Chapter 4

Interior Two-Level Cycles

Having computed the smoothing factor $\overline{\mu}$, one should expect the asymptotic convergence factor per multigrid cycle to approach $\overline{\mu}^{\nu}$, where ν is the number of relaxation sweeps (on the fine grid h) per cycle. This ideal figure does not take into account the exact nature of the inter-grid transfers. The next task then is to design those transfers so as to approach the ideal figure. To separate their design from questions related to boundary conditions (which are taken up at the next chapter), we still think in terms of fully-periodic or full-space problems; we still, that is, restrict our attention to *interior* processes, because it is there that most of the computational work is invested. Furthermore, we simplify the multigrid situation at this stage by restricting our attention to two grids only, the finest grid $h = (h_1, \ldots, h_d)$ and the next coarser grid $H = (H_1, \ldots, H_d)$, where usually $H = 2h$. That is, we assume in our analysis that the grid-H equations are solved (exactly) each time the algorithm gets to that grid, without analyzing how and how expensively that solution is obtained, hence without involving grids coarser than H in the analysis.

These assumptions indeed simplify our studies very much. First, the error can be expanded in a Fourier integral (or series) and the transformations of the amplitudes of different Fourier components by multigrid operations can be calculated. Indeed, for linear systems with constant coefficients only few Fourier components at a time are coupled to each other by these two-level interior processes, hence transformations of Fourier amplitudes are expressed as small matrices (§4.1). In case of nonconstant coefficients, we usually freeze them at some values (treated then as parameters of the analysis). In case of nonlinear equations, their Newton linearization is analyzed (although no such linearization is needed in the actual processes; see §8.3). The parameters of the analysis then depend on the solutions around which linearization is made.

This freezing of coefficients is reasonable as long as the real coefficients do not change too drastically over a meshsize. Where they do, we can sometimes model them as changing periodically, again making mode analysis with small matrices possible [BD79, §4.7]. See more about the rigor of this analysis in §14.

When mode analysis becomes too difficult or dubious, or if one simply wishes to skip it, **experiments with periodic boundary conditions** can be used instead. One can in fact do such experiments even when mode analysis *is* available, and *compare* the analysis with the experiments. This is an accurate debugging technique, completely separating away issues related to boundary conditions. Moreover, such periodic-boundary-condition program could serve as an excellent preliminary stage in developing your real multigrid program. For this preliminary program, the various advices given in §5.1 concerning the full program could be used, including in particular the use of *multi*grid program to simulate a *two*-grid algorithm, and the trick of near-linearization of non-linear equations.

Some warnings, however, concerning this use of periodic boundary conditions: First, relaxation should better be restricted to simultaneous-displacement (Jacobi, red-black Jacobi, zebra Jacobi, etc.) schemes, to avoid the special non-smoothness created along the starting line (or termination line) of the relaxation sweep. (This non-smoothness would be particularly bad if downstream ordering were used.) Second, periodic boundary conditions often require additional global conditions to be added so as to make the problem well posed. In some cases these extra conditions are easy to implement, involving for example just an adjustment of the solution average by adding a constant. (Multigrid treatment of global conditions is discussed in §5.6.) Other cases, especially nonlinear ones, are less straightforward. A full section could, and perhaps will, in fact be written about the art of using periodic problems. It is also important to realize that interior studies in general have more limited value for non-elliptic problems, where the interplay with boundaries is more essential (see end of §4.1).

One should also make sure that at this stage (whether mode analysis or periodic numerical experiments are used) both grid h and grid H are fine enough, and grid-H equations are solved accurately enough (without taking into account the work this accurate solution requires), in order to separate away questions related to coarser grids (see §6.3). Do not forget, however, in the process of optimizing the grid-H operator L^H, that this is a modeling for a multigrid solution, hence your model must be recursive: The H equations should have the same general form as the original h equations, with the same range of possible parameters.

