MULTIGRID SOLVERS FOR NON-ELLIPTIC AND SINGULAR-PERTURBATION STEADY-STATE PROBLEMS

Achi Brandt

Department of Applied Mathematics
The Weizmann Institute of Science
76100 Rehovot, Israel

This research is sponsored by the Air Force Wright Aeronautical Laboratories, Air Force Systems Command, United States Air Force, under Grant AFOSR 81-0001.

CONTENTS

1. INTRODUCTION

This is an intermediate report of an on-going research. Its purpose is to examine the fundamental questions related to numerical stability and fast multi-grid solutions for boundary-value problems (BVPs) which are not elliptic, or in which the elliptic principal part is small. We will refer to the latter as singular perturbation problems. We are particularly interested in genuine BVPs, i.e., problems in which there is no one particular coordinate that can be singled out as the direction of evolution (time). Many steady-state problems in fluid dynamics, and in other fields, are of this type. Fast multigrid solvers are by now fairly developed for elliptic BVPs, including elliptic systems arising in steady-state fluid dynamics [14], [9]. They are far less developed and far less understood for BVPs not dominated by ellipticity.

We will consider the non-elliptic and the singular perturbation cases as one and the same. Namely, non-elliptic BVPs will be considered as limits of elliptic BVPs. In all physical situations familiar to us there is a natural, physical way of doing this. For example, the steady-state Euler equations (inviscid flows) can be considered as a limit of the full (viscous) compressible steady-state Navier-Stokes equations, with static diffusion added to the continuity equation to gain full ellipticity (see [13]). This limit is no artifice; only solutions which are obtainable as such limits are physically valid. In this way we can often avoid dealing with the time-dependent problem (except when the steady state really depends on the initial conditions) and develop very fast multigrid solvers, solving the steady-state problem in computational work equivalent to just few explicit time steps. We also avoid the need for dealing explicitly with entropy conditions and their numerical enforcement. We can indeed regard our multigrid algorithms (e.g., the FMG algorithm) as including a continuation process in which we start with viscous solutions, and gradually eliminate viscosity (see Sec. 4.4).

Artificial elliptic singular perturbations, mainly known as artificial viscosity, have long been used in numerical solutions of non-elliptic problems, either directly or in the form of up-stream differencing. They are needed because locally, on the scale of the grid, characteristic directions cannot be approximated, unless they coincide with grid directions. Here we prefer the explicit use of the physical artificial viscosity. Its use is often less expensive then the use of upstream differencing, especially in complicated systems (see [13]). Also, in multi-dimensional problems, upstream differencing is rarely the most efficient discretization to be used in multigrid relaxation, since it requires more complicated and expensive relaxation schemes (owing to accidental alignments of characteristic directions with grid directions). What really counts is not the static properties of the discretization, but the dynamic processes associated with it. We have found it advantageous to use artificial viscosity (anisotropic only in as much as the alignment between grid and characteristic lines is not accidental) in the relaxation process, and then to employ straight central differencing, with no artificial terms, in calculating the residuals to be transferred from finer grids to coarser ones. This "double discretization" scheme provides us with the accuracy of the central differencing without its instability.

Fast multigrid solvers for non-elliptic and singular perturbation fluid-dynamics BVPs were discussed before [3] [5] [17] [18] [25] [28] [30] [31] [35] [38] . The present research is a continuation of [7] and [8], and it studies more systematically and in more generality some of the basic issues concerning multigrid processes for problems not dominated by ellipticity. Starting with general concepts of stability and multigrid efficiency on one hand, and with the examination of the simplest examples on the other hand, we study the basic differences between treating such problems and the usual treatment of regular elliptic problems.

They differ in many details: The discretization methods are different, the role of relaxation is put in a new perspective, the need to treat discontinuities introduces new considerations, new techniques of residual transfer and of interpolation, new algorithms with more coarse-grid corrections and less relaxation, etc. The local mode analysis, regularly used to choose algorithms, predict their behavior and debug the programs, has to be modified, too. We have found, through this systematic study, a number of conceptual mistakes in our previous multigrid codes for non-elliptic problems, even in those which gave vast improvements over other approaches.

The insights gained, the methods developed, the rules realized, and the know how accumulated so far, are summarized in this article (see the table of content above). The study is not yet completed, various alternatives have not yet been compared, and the numerical experiments are still running. A twin article is [13], which consists of a major fluid-dynamics application, while here we present the material in terms of much simpler examples.

It is assumed that the reader is familiar with basic multigrid ideas and algorithms. Otherwise he is referred to other papers such as [5]. If he is interested in developing codes himself, he may like to consult [10] or the more updated [12].

