j = f_\varepsilon \tag{20.6c}$$

$$p = \overline{p}(\varepsilon, \rho) \tag{20.6d}$$

which, in view of (20.2), is a system of n equations for the n unknowns U. In terms of this simpler system we will now study the principal and the inviscid subprincipal parts. This will tell us what boundary conditions are appropriate and which terms are locally dominant, which is important for designing the relaxation scheme and the form of the artificial viscosity terms.

20.1.2 The viscous principal part

The principal part of (20.6), i.e., the part of the linearized operator which contributes to the highest-order terms of its determinant, is the operator

$$L_p = \begin{pmatrix} -\mu\Delta - \overline{\lambda}\partial_{11} & \cdots & -\overline{\lambda}\partial_{1d} & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ -\overline{\lambda}\partial_{d1} & \cdots & -\mu\Delta - \overline{\lambda}\partial_{dd} & 0 & 0 & 0 \\ 0 & \cdots & 0 & \underline{u} \cdot \underline{\nabla} & 0 & 0 \\ 0 & \cdots & 0 & 0 & -\kappa\Delta & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix} \tag{20.7}$$

where $\partial_{ij} = \partial_i \partial_j$, $\Delta = \partial_{11} + \cdots + \partial_{dd}$, $\underline{u} \cdot \underline{\nabla} = u_1 \partial_1 + \cdots u_d \partial_d$ and $\overline{\lambda} = \lambda + \mu$. It can be shown by dimensional analysis that on a small enough scale the behavior of solutions to the original system depends only on L_p. Thus, on a sufficiently small scale, viscosity is the main mechanism that determines velocities, convection determines density, diffusion determines the internal energy, and they are all locally independent. It also follows from (20.7)

that the full CNS system requires d boundary conditions for the velocities (usually $\underline{u}$ is given) and one for ε (usually given in the form of boundary temperature or temperature gradient) on all boundaries, and an additional condition for ρ (or for p) at one end of each streamline.

20.1.3 Elliptic singular perturbation

Since $\underline{u}\cdot\underline{\nabla}$ is a factor of $\det L_p$, the steady-state CNS system is not elliptic: The streamlines p are its characteristic lines (the *only* characteristics, as long as the flow is viscous). This means that an addition of artificial h-ellipticity will be needed in local numerical processes, unless the grid exactly aligns with the stream. Therefore, and for uniform treatment of all artificial terms in the inviscid limit, we will regard already the CNS *differential* system as a limit, $\nu \to 0$ say, of an elliptic system, obtained by adding a singular-perturbation term to the continuity equation (20.6b), rewriting it (times ρ) as

$$\sum_{j=1}^{d} [\rho\partial_j(\rho u_j) - \partial_j(\nu\partial_j\rho)] = f_\rho. \tag{20.8}$$

ν should be positive for the additional term to be compatible (i.e., give a well-posed problem together) with the *time-dependent* system (20.1), so as to make the limits $\nu \to 0$ and $t \to \infty$ interchangeable. With $\nu > 0$, the additional term indeed represents a physical effect, namely, static molecular diffusion, which could normally be neglected.

The system (20.6) with (20.8) replacing (20.6b) is called the *augmented CNS (ACNS) system*. Its principal-part determinant is $\kappa\nu\mu^d\Delta^{d+2}$, so it is indeed elliptic. It requires the same $d+1$ boundary conditions (on $\underline{u}$ and ε) as before, plus a boundary condition on ρ (or p), on all boundaries. But the sign of ν ensures that the latter condition will affect the solution in the limit $\nu \downarrow 0$ only at points where the flow *enters* the domain. At non-entry boundaries, an artificial boundary layer (discontinuity in the limit) would be formed; but it can be avoided by using the original continuity equation (20.6b) as the extra boundary condition at such points. If *all* boundaries are such, however, we will have only *gradient* conditions on ρ along the boundaries, hence we will need an extra integral condition to uniquely determine the solution to the ACNS system, and also, in the limit, to the CNS system. This condition is usually the *total mass*, or some equivalent datum. Indeed, if the flow nowhere enters the domain, rigid walls are all around, then the total mass is determined only by the initial conditions, and therefore should be added as an extra condition to the steady-state equations.

20.1.4 Inviscid (Euler) and subprincipal operators

The inviscid case (Euler equations) is the system (20.6) with vanishing viscosities and heat conduction: $\lambda = \mu = \kappa = 0$. More precisely, the flow is inviscid (free of viscous and heat-conduction effects) where λ, μ and κ are small compared with $\rho l q$, where l is a typical length of change of $\underline{u}$ and ε. Usually there will be some particular narrow subdomains, such as boundary layers, where l will be just small enough to make the flow viscous. Thus, viscosity effects can seldom be completely neglected.

