Chapter 18

Steady-State Stokes Equations

18.1 The differential problem

As a prelude to the treatment of the full Navier-Stokes equations, we consider now the steady-state Stokes equations in a d-dimensional domain

$$\underline{\nabla} \cdot \underline{u} = F_0 \tag{18.1a}$$

$$-\Delta \underline{u} + \underline{\nabla} p = \underline{F}, \tag{18.1b}$$

where $\underline{u} = (u_1, \ldots, u_d)$ represents the velocity of a fluid and p represents the pressure, $\underline{\nabla} = (\partial_1, \ldots, \partial_d)$ is the gradient operator, $\Delta = \partial_1^2 + \cdots + \partial_d^2$ is the Laplace operator, and F_0 and $\underline{F} = (F_1, \ldots, F_d)$ are given forcing functions. (18.1) are the equations of "creeping" flows (vanishing Reynolds number). (18.1a) is the "continuity equation" (usually with vanishing source term: $F_0 = 0$), and (18.1b) is the vector of d momentum equations.

The matrix-operator form of (18.1) is

$$L \begin{pmatrix} p \\ u_1 \\ \vdots \\ u_d \end{pmatrix} := \begin{pmatrix} 0 & \partial_1 & \cdots & \cdots & \partial_d \\ \partial_1 & -\Delta & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \partial_d & 0 & \cdots & 0 & -\Delta \end{pmatrix} \begin{pmatrix} p \\ u_1 \\ \vdots \\ u_d \end{pmatrix} = \begin{pmatrix} F_0 \\ F_1 \\ \vdots \\ F_d \end{pmatrix} \tag{18.2}$$

and the operator determinant is

$$\det(L) = (-\Delta)^d. \tag{18.3}$$

Hence (18.1) is a $2d$-order elliptic system and will require d boundary conditions. These are usually given by specifying the velocity on the boundary

$$\underline{u}(\underline{x}) = \underline{G}(\underline{x}), \qquad (\underline{x} \in \partial\Omega) \tag{18.4}$$

where $\underline{G} := (G_1, \ldots, G_d)$ and $\underline{x} := (x_1, \ldots, x_d)$.

Equations (18.1) with the boundary conditions (18.4) constitute a well-posed problem, provided the compatibility condition, obtained from (18.1a) and the divergence theorem,

$$\int_\Omega F_0 d\underline{x} = \int_{\partial\Omega} \underline{G} \cdot \underline{d\sigma} \tag{18.5}$$

is satisfied, where $\underline{d\sigma}$ is boundary element multiplying an outward normal unit vector.

18.2 Finite-difference equations

By arguments similar to those in §17.2 and 17.6 we find it best to discretize (18.1) on a staggered grid (but see §18.6). Such a grid, in the two-dimensional case, is shown in Fig. 18.1.

In the general d-dimensional case, the grid hyperplanes (planes if $d = 3$, lines if $d = 2$) define cells, each cell with $2d$ faces. The discrete velocity $u_j^h(\underline{x})$ values are defined at centers of j-faces, i.e., faces perpendicular to the j-*th* coordinate. The discrete pressure p^h and its computed approximation p^h are located at cell centers. The discrete approximation to (18.1) can then be written as

$$\sum_{j=1}^{d} \partial_j^h u_j^h = F_0^h \qquad \text{at cell centers} \tag{18.6a}$$

$$-\Delta^h u_j^h + \partial_j^h p = F_j^h \quad \text{at centers of } j\text{-faces}, \quad (j = 1, \ldots, d) \tag{18.6b}$$