In addition to the relaxation scheme, studied above, the main issues to be studied at this interior-two-level stage are when to switch (under what criteria, or after how many relaxation sweeps) from grid h to grid H; what should be the coarse-grid variables; and the type (in the interior) of three multigrid operators: The fine-to-coarse transfer of residuals I_h^H, the coarse

grid operator L^H, and the coarse-to-fine interpolation of corrections I_H^h. These issues are one-by-one discussed in the subsections below. They are later reviewed again, from a more general perspective, in §11. Relevant to these issues are also §8.5 and §10.2 (nonlinear problems and higher-order techniques).

4.1 Two-level cycling analysis. Switching criteria

Details of the two-level mode analysis are described in [BD79, §4.6–4.8] and in [ST82, §§3.3–3.5, 7, 8, 9]. The former also discusses modifications of the analysis to account for the fact that in practice the grid-H equations are only approximately solved, modification for the case of equations with highly oscillatory coefficients, and ways to make precise comparisons between mode analysis and numerical experiments (for debugging purposes). The main things to know are the following.

On grid $2h$ the Fourier mode $\exp(i\underline{\theta} \cdot \underline{x}/\underline{h})$ aliases (coincides with) the mode $\exp(i\underline{\theta}' \cdot \underline{x}/\underline{h})$ if $|\theta_j - \theta'_j| = 0$ or π for $j = 1, \ldots, d$. Hence each set of so aliasing components usually includes 2^d components $\{\underline{\theta}^1, \ldots, \underline{\theta}^{2^d}\}$, called **harmonics** of each other. They are coupled to each other by the two-level processes. (The special sets with less than 2^d different components do not require special analysis, since they are limits of regular sets.)

We define the **two-level cycle** as follows: Make ν_1 relaxation sweeps on grid h, then transfer the residual problem to grid H and solve it there exactly, then interpolate that grid-H solution to grid h and add it as a correction to the former grid-h solution, then make ν_2 more relaxation sweeps on grid h. It is easy to see that in the infinite space, if L^h, L^H, I_h^H and I_H^h are all constant operators, and if the error in the solution before such a cycle has the form $\sum_j A_j \exp(i\underline{\theta}^j \cdot \underline{x}/\underline{h})$, where the sum is over a set of 2^d harmonics, then the error after the cycle will have a similar form, and the new A_j's will be linear combinations of the old ones. If we deal with a system of q grid equations then each amplitude A_j is a q-vector, hence the overall transformation of the 2^d amplitudes by the two-level cycle is a $(2^d q) \times (2^d q)$ matrix M, which can be denoted $M(\underline{\theta})$ where $\underline{\theta}$ is the lowest harmonic ($|\underline{\theta}| \leq \frac{\pi}{2}$).

This matrix $M(\underline{\theta})$ is called **the two-level amplification matrix**. The easiest and most modular program for calculating it is to write a different routine for the general matrix-element of each of the five involved processes: relaxation, L^h, I_h^H, L^H and I_H^h. Their respective matrices $\check{S}^h$, $\check{L}^h$, $\check{I}_h^H$, $\check{L}^H$ and $\check{I}_H^h$ have dimensions $(2^d q) \times (2^d q)$, $(2^d q) \times (2^d q)$, $q \times (2^d q)$, $q \times q$, and $(2^d q) \times q$, respectively, and each of their elements is a function of $\underline{\theta}$. Then program

$$M(\underline{\theta}) = \left(\check{S}^h\right)^{\nu_2} \left[I - \check{I}_H^h \left(\check{L}^H\right)^{-1} \check{I}_h^H \check{L}^h\right] \left(\check{S}^h\right)^{\nu_1}. \tag{4.1}$$

The main performance measure of the two-level cycle is the **two-level asymptotic convergence factor** (per cycle)

$$\overline{\lambda} := \max_{|\underline{\theta}| \leq \frac{\pi}{2}} \rho\left(M\left(\underline{\theta}\right)\right) , \tag{4.2}$$

where $\rho(M)$ is the spectral radius of M. Note that $\overline{\lambda}$ depends on the sum $\nu = \nu_1 + \nu_2$, but not on the separate values of ν_1 and ν_2. In fact, when many cycles are performed the separate values are immaterial. Various other performance measures can similarly be defined. (See [ST82, §3.4-3.5], where the notation M_h^{2h} and $\rho(M_h^{2h})$ is used for our M and $\overline{\lambda}$, respectively. Additional two-level measures will be discussed in §7.4.)