Anyway, wherever the flow is inviscid, the scale where viscosity dominates is much smaller than the scale of changes in the flow, which will later also be the scale of our grid h. So we like to isolate the terms which dominate the flow in that intermediate scale (small scale in terms of the flow geometry, but large enough to neglect viscosity and heat conduction). These are the *sub-principal* terms, defined as all the terms that are either principal or become principal when λ, μ and κ, or some of them, vanish. They form the following *sub-principal operator* L_s

$$L_s = \begin{pmatrix} Q_\mu - \overline{\lambda}\partial_{11} & \cdots & -\overline{\lambda}\partial_{1d} & 0 & 0 & \partial_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ -\overline{\lambda}\partial_{d1} & \cdots & Q_\mu - \overline{\lambda}\partial_{dd} & 0 & 0 & \partial_d \\ \rho^2\partial_1 & \cdots & \rho^2\partial_d & Q_\nu & 0 & 0 \\ p\partial_1 & \cdots & p\partial_d & 0 & Q_\kappa & 0 \\ 0 & \cdots & 0 & -p_\rho & -p_\varepsilon & 1 \end{pmatrix}, \tag{20.9}$$

where generally

$$Q_\alpha := -\underline{\nabla} \cdot (\alpha \underline{\nabla}) + \rho \underline{u} \cdot \underline{\nabla}. \tag{20.10}$$

This is the operator that should be kept in mind in the *local* processing, such as relaxation, and in the choice of discretization to be used with relaxation. The coefficients $\underline{u}$, ρ and p appearing in L_s are actually the values of some solution around which the flow is examined through principal linearization (see §3.4); they will always be derived from the *current approximate* solution $\tilde{U}$ (see §20.3.5). It can always be assumed that the current approximation is close enough to the solution, by employing continuation if necessary (see §8.3.2). The determinant of L_s , developed by its last row, is

$$\det L_s = Q_\mu^{d-1} \left\{ Q_\kappa Q_\mu \left(Q_\mu - \overline{\lambda}\Delta \right) - \left(\rho^2 p_\rho Q_\kappa + p p_\varepsilon Q_\nu \right) \Delta \right\}. \tag{20.11}$$

The *reduced (principal) operator* L_r is defined as L_s for $\lambda = \mu = \kappa = \nu = 0$, i.e., the principal part of the inviscid limit, namely,

$$L_r = \begin{pmatrix} Q_0 & \cdots & 0 & 0 & 0 & \partial_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & Q_0 & 0 & 0 & \partial_d \\ \rho^2\partial_1 & \cdots & \rho^2\partial_d & Q_0 & 0 & 0 \\ p\partial_1 & \cdots & p\partial_d & 0 & Q_0 & 0 \\ 0 & \cdots & 0 & -p_\rho & -p_\varepsilon & 1 \end{pmatrix}, \tag{20.12}$$

$$\det L_r = Q_0^d \left(Q_0^2 - \rho^2 a^2 \Delta\right) = \rho^{d+2} (\underline{u} \cdot \underline{\nabla})^d \left[(\underline{u} \cdot \underline{\nabla})^2 - a^2 \Delta\right], \qquad (20.13)$$

where $a := (p_\rho + \rho^2 p p_\varepsilon)^{\frac{1}{2}}$ is the *speed of sound.* (In the time-dependent inviscid case the operator in brackets in (20.13) is replaced by $[(\frac{\partial}{\partial t} + \underline{u} \cdot \underline{\nabla})^2 - a^2 \Delta]$, showing that a is the speed relative to the flow at which small disturbances would propagate.) The ratio $M = q/a = (\underline{u} \cdot \underline{u})^{\frac{1}{2}}/a$ is called the *Mach number.* Where $M < 1$ the flow is called *subsonic*, where $M > 1$ it is called *supersonic*, and the line where $M = 1$ is the *sonic line.* We can see from (20.13) that the steady state inviscid supersonic equations are hyperbolic, regarding the stream direction $\underline{u}$ as the time-like direction, with three families of characteristic lines and three characteristic speeds: q, $|a - q|$ and $|a + q|$. The steady-state inviscid subsonic equations are neither hyperbolic nor elliptic, and have only one family of characteristic lines, namely, the streamlines. In either case the equations are of order $d + 2$, hence require $d + 2$ boundary conditions per streamline. The only restriction on these imposed by (20.13) is that, in the subsonic case, at least one condition should be given all along the boundary (on both sides of each streamline). Actually the situation is more complicated since the flow can be partly subsonic and partly supersonic and, more importantly, acceptable solutions of the inviscid equations are only those obtainable as limits of solutions of the viscous equations. This latter requirement determines which of the boundary conditions of the full CNS equations will affect the inviscid flow away from the boundary, and which will be ignored in the inviscid limit, creating a discontinuity (boundary layer). It also determines what type of discontinuities (shocks) are permissible in the interior. The derived rules for permissible discontinuities are sometimes expressed as "entropy conditions".

Our approach here to the inviscid case will generally be to imitate the physics. Instead of deriving entropy conditions and then imposing them numerically, and instead of getting fully into the question of correct boundary conditions, we will locally (i.e., in relaxation) use a numerical scheme which contain artificial viscosity exactly analogous to the physical viscosity, thus ensuring correct selection of discontinuities. Our local processes need artificial viscosity anyway, to eliminate high-frequency errors.

The required magnitude of the artificial viscosity coefficients can be seen from $\det L_s$. They should be as effective on the scale of the meshsize h as other terms, hence, regarding each differentiation symbol as $O(h^{-1})$, the coefficients (λ,μ,ν,κ) should be chosen so as to make the order of $\det L_s$ homogeneous in h. It is easy to see that this is obtained if and only if the artificial λ, μ, ν and κ are all $O(h\rho|\underline{u}|)$, in which case $\det L_s$ is homogeneously of order h^{-d-2}.