Although we basically treat here boundary-value problems, much of this work is also relevant for time-dependent problems. For example, integrating a time-dependent problem by a difference equation of the form $Q_1^h\, u(t+\Delta t) = Q_0^h\, u(t)$ over a long time interval, the operator $L^h = Q_1^h - Q_2^h$ should be semi h-elliptic (see Sec. 3), lest the solution will eventually develop strong numerical spatial oscillations. Fast multigrid solvers can of course be used to solve the systems of equations arising from implicit time steps [5], [15]. Sometimes these solvers need to use finer grids only once in many time steps, making these implicit steps far less expensive than the usual explicit ones (see Sec. 3.9 in [36]). The multigrid adaptation structure (see again Sec. 3.9 in [36]) and coarse-grid higher-order correction techniques (see Sec. 5.5 below) are also very useful to evolution problems.

One case of singular perturbation problems covered by a separate study [16] is that of problems with highly-oscillating solutions, as encountered in acoustics, electromagnetic wave theory, etc., which requires a totally different multigrid approach.

The present work has benefited from numerical experiments performed by C. Börgers, N. Dinar, M. Donovang and D. Sidilkover. Some more experiments are reported by Börgers in [37].

2. MULTIGRID DOUBLE DISCRETIZATION

On any given grid participating in multigrid interactions, discrete approximations to the continuous operator L are used in two different processes: in the relaxation sweeps, and in calculating residuals to be transferred to coarser grids. The two discretization schemes need not be the same (Sec. 3.11 in [14]). The discretization L_0^h employed in the relaxation sweeps must be stable, but its accuracy may be lower than the one we wish to generate. The discretization L_1^h used in calculating the transferred residuals determines the accuracy of our numerical solution, but it need not be stable. This double discretization scheme is especially useful in dealing with non-elliptic and singular perturbation problems, where a certain conflict arises between stability (needed for high-frequencies, hence treated at relaxation) and accuracy (a low-frequency property, hence treated mainly by coarse-grid corrections). Thus we can use the most convenient (but sometimes unstable) central differencing for L_1^h, and add artificial viscosities (as described in Sec. 3) only to L_0^h. This will ensure stable solutions which still have the accuracy of the central differencing.

Note that such a multigrid process will not converge to zero residuals, since it uses two conflicting difference schemes. The very point is, indeed, that the solution produced is a better approximation to the *differential* solution than can be produced by either scheme.

Double discretization schemes can of course similarly be applied to *boundary conditions*; e.g., to Neumann conditions: Simple first-order schemes can be used in relaxation, while second-order Neumann conditions (which are sometimes complicated and may sometimes be unstable) can be used to transfer boundary-condition residuals to coarser grids. (There should of course be a clear separation between these residuals and interior-equation residuals.)

Two-grid double discretization schemes can also be useful in time-dependent problems (see Sec. 5.5).

Whenever a double discretization scheme is used on the finest level, it can also be used on coarser levels. This will give better coarse-grid corrections, and hence faster algebraic convergence. (In non-elliptic and singular perturbation cases the algebraic convergence is most often determined by the quality of the coarse-grid correction. See Sec. 5.1.)

2.1 FAS equations

A point to notice is how to use this double scheme on coarser levels together with the Full Approximation Scheme (FAS, described for example in [5], [7], [14], [9], and used to treat nonlinear problems or to construct local refinements). One way to write these FAS equations is as follows. Let

$$L_i^H U^H(\underline{x}) = F_i^H(\underline{x}), \quad (i = 0,1) \qquad (2.1)$$

be the two discretized equations on the finest grid. That is, these are the equations used at any point $\underline{x}$ where a finer-grid approximation $L_1^h u^h$ does not (so far) exist. Then at points $\underline{x}$ where a finer-grid approximation $L_1^h u^h$ does exist (usually for $h = H/2$), the coarse-grid equations (2.1) will use modified right-hand sides, defined by

$$F_i^H = L_i^H(I_h^H u^h) + {}_iI_h^H(F_1^h - L_1^h u^h), \qquad (2.2)$$

where I_h^H and ${}_iI_h^H$ denote fine-to-coarse transfer operators, not necessarily the same (see Sec. 6.4). Equation (2.1) with $i=0$ is used in relaxation on grid H, while $i = 1$ is used, as in the last term of (2.2), to calculate residuals $(F_1^H - L_1 u^H)$ to be transferred to the next coarser grid (2H). Otherwise FAS processing proceeds as usual (but see also Sec. 4.3).

This scheme seems to require the calculation of both F_0^H and F_1^H, and perhaps storing both. That, however, does not cost much more than usual, since F_1^H and F_0^H normally differ only on coarser grids, not on the locally-finest. Even on coarser grids, the difference between F_1^H and F_0^H in applications considered below is only in few terms (viscosity terms), hence calculating F_0^H from F_1^H is inexpensive indeed. In many cases F_1^H is calculated anyway in the process of calculating F_0^H (provided ${}_0I_h^H = {}_1I_h^H$).