The two-level analysis is used to (roughly) optimize the involved processes; namely, the objective is to maximize $w^{-1}\log(1/\overline{\lambda})$, where $w = A(\nu w_0 + w_1 + w_2)$, w_0 is the work in one relaxation sweep, w_1 is the work of calculating and transferring the residuals, w_2 is the work of the I_H^h interpolation, and A is a factor through which the work on coarser grids is taken into account. For our objective here the value of A is really immaterial, but to have a definite value in later uses (see §6.2) we observe that for V cycles we can assume similar operations on each of the grids, hence $A = (1 - \hat{\rho}_1 \cdots \hat{\rho}_d)^{-1}$, where $\hat{\rho}_j = h_j/H_j$ (usually $\hat{\rho}_j = .5$) while for W cycles $A = (1 - 2\hat{\rho}_1 \cdots \hat{\rho}_d)^{-1}$. To avoid the laborious count of operations and the arbitrary assignment of proper weights to different arithmetic and non-arithmetic operations (which are really machine-dependent), one can use the work of a standard relaxation sweep as the work unit. In complicated problems, where calculating L^h outweighs interpolations, one can then neglect w_2 and take $w_1 = 1$ for full residual weighting and $w_1 = 2^{-d}$ for residual injection. The convergence factor per work unit is then denoted by $\mathring{\mu} = \overline{\lambda}^{1/w}$. As above (§3.2), in addition to the goal of minimizing $\mathring{\mu}$ we should take robustness and simplicity into account.

One can also partly separate the study of I_h^H, L^h and I_H^h from that of relaxation by the **Coarse-Grid Correction (CGC) mode analysis**, as in [Bra77a, §A.1]. But this is not simpler than the full two-level analysis, especially since relaxation schemes have already been selected in the previous stage. We use a CGC analysis in §4.3 below.

The ideal factor $\overline{\lambda} = \overline{\mu}^{\nu}$ is not always obtainable. If $\overline{\mu}^{\nu}$ is too small we will get $\overline{\lambda} > \overline{\mu}^{\nu}$, because of significant high-frequency amplitudes generated from low ones by interpolation or by RB-type relaxation (see §4.3). Even when obtainable, too small values of λ will require too precise interpolations, hence too much investment in w_1 and w_2, and will at a later stage be frustrated by other interactions (boundaries and non-constant coefficients). Also, such small $\overline{\lambda}$ will not usually be needed in the final FMG algorithm (see §§7.2-7.3). Hence, the optimal cycle always employs a small ν, typically $\nu \leq 3$.

In regular elliptic problems $\nu = 1$ is too small to be optimal (unless the sweep includes several passes, as in symmetric and alternating-direction

schemes), since the overhead of w_1 and w_2 weights too much against it. Hence usually the optimal number is $\nu = 2$ for very efficient smoothers ($\overline{\mu} \leq .3$ or so), and $\nu = 3$ otherwise. A small change in ν does not disturb the overall efficiency very much. Considerably larger ν are less efficient, because they bring the process into the range of larger feeding from low to high frequencies, while not much more is gained in reducing the overhead (already at $\nu = 3$, $w_1 + w_2$ is quite small compared with νw_0).

A possible approach is **accommodative**: do not fix ν in advance, but continue relaxation as long as it exhibits the fast convergence of high frequencies, e.g., as long as the convergence factor (some norm of the residuals divided by the same norm a sweep earlier) is smaller than the smoothing factor $\overline{\mu}$. For non-scalar ($q > 1$) systems, such a criterion can separately be applied to each equation, possibly resulting in more passes for part of the equations. Similarly it may separately be applied at different subdomains (since smoothing is a local process), possibly giving partial relaxation sweeps.