where $\partial_j^h \Phi(\underline{x}) := \frac{1}{h}[\Phi(\underline{x} + \frac{1}{2}\underline{h_j}) - \Phi(\underline{x} - \frac{1}{2}\underline{h_j})]$, $\underline{h_j}$ is h times the unit vector in the direction x_j, and the discrete approximation Δ^h to the Laplace operator is the usual $(2d+1)$-point approximation $\sum_{j=1}^{d} (\partial_j^h)^2$. For a point $\underline{x}$ near a boundary, however, $\Delta^h u_j^h(\underline{x})$ may involve an exterior value $u_j^h(\underline{x}^e)$. This value is defined by quadratic extrapolation from $u_j^h(2\underline{x} - \underline{x}^e)$, $u_j^h(\underline{x})$ and $u_j^h(\underline{x}^b) = G_j^h(\underline{x}^b)$, where $\underline{x}^b$ is a boundary point on the segment $(\underline{x}, \underline{x}^e)$. This definition is used to eliminate the exterior value from $\Delta^h u_j^h(\underline{x})$, so that the discrete Laplacian is modified and includes a boundary value of u_j^h.

The matrix operator of (18.6) is

$$L^h := \begin{pmatrix} 0 & \partial_1^h & \cdots & \cdots & \partial_d^h \\ \partial_1^h & -\Delta^h & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \partial_d^h & 0 & \cdots & 0 & -\Delta^h \end{pmatrix}, \tag{18.7}$$

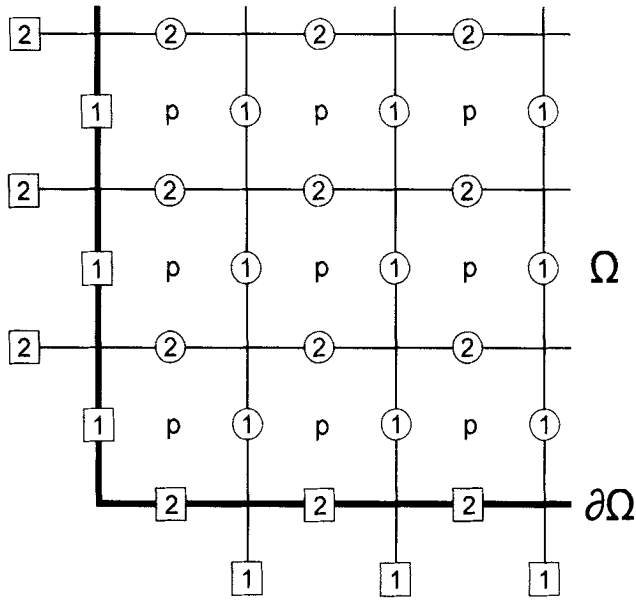

Figure 18.1. ***Discretization of two-dimensional Stokes equations.***

A typical part of the grid is shown. The discrete pressure p^h is defined at cell enters (p). The discrete velocity u_1^h is defined at centers of vertical links (① = interior centers; 1 = boundary and exterior centers), and u_2^h is defined at centers of horizontal links (② and 2). The discrete continuity equations are centered at cell centers (p). The j-th momentum equation is centered at interior values of u_j^h (ⓙ), $j = 1, 2$. The exterior values of u_1^h and u_2^h (at 1 and 2, respectively, but not on the boundary) are fictitious.

hence $\det(L^h) = (-\Delta^h)^d$ and its symbol is

$$\tilde{\det} L^h(\underline{\theta}) = h^{-2d} \left\{ \sum_{j=1}^{d} \left(2 \sin \frac{\theta_j}{2} \right)^2 \right\}^d , \tag{18.8}$$

which is positive for $0 < |\underline{\theta}| \leq \pi$. The difference system (18.6) is therefore h-elliptic (see §2.1) and even R-elliptic [BD79, §3.6].

The boundary condition (18.4) is approximated by the above way for treating boundary and exterior values of u_j^h. For simplicity, consider the case of domains whose boundary is contained in grid lines (or grid planes). In this case the velocity normal to the boundary is conveniently defined at the center of boundary faces, and the discrete analog to (18.5) is naturally

written as

$$\sum_{\underline{x}} F_0^h(\underline{x}) = \sum_{\underline{y}} G_n^h(\underline{y}), \tag{18.9}$$

where $\underline{x}$ runs over all cell centers, $\underline{y}$ runs over all centers of boundary faces, and $G_n^h(\underline{y})$ is the (given) normal velocity at $\underline{y}$.