20.1.5 Incompressible and small Mach limits

The incompressible limit is the case of vanishingly small $\partial\rho/\partial p$, or indefinitely large p_ρ. The main operator in this case is the cofactor of p_ρ in (20.9)

(reducing by one the number of equations, corresponding to the fact that ρ is no longer unknown). The resulting system is reducible: the momentum and continuity equations form a separate system of equations for $\underline{u}$ and p, easily simplified to (19.3) above. An exactly analogous situation arises if p_ε, instead of p_ρ, is large, and, more generally, whenever the Mach number is small. Thus, if one develops a multigrid solver for cases which include regions with small Mach, its discretization and relaxation should be efficient in the incompressible limit (19.3).

20.2 Stable staggered discretization

20.2.1 Discretization of the subprincipal part

The stable discretization constructed here is for use in relaxation processes, and will thus (see §2.1) be based on considerations concerning the subprincipal operator (20.9) with the *coefficients* $\underline{u}$, ρ, p, p_ρ, p_ε regarded as fixed. For all admissible values of these coefficients, the discretization and relaxation processes should be uniformly effective. In particular, they should accommodate the incompressible limit (cf. §20.1.5). We will therefore use a staggered grid as in §18.2 and 19.2, with ρ^h, ε^h and p^h (the discrete approximations to ρ, ε and p, respectively) defined at

$$x^h_k = x^h_0 + (k_1 h_1, \ldots, k_d h_d) \qquad \text{(cell centers)}, \tag{20.14}$$

and u^h_i (approximating u_i) at

$$x^{h,i}_k = x^h_k + .5\underline{h_i} \qquad (i\text{-face center}), \tag{20.15}$$

where $k = (k_1, \ldots, k_d)$ are vectors of integers, $h_1 \times \cdots \times h_d$ is the size of the grid-h cells, $\underline{h_i} = h_i \underline{e_i}$, and $\underline{e_i}$ is the i-*th* coordinate unit vector. In two dimensions ($d = 2$) the staggering is depicted in Fig. 20.1. With this staggering (20.9) can indeed be approximated by replacing ∂_i by its *short* central analog ∂^h_i,

$$\partial^h_i \Phi(\underline{x}) := \frac{1}{h_i}\left[\Phi(\underline{x} + .5\underline{h_i}) - \Phi(\underline{x} - .5\underline{h_i})\right],$$

replacing ∂_{ij} by $\partial^h_{ij} = \partial^h_i \partial^h_j$, and each Q_α by a proper approximation Q^h_α. Calling the resulting operator L^h_s, we find, similarly to (20.11), that

$$\det L^h_s = (Q^h_\mu)^{d-1}\left\{Q^h_\kappa Q^h_\mu\left(Q^h_\mu - \overline{\lambda}\Delta^h\right) - \left(\rho^2 p_\rho Q^h_\kappa + p p_\varepsilon Q^h_\nu\right)\Delta^h\right\}. \tag{20.16}$$

where $\Delta = \sum_j \partial^h_{jj}$ is the usual $(2d+1)$-point discrete Laplacian. Thus, the approximation is h-elliptic, provided Q^h_α are h-elliptic, $\alpha = \mu, \nu, \kappa$.

The approximation of these diffusion-convection operators is generally similar to that of Q^h in §19.2, with ρ/μ replacing R, with $O(h\rho|\underline{u}|)$ replacing $O(h|\underline{u}|)$, and with some modifications in case of shocks. Such modifications,

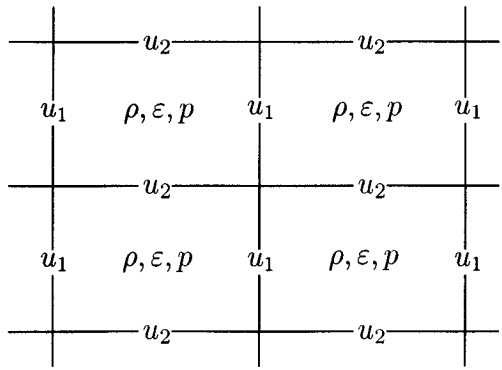

Figure 20.1. ***Grid staggering for compressible Navier-Stokes discretization.***

introduced of course in the conservative formulation, have been studied and the main emerging rule, as with boundary layers, is to avoid differencing across discontinuities: the stronger the shock, the more precisely rotated and upwinded the calculation of fluxes, hence the more weakly it straddles the shock. Where boundary layer and shocks are *not* present, we will thus approximate any Q_α at any gridpoint $\underline{x}$ by

$$Q_\alpha^h(\underline{x}) := \sum_{j=1}^{d} \left(-\partial_j^h \left(\alpha_j^h(\underline{x})\partial_j^h\right) + \overline{\rho}(\underline{x})\overline{u}_j^h(\underline{x})\partial_j^{2h}\right), \tag{20.17}$$

where $\overline{\rho}(\underline{x})$ and $\overline{u}_j^h(\underline{x})$ are central averages of ρ^h and u_j^h, respectively, over points nearest to $\underline{x}$, and

$$\alpha_j^h(\underline{x}) := \max\{\alpha, \beta h_j \rho^h |u_j^h|\}. \tag{20.18}$$

The max in (20.18) is taken in principle over all values of h_j, ρ^h and u_j^h in some neighborhood of $\underline{x}$. Usually *any* neighboring values, not necessarily exactly maximal ones, can be taken, except near *stagnation point*, where $\underline{u}$ nearly vanishes. β is an $O(1)$ parameter; $\beta = \frac{1}{2}$ normally gives upstream differencing, but slightly larger values ($\beta = .6$ or $.7$) may give better results (cf. [Bra81a]).