In the case of non-elliptic and singular perturbation problems there are some particular smooth error components (smooth characteristic components of the differential operator or the reduced differential operator) for which L^H is a bad approximation to L^h, hence $\overline{\lambda}$ cannot be much smaller than .5, no matter how small $\overline{\mu}^\nu$ is [Bra81a, §5.1], [Bör81]. But for exactly the same components and the same reason, L^h itself is not a good approximation to the differential operator L. Hence, exactly for these components, we do not *need* much *algebraic* convergence (convergence to the discrete solution), since the discrete solution itself is far from the differential solution. Hence, for such cases the asymptotic convergence factor $\overline{\lambda}$ is not really the measure we need.

The one we need is obtained by the two-level FMG analysis (see §7.4). Moreover, for non-elliptic or singular perturbation problems the usual assumption that high-frequency components are local does not hold. It is violated by high-frequency characteristic components in cases of strong alignment (§2.1). The interior mode analysis should then be supplemented with a half-space analysis (§7.5).

4.2 Choice of coarse grid

When the fine grid, with meshsize $h = (h_1, \ldots, h_d)$, is given, the choice of a coarse grid, with meshsize $H = (H_1, \ldots, H_d)$, is often straightforward: Take every other line (every other hyperplane, for $d > 2$) of the fine grid in each direction. The coarsening ratio $H_j/h_j = 2$ is usually optimal: it is the smallest recursively convenient number, and it is already big enough to make the coarser-grids work quite small relative to the fine-grid work; larger H_j/h_j will not save significantly more work, but will significantly degrade the smoothing factors (see (3.3)). The smaller ratio $H/h = 2^{\frac{1}{2}}$ may be as efficient (trading larger A for smaller ν), and it is recursively

convenient in some two dimensional problems with rotatable operators; see [ST82, §2.5], [RTW83].

When the fine-grid discretizations are done in terms of "cells" with the discrete variables defined at certain cell positions (e.g., cell centers, or centers of vertical cell boundaries, etc.), and especially when the grid is **staggered** (different grid functions are defined at different cell positions), it is more convenient to coarsen in terms of the cells: Take every 2^d fine cells as a coarse cell, and then place coarse-grid variables at coarse-cell positions analogous to their positioning in the fine cells. The coarse grid points then are not a subset of the fine grid points. See examples in §17.4, 18.4 and another approach in [Den82a].

In some cases the "fine-grid" is not a well-organized grid at all; e.g., a general finite-element triangulation, not based on any grid lines. Then one can still construct the coarse grid as a uniform grid, placed over the domain with no particular relation to the fine grid. Another approach is to base the choice of coarse-grid variables on purely algebraic considerations (§13.1). Mode analysis is of course not very suitable for analyzing such situations.

4.2.1 Semi coarsening

Semi coarsening, or more specifically S-coarsening, is the technique of using grid H which is coarser than h in only a partial set S of coordinates; i.e., $H_j = 2h_j$ for $j \in S$ and $H_j = h_j$ for $j \notin S$. This means some more work is done on coarse grids; but either this or block relaxation are needed in some cases – see the rule in §3.3. Semi coarsening is sometimes preferable to block relaxation. For example, in three-dimensional problems where there are two fixed coordinates with stronger couplings, full coarsening would require plane relaxation, which is inconvenient. (Solving these plane equations approximately by one multigrid cycle, if done simultaneously at all planes, will look very much like semi coarsening.) Also, exactly in those cases, semi coarsening involves relatively small work on coarser grids, since two coordinates are still coarsened, hence the total number of points on all coarser grids is at most one third the number of points on the finest grid.

Sometimes, a combination of block relaxation and semi coarsening may be the best. For example, the equation $aU_{xx} + bU_{yy} + cU_{zz}$ with $a \ll b \ll c$, discretized on a cubic grid ($h_x = h_y = h_z$), will best be solved by z-line relaxation and $y-z$ semi coarsening. Generally, rough calculations of S-smoothing factors (§3.3) immediately show what procedures can be taken.