Theorem 18.1. *The discrete Stokes equations (18.6), with exterior and boundary values determined by the boundary conditions as above, have a unique solution, up to an additive constant in p^h, if and only if (18.9) is satisfied.*

The proof is simple. The number of equations is the same as the number of unknowns, since for each interior $u_j^h(\underline{x})$ there corresponds an equation (18.6b) at $\underline{x}$, and for each unknown $p^h(\underline{y})$ there corresponds an equation (18.6a) at $\underline{y}$. The pressure values p^h are determined only up to an additive constant, but, on the other hand, the equations are dependent; summing (18.6a) over all cell centers we get (18.9). That is to say, if (18.9) is not satisfied we get a contradiction. If (18.9) *is* satisfied, it is enough to show that in the homogeneous case ($F_0^h \equiv 0$, $\underline{F}^h \equiv 0$, $\underline{G}^h \equiv 0$), the only solution is the trivial one ($\underline{u}^h \equiv 0$, $p^h \equiv$ constant). Indeed, if $\underline{F}^h \equiv 0$, it is easy to see from (18.6b) that

$$\begin{aligned} 0 &= \textstyle\sum_{j=1}^d \sum_1^j \left[-\Delta^h u_j^h(\underline{x}) + \partial_j^h p^h(\underline{x})\right] u_j^h(\underline{x}) \\ &= \textstyle\sum_{j=1}^d h^{-2} \sum_2^j \left[u_j^h(\underline{x}) - u_j^h(\underline{y})\right]^2 + \\ &\quad \textstyle\sum_{j=1}^d h^{-2} \sum_3^j \left[u_j^h(\underline{x}) - u_j^h(\underline{z})\right] u_j^h(\underline{x}) - \\ &\quad \textstyle\sum_4 p^h(\underline{x}) \sum_{j=1}^d \partial_j^h u_j^h(\underline{x}), \end{aligned}$$

where the point $\underline{x}$ in $\sum_1^j$ runs over all interior positions of of $u_j^h(x)$ (points ⓙ in Fig. 18.1, $j = 1, 2$); the pair $\{\underline{x}, \underline{y}\}$ in $\sum_2^j$ runs over all pairs of neighboring interior positions of u_j^h; the pair $\{\underline{x}, \underline{z}\}$ in $\sum_3^j$ runs over all pairs of neighboring positions u_j^h, with $\underline{x}$ being an interior position (ⓙ in Fig. 18.1, $j = 1, 2$) and $\underline{z}$ being a boundary or exterior position (j in Fig. 18.1); and $\underline{x}$ in $\sum_4$ runs over all cell centers (p in Fig. 18.1), including boundary cells due to the vanishing of the boundary conditions. The term with $\sum_4$ vanishes by (18.6a), since $F_0^h \equiv 0$. In the $\sum_3^j$ term, either $\underline{z}$ is a boundary point, therefore $u(\underline{z}) = 0$ and the term is non-negative; or $\underline{z}$ is an exterior point. By the definition of exterior values, we get (for $G^h \equiv 0$) $u_j^h(\underline{z}) = 2u_j^h(\underline{x}) - \frac{1}{3}u_j^h(\underline{y})$, where $\underline{y}$ is the interior neighbor of $\underline{x}$ opposite to $\underline{z}$. Hence,

$$\sum_{j=1}^d \left\{ \sum_5^j \left[u_j^h(\underline{x}) - u_j^h(\underline{y})\right]^2 + \sum_3^j \left[3u_j^h(\underline{x})^2 + u_j^h(\underline{y})^2 - \frac{7}{3}u_j^h(\underline{x})u_j^h(\underline{y})\right] \right\} \leq 0,$$

where $\sum_5^j$ runs as $\sum_2^j$ does, except for terms added to $\sum_3^j$. This form is positive definite, hence $u_j^h \equiv 0$. By (18.6b), $p^h \equiv$ const..