20.2.2 The full quasi-linear discretization

Guided by the above discretization scheme for the sub-principal part, the discrete approximation to the full CNS system (20.6) on the staggered grid

(20.14)-(20.15) will be written as

$$Q^h_\mu u^h_i - \overline{\lambda}\sum_j \partial^h_{ij} u^h_j + \partial^h_i p^h = f^h_i \text{ at } x^{h,i}_k,\ (i = 1,\dots,d) \quad (20.19a)$$

$$Q^h_\nu \rho^h + (\rho^h)^2 \sum_j \partial^h_j u^h_j = f^h_\rho \text{ at } x^h_k \quad (20.19b)$$

$$Q^h_\kappa \varepsilon^h + p^h \sum_j \partial^h_j u^h_j - B^h(\underline{u}^h) = f^h_\varepsilon \text{ at } x^h_k \quad (20.19c)$$

$$p - \overline{p}(\varepsilon, \rho) = 0 \text{ at } x^h_k, \quad (20.19d)$$

where all Q^h are defined by (20.17)–(20.18) and $B^h(\underline{u}^h)$ is the simplest central approximation to

$$B(\underline{u}) := \mu \sum_{i,j} (\partial_i u_j + \partial_j u_i)\partial_j u_i + \lambda \left(\sum_i \partial_i u_i \right)^2 . \quad (20.20)$$

The exact form of B^h is not important, since it is neither a principal nor a subprincipal term. $\underline{f}^h := (f^h_1, \dots, f^h_d, f_\rho, f_\varepsilon)$ are some local averages of $\underline{f}$; injection $\underline{f}^h = \underline{f}$ is usually used, except in some cases where this fails to give good enough approximations on the finest grid (relative to the grid-$2h$ solution, whose right-hand side is $\underline{f}^{2h} = I^{2h}_h \underline{f}^h$).

The scheme (20.19) will be used in the relaxation processes. The same scheme, but without the artificial viscosity terms ($\alpha^h_j = \alpha$ in (20.17)), will be used to calculate residuals transferred to coarse grids, thus making the overall approximation $O(h^2)$.

In the inviscid case ($\lambda, \mu, \kappa \ll \rho h|\underline{u}|$) the term with $\overline{\lambda}$ in (20.19a) and the term $B^h(\underline{u}^h)$ in (20.19c) may be omitted. They do not contribute to the h-ellipticity of the system. The resulting scheme is nothing but Euler equations with a simple form of artificial viscosity, derived from the viscous (Navier-Stokes) equations.

20.2.3 Simplified boundary conditions

At this stage of development, to separate away various algorithmic questions (cf. §4), the numerical experiments were conducted with known smooth solutions U, employing first periodic boundary conditions and then Dirichlet conditions in two dimensions ($d = 2$). In the periodic case no boundaries are actually present; gridpoint (x_1, x_2) is simply identified with gridpoint $(x_1 + 2\pi, x_2 + 2\pi)$. This enables us later to check that no slow-down is caused by boundary conditions. The Dirichlet case at this stage is the square domain $\{|x|, |y| \le \pi\}$, with $\underline{u}$, ρ and ε given on its boundary. The staggered grid is square and is placed so that the bounary of the domain coincides with cell boundaries; i.e., $h_1 = h_2 = h = 2\pi/N$ and $x^h_0 = (h/2, h/2)$ (see (20.14)).

Moreover, to simplify the code development, the Dirichlet boundary conditions are at this stage placed not exactly on the boundary but in their natural staggered-grid positions. For example, ρ and ε are specified at the cell centers immediately outside the boundary, i.e., on the lines $\{|x|, |y| \leq 2\pi + h/2\}$. This is easy to do at this stage since the numerical tests are made with known solutions U, whose values are in particular known on that staggered boundary. Ultimately, these staggered-boundary conditions will be obtained by (quadratic) extrapolation from the real boundary conditions as well as interior values; the present type of conditions is only employed in order to separate away questions as to exactly when and how this extrapolation should be made.

In the inviscid case some of the above conditions are redundant, but the code can handle this automatically (cf. §20.1.4).

20.3 Distributive relaxation for the simplified system

20.3.1 General approach to relaxation design

Since the problem at hand is not elliptic, one should not attempt obtaining "perfect smoothers" (see §3.3, §7 and end of §4.1). So the question is how to use the usual measure of relaxation performance, namely, the smoothing factor $\overline{\mu}$, in selecting the relaxation scheme. We do it by dividing the relaxation *design* into the following three stages.

(A) First, a *pointwise* (not-block) and direction-free relaxation, with low (per operation, of course, and with the other considerations of §3.2) is constructed for the *uniformly h-elliptic* operator, denoted L_e^h, obtained from L^h when sufficiently large and *isotropic* artificial viscosity terms are used. This means replacing (20.18) by

$$\alpha_j^h = \max\{\alpha, \beta\rho^h \max_l h_l |u_l^h|\}, \tag{20.21}$$

and choosing β just large enough to make excellent $\overline{\mu}$ obtainable independently of relaxation marching directions. It means that β should be appreciably larger than the minimal value $\beta = .5$ needed for stability of simple relaxation schemes. For example one can take for this purpose $\beta = 1$; any larger value of β will not substantially change the smoothing factors, and will especially not change the comparison between different relaxation schemes.