In some cases block relaxation is of course preferable to semi coarsening. For example, when directions of strong alignment are different at different subdomains. To change accordingly the directions of semi coarsening would be much messier than changing block directions.

4.2.2 Modified and multiple coarse-grid functions

When a difference operator L^h is given which has no good h-ellipticity or semi-h-ellipticity measure, then no relaxation can be efficient in reducing all high-frequency error components. To reduce all components efficiently we can then often use modified coarse-grid correction functions.

Suppose for example that the slow components (i.e., the components for which relaxation is inefficient) are all clustered around some known modes $\phi_j(\underline{x})$, $(j = 1, \ldots, J)$. This means that the error $v^h = \tilde{u}^h - u^h$ can be written as $v^h(x) = \sum_j v_j^h(\underline{x})\phi_j(\underline{x})$, where v_j^h are smooth functions. It is then these smooth functions which we try to approximate by coarse-grid functions v_j^{2h}. See [Bra80c, §3.2]. Sometimes, each of these functions can most efficiently be approximated on a different (e.g., differently rotated) grid [BL97].

4.3 Orders of interpolations and residual transfers

The most important aspect of the coarse-to-fine correction interpolation I_H^h and the residual transfer I_h^H is their orders, defined as follows: The order of I_H^h is m if the interpolation of the low frequency Fourier component $\exp(i\underline{\theta} \cdot \underline{x}/\underline{h})$, with amplitude 1 on the coarse grid H, creates on the fine grid h high-frequency components (the harmonics of the low frequency) with amplitudes $O(|\underline{\theta}|^m)$. It also reproduces the $\underline{\theta}$ component itself on grid h with an amplitude $1 + O(|\underline{\theta}|^m)$. The order of the fine-to-coarse transfer I_h^H is said to be m, and its secondary order $\overline{m}$, if a high-frequency harmonic with amplitude 1 on grid h contributes $O(|\underline{\theta}|^m)$ to the amplitude of the corresponding low frequency $\underline{\theta}$ when transferred to grid H, while a low frequency with amplitude 1 on grid h contributes $1+O(|\underline{\theta}|^{\overline{m}})$ to its grid-H amplitude. Thus, linear and bilinear interpolations have order 2, while cubic interpolation is fourth order. Residual transfer by injection ($I_h^H \equiv 1$) has order 0 and infinite secondary order, whereas the usual full-weighting residual transfer ((4.6) below) is of order 2 and secondary order 2.

What orders should be used in the multigrid cycle? This depends on the orders of derivatives appearing in our equations. Suppose we have a system of q differential equations in q unknown functions, and let m_{ij} be the highest order of differentiation (or differencing) of the j-*th* unknown in the i-*th* equation, $(i, j = 1, \ldots, q)$. We assume, and this is usually the case, that the q unknown functions are interpolated independently of each other and that the residuals of each of the q grid equations are transferred separately from the others. Denote by m^j the order of I_H^h used in interpolating the j-*th* correction (correction to the j-*th* unknown function) and by m_i and $\overline{m}_i$ the order and secondary order, respectively, of the I_h^H used in transferring the i-*th* residual (residuals of the i-*th* equation).

What m^j, m_i and $\overline{m}_i$ $(i, j = 1, \ldots, q)$ should be used? Examining

orders of magnitude in the CGC mode-analysis operator (the operator in brackets in (4.1)), under the assumption that all $m^j > 0$, we find the following basic rules and observations:

(A) The high-frequency harmonics of the lowest frequencies (those with $|\theta| = O(h)$), are amplified by the CGC operator by a factor with a spectral radius $1 + O\left(\sum_{i,j} h^{m_i+m^j-m_{ij}}\right)$. Hence, to avoid large magnification of high-frequencies, we should have

$$m_i + m^j \geq m_{ij}, \tag{4.3}$$

preferably even $m_i + m^j > m_{ij}$. On the other hand, larger values ($m_i+m^j > m_{ij}+1$) would not significantly further reduce the spectral radius, hence they are *asymptotically* (when many cycles are made) not needed [Bra94].