2.2 Role of relaxation and algorithmic implications

The role of relaxation in multigrid processes is to smooth the error, i.e., to reduce high-frequency error components wherever their amplitude is large (large compared with the high-frequency errors produced when the low-frequency errors are interpolated). The "error" in this statement is usually thought of as the algebraic error, i.e., the difference $U^h - u^h$ between our calculated solution u^h and the exact discrete solution U^h. In view of the double discretization scheme, however, it becomes clear that what relaxation should really do is to reduce the high-frequency differential error, i.e., the difference $U - u^h$, where U is the solution to the differential equations. In fact, this is the true role of relaxation even when double discretization is not used, if what we want is to approximate U, not U^h.

Thus the important measure of efficiency of relaxation is not the algebraic smoothing rate, but the <u>differential smoothing rate</u>, the rate of reducing $\|U-u^h\|$. This is not usually recognized because the later rate is not constant; it is not solely determined by the relaxation scheme, but also depends on u^h. When u^h is closer to U than to U^h, the differential smoothing rate can in fact be negative. The algebraic rate, on the other hand, depends only on the relaxation scheme and can easily be calculated. But it should be kept in mind that this calculated rate is important only as long as the high-frequencies in $U-u^h$ are large compared with those in $U-U^h$. Below this level the algebraic rate may mislead (see end of Sec. 3.3 and Sec. 7.2).

This observation leads to another: The stability of L_0^h is not important by itself. This is a static property, while what counts in the final analysis is the overall dynamic process of relaxation. Good stability measures (see Sec. 3) are only <u>tools</u> for obtaining efficient (differential) smoothing rates, which are the only ultimately important measures. Such rates are relatively easy to calculate, since they are local. Thus, to study L_0^h and associated relaxation schemes one can use local mode analysis (see Sec. 5), and it is enough to consider principal and sub-principal terms (see Sec. 5.6).

One qualification, though. If p, the overall approximation order obtained, is higher than the approximation order p_0 of L_0^h (due to double discretization), then the influence of relaxation on low-frequency components may be significant. Suppose the relaxation has <u>convergence order</u> r_0; that is, every mode $\exp(i\,\underline{\theta}\cdot\underline{x}/h)$ has convergence factor $\mu(\underline{\theta})$ per relaxation sweep such that $|1-\mu(\underline{\theta})| \leqslant O(|\theta|^{r_0})$. Each relaxation sweep then introduces a relative change $O(|\underline{\theta}|^{p_0+r_0})$ to a higher-order approximations. To retain the approximation order p we must therefore have $r_0+p_0 \geqslant p$. To avoid degrading even the <u>coefficient</u> in the p-order approximation, it is even desired to have $r_0 + p_0 > p$. This is particularly important in non-elliptic and singular perturbation problems, where some schemes may have $r_0 = 0$; e.g., (locally) downstream relaxation schemes with upstream differencing for (locally) hyperbolic systems (or reduces systems).

There are three important algorithmic implications to this in case the approximation order p used in L_1^h does exceed, or even equals $r_0 + p_0$. First, one should <u>not perform relaxation sweeps after returning from the coarse grid</u>, only before it or in between two switches to the coarser grid. Secondly, since the coarse-grid correction should improve the approximation order, not just reduce the algebraic error by a fixed factor $(2^p$ or so), it is safer to use <u>W cycles</u> rather than V cycles (see Sec. 6.1). Thirdly, since the correction we interpolate from the coarser grid is $O(h^{p_0})$, and we like it to have errors no larger than $O(h^p)$, <u>the interpolation order must be at least</u> $p - p_0$, preferably $p - p_0 + 1$.

Other implications of the double discretization is the need to use <u>full</u> weighting in transferring residuals to coarser grids (see Sec. 6.4), and some advantages and disadvantages of red-black relaxation schemes (see Sec. 6.3).

In case $p > 2p_0$ it is not enough to have an order-p_0 solution on the coarser grid. Hence, if L_0^h is used in relaxing on the coarser grid, a double discretization scheme must be used there, too, otherwise the approximation order p will not be attained. To avoid degrading even the <u>coefficient</u> of the order-p approximation, it may be desired to use coarse-grid double discretization even when $p = 2p_0$. It is not needed when $p < 2p_0$.

2.3 <u>Fine-grid defect corrections are not useful</u>.