(B) Having designed the relaxation scheme, it is then actually used with the *anisotropic* artificial viscosities (20.18), rather than (20.20). Note that if the flow is not (nearly) aligned with the grid, there is no fundamental difference between the two. If the flow *is* aligned with the grid, there is one kind of high-frequency error components V which

are not deflated in the anisotropic as in the isotropic case because L^hV is much more closely singular than L_e^hV: These are the "characteristic components", i.e. high-frequency components which are smooth in the flow direction. When the flow is not aligned with the grid, neither scheme approximate these components well. If there is alignment, only the anisotropic scheme approximates them well, but exactly for this scheme and these components the pointwise relaxation is not effective. (The effectiveness of relaxation in the isotropic case is obtained for a characteristic component at the price of not really approximating its amplitude in the *differential* solution, which makes the fast convergence for this component meaningless; cf. §12). It is *meaningful* to get good smoothing for the characteristic components only if the alignment is *strong* (i.e., intended and consistent; see §3.3), and exactly then it is *possible* to do that, via block relaxation.

Thus, *the pointwise relaxation scheme, designed in terms of the isotropic artificial viscosities (20.21), is actually used with the anisotropic viscosities (20.18), and it is modified to the corresponding block scheme in the case of strong alignment.* "Corresponding" means that the same distribution matrix M^h, and a similar relaxation ordering, are maintained while the "ghost functions" w_i (see §3.7) are relaxed in blocks instead of pointwise, where the blocks are in the specific strong-alignment direction.

(C) To this basic relaxation scheme, several improvements can be added. First, the value of β can be lowered, either experimentally, or theoretically through the modified smoothing range (12.1). (As explained there (following (12.1)), if β is lowered to near its minimal stable value ($\beta = .5$) , the final result may be improved by averaging. Also note that near $\beta = .5$, downstream relaxation ordering may become badly divergent for some special *smooth* component which may not show up in one experiment but badly affect another; see §20.3.4). Then the equations may more precisely be rotated and upwinded near strong shocks, double discretization schemes may be introduced (see §10.2), etc.

In the two latter stages, (B) and (C), efficiency should of course mainly be measured not in terms of asymptotic factors, but in terms of FMG performance, whether experimentally (see §1.6) or theoretically (see §7.4, 7.5).

20.3.2 Possible relaxation scheme for inviscid flow

In the inviscid case the principal difference operator is L_r^h (cf. (20.12)), and the usual distribution operator (cf. §3.7 and §19.3) would be

$$M_r^h := \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & -\partial_1^h Q_0^h \\ 0 & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 & 0 & -\partial_d^h Q_0^h \\ 0 & \cdots & \cdots & 0 & 1 & 0 & -\rho^2 \Delta^h \\ 0 & \cdots & \cdots & 0 & 0 & 1 & -p\Delta^h \\ 0 & \cdots & \cdots & 0 & 0 & 0 & (Q_0^h)^2 \end{pmatrix}, \tag{20.22a}$$

where the last column is made of the cofactors of the last row in L_r^h, divided by their common factor $(Q_0^h)^2$. Since

$$L_r^h M_r^h = \begin{pmatrix} Q_0^h & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 & \vdots \\ 0 & \cdots & 0 & Q_0^h & 0 & 0 & 0 \\ \rho^2\partial_1 & \cdots & \cdots & \rho^2\partial_d & Q_0^h & 0 & 0 \\ p\partial_1 & \cdots & \cdots & p\partial_d & 0 & Q_0^h & 0 \\ 0 & \cdots & \cdots & 0 & -p_\rho & -p_\varepsilon & (Q_0^h)^2 - \rho^2 a^2 \Delta^h \end{pmatrix}, \tag{20.22b}$$

the relaxation process is essentially decoupled into relaxing the convection operator Q_0^h and, separately, the potential-flow operator $L_{\text{pot}}^h := (Q_0^h)^2 - \rho^2 a^2 \Delta^h$. The former has been discussed in §19.3; the latter has been studied via numerical experiments with potential flows (see remarks in §21).

For low Mach numbers the potential operator is nicely elliptic and the performance of the solver is essentially the same as in the incompressible limit (see §19, 20.1.5).

In the transonic and supersonic case, L_{pot}^h is not elliptic, so the approach outlined above (§20.3.1) is applied. Taking larger artificial viscosities shows, however, that the present approach is not optimal. This is easy to see by observing the limit case of large artificial viscosities, where $L_{\text{pot}}^h \approx (\Delta^h)^2$, i.e., the relaxed operator is essentially the biharmonic operator, for which Gauss-Seidel smoothing is relatively slow: $\overline{\mu} = .80$. Better smoothing schemes for Δ_h^2 exist (see [Bra77a, §6.2]), but they are more complicated, and not fully effective, too. The best scheme for Δ_h^2 is obtained by writing it as a system of two Poisson equations, each relaxed by red-black Gauss-Seidel, yielding $\overline{\mu}_1 = \overline{\mu}_2 = .25$ (cf. (3.2)), with operation count considerably smaller than for Gauss-Seidel relaxation of Δ_h^2. However, this requires the introduction of an auxiliary function and some special care near boundaries, and is much less convenient in case of the actual op-

erator L^h_{pot}, especially in the present framework of the overall distributive relaxation.

Two strategies to avoid this trouble are (A) Introduce the auxiliary function only during relaxation and use an operator product relaxation scheme [LB04], with additional local relaxation sweeps near boundaries (see §5.7); or (B) Choose another distribution operator M^h, where care is taken not to distribute as far as to create the need to relax the *square* of Q^h_0. Moreover, the new approach, to be described next, is applicable to the general CNS system, whereas (20.22) applies to the inviscid limit only, since it assumes $Q_\mu = Q_\kappa = Q_\nu$.