(B) Every high-frequency harmonic (before the CGC cycle) contributes to the corresponding low-frequency (after the cycle) through a $q \times q$ transformation matrix $(L^H)^{-1}B$, where $B_{ij} = O(h^{m_i-m_{ij}})$. This is usually not important asymptotically (for many cycles), but *if only one cycle is performed* (as in FMG algorithms), that transformation may cause large errors unless

$$m_i \geq m_{ij}. \tag{4.4}$$

For relaxation schemes with interactions between high and low frequencies (e.g., RB schemes), this transformation may also cause *asymptotic* degradation unless $m_i > \sum_k (m_{ij} - r_{kj})$, where $O(h^{r_{kj}})$ is the size of the high-frequency errors in the *k-th* function generated by relaxation from an $O(1)$ low-frequency error in the *j-th* function. RB and zebra schemes for $q = 1$ give $r_{11} = m_{11}$, hence the rule requires $m_1 > 0$, i.e., full weighting (see §4.4). This requirement can however be slackened by a more precise look at the nature of these particular schemes (allowing the use of simpler transfers such as the "half injection" $I_h^H \equiv .5$ or "half weighting"; see [FST81, §2], [ST82, §8.1]).

(C) The low-frequency error components themselves are reduced by a factor $O(h^{\tilde{m}})$, where $\tilde{m} := \min\{\tilde{p}, \overline{m}_1, \ldots, \overline{m}_q, m^1, \ldots, m^q\}$ and $\tilde{p}$ is the lowest of the approximation orders on levels h and H. Hence $\tilde{m}$ must be positive, which is indeed the case for any consistent differencing and interpolation schemes. Larger values of $\tilde{m}$ may of course give better cycle performance. Our experience indicates that $\tilde{m} = 2$ gives considerably better $\overline{\lambda}$ than $\tilde{m} = 1$. Since this is a low-frequency matter, hence non-local, higher $\tilde{m}$ may be effective only if they are carefully matched by corresponding high-order approximations and

interpolations at *boundaries*. But one usually does not have to go into the trouble of $\tilde{m} > 2$. Rather, employ more cycles with $\nu \leq 3$ (see §4.1). As a result, the factor $O(h^{\tilde{m}})$ will usually be dominated by $\overline{\mu}^{\nu}$ in determining $\overline{\lambda}$.

(D) We also note that every low-frequency error component (before the CGC cycle) contributes to every one of its harmonics (after cycle) through a $q \times q$ transformation matrix D, where $D_{jj} = O(h^{m^j})$ and for $i \neq j$, D_{ij} has higher orders in h. This tells us something about the range where relaxation should be efficient (see §12).

4.4 Variable operators. Full weightings

The above mode-analysis rules are insufficient in case L^h is highly-varying, i.e., its coefficients substantially change between two neighboring grid-points. For such L^h the residuals after relaxation are also highly varying, hence to represent them correctly on grid H, *full residual weighting* should be used, i.e., I_h^H should satisfy, for any residual function r^h,

$$(H_1 \cdots H_d) \sum_{x^H} \left(I_h^H r^h\right)(x^H) = h_1 \cdots h_d \sum_{x^h} r^h(x^h), \tag{4.5}$$

where x^h are the fine-grid points and x^H are the coarse-grid points. In other words, full weighting "preserves integrals". (Throughout this discussion it is assumed that the difference equations on all grids are written in their *divided* form, analogous to the differential equations. If, however, they are multiplied through by factors which depend on the meshsize, then one should not forget to have those factors in (4.5), too.) One can regard full weighting as a scheme in which each residual $r^h(x^h)$ on the fine grid is being distributed to several coarse grid points, with weights whose sum is $\hat{\rho} = h_1 \cdots h_d/(H_1 \cdots H_d)$. Hence each residual r^h is a weighted average of its transferred values on grid H , times $\hat{\rho}$. This weighted average represents a certain interpolation, $\check{I}_H^h$ say. Thus every full weighting I_h^H is the *adjoint* (or, in matrix terminology, the *conjugate transpose*) of an interpolation $\check{I}_H^h$, times $\hat{\rho}$. $I_h^H = \hat{\rho}(\check{I}_H^h)^*$. For instance, the d-dimensional analogue of the standard 9-point symmetric full weighting, defined by

