Chapter 14

Practical Role of Rigorous Analysis and Quantitative Predictions

14.1 Rigorous qualitative analysis

A good deal of the literature on multigrid consists of articles with *rigorous* analyses of the algebraic convergence. See for example the pioneering papers [Fed64a] and [Bak66b], the classical book [Hac85] and many articles in the proceedings of over thirty Copper Mountain and European conferences on multigrid methods. For a growing class of problems the basic multigrid assertion is rigorously proven, namely, that an FMG algorithm will solve the algebraic system of n equations (n unknowns on the finest grid) to the level of truncation errors in less than Cn computer operations; or at least, that $Cn \log \frac{1}{\varepsilon}$ operations are enough to reduce the l_2 norm of the error by any desired factor ε. The emphasis is on C being independent of n; it may depend on various parameters of the given differential problem. This is clearly the best one can do in terms of the order of dependence on n, hence the result is very satisfying.

The question discussed below is what role such rigorous analyses can have in the *practical* development of multigrid techniques and programs. It is an important question for the practitioner, who may wonder how much of those proofs he should try to understand. The rigorous proofs apply only to relatively simple problems and synthetic algorithms (often different from the best practical algorithms), but their main shortcoming is that they are usually unrealistic in predicting the true size of C. In most proofs C is not even estimated. This does not change the important fact that the best the proof could do in terms of C is very unrealistic: In most cases the provable constant is many orders of magnitude larger than the one obtainable in practice. In some typical cases the rigorous bound is $C \approx 10^8$, while the practical one is $C \approx 10^2$. Only in the very simplest situation (equations with constant coefficients in a rectangle) one can obtain realistic values of C,

by Fourier methods [Fre75], [Bra77a, App. C], [ST82, §8]. Recently, some analyses have been made which obtain reasonable (although still several times larger than the practical) values of C for more general problems [Bra82a],[Ver84a],[Bra86].

What can then be the practical value of the Cn results, especially those where C is unreasonably large? Usually in complexity analyses results with undetermined constants are sought in cases where the size of the constants is indeed less important. A typical result would for example be that a solution to some problem, depending on some parameter n, is obtained in $Cn!$ operations. Here C may be unimportant, since changing C by orders of magnitude will only slightly increase the range of n for which the problem is solvable. But this spirit of undetermined constants is clearly pushed way too far when the estimate is Cn, the typical constant is $C = 10^8$ and the typical value for n is 10^4 to 10^6. Here C becomes more important than n. In the practical range of n, the provable Cn result is then vastly inferior to results obtained by simpler algorithms (such as banded elimination with typically $4n^2$ operations; not to mention drastic improvements obtainable by modern sparse-matrix packages [Duf82]). Thus, the values of n for which the unrealistic rigorous result can compete with much simpler solution methods is very far out in the range of overkilling the problem. In a sense, one proves efficiency of an algebraic solution process by taking an extremely unreasonable algebraic problem.

The usual rigorous theory, being too concerned on making C independent of n, is often careless about its dependence on various problem parameters. This dependence can be hair-raising indeed, something like $\exp(\exp(\cdot))$, with as many compounded exponentials as there are stages in the proof. Hence, a very distorted picture is in fact supplied about the real complexity in solving the given differential problem.

The implied intention of "Cn" theorems with unspecified or unrealistic C is sometimes understood as follows: the rigorous analysis only tells us that a constant C *exists*, its actual value can then be determined empirically. That is, if we have calculated with $n = 10^3$ and solved the problem in 10^5 operations, say, then the rigorous proof guarantees that for n = 10^4 we would solve the problem in 10^6 operations. This understanding is wrong: The nature of the rigorous proofs is such that the information for $n = 10^3$ does not help the estimates for $n = 10^4$. The only rigorous estimate is still $C10^4$ operations, with the same unrealistic C. The guess that the number of operations for $n = 10^4$ will be 10^6 is purely non-rigorous. Even *heuristically* it does not follow in any way from the "Cn" theorem. Nothing in that theorem excludes, even heuristically, an operation count such as $Cn/(1 + 10^4C/n^2)$, for example with an astronomically large C. Thus, if one literally believed these rigorous bounds, one would not use the multigrid method in the first place. This indeed historically happened: The estimates in [Fed64b] are so bad (although only the simplest problem is considered; cf. [Bra77a, §10]), and those of [Bak66a] so much worse (even

though his constants are undetermined), that nobody was encouraged to use such methods. They were considered to be merely of asymptotic curiosity.