18.3 Distributive relaxation

Designing the relaxation scheme for (18.7) by the general approach of §3.7, note that L^h is almost triangular. To triangularize it, it is enough to transform its first column by multiplying L^h on the right with

$$M^h = \begin{pmatrix} -\Delta^h & 0 & \cdots & \cdots & 0 \\ -\partial_1^h & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -\partial_d^h & 0 & \cdots & 0 & 1 \end{pmatrix}, \tag{18.10}$$

(The first column of M^h could be obtained as the cofactors of the first row of L^h, divided by their maximal common divisor $(-\Delta^h)^{d-1}$.) This distribution operator M^h implies the following relaxation scheme.

First, for each $1 \le j \le d$, the *j-th* momentum equation (18.6b) is relaxed by scanning, in some prescribed order (RB ordering is best), all the interior j-face centers, changing at each such point $\underline{x}$ the current value $\tilde{u}_j^h(\underline{x})$ so as to satisfy the momentum equation at $\underline{x}$. Then, the continuity equation (18.6a) is relaxed by scanning the cell centers (preferably in RB ordering), introducing at each such center $\underline{x}$ the following changes:

$$\tilde{u}_j^h(\underline{x} + \tfrac{1}{2}\underline{h_j}) \leftarrow \tilde{u}_j^h(\underline{x} + \tfrac{1}{2}\underline{h_j}) + \delta, \quad (j = 1, \ldots, d), \tag{18.11a}$$

$$\tilde{u}_j^h(\underline{x} - \tfrac{1}{2}\underline{h_j}) \leftarrow \tilde{u}_j^h(\underline{x} - \tfrac{1}{2}\underline{h_j}) - \delta, \quad (j = 1, \ldots, d), \tag{18.11b}$$

$$\tilde{p}^h(\underline{x}) \leftarrow \tilde{p}^h(\underline{x}) + \frac{2d}{h}\delta, \tag{18.11c}$$

$$\tilde{p}^h(\underline{x} + \underline{h_j}) \leftarrow \tilde{p}^h(\underline{x} + \underline{h_j}) - \frac{1}{h}\delta, \quad (j = 1, \ldots, d), \tag{18.11d}$$

$$\tilde{p}^h(\underline{x} - \underline{h_j}) \leftarrow \tilde{p}^h(\underline{x} - \underline{h_j}) - \frac{1}{h}\delta, \quad (j = 1, \ldots, d), \tag{18.11e}$$

where

$$\delta = \frac{h}{2d} r_0^h(\underline{x}) = \frac{h}{2d} \left(F_0^h - \sum_{j=1}^{d} \partial_j^h \tilde{u}_j^h \right)_{\text{before the changes}}, \tag{18.12}$$

so that *after* these changes, the new residual $r_0^h(\underline{x})$ vanishes.

The relaxation of the continuity equation, and its modification near a boundary, are shown in Fig. 18.2 on the next page.