$$\left(I_h^{2h} r^h\right)(x^{2h}) = \sum_{\max_j |\nu_j| \leq 1} 2^{-d-\sum |\nu_j|} r^h \left(x^{2h} + (\nu_1 h_1, \ldots . \nu_d h_d)\right), \tag{4.6}$$

is the adjoint of d-linear multivariate interpolation (the tensor product of linear interpolations along one dimension at a time), times 2^{-d}.

The requirement (4.5) is equivalent, in terms of the Fourier analysis, to the requirement that I_h^H has a positive order (see §4.3). Such full weightings should perhaps be used in almost any case. Only in some particular cases non-full weighting operators happen to be asymptotically somewhat better.

An example is injection in case of the standard 5-point Poisson operator, which yields lower $\overline{\lambda}$ as well as lower w_1 than the full weighting (4.6) (see [BD79, §4.8]). But even in those cases, for the purpose of Full Multigrid (FMG) algorithms (see §7), full weightings may be preferable. (See rule (4.4) above and [ST82, §3.6]).

4.5 Coarse-grid operator. Variational and Galerkin coarsening

The coarse grid operator L^H should be a proper homogenization of the fine-grid operator L^h. In smooth problems this is easily obtained by good discretizations of both L^h and L^H. In nonlinear problems this is effectively obtained by a suitable FAS averaging of the fine-grid *solution* (see $\check{I}_h^H$ in §8.5). Sometimes one needs to derive L^H from L^h, not from the differential operator L, either because L is not available or because one wants an automatic program, for some *general* class of L^h, or without having to treat separately boundary conditions and whatever other features of the differential problem. That is, one wants to regard the fine grid equations simply as a matrix equation

$$\underline{L}^h \underline{u}^h = \underline{f}^h \tag{4.7}$$

where the underlines signify matrix notation: $\underline{L}^h$ is an $n^h \times n^h$ matrix, where n^h is the number of unknowns on grid h. The geometry of the grids is only used to construct the inter-grid transfers I_H^h and I_h^H. (A multigrid treatment without any geometrical structures is discussed in §13.1.).

In case $\underline{L}^h$ is **symmetric**, a general way to derive L^H is to regard (4.7) as the equivalent problem of finding $\underline{u}^h$ which minimizes the functional

$$\Phi^h(\underline{u}^h) := \frac{1}{2}(\underline{u}^h)^* \underline{L}^H \underline{u}^h - (\underline{f}^h)^* \underline{u}^h, \tag{4.8}$$

where stars stand for transposing. Given now an approximate solution $\tilde{\underline{u}}^h$, with a (smooth) error $\underline{v}^h = \underline{u}^h - \tilde{\underline{u}}^h$ which is to be approximated by the coarse-grid correction $I_H^h \underline{v}^H$, the-equations for $\underline{v}^H$ are fully specified by requiring it to yield a correction which reduces Φ^h as far as possible, i.e., $\underline{v}^H$ should minimize $\Phi^h(\tilde{\underline{u}}^h + I_H^h \underline{v}^H)$. This immediately gives the coarse grid equations

$$\underline{L}^H \underline{v}^H = I_h^H (\underline{f}^h - \underline{L}^h \underline{u}^h), \tag{4.9}$$

where

$$I_h^H = (I_H^h)^* \tag{4.10}$$

and

$$\underline{L}^H = I_h^H L^h I_H^h. \tag{4.11}$$

Equation (4.9) has the general form (1.14). The specific prescription (4.10)-(4.11) is called *variational coarsening*, since it results from the variational

formulation of the problem. It is automatically determined as soon as the interpolation operator I_H^h is selected. I_h^H given by (4.10) is automatically "full" in the sense of §4.4. (By (4.11), the coarse grid equations (4.9) are not basically changed if I_h^H is multiplied by any scalar, such as $\hat{\rho}$ in §4.4.)