Several other cases from the multigrid history are known where wrong practical conclusions were derived from the asymptotic rigorous analysis. For example, non-smooth boundaries, reentrant corners in particular, gave troubles in the rigorous proofs. This led to the wrong conclusion that there are real troubles there. The practical fact is that such problems are solved to within truncation errors as easily as regular problems; even the asymptotic algebraic convergence rates in such cases can be made to attain the interior rates (see §5.7). The difficulties are purely difficulties of the proof, not of the computational process. The proof made us too pessimistic. In other cases similar proofs made people too optimistic, because their asymptotic relations did not show the real difficulties encountered in the real range. Some people did not realize, for example, the very real difficulties in solving degenerate and singular perturbation equations (in particular, indefinite problems such as $\varepsilon\Delta u + k^2 u = f$, where ε is positive but small), because these difficulties disappear for sufficiently small meshsizes. But such meshsizes are far too small to be used in practice. (The terrible growth of C as function of ε is not seen if all we are interested in is that C will not depend on n.) Fedorenko had a completely wrong idea about the *practical* meshsize ratios and the number of grids to be used. He writes: "The proposed method thus consists of a solution with the aid of an auxiliary net; if this latter is extremely large, the problem can also be solved on it by using a net of a particular type for the problem, and so on". Several similar historical examples could be given.

It is indeed not reasonable to expect unrealistic performance estimates to be of practical value. In practice we are interested in understanding the difference between one algorithm which solves the problem in few minutes CPU time and another algorithm which solves it in a few more minutes, or in hours. A rigorous result that tells us that the solution will surely be obtained within a few weeks (even years) of CPU time, cannot explain that difference. *The factors important in the proof may only remotely and non-quantitatively be related to those operating in practice.* Even in cases of much more reasonable C (such as [Bra82a]), the relative values of C in two competing approaches (e.g., V cycles vs. W cycles) does not point to their relative efficiency in practice. The rigorous proof tells us more about the efficiency of the proof than about that of the actual algorithm.

In sum, for all its pure-mathematical interest and intellectual challenge, much of the existing rigorous approach is *not a practical tool.* It has played no significant role in developing the various algorithms and insights described in this book. Its only role has generally been to *enhance our confidence in the method,* a psychological role that should not be slighted.

14.2 Quantitative predictors

On the other hand, it is strongly recommended not to restrict oneself to numerical experiments only, without *any* supporting theory. The experiments can be, and have been, quite misleading: they happen to show, for some particular cases, much better results than should generally be expected. More often, they show results much inferior to those that could be obtained, because of some conceptual mistakes and/or programming bugs. Experience has taught us that **careful incorporation of (usually non-rigorous) theoretical studies is necessary for producing reliable programs that fully realize the potential of the multigrid method.**

The purpose of the analysis should be borne in mind. We are not trying here to prove any central mathematical idea. We are engaged in a very practical problem, namely, how to solve the equations *fast.* This is in its nature as practical a problem as, say, building an airplane or understanding nuclear fission. (In fact the main purpose of the fast solvers is to aid solving such engineering and scientific problems.) One would not postpone building airplanes until rigorous proofs of their flight capabilities are furnished. Clinging to rigorous mathematics, like clinging to any secure images, may have wrong contexts. Moreover, in this business of fast solvers what one tries to a-priori estimate is nothing but the computer time (and other computer resources), which is after all exactly known in each particular case, even though *a-posteriori.* The main practical aims of theoretical understanding should therefore be:

(A) To give us realistic and **quantitative** insights to the important factors affecting the overall efficiency. The insight should be simple enough and still precise enough so that one can use it to improve our algorithms, and perhaps even to debug his programs.

(B) Even more importantly than quantitative performance prediction, one wants to know whether the performance (predicted or found empirically) is as good as one could *hope* to get (see the situation described in §0.2). Hence, the main theoretical task is to provide us with **ideal performance** figures, which the practical algorithm should then *attempt* to approach.

Local Mode Analysis (LMA) is an example of a theory constructed with these aims in mind. This is amply emphasized throughout Part I of the present book. The easiest and most practical LMA predictor is the smoothing factor (§3). A more elaborate predictor is obtained by a similar Fourier analysis of a several-level (most often two-level) multigrid cycle, thus analyzing both the relaxation and inter-grid transfers. (See §4.1 and detailed results and software for calculating such convergence factors in [Wei01]). LMA is also applicable to the FMG algorithm (§7.4), even in non-elliptic cases where the boundary plays an important role (§7.3). Although the em-

ployed Fourier analysis is rigorously valid only for equations with constant coefficients in an infinite or rectangular domains, in practice the predictions hold in a much wider class of problems, so they are routinely used in algorithm development and program debugging, even for complicated nonlinear systems.