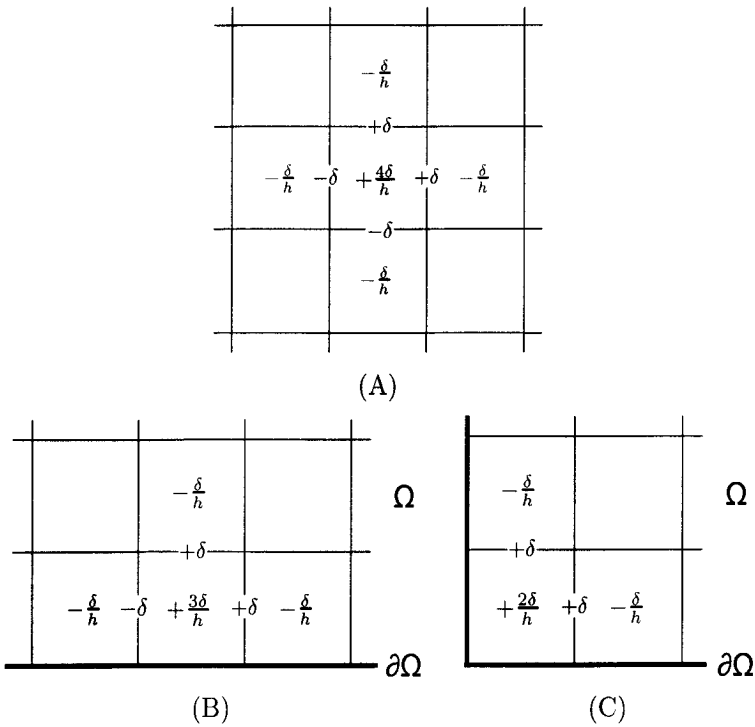

Figure 18.2. ***Continuity-equation relaxation step in two-dimensional Stokes equations.***

(A) The central cell is relaxed by 9 simultaneous changes. The amount of change is displayed at the position of the changed variable (cf. Fig. 18.1). $\delta = hr_0^h(\underline{x})/4$, where $r_0^h(\underline{x})$ is the dynamic residual at the relaxed cell.

(B) Configuration of changes in a boundary cell. $\delta = hr_0^h(\underline{x})/3$.

(C) Configuration of changes in a corner cell. $\delta = hr_0^h(\underline{x})/2$.

Observe that *near* the boundary, unlike the situation away from it, the relaxation of the continuity equation at a point does introduce slight changes in neighboring momentum residuals, as if locally contradicting the triangularity of L^hM^h. But these slight changes do not cause later (when the momentum equations are relaxed) significant "feed-back", i.e., too large changes back in r_0^h, because near the boundary feed-back changes are partly "absorbed" by the boundary.

The **smoothing factor** is the same as for GS relaxation for L^hM^h, i.e., the same as for Δ^h. Hence, in two dimensions, for lexicographic order-

ing $\overline{\mu} = .5$ and for RB ordering, if used in all three passes, $\overline{\mu}_1 = \overline{\mu}_2 = .25$ and $\overline{\mu}_3 = .32$ (cf. (3.2)). For $d = 3$ and lexicographic ordering, $\overline{\mu} = .563$.

18.4 Multi-grid procedures

For multi-grid processing of Stokes equations we use a sequence of grids (levels) with meshsizes $h_1, \ldots, h_M$, where $h_{k+1} = \frac{1}{2}h_k$, and where the grid lines (or grid planes) of level k are every other grid line (plane) of level $k+1$. Hence, each cell of level k is the union of 2^d cells of level $k+1$. In two dimensions ($d = 2$) the configuration is shown in Fig. 18.3. Instead of $\underline{F}^h$, $\underline{u}^h$, p^h, $\underline{r}^h$, μ_j^h and ∂_j^h used in §18.2–18.3 and in (17.7), the discrete functions and operators on the *k-th* level are now denoted by $\underline{F}^k$, $\underline{u}^k$, p^k, $\underline{r}^k$, μ_j^k and ∂_j^k, respectively.

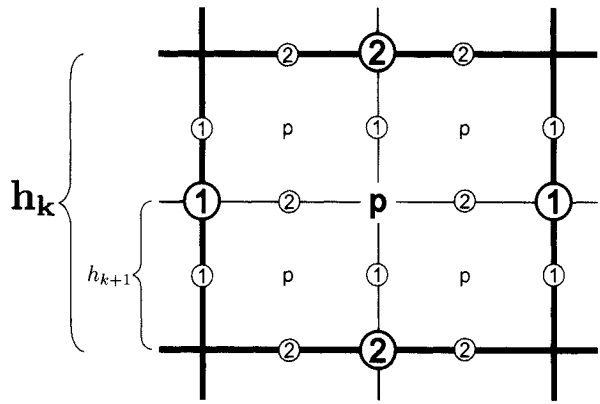

Figure 18.3. ***A coarse-grid cell divided into fine-grid cells.*** *The notation of Fig. 18.1 is used, with heavy type for the coarse grid and light type for the fine grid.*

We have solved Stokes equations both with the Correction Scheme (CS) and with the Full-Approximation Scheme (FAS), getting of course identical results. We describe here the procedures in terms of FAS, since CS is not extendible to the nonlinear Navier-Stokes equations.