In case $\underline{L}^h$ is not symmetric, (4.10) is not always advisable (see §4.6), but (4.11) can generally be used. This L^H is called the *Galerkin operator*, since it is equivalent to requiring the coarse-grid correction to be a *projection*, i.e., requiring the residuals of the corrected solution $\tilde{\underline{u}}^h + I_H^h \underline{v}^H$ to vanish when transferred back to the coarse grid:

$$I_h^H \left(f^h - L^h(\tilde{\underline{u}}^h + I_H^h \underline{v}^H) \right) = 0.$$

The reason Galerkin operators and variational coarsening are not always advisable is the amount of work involved in the construction (4.11), which could be considerably larger than the entire solution work (e.g., when solving by the algorithm in Fig. 1.2). Also, once constructed, the Galerkin operator is often much more complicated than the simpler L^H derived directly from L (e.g., 9-point instead of 5-point formulae), and requires much larger storage for storing all its coefficients, whereas the simpler L^H may require no storage (e.g., whenever L is autonomous, whether linear or not; this includes all fluid dynamic equations, if they are not linearized). See more about this issue in §11.

4.6 Strongly discontinuous, strongly asymmetric operators

As long as the fine-grid operator does not vary drastically, the above rules for I_H^h, L^H and I_h^H work fine. A more difficult case is that of a strong discontinuity in L^h, i.e., where its coefficients change their *order of magnitude* within a meshsize. Orders of interpolations are not so important then; rather, special forms should be used which take into account the particular nature of the discontinuity. The rule is first to analyze the behavior, near the discontinuity, of the error which is inefficiently reduced by relaxation. This error is approximately a solution to the homogeneous equations. (If it is not, then it has large residuals and therefore there locally exists a relaxation scheme for which it will be reduced efficiently; cf. §1.1). Hence its general behavior is like that of solutions to the homogeneous differential equations. The interpolation I_H^h of corrections should take this behavior into account. For example, if we have a diffusion problem $\nabla(a\nabla u) = F$, near a strong discontinuity of the diffusion coefficient $a(x)$ the derivatives of the solution to the homogeneous equation are not continuous. Instead, $a\nabla u$ is continuous there, and this can be used to design good interpolation schemes [ABDP81]. In the case of singular perturbation or non-elliptic problems, solutions to the homogeneous equations are continuous along (sub) characteristics, hence interpolation should be as much as possible in

the characteristic directions. This is possible exactly where it is most important, namely, cases of intended strong alignment (cf. §2.1). Thus, for example, avoid interpolating across a boundary layer which is intended to be sharply reproduced.

It is less clear how to generally design the residual transfers I_h^H and the coarse grid operators L^H near a strong discontinuity. In the symmetric case the variational rules (4.10)–(4.11) are most robust [ABDP81], even though expensive. For cases which are not essentially symmetric the Galerkin operator (4.11) can still be used, but instead of (4.10) one should take

$$I_h^H = \left({}^*\underline{I}_H^h\right)^* \tag{4.12}$$

where ${}^*\underline{I}_H^h$ is an interpolation appropriate in the above sense for $(L^h)^*$, the adjoint of L^h. See [Den82b].

For non-elliptic and singular perturbation problems, the considerations and experiments in [Bra81a] indicate that improved results are obtained by a full residual weighting in which residuals, on being transferred from a fine gridpoint to a different point (or points) on the coarse grid, are transferred roughly in the downstream direction. As for correction interpolation for such problems, however, it seems that the symmetric schemes are preferable to schemes with upstream bias. Coarse grid operators identical with the fine-grid ones (hence much cheaper than (4.11)) were used, with excellent FMG results (even where the asymptotic rates were slow).

A general perspective on these questions of coarsening a problem (designing I_h^H, L^H, I_H^h) is given in §11 below.