Instead of raising the order p_0 of the relaxation operator L_0^h, an alternative approach seems to be the use of defect corrections on the fine grid. We correct, that is, the right-hand side of the equation by the function $L_0^h u_0 - L_2^h u_0$, where L_2^h is a higher-order (order $p_2 > p_0$) operator and u_0 is our last approximate solution, and then we relax on the resulting equation with its low-order operator L_0^h. This would introduce changes $O(|\underline{\theta}|^{p_2+r_0})$ instead of $O(|\underline{\theta}|^{p_0+r_0})$ per relaxation sweep. On closer examination, however, this idea is not useful: After the defect correction, high-frequency errors are no longer suitably reduced by further relaxation sweeps -- defeating the very purpose of relaxation. For example, if L_2^h is unstable, i.e., if there is a high-frequency error v such that $L_2^h v = 0$, then it is easy to see that, after the defect correction, the error v shows no residuals and hence it cannot be affected by relaxation.

3. STABILITY MEASURES AND RELAXATION

The discrete operator L_0^h used in relaxation needs to be stable, although it is not exactly the stability of this static operator which is important, but rather the overall process of relaxation: What counts in the final analysis is the overall efficiency with which the relaxation sweeps reduce large high-frequency error components and avoid from amplifying low-frequency ones (cf. Sec. 2.2). Still, the first step in constructing good relaxation schemes is to have discrete operators with good measures of stability.

The discretization of singular-perturbation BVPs is usually guided by one-dimensional ideas which are sometimes useful in higher dimensions, but are also misleading. These include exponential fitting methods (extendable only to very special higher dimensional situations), the concept of positive-type difference operators (which is often a useful guide, but is generally neither necessary nor sufficient for stability, nor is it always desirable), and up-stream differencing techniques. The latter are usually very useful, but often they call for unnecessarily complicated relaxation schemes (see Sec. 3.3), and they should be replaced by a more precise rule near discontinuities (see Sec. 4.2). Moreover, none of the above concepts gives general and precise rules for numerical stability of BVPs. A general stability theory for purely elliptic singular perturbation equations (where both the reduced equations and the singular perturbation are elliptic) has been advanced by L.S. Frank [20], [21], but we are mostly interested in cases where the reduced problem is not elliptic, which is the usual situation in fluid dynamics and other applications.

Like Von-Neumann analysis (and its extensions by Kreiss and others) for time-dependent problems, precise rules for stability are obtained by mode analyses. It turns out however that, especially for singular-perturbation problems, the distinction between stable and unstable discrete operators is not so important. More important is the measure of stability (at a given meshsize): When the stability measure is low the scheme is still formally stable, but its actual behavior can be intolerably bad.

A simple example, typical to more complicated systems in fluid dynamics and other fields, is the d-dimensional diffusion-convection equation

$$L_\varepsilon U \equiv -\varepsilon\Delta U + \sum_{j=1}^{d} a_j \frac{\partial U}{\partial x_j} = F(\underline{x}), \qquad (3.1)$$

discretized by central differencing as

$$-\varepsilon\Delta^h U^h + \sum_{j=1}^{d} a_j \hat{\partial}_j^h U^h = F^h , \qquad (3.2)$$

where Δ^h is the (2d+1)-point Laplacian and $\hat{\partial}_j^h U^h(\underline{x}) = [U^h(\underline{x}+\underline{h}_j)-U^h(\underline{x}-\underline{h}_j)]/(2h_j)$, $\underline{h}_j = h_j\underline{e}_j$, $\underline{e}_j$ being the unit vector, and h_j the meshsize, in direction x_j. For any $\varepsilon > 0$ (3.2) is still formally stable: In contrast to a common folklore, no dramatic loss of stability occurs when the equation loses its "positive-type", i.e., when ε becomes just smaller than $\bar{\varepsilon}^h = \frac{1}{2}\max(h_j|a_j|)$. But for $0 < \varepsilon << \bar{\varepsilon}^h$ the solution may show large (even if formally bounded) numerical oscillations; its behavior is evidently deteriorated.

A general measure of numerical stability can be defined as the minimal amplification ratio among all acceptable modes, where the amplification ratio of a mode is defined as its differential amplification factor divided by its discrete one. For low-frequency modes (large wavelengths compared with meshsize) this ratio is kept close to 1 by consistency. The stability measure can therefore be defined in terms of high-frequency modes alone, where no direct reference to the differential operator is required other than the distinction between several different situations concerning the alignment of grid lines with characteristic lines.