Moreover, it was rigorously proved in [Bra91] and [Bra94] that in general uniformly-discretized linear elliptic PDE systems with piecewise smooth coefficients on general domains, the quantitatively sharp convergence factors predicted by LMA are indeed attained in the limit of small meshsizes, provided the multigrid cycle is supplemented with a proper processing at and near the boundaries. That processing was proved to cost negligible extra computer work.

Apart from mode analysis, a Coarse Grid Approximation condition has been introduced in [Bra91] and [Bra94] which is both necessary and sufficient for the multigrid algorithm to work properly. Various error norms and their relations to the orders of the inter-grid transfer operators are analyzed. Global mode analysis, required to supplement the local analysis in various border cases, is developed, and practical implications of the analysis, including practical ways for constructing and debugging multigrid solvers, are generally reviewed. A major emphasis is on the importance and practicality of adding partial (local) relaxation passes to the multigrid algorithm [Bra77a, App. A.9]: both theory and practice demonstrate that multigrid efficiency is greatly enhanced by adding special relaxation steps at any local neighborhood exhibiting unusually large residuals (cf. the adaptive relaxation rule in §5.7).

14.3 Direct numerical performance predictors

LMA played a key role in the initial understanding and the subsequent development of fully efficient multigrid algorithms. Its application has however proved increasingly cumbersome in advancing to variable-coefficient and nonlinear problems, in particular in analyzing unstructured or near-boundary processes. Much attention has therefore been given in recent years to more generally applicable and simpler-to-implement direct numerical tools for performance prediction and program debugging.

14.3.1 Compatible relaxation

Introduced in [Bra00], Compatible Relaxation (CR) is a tool to assess the potential multigrid efficiency of a given combination of a relaxation scheme and a set of coarse variables – prior to the actual construction of the inter-grid and coarse-grid operators. It is a special case of a general approach for constructing coarse-level descriptions of a given fine-level system, including non-stationary, highly nonlinear and also non-deterministic systems [Bra10]. The general coarsening criterion in all these systems states that *a*

set of coarse-level variables is considered adequate if, and to the extent that, a local processing is available to rapidly reconstruct any fine-level solution from its coarse-level values.

In the case of solving a system of local equations, such as discretized PDE, the local processing is *Compatible Relaxation* (CR), defined as a relaxation scheme that keeps the fine-level configuration compatible with the coarse one (i.e., coarsening the fine configuration yields the given coarse-level equations), at least asymptotically. For example, if coarsening is done by injection, i.e., if the coarse-level values consist of a subset of the set of fine-level value, then one kind of compatible relaxation can be a Gauss-Seidel relaxation that avoids relaxing the values of that subset. Another, more generally useful kind of CR is *Habituated Compatible Relaxation* (HCR) [Liv04], in which one alternates between relaxation sweeps and passes that reset (or asymptotically tend to reset) the coarse variables to their given values. Given a fine-level system of equations $L^h u^h = f^h$ and a corresponding relaxation scheme, together with a coarsening rule $u^H = \tilde{I}_h^H u^h$, HCR is applied as follows:

(1) Choose a particular problem that has a known solution, u_0^h, by setting $f^h = L^h u_0^h$. In the case of a linear system, simply choose $f^h = 0$, so the known solution is $u_0^h = 0$.

(2) Calculate the corresponding set of coarse values $u_0^H = \tilde{I}_h^H u_0^h$. The task of the next steps is then to rapidly reconstruct u_0^h by local processing, given f^h and u_0^H.

(3) Choose an arbitrary (e.g., random) initial approximation u^h. In the nonlinear case, u^h should be sufficiently close to u_0^h, but otherwise arbitrary.

(4) Modify u^h so that $\tilde{I}_h^H u^h$ gets much closer to u_0^H. If $\tilde{I}_h^H$ is an injection, simply introduce the values of u_0^H into u^h. If $\tilde{I}_h^H$ is such that each value of u_0^H is some local average (or any other linear combination) of values of u_0^h, make one or two passes of Kaczmarz relaxation (see §1.1) on the system of equations $\tilde{I}_h^H u^h = u_0^H$. (One sweep would suffice in the case of no overlap between these equations. In any case, the iterations should converge very fast. The important advantage here of the Kaczmarz relaxation is that it solves such a vastly-underdetermined system with minimal changes to u^h.)

