Chapter 11

Coarsening Guided By Discretization

The term "coarsening" is used here for the entire process of transferring a residual problem $L^h v^h = r^h$ from a fine grid h to the next coarser grid $H(= 2h)$. This includes the formulation of the coarsegrid problem $L^H v^H = I_h^H r^h$, where the coarse-grid operator L^H and the fine-to-coarse transfer I_h^H should be determined both in the interior and near boundaries, and similar equations should be transferred for the boundary conditions themselves, and for any other side conditions the problem may have. Any sufficiently general method of coarsening implies a discretization method, in the following sense: If the differential problem $Lu = f$ is discretized by any method, giving the problem $L^h u^h = f^h$ on grid h, and if this problem is then successively coarsened to $L^{2h} u^{2h} = f^{2h}$, $L^{4h} u^{4h} = f^{4h}$, etc. by successive applications of that same coarsening method, then in the limit (for a sufficiently coarse grid) we obtain a discretization of $Lu = F$ which does not depend on the original discretization $L^h u^h = f^h$, but only on the method of coarsening. The limit discretization, in this sense, is the **fixed point of the coarsening method**. (In practice the limit is almost fully established after just a couple of coarsening levels).

This rather trivial observation implies that coarsening is at least as rich and difficult as discretization. It also implies that controversies and competing techniques will emerge concerning coarsening techniques similar to the ones in the field of discretization. Indeed such competitions have already surfaced. For example, the competition between finite-difference and finite-element methods, a dispute which in fact consists of several separate issues: The variational derivation of discrete equations (or coarse-grid equations) vs. direct differencing; the interpolation issue (finite-elements insist on using the same coarse-to-fine interpolation - the same "element" - as used in deriving the discrete equations, while finite-differences allow more freedom in interpolation, sometimes gaining higher accuracy in some

error norms); the issue of general triangulation vs. uniform grids; and the issue of compactness of high-order approximations. These issues should not be confused with each other: variational derivation is possible and natural even without the use of elements [FW60, §20.5]. Uniform grids can be used with finite-element solutions, too, changing the elements only near boundaries, a structure offering high computational efficiency, especially in conjunction with multigrid methods [Bra79a]. High-order compact operators arise quite naturally in the finite-element method, but such operators can also be derived by finite-difference approaches, such as the operator compact implicit method [CLW78] and also the Hodie method [LR78].

All these and other issues arise as well with regard to coarsening, and the competing approaches are generally successful in coarsening wherever they are successful as discretization procedures – which is usually in problems.where they are more natural. Variational approaches [Nic75], [Bra77a, App. A.5], [Hac82] are natural for self-adjoint problems, and have provided the most robust and automatic coarsening procedures for such problems [ABDP81], [Den82a], although they can be replaced by much less expensive procedures (analogous to direct differencing) if the self-adjoint problem is not particularly complicated (cf. §4.5–4.6). In singular perturbation problems, such as those arising in fluid dynamics, discretization as well as coarsening are most successfully guided by physical understanding (artificial viscosity, upstream differencing, etc.); and so on.

The attempt to devise general fine-to-coarse transfers, good for all problems, is as hopeless (or as hopeful) as the attempt to have general, completely problem-independent discretization procedures.

Notwithstanding, while this argument tells us how complicated coarsening can be, it also elucidates a general way to handle this difficulty. Namely, the coarsening method can always be guided by the discretization scheme.

Indeed, conversely to the statement above (that every coarsening method implies a discretization scheme), one can say that every discretization scheme can be used to derive a coarsening procedure. This is done by imitation or analogy: Think about *discretizing* the problem $Lv = r$ on the coarse grid H; then replace the operations done on the continuous domain by analogous operations done on the fine-grid h; e.g., replace integrations by analogous summations (or by integrations by elements, in case v^h is given in terms of finite elements). Galerkin discretization schemes, for example, are easily translated in this way into analogous coarsening formulae of the type (4.11) [Nic75, §3].

A coarsening procedure analogous to the finest-grid discretization scheme is called **compatible coarsening**. It is not necessary to use compatible coarsening, but it usually makes a good sense to do so. In case the discretization scheme is a bad one, this would give a bad coarsening and hence slow asymptotic convergence rates of the multigrid cycling. But experience with several such cases (e.g., boundary singularities improperly

treated) show that, if compatible coarsening is used, this slowness does not matter, because the source of slowness (bad discretization) is also, and for the same solution components, a source for large truncation errors, hence an FMG algorithm (with the same discretization scheme on all currently-finest levels) still solves *below truncation errors* in the usual number of cycles (one or two, depending on the interior processes). Moreover, the slower *asymptotic* algebraic convergence rates can in this way serve as a detector for the bad discretization, which otherwise may be passed unnoticed.

Compatible coarsening makes sense also from the point of view of computer resources and programming effort. For example, if a great generality and simplicity of programming is obtained by a discretization scheme (e.g., finite elements) which on the other hand spends a lot of computer time and storage to assemble the discrete equations and store them, the coarsening procedure can do the same since the time and storage it spends will be smaller than those already spent on discretizing on the fine grid.

There are some special cases in which compatible discretizations are not quite available. These are cases where the discretization scheme is not general enough, because it specifically uses features of the finest grid not present on coarser ones. It uses for example a finest grid exactly laid so that its lines coincide with special lines of the problem, such as boundaries or lines of strong discontinuities (as in [ABDP81]). In such situations compatible discretization is not well defined. To define it we must think in terms of a more general discretization scheme. (Again, the coarsening process serves to detect a certain flaw in discretization: In this case the flaw is the lack of generality.)

When double discretization is used (§10.2), compatible coarsening means the use of such a double discretization on coarser levels, too (as indeed recommended in §10.2).