3.1 h-Ellipticity

One situation is the general, or indiscriminating grid, where no particular relation can be assumed between grid directions and characteristic directions. Numerical stability of the difference operator L^h can then be required to be isotropic, and can therefore be measured by its h-ellipticity measure

$$E^h(L^h) = \min_{\rho\pi\leq|\underline{\theta}|\leq\pi} |\tilde{L}^h(\underline{\theta})| \; / \; |L^h| \; , \qquad (3.3)$$

where $\tilde{L}^h(\theta)$ is the symbol of L^h (i.e., $L^h \exp(i\underline{\theta}\cdot\underline{x}/\underline{h}) = \tilde{L}^h(\theta)\exp(i\underline{\theta}\cdot\underline{x}/\underline{h})$), $\underline{\theta} = (\theta_1,\ldots,\theta_d)$, $|\underline{\theta}| = \max|\theta_j|$, $\underline{\theta}\cdot\underline{x}/\underline{h} = \sum\theta_j x_j/h_j$ and $|L^h|$ is some measure of the size of the coefficients of L^h, e.g., $|L^h| = \max|\tilde{L}^h(\underline{\theta})|$. In case L^h is a system of q difference operators (operating on q unknown grid functions) then $\tilde{L}^h(\underline{\theta})$ is a $q\times q$ matrix and $|\tilde{L}^h(\underline{\theta})|$ should be interpreted as a measure of its non-singularity (e.g., its determinant, or its smallest eigenvalue). The range $|\underline{\theta}| \leq \pi$ is the range of frequencies $\underline{\theta}$ for which the Fourier component $\exp(i\underline{\theta}\cdot\underline{x}/\underline{h})$ is visible to a grid with meshsize $\underline{h}$ (i.e., any component outside this range coincides on the grid with a lower component inside this range). The range $\rho\pi \leq |\underline{\theta}| \leq \pi$ in (3.3) is the range of "high-frequency" components on grid h; i.e., components visible to grid $\underline{h}$ but invisible to a coarser grid with meshsize $\underline{h}/\rho$. In multigrid applications $\rho = \frac{1}{2}$. See more about ellipticity and h-ellipticity in [14] and [8].

A non-elliptic differential operator L can be approximated by an h-elliptic difference operator L^h (i.e., such that $E^h(L^h)$ is not small), by adding an $O(h^p)$ elliptic perturbation to a higher-order (e.g., central) approximation L_1^h . The simplest such perturbations are the physical perturbations, defined on a square grid $(h_j = h)$ as follows: Let $L = \lim_{\varepsilon \to 0} L_\varepsilon$, where L_ε is elliptic and where the only physically acceptable solutions to L are those obtainable as limits of solutions to L_ε . The h-elliptic approximation L^h to L is obtained as a higher-order (e.g., central) approximation to $L_{\varepsilon(h)}$, where $\varepsilon(h)$ is just large enough to yield a good h-ellipticity measure. For example, (3.2) is made h-elliptic by taking

$$\varepsilon(h) = \max[\varepsilon, \beta h a] , \quad a = \max_{1 \leq j \leq d} |a_j| , \qquad (3.4)$$

where $0 < \beta = O(1)$. A generalization for rectangular grids is (3.7) below.

Such physical perturbations are the safest device to ensure physicality of solutions, i.e., to allow only physically acceptable discontinuities. By magnifying the physical singular perturbation terms to h-ellipticity we smear the width of these discontinuities to the scale h. We let the physics play its full role on a magnified scale visible to our grid. In the above example, if ε is sufficiently small, then using $-\varepsilon(h)$ instead of $\varepsilon(h)$ in (3.2) would still produce an h-elliptic operator. But only preserving the correct sign ensures that the discrete boundary layers will appear exactly at the same end of each characteristic curve as in the differential solution.

The physical perturbations can easily be written in conservation form. It is in fact their natural form, since all the terms are derived from the physical conservation laws. Higher-order physical approximations (cf. Sec. 2.4) can therefore also be written in conservative form (see Sec. 3.10.4 in [14]). Generally, these physical schemes are simple, direct and inexpensive to implement, especially in complicated systems. For compressible Navier-Stoke equations, for example, they are considerably simpler than upstream differencing (See [13]).

The main advantage of h-elliptic operators in multigrid processing is the simple and efficient relaxation schemes available with them, as expressed in the following theorem.

THEOREM 3.1. Good (i.e., O(1)) h-ellipticity measure is a necessary and sufficient condition for the existence of "purely local" (i.e., pointwise and direction-free) relaxation schemes with good (i.e., bounded away from 1) smoothing factors.

The general proof (see [8]) uses a general-purpose relaxation scheme, called simultaneous fully-distributed under relaxation, which is not usually the best one for any specific case. But in all specific examples of h-elliptic operators which we have so far considered, including complicated ones like full Navier-Stoke systems [13], the best relaxation schemes are also purely local, and their smoothing factors are typically between .3 and .4 . Moreover, these relaxation schemes are highly parallelizable and easily vectorizable: The relaxation can in fact be carried out simultaneously at (typically) half the grid points (see [11]). And when the method of "physical" h-ellipticity is used, the discrete operators are all central, containing no branching according to velocity directions, hence directly vectorizable.