(5) Relax the equation $L^h u^h = f^h$ by ν sweeps of the given relaxation scheme.

(6) Repeat steps 4 and 5, each repetition representing a cycle. Measure the rate of convergence per cycle of u^h to u_0^h.

Fast convergence (usually first tested with $\nu = 1$) implies that the set of coarse variables $\tilde{I}_h^H$, together with the given relaxation scheme, can produce

an efficient two-grid cycle. This test can therefore be a very effective tool in choosing the coarse set. It is extensively used for that purpose, particularly in constructing AMG solvers for problems on unstructured grids.

Moreover, suitable versions of HCR can be used to accurately predict the convergence rate obtainable for an actual two-grid cycle that includes the same number ν of relaxation sweeps. This in particular is achieved when $\tilde{I}_h^H$ is a Full Weighting operator (the adjoint of an interpolation operator; see §4.4). The prediction is especially accurate when $\tilde{I}_h^H = (I_H^h)^T$, the transposed of the interpolation operator used in the actual cycle, but even for other reasonable choices of $\tilde{I}_h^H$ the predictions are as accurate as those of the two-level LMA – when the latter is at all applicable. HCR offers several advantages over LMA:

(A) *Easy implementation.* Unlike LMA, which requires a substantial separate programming effort, HCR is simple to implement as soon as the relaxation routine has been constructed and a set $\tilde{I}_h^H$ of coarse variables has been proposed.

(B) *Idealized analysis.* Similar to the smoothing-rate LMA predictor (see §3.1), HCR does not really depend on the inter-grid transfers (I_H^h, I_h^H and $\tilde{I}_h^H$) or on the coarse-level operator (L^H). It thus predicts an ideal efficiency that can be attained once these operators are correctly built. HCR can therefore guide the actual construction of these operators by detecting wrong choices and implementation bugs.

(C) *Generality.* Unlike LMA, HCR is directly applicable for complex domains and/or disordered grids and/or disordered *coarse* grids and/or disordered relaxation schemes (including adaptable schemes, e.g., with extra steps near singularities) and/or irregular equations (possibly with strongly discontinuous coefficients) and/or nonlinear problems.

The experience and understanding is that the HCR analysis, with some quite obvious possible modifications, can generally predict *quantitatively well* the ideal efficiency of any normal two-grid cycle, where by "normal" we mean a cycle in which relaxation is used for reducing errors with "large residuals", while the coarse-grid correction is employed for reducing all other errors. For a (linearized) operator L^h, an error v^h is said to have "large residuals" if $\|L^h v^h\|$ is comparable with $\|L^h\| \cdot \|v^h\|$. the norms being the discrete l_2 norms. The HCR predictions should be quantitatively adequate also for the case that the cycle, together with these norms, operate only at some *subdomain*. (An example of cycling which is *not* normal is the one designed for solving non-uniformly elliptic problems, where the accuracy in approximating smooth component propagates a finite additional distance away from the boundary upon each coarse-grid correction. See §7.5.)

Similar predictions can also be extended to other types of cycles; for example, three-grid cycles. Most such predictions are not rigorous. But

their quantitative accuracy in predicting the ideal performance is no less reassuring, and certainly more directly useful, than non-quantitative rigorous analyses.

14.3.2 Other idealized cycles

In a similar spirit, a quite general numerical approach for isolating sources of inefficiency in an existing multigrid program has been developed in [BLE05]. Running that program on a particular case where the solution is known, so that the error function at each stage is also known, the performance of each part of the multigrid cycle is separately evaluated by replacing that part with an "ideal" part and comparing the (asymptotic) behavior of the original cycle with that of the idealized one. For example, as an idealized relaxation one can use an error averaging similar to that produced when relaxing the Poisson equation. Alternatively, one can apply to the error function the operator product $I_H^h I_h^H$, i.e. restriction followed by interpolation. As an idealized coarse-grid correction, one can multiply the error vector by the matrix $I - I_H^h I_h^H$, where I is the identity matrix; and so on. A collection of examples in [BLE05] shows the wide applicability and accuracy of the approach, successfully analyzing cases for which LMA is inapplicable.