Coarse-to-fine interpolations. In the FMG algorithm, to obtain residuals smaller than their truncation errors, the *first* coarse-to-fine interpolation has to be of order at least four for the velocities and at least three for the pressure (see §7.1). The design of such interpolations is straightforward, although it turns out somewhat cumbersome near boundaries. The coarse-to-fine interpolation of *corrections* has to be of orders at least two for the velocities and one for the pressure (see §4.3). We used bilinear (i.e., order two) interpolations for both.

The **fine-to-coarse transfers** are made by averaging. For the FAS transfer of u_j^{k+1} we can use the same averaging as for r_j^{k+1}, $(j = 1, \ldots, d)$,

which can be either the minimal-operation transfer

$$I_{k+1}^k r_j^{k+1} = \mu_1^{k+1} \cdots \widehat{\mu_j^{k+1}} \cdots \mu_d^{k+1} r_j^{k+1}, \quad (j = 1, \ldots, d), \tag{18.13}$$

or the full weighting

$$I_{k+1}^k r_j^{k+1} = \mu_j^{k+1} \mu_1^{k+1} \cdots \mu_d^{k+1} r_j^{k+1}, \quad (j = 1, \ldots, d), \tag{18.14}$$

where the hat in (18.13) indicates the term to be skipped in the sequence. The residual-weighting (18.13) is less expensive than (18.14), especially since it requires calculating only one half of the fine-grid residuals. But (18.14) is more reliable in the nonlinear case and near boundaries, since it is "full" (see §4.4).

The FAS transfer of $\tilde{p}^{k+1}$ can be made with the same weighting as the transfer of the continuity-equation residuals

$$I_{k+1}^k r_0^{k+1} = \mu_1^{k+1} \cdots \mu_d^{k+1} r_0^{k+1}, \tag{18.15}$$

which is both simplest and "full". In fact, if the minimal-operations transfer (18.13) is used for the velocities $\tilde{u}_j^{k+1}$, then (18.15) need not really be calculated: If the FAS continuity equation at level k is written in the form

$$\sum_{j=1}^{d} \partial_j^k u_j^k = f_0^k \tag{18.16a}$$

(where $f_0^l = F_0^l$ at the currently finest level l), it is easy to see that (18.15) is equivalent to

$$f_0^k = \mu_1^{k+1} \cdots \mu_d^{k+1} f_0^{k+1}, \quad (k < l), \tag{18.16b}$$

which does not depend on the current approximation $\underline{\tilde{u}}^{k+1}$.

The compatibility condition (18.9) is automatically obtained (up to round-off errors) on all levels provided it holds on the finest one. This results directly from (18.15).

18.5 Numerical results

Our early (1978) experiments with the above procedures, in various cycling and FMG algorithms, on two-dimensional rectangular domains (chosen only for programming simplicity), are described in [BD79] and in more details in [Din79]. The program itself is available [MUG84]. The experiments were not optimal because we have used lexicographic instead of RB ordering in relaxation (RB was used in our recent experiments with non staggered grids; see §18.6). The asymptotic convergence rates were 20% slower than those predicted by the smoothing rate $\overline{\mu}$, but not more than 6% slower than predicted by the two-level analysis. The fact that the smoothing rate is not fully exploited indicates that better convergence rates may be

obtainable with better inter-grid transfers, but this does not seem to worth the extra work: The obtained rates are good enough to give a solution below truncation errors by an FMG algorithm with one V(2,1) cycle per level (cf. Fig. 1.2). With RB relaxation, V(1,1) should already suffice (see §18.6).