A known disadvantage of the above operators is their low accuracy: their error is O(h). In our double discretization scheme (Sec. 2) this is not crucial; the error is corrected to $O(h^2)$ or even $O(h^3)$ by the coarse-grid correction. Still higher orders can be produced by certain generalizations discussed in Sec. 3.4 .

A more serious disadvantage of h-elliptic operators is their large cross-stream viscosity, which is especially undesired near boundary layers and other discontinuities. At such locations one often needs semi-h-elliptic operators (Secs. 3.2 and 3.3) or special formulae (Sec. 4.2), or local refinements (Sec. 4.4), depending on the situation.

3.2 Semi h-ellipticity

We have discussed the numerical stability on an indiscriminating grid. Another situation is that of a characteristic grid, namely, when characteristics of the differential operator (or those of the reduced operator, in case of singular perturbations) consistently coincide (or nearly coincide) with grid directions. We can then allow high-frequency modes to be carried from boundaries, and be even amplified, along characteristics, as they are in the differential case. That is, the numerical solution can be allowed to change wildly in cross-characteristic directions, but not in the characteristic direction itself. More generally, in a d-dimensional domain, suppose characteristic lines proceeds only (or mainly) in a subset $S \subset \{1,\dots,d\}$ of grid directions; i.e., if $k \notin S$ then x_k is constant (or slowly varying) along every characteristic line. We would then like to forbid high-frequency oscillations only in the directions of S. This kind of stability we will call semi h-ellipticity *in directions* S, or briefly S-*h-ellipticity*. Its measure can be defined as

$$E^h_S(L^h) \;=\; \min_{\rho\pi \leq |\underline{\theta}|_S \leq \pi} |\tilde{L}^h(\underline{\theta})|/|L^h| , \qquad (3.5)$$

where $|\underline{\theta}|_S = \max_{j\in S}|\theta_j|$. Full h-ellipticity (3.3) is the special case $S=\{1,\ldots,d\}$. We say that L^h is S-*h-elliptic* if it has good S-h-ellipticity measure: $E^h_S(L^h) = O(1)$. If $S_2 \subset S_1$ then clearly $E^h_{S_1} \leq E^h_{S_2}$, hence S_1-h-ellipticity entails S_2-h-ellipticity. If $E^h_j(L^h) = E^h_{\{j\}}(L^h) << O(1)$ we will say that L^h is *weakly coupled in direction* j .

A simple and safe way to construct semi-h-elliptic operators is again by magnifying the elliptic singular perturbation (see Sec. 3.1), except that the magnification can be anisotropic, with "viscosity" added mainly (or only) in the semi-h-ellipticity directions S.

For example, a semi-h-elliptic operator for approximating (3.1) is

$$-\varepsilon \sum_{j\notin S} \partial^h_{jj} - \sum_{j\in S} \overline{\varepsilon}^h_j(\underline{x})\,\partial^h_{jj} + \sum a_j(\underline{x})\,\hat{\partial}^h_j \; , \qquad (3.6)$$

where

$$\partial^h_{jj}\, U^h(\underline{x}) \;=\; [U^h(\underline{x}-\underline{h}_j) - 2U^h(\underline{x}) + U^h(\underline{x}+\underline{h}_j)]/h_j^2 \; ,$$

$$\overline{\varepsilon}^h_j(\underline{x}) \;=\; \max[\varepsilon,\, \beta\frac{h_j^2}{h_k}|a_k(\underline{y})| ,\; \overline{\beta}\,\frac{h_j^2}{h_k^2}\,\varepsilon] , \qquad (3.7)$$

with $\beta = O(1)$, $\overline{\beta} = O(1)$, and the max being taken over all $1 \leq k \leq d$ and over all points $\underline{y}$ in a neighborhood of $\underline{x}$. Usually it is enough to take $\underline{y} = \underline{x}$, except near a *stagnation point* (at which $\max|a_j|$ is much smaller than at neighboring points), where the larger ε^h_j is needed both for stability and for multigrid convergence (see [37]). On a square grid $(h_j=h)$, (3.7) simplifies to (3.4).

Such physical artificial semi-elliptic terms are again important in order to obtain correct selection of discontinuities. For that purpose it is usually enough to have the correct sign in the streamwise viscosity; cross-stream discontinuity (contact discontinuity) does not depend on the sign of such terms. Hence all we need is that S contains the stream direction.