20.3.3 Distributed collective Gauss-Seidel

In view of the subprincipal operator L^h_s (cf. (20.9)), the distribution operator

$$M^h_s := \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & -\partial^h_1 \\ 0 & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 & 0 & -\partial^h_d \\ 0 & \cdots & \cdots & 0 & 1 & 0 & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 1 & 0 \\ \overline{\lambda}\partial^h_1 & \cdots & \cdots & \overline{\lambda}\partial^h_d & 0 & 0 & Q^h_{\mu+\overline{\lambda}} \end{pmatrix} \tag{20.23}$$

yields

$$L^h_s M^h_s := \begin{pmatrix} Q^h_\mu - \overline{\lambda}\partial^h_{11} & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & 0 & \vdots \\ 0 & \cdots & 0 & Q^h_\mu - \overline{\lambda}\partial^h_{dd} & 0 & 0 & 0 \\ \rho^2\partial_1 & \cdots & \cdots & \rho^2\partial_d & Q^h_\nu & 0 & -\rho^2\Delta^h \\ p\partial_1 & \cdots & \cdots & p\partial_d & 0 & Q^h_\kappa & -p\Delta^h \\ \overline{\lambda}\partial^h_1 & \cdots & \cdots & \overline{\lambda}\partial^h_d & -p_\rho & -p_\varepsilon & Q^h_{\mu+\overline{\lambda}} \end{pmatrix} \tag{20.24}$$

With few operations expended on distribution (relative to those expended on calculating residuals) the relaxation is thus "geographically" decoupled: each of the $d+1$ uniform grids composing our staggered grid (cf. (20.14)-(20.15)) is separately relaxed (in terms of the ghost functions; see §3.7). The relaxation for each i-face-centers grid is a simple Gauss-Seidel with the operator $Q^h_\mu - \overline{\lambda}\partial^h_{ii}$, which actually behaves better (has stronger h-ellipticity) than the convection-diffusion operator Q^h_μ. The relaxation of

the cell-centers grid is a relaxation on the 3×3 system

$$\begin{pmatrix} Q_\nu^h & 0 & -\rho^2\Delta^h \\ 0 & Q_\kappa^h & -p\Delta^h \\ -p_\rho & -p_\varepsilon & Q_{\mu+\overline{\lambda}}^h \end{pmatrix}. \tag{20.25}$$

We could relax this system itself by distributive relaxation. But this would yield in the inviscid case the relaxation of §20.3.2, which we have rejected, and in the viscous case would be worse yet (requiring higher order distribution, because $Q_\nu \neq Q_\kappa \neq Q_{\mu+\overline{\lambda}}$). Instead, we relax (20.25) by collective Gauss-Seidel (CGS), i.e., all the three equations defined at each cell center are relaxed simultaneously. This means a solution of a 3×3 linear system, with two convenient zeros already in it, which can be done in 13 operations, a small number compared with the work of calculating the three residuals at each cell center.

20.3.4 Relaxation ordering and smoothing rates

The relaxation of §20.3.3 is examined through the approach of §20.3.1, i.e., by calculating smoothing factors for the *isotropic* artificial viscosity (20.21). Direction-free robust schemes are sought. For such schemes $\overline{\mu}$ always improves (decreases) with increased physical viscosities $(\mu, \overline{\lambda}, \kappa, \nu)$. (The only case in which $\overline{\mu}$ *increases* with increased viscosities is the special case of large Mach numbers and downstream relaxation, a case which is not direction-free and which is also ruled out below for other reasons.) Hence, examined here is mainly the inviscid case $\mu = \overline{\lambda} = \kappa = \nu = 0$.

By (20.24), the relaxation can be performed as $d+1$ separate passes, d of them with the operator Q_0^h, and one with the system (20.25). Hence $\overline{\mu} = \max\{\overline{\mu}^Q, \overline{\mu}'\}$, where $\overline{\mu}^Q$ is the smoothing factor of relaxing Q_0^h and $\overline{\mu}'$ is the smoothing factor in the collective relaxation of (20.25). Both.depend on the order in which the corresponding passes are made, on the value of β in (20.21), and on the direction of $\underline{u}$, and $\overline{\mu}'$ also depends on the Mach number M. This dependence is shown in Table 20.1. Notice that, except for RB(2), $\overline{\mu} = \overline{\mu}' \geq \overline{\mu}^Q$, and in most cases $\overline{\mu} \approx \overline{\mu}' \approx \overline{\mu}^Q$ (assuming the same ordering is used in all passes). This results from the fact that Q_0^h is a divisor of the determinant of (20.25).

The table shows that it very efficient to use Red-Black (RB) ordering in all passes. First, because of its usual advantage of being vectorizable (cf. §3.6). Secondly, for sufficiently large β (and hence also for viscous flows), RB smoothing rates are superior to all others. Only when β approaches its minimal value (e.g., for $\beta \leq 1$), downstream ordering (e.g., lexicographic ordering in case all $u_i \geq 0$) shows better smoothing rates. But this is not direction-free: it is complicated to maintain downstream ordering in case the flow directions (i.e., the signs of $u_1, \ldots, u_d$) change with location. Also, the components for which RB is not so good at smaller β

are exactly the characteristic components one needs not care about (see §20.3.1).