18.6 Non-staggered grids

Having received complaints about the inconvenience of staggered grids despite their advantages (see §17.6), we later experimented with Stokes (and Navier-Stokes) solvers on conventional, non-staggered grids, where all unknowns are defined at the same gridpoints. The approach is to use a quasi-elliptic approximation, but to properly average the results. Thus, ∂_j^h in (18.6) or (18.7) has throughout been changed into the (long) central differencing $\partial_j^{2h} = \mu_j^h \partial_j^h$, replacing (18.8) by

$$\tilde{\det} L^h(\underline{\theta}) = h^{-2d} \left\{ \sum_{j=1}^{d} \left(2 \sin \frac{\theta_j}{2} \right)^2 \right\}^{d-1} \left\{ \sum_{j=1}^{d} \sin^2 \theta \right\}. \tag{18.17}$$

This symbol vanishes for some $|\underline{\theta}| = \pi$, showing L^h to be only quasi-elliptic.

More precisely, there are the high-frequency components

$$\begin{pmatrix} p \\ \underline{u} \end{pmatrix} = \begin{pmatrix} 1 \\ \underline{0} \end{pmatrix} e^{i\underline{\theta}\cdot\underline{x}/h} \quad (\theta_j = 0 \text{ or } \pi,\ 1 \le j \le d;\ |\underline{\theta}| = \pi), \tag{18.18}$$

which satisfy the homogeneous difference equations. These components by themselves do not really matter more than adding a constant to the pressure: They do not affect $\partial_j^{2h} p^h$, nor any other term in the difference equations. But other components, in the *neighborhood of* (18.18), do matter for the solution. They constitute high-frequency components which are not locally controlled. They will give rise to a subtle kind of instability (see §17.6), and they will not efficiently be reduced by relaxation, hence will be slow to converge in conventional multigrid algorithms. (Fast convergence can still be obtained by modified coarse-grid functions or by AMG algorithms; see §4.2.2 and 13.1.)

Both troubles (instability and slow smoothing) are closely related and easily cured by the same simple device: The bad components can be eliminated by *averaging the pressure*, i.e., replacing

$$\tilde{p}^h \to (\mu_1^h \dots \mu_d^h)^2 \tilde{p}^h. \tag{18.19}$$

This operation need to be done only on the final result. Similar bad convergence of some high-frequency components of an *intermediate* level does not matter, since those components are efficiently reduced by the next-finer-level relaxation (cf. §12).

To test this approach, without any interference of questions related to boundary conditions, we have studied the two-dimensional ($d = 2$) case, on the square $\{|x|, |y| \leq \pi\}$, with periodic boundary conditions. With such boundary conditions, u_j and p are determined only up to an additive constant each. We have used distributive Gauss-Seidel relaxation based on the distribution operator

$$M^h = \begin{pmatrix} -\Delta^h & 0 & 0 \\ -\partial_1^{2h} & 1 & 0 \\ -\partial_d^{2h} & 0 & 1 \end{pmatrix}, \tag{18.20}$$

(cf. (18.10)), with RB ordering in each of its three passes (corresponding to the three differential equations). Inter-grid transfers were standard: Full weighting (4.6), bilinear interpolation of corrections and cubic FMG interpolation of the first approximation. The coarse-grid operator is the same non-staggered, central Stokes operator as employed on the finest grid.

The compatibility condition, analogous to (18.5), seems at first to cause some troubles. In the discrete approximation it breaks up into 4 different conditions, each obtained by summing the discrete continuity equation on one of the four staggered sub-grids into which that equation is decoupled. Even if we take the trouble to satisfy these four conditions on the finest grid, they will not be satisfied on coarser grids, unless the coarse-grid equations are adjusted, e.g., by adding four suitable constants to F_0^k, one constant on each subgrid. Actually, all this is not needed. The said adjustment is below the level of the coarse-grid truncation, hence it does not improve the quality of the coarse-grid contribution to the fine-grid convergence. The fine-grid compatibility condition is itself similarly unimportant, unless one wants to solve the algebraic system below truncation errors. We simply ignore this condition at all levels.