We distinguish between two kinds of semi-h-elliptic operators, depending on two ways of choosing the set of strong-coupling directions S. First, S can be chosen separately at each point, taking into account only the characteristic directions at that point. We can call it *pointwise semi h-ellipticity*. This is the kind of semi h-ellipticity produced by *upstream differencing*. It gives full h-ellipticity at points where the flow happens to be oblique to all grid lines, and $\{j\}$-h-ellipticity at points where the flow is in the x_j direction. Thus the cross-stream viscosity in upstream differencing is inconsistent. Its size depends on the alignment between grid lines and stream lines, which may vary over the flow domain.

Another way is to choose S *uniformly* over a domain. For example, in (3.6) the uniform set S will be the set of all directions in which the reduced equation $(\varepsilon = 0)$ has strong couplings at some point; i.e., $k \notin S$ only if, at *all* points $\underline{x}$ of the domain, $|a_k(\underline{x})|/h_k$ is small compared with some $|a_j(\underline{x})|/h_j$. Such *uniform* S-*h-elliptic* operators produce consistent cross-stream viscosity.

Pointwise relaxation schemes cannot generally provide good smoothing rates with S-h-elliptic operators. They can only provide good S-*smoothing rates*. That is, they can guarantee amplification factors bounded away from 1 only for modes $\exp(i\underline{\theta} \cdot \underline{x}/\underline{h})$ highly oscillating *in* S *direc-*

tions (i.e., with $\max_{j\in S}|\theta_j| \geq \rho\pi$). Indeed, we have the following extension of Theorem 3.1.

THEOREM 3.2. S-*h-ellipticity is a necessary and sufficient condition for the existence of pointwise relaxation with good* S-*smoothing factors*.

S-smoothing is enough for a multigrid algorithm which employs the corresponding semi coarsening (i.e., the coarser grid is coarser only in directions S. See Sec. 3.1 in [10]). To efficiently obtain *full* smoothing, however, relaxation schemes for S-h-elliptic operators should use the distinguished direction S , by relaxing in S-*blocks*; that is, relaxing simultaneously grid points which only differ in their S coordinates. For such schemes we can further generalize the above theorems as follows.

THEOREM 3.3. *Let* S *and* S' *be two sets of directions*: S, S' $\subset\{1,\ldots,d\}$. *A necessary and sufficient condition for the existence of an* S-*block relaxation scheme with good* S'-*smoothing rates is that the discrete operator* L^h *is uniformly coupled in all* S'-S *directions; that is*, $E^h_{S',S}(L^h) = O(1)$.

$E^h_{S',S}$ is the *measure of uniform coupling in* S' *modulo* S. It can be defined by

$$E^h_{S',S}(L^h) = \min_{\substack{\rho\pi\leq|\underline{\theta}|_{S'}\leq\pi \\ \theta'_j=\theta_j \text{ for } j\in S}} |\tilde{L}^h(\underline{\theta})| \,/\, |\tilde{L}^h(\underline{\theta}')| \,. \qquad (3.8)$$

Theorem 3.3 states, in other words, that the S'-smoothing factors, produced for L^h by a suitable S-block relaxation, are bounded away from 1 by a quantity which depends only on $E^h_{S',S}(L^h)$.

3.3 What to use where

All the above theorems deal of course with constant-coefficients operators. They still describe the essential smoothing properties even when the coefficients of L^h gradually change over the grid, because smoothing is essentially a local process.

A basic difficulty with pointwise S-h-ellipticity seems to emerge: When the coefficients change over the domain, the set S changes too, hence full smoothing may require different S-block relaxation in different subdomains. Thus, for example, upstream differencing in a flow which occasionally aligns itself with all possible grid directions would require (in a 3-dimensional problem) three plane relaxation sweeps, which is inconvenient to program and less efficient. In that situation full h-ellipticity seems preferable, because it allows simple pointwise relaxation, every sweep being fully efficient at all subdomains. Moreover, in that situation full h-ellipticity does not introduce considerably more artificial viscosity than pointwise S-h-ellipticity, since the latter avoids cross-stream viscosity only rarely and accidentally, at those particular points where the stream happens to be aligned with the grid.

On the other hand, in situations where the alignment between characteristic directions and grid directions is not accidental, we can apply the corresponding S-h-ellipticity without having to use complicated relaxation. If the streamlines are all in the x_j direction, for example, we can use {j}-h-elliptic approximations throughout, together with {j}-block relaxation (line relaxation with lines in the x_j directions).

Another example: If the stream is always in planes $\{x_3 = \text{constant}\}$ but otherwise has general directions (not always aligned with either x_1 or x_2), then uniformly {1,2}-h-elliptic approximations should be used, together with the corresponding plane relaxation. Instead of plane relaxation it may be more convenient to use point relaxation together with semi-coarsening, the next coarser grid being coarser only in the x_1 and x_2 directions, but retaining the same meshsize h_3. (See Sec. 3.1 in [10]). Notice that each of these two alternatives depends on uniform S-h-ellipticity. If upstream differencing were used we would have to use either several directions of plane relaxation or several semi-coarsening procedures successively.