Moreover, the main advantage of the downstream relaxation is shown as *super-fast smoothing* in the case of small β and large M; indeed, $\overline{\mu} \to 0$ as $\beta \searrow .5$ and $M \to \infty$. But this small $\overline{\mu}$ shows only the behavior of *high*-frequency components. There are *low*-frequency ones for which exactly this relaxation badly diverges: the amplification is about $2^{-\frac{1}{2}}M$ for a component $(\theta_1, \theta_2) \approx 2^{-\frac{1}{2}}M^{-1}(u_1, u_2)/q$. This does not happen when the slower schemes are used, such as RB, or even the downstream ordering with larger β ($\beta \geq .8$, say): For these later schemes some low-frequency components may still diverge, but the divergence is slow and easily corrected by the multigrid coarse-grid corrections.

Table 20.1 also shows *symmetric ordering* (SGS) to yield a very efficient smoother. The bad behavior of *low* frequencies for large M and $\beta \approx .5$ is still theoretically present here, although more weakly. One can eliminate this trouble simply by switching to larger β (e.g., $\beta = 1$) wherever M is large and the relaxation marching happens to be downstream. In fact, there is no reason to use the same β at the same location in all passes, especially as still another value ($\beta = 0$) is used for fine-to-coarse residual transfers.

20.3.5 Summary: relaxation of the full system

The scheme outlined above for the subprincipal part L_s^h easily translates to the following relaxation procedure for the full quasi-linear system (20.19). We denote by $\tilde{U}^h = (\tilde{\underline{u}}^h, \tilde{\rho}^h, \tilde{\varepsilon}^h, \tilde{p}^h)$ the dynamic approximation to the solution $U = (\underline{u}, \rho, \varepsilon, p)$, i.e., the approximation just prior to any described relaxation step, while $U^h = (\underline{u}^h, \rho^h, \varepsilon^h, p^h)$ will denote the exact solution of the stable difference equations (20.19). $r^h = (r_1^h, \ldots, r_d^h, r_\rho^h, r_\varepsilon^h, r_p^h)$ will denote the dynamic residuals, i.e. the left-hand sides of (20.19) applied to $\tilde{U}^h$ instead of U^h and subtracted from the right-hand sides. Thus, at each step of each relaxation pass, $\tilde{U}^h$ and r^h change.

The relaxation steps are first described for the case where there is no strong alignment between the grid and the flow direction.

A relaxation sweep consists of $d+1$ passes. The recommended ordering within each pass is either the red-black ordering (which we used throughout our numerical experiments) or the symmetric lexicographic (SGS).

First, one pass is made for each momentum equation. The *i-th* equation at the point $x_k^{h,i}$ is relaxed by the replacements

$$\tilde{u}_i^h \leftarrow \tilde{u}_i^h + \psi_{i,k}^h \tag{20.26a}$$

$$\tilde{p}^h \leftarrow \tilde{p}^h + \overline{\lambda}\partial_i^h \psi_{i,k}^h, \tag{20.26b}$$

where $\psi_{i,k}^h$ is a function defined at all i-face centers (see (20.15)), vanishing on all of them except at the relaxed point $x_k^{h,i}$. Its value at $x_k^{h,i}$ is determined

Relax. Order	β / M	$(u_1,u_2)=(1,1)$ 2.0	1.0	0.5	$(u_1,u_2)=(0,1)$ 2.0	1.0	0.5
RB(1)	*	.27	.33	1.00	.27	.31	.50
	0.0–0.1	.27	.43	1.00	.27	.31	.50
	1.0–5.0	.40–.41	.46–.48	1.00	.27–.29	.33–.36	.50–.55
RB(2)	*	.39	.50	1.00	.36	.39	.42
	0.0–0.1	.27	.65	1.00	.27	.31	.50
	1.0–5.0	.33–.35	.41–.42	1.00	.27–.29	.33–.39	.50–.55
Lex+	*	.38	.25	0	.46	.42	.47
	0.0–1.0	.44–.50	.42–.50	.42–.50	.48–.50	.47–.50	.49–.50
	2.0	.41	.35	.24	.47	.45	.48
	5.0	.39	.29	.056	.46	.44	.50
Lex−	*	.63	.75	1.00	.58	.66	.72
	0.0–5.0	.63	.75–.76	1.00	.58	.66–.67	.84–.86
Lex±	*	.54	.63	1.00	.58	.66	.72
	0.0–5.0	.54–.55	.63–.64	1.00	.58	.66–.67	.84–.86
SGS	*	.48	.42	0	.50	.49	.49
	0.0–0.1	.50	.50	.50	.50	.50	.50
	1.0	.51	.54	.60	.51	.54	.60
	5.0	.49	.46	.24	.50	.51	.53

Table 20.1. ***Smoothing factors for two-dimensional Euler equations.***

$\overline{\mu} = \max\{\overline{\mu}^Q, \overline{\mu}'\}$. *In the rows where* * *stands for* M, $\overline{\mu}^Q$ *is displayed; in the others,* $\overline{\mu}'$ *is displayed.* RB(i) *is red-black relaxation ordering with* i *sweeps per cycle (*$\overline{\mu}_i$ *derived by (3.2)).* Lex+ *is lexicographic ordering,* Lex− *is reversed lexicographic,* Lex± *is lexicographic with only the* y *coordinate reversed, and SGS is Symmetric GS (a* Lex+ *alternating with a* Lex−, $\overline{\mu}$ *calculated per pass).*

so that

$$\left(Q_\mu^h - \overline{\lambda}\partial_{ii}^h\right)\psi_{i,k}^h(x_k^{h,i}) = r_i^h(x_k^{h,i}), \tag{20.27}$$

where the coefficients of Q_μ^h are evaluated at $x_k^{h,i}$ (see (20.17)), based on $\tilde{U}^h$. Neglecting non-principal effects, this means that after the changes (20.26), the *i-th* momentum equation (20.19a) at $x_k^{h,i}$ will be satisfied.