Two additional compatibility conditions emerge because of our periodic boundary conditions, each obtained by summing one of the momentum equations over the grid. In the differential equations chosen by us these conditions are automatically satisfied, since we calculate a right-hand side $\underline{F}$ from a known solution (so that we know the exact differential solution and can compare our results with it). For reasons similar to the above, we ignored satisfying these compatibility conditions, too, in our discrete approximations, whether fine or coarse.

Components in the *neighborhood* of (18.18), eventually almost eliminated by (18.19), are still present, even though their influence on the residuals must *initially* be very small. Hence, *asymptotically* (after many cycles) we must get slow convergence. Indeed, on large $N \times N$ grids we have obtained asymptotic convergence factors of about $1 - 160N^{-2}$ per either V(1,1) or W(1,1) cycle. Initially, however, those components are not important and convergence factors are excelent; e.g., in the experiments of Table 18.1 the average convergence factor for the first five cycles was always better than .2 per V(1,1) cycle.

The main point of course is that the convergence rate of high-frequency components is immaterial by itself, since we do not need to converge them much to get below their large truncation errors. What does then matter is how many cycles per level an FMG algorithm needs in order to solve the problem to below truncation. Typical results are summarized in Table 18.1. They show that *convergence well below truncation errors is already obtained for an algorithm with only* 3.6 *RWUs* (relaxation work units). Being based on the RB relaxation, the algorithm is fully vectorizable.

FMG Algorithm			Results			
Grid Size	Cycle Type	n	$\|u-\tilde{u}^h\|_\infty$	$\|v-\tilde{v}^h\|_\infty$	$\|p-\tilde{p}^h\|_\infty$	
					Without Averaging	With Averaging
16×16	V	1	.0259	.0379	.4589	.1984
		2	.0217	.0224	.3306	.1671
		3	.0209	.0213	.3247	.1662
	W	1	.0259	.0356	.4400	.1873
		2	.0259	.0219	.3287	.1666
		3	.0259	.0216	.3275	.1662
32×32	V	1	.0056	.0107	.1216	.0821
		2	.0044	.0069	.0774	.0556
		3	.0043	.0068	.0627	.0541
	W	1	.0047	.0090	.1248	.0813
		2	.0045	.0069	.0789	.0564
		3	.0043	.0068	.0630	.0543
64×64	V	1	.0019	.0036	.0345	.0172
		2	.0020	.0029	.0180	.0141
		3	.0020	.0028	.0154	.0141
	W	1	.0020	.0032	.0367	.0187
		2	.0020	.0029	.0184	.0143
		3	.0020	.0028	.0158	.0141

Table 18.1. ***Stokes solutions on non-staggered grid.***

Results are shown for an FMG solution of a problem whose exact differential solution is $u = v = p = \sin(\cos(x+2y))$, *the right-hand sides* F_0 *and* $\underline{F}$ *being calculated accordingly.* F_0^h *and* $\underline{F}^h$ *at all levels are obtained by averaging* F_0 *and* $\underline{F}$ *on* $h \times h$ *squares. The approximate solution* $(\tilde{u}^h, \tilde{v}^h, \tilde{p}^h)$ *is obtained by the algorithm explained in the text, with relaxation counts* $\nu_1 = \nu_2 = 1$ *(see §1.4, 1.6; the V-cycle algorithm is exactly that of Fig. 1.2). The undetermined additive constants in* $\tilde{u}^h$, $\tilde{v}^h$ *and* $\tilde{p}^h$ *were of course properly subtracted off.* n *denotes the number of cycles per level.*