Another example, most common in applications, is the non-accidental alignment between flow and grid directions <u>near boundaries</u>. Boundaries are often set to lie in grid planes, and the flow is often parallel to the boundary. Assume now that all boundaries parallel to the flow are in grid planes, whereas the main flow away from boundaries has no one particular alignment with the grid (vortex flow, for instance). In (3.1), for example, this is modelled by the case of a 3-dimensional $(d = 3)$ box domain, with $|a_j(\underline{x})| << \max_k |a_k(x)|$ near the boundaries $\{x_j = \text{constant}\}$. Can we then ignore the alignment near boundaries and use full h-ellipticity throughout? Usually we cannot: Usually there are boundary layers (i.e., discontinuities) along the boundaries, hence the cross-stream atrificial viscosity that will be created there by full h-ellipticity is by far larger than any other artificial viscosity introduced anywhere in the problem. It will therefore be far better to use near such a particular plane the corresponding plane-h-ellipticity (e.g., {1,2}-h-ellipticity near the boundary $\{x_3 = \text{constant}\}$. Since we may have several such boundaries, in different directions, we cannot use just one plane-h-ellipticity throughout. Also, away from the boundary we prefer, as before, the full h-ellipticity with its simpler pointwise relaxation. The best procedure seems thus to be to use full h-ellipticity throughout the domain, except on particular grid planes adjacent to parallel boundaries, where the corresponding plane-h-ellipticity should be used, avoiding straddling the boundary layer. (This is a special case of a more general rule discussed in Sec. 4.2.) The relaxation will then be a simple point relaxation throughout the bulk of our grid, supplemented with a special plane relaxation at each of the particular grid planes adjacent to parallel boundaries. We do not have to sweep with all these particular plane relaxation across the entire grid. This scheme is convenient to program: The special relaxation is used exactly where the special differencing is used, and only there.

Often in the above situation one can avoid plane relaxation altogether. The boundary layers are often thin compared with the meshsize. We can then suppress them completely and let the external flow extend up to (and including) the boundary. The plane-h-elliptic operator can then be used only on the boundary itself. If the same procedure is also used for the same boundary on the coarser grid, then no cross-stream smoothing will be needed there, hence simple point relaxation throughout the domain, including the boundary, will efficiently give us all the smoothing we need.

Finally, contrary to a possible impression from the discussion above, we should point out that pointwise direction-free relaxation (red-black scheme, for example) with <u>fully</u> h-elliptic operator, L_f^h say, is not always better than with <u>semi</u>-h-elliptic operator, L_s^h say. The latter (with only streamwise artificial viscosity) has a worse smoothing rate, that is true; but this is an algebraic smoothing (see Sec. 2.2). If the cross-stream high-frequency differential error components have amplitudes comparable to those of the cross-stream high-frequency solution components (as it is usually the case when an FMG algorithm with suitable interpolations is used) then the differential smoothing by L_s^h is generally not worse, and near boundaries it is even better, than by L_f^h.

The optimal amounts of upstream and cross-stream artificial viscosity in FMG algorithms (Sec. 6.2) should be studied quantitatively by local mode analyses and numerical experiments (see Sec. 7.1).

3.4 Higher-order techniques

The stable approximations L_0^h discussed above are (for non-elliptic or singular perturbation problems) all first-order methods, carrying $O(h)$ errors. In view of the multigrid double-discretization option, they may also be used in schemes that obtain higher approximation orders. To obtain still higher orders, however, we have to increase either the approximation order of L_0^h or the convergence order of the associated relaxation scheme (see Sec. 2.2). The convergence order can be raised by using distributive relaxation schemes equivalent (in the sense of Sec. 4.1 in [8]) to a classical relaxation of a higher-order operator. It seems simpler to directly use a higher-order approximation to L_0^h.

Higher-order semi-h-elliptic approximations can be constructed by using formulae from Sec. 5.2 of [7] and also, for conservation forms, from Sec. 3.10.4 of [14]. They give stable operators of any order (which for orders higher than 2 cannot be completely one-sided). For reasons explained above, however, one would often prefer "physical" approximations. Higher-order physical approximations are constructed similarly to the first-order ones (Sec. 3.1), with one important difference: Instead of artificial viscosity of the same form as the physical one, higher-order artificial viscosity, with the same sign, should be used. For example, a p-order S-h-elliptic approximation to (3.1) is given, in case of a square grid $(h_j = h)$, by the operator

$$L^{(p)} = -\varepsilon\Delta^{(p