Then, a pass is made on cell centers. The three equations defined at

the cell center x_k^h are relaxed simultaneously by replacements of the form

$$\tilde{u}_i^h \leftarrow \tilde{u}_i^h - \partial_i^h \Phi_k^h, \qquad (i = 1, \ldots, d) \tag{20.28a}$$

$$\tilde{p}^h \leftarrow \tilde{p}^h + Q^h_{\mu+\overline{\lambda}} \Phi_k^h \tag{20.28b}$$

$$\tilde{\rho}^h \leftarrow \tilde{\rho}^h + \hat{\rho}_k^h \tag{20.28c}$$

$$\tilde{\varepsilon}^h \leftarrow \tilde{\varepsilon}^h + \hat{\varepsilon}_k^h, \tag{20.28d}$$

where Φ_k^h, $\hat{\varepsilon}_k^h$ and $\hat{\rho}_k^h$ are functions defined at cell centers (see (20.14)) vanishing everywhere except at the relaxed point x_k^h. Their values at x_k^h are chosen so as to satisfy the three equations

$$\left(\tilde{Q}_\nu^h \hat{\rho}_k^h - \tilde{\rho}^2 \Delta^h \Phi_k^h\right)(x_k^h) = r_\rho^h(x_k^h) \tag{20.29a}$$

$$\left(\tilde{Q}_\kappa^h \hat{\varepsilon}_k^h - \tilde{p} \Delta^h \Phi_k^h\right)(x_k^h) = r_\varepsilon^h(x_k^h) \tag{20.29b}$$

$$\left(-\tilde{p}_\rho \hat{\rho}_k^h - \tilde{p}_\varepsilon \hat{\varepsilon}_k^h + \tilde{Q}^h_{\mu+\overline{\lambda}} \Phi_k^h\right)(x_k^h) = r_p^h(x_k^h). \tag{20.29c}$$

where $\tilde{\rho}$, $\tilde{p}$, $\tilde{p}_\rho$, $\tilde{p}_\varepsilon$ and the coefficients of $\tilde{Q}_\nu^h$, $\tilde{Q}_\kappa^h$ and $\tilde{Q}^h_{\mu+\overline{\lambda}}$ are all evaluated at x_k^h, based on $\tilde{U}^h$.

The functions $\psi_{i,k}^h$, Φ_k^h, $\hat{\rho}_k^h$ and $\hat{\varepsilon}_k^h$ mentioned above do not actually appear in the program, of course. They just serve to concisely describe the relaxation steps.

Instead of the separate $d+1$ passes just described, they could of course be merged in any desired fashion.

In the case of strong alignment, i.e., if one grid direction nearly coincides with the flow direction throughout a substantial subdomain, relaxation should be done in the corresponding blocks (lines in the alignment direction). This means that each unknown function (ψ^h in the case of (20.26), and Φ^h, $\hat{\rho}^h$ and $\hat{\varepsilon}^h$, in the case of (20.28)) are free (i.e., not fixed to be zero) not just at one gridpoint at a time, but at all gridpoints of the relaxed block, thus giving exactly the number of parameters needed to simultaneously satisfy the equations (20.27) or (20.29)) at all gridpoints of the block. In the case of *plane alignment* (cf. §19.2) it is advisable to coarsen the grid only in that plane directions (see §4.2.1), in which case no block relaxation will be needed.

The total work of the relaxation sweep is only a fraction (20% or so) larger than the work of *expressing* the differences equations (20.19), or calculating their residuals, at all gridpoints.

20.4 Multigrid procedures

The grids, their relative positions and the interpolation procedures between them are generally as for Stokes equations (§18.4), with ρ and ε transferred similarly to p, and the residuals of the energy and state equations transferred similarly to the residuals of the continuity equations. Because of the

nonlinearity, FAS is of course used (see §8), and the full weighting (18.14) is preferred over (18.13) in the fine-to-coarse transfers of both the velocity u_i and the *i-th* momentum residual.

An option for *double discretization* (cf. §10.2) is included. Namely, the artificial viscosity terms may be omitted in calculating the residuals to be transferred from any grid to the next coarser one. More generally, the artificial-viscosity coefficient β (see (20.18)), may have different values at different stages of the algorithm.

Whatever the value of β, in any stage of the solver, the discretization should also attempt not to straddle strong discontinuities, by calculating one-sided fluxes.

Note that in FAS, Dirichlet boundary conditions appear the same on all grids, whether the grid is the currently-finest or a correction grid. This can also be the case in the simplified, staggered boundary conditions (see §20.2.3): The exact differential solution is enforced at the staggered boundary points of the coarser, correction grid H, too. On those points, the value of $\hat{I}_h^H \tilde{U}^h$ is also defined to be the exact solution, hence, for the purpose of interpolating back to the finer grid (like (8.6)), the difference $\tilde{U}^H - \hat{I}_h^H \tilde{U}^h$ is defined to be zero at the staggered boundary points.

As in other non-elliptic cases, W cycles are generally preferred to V cycles (see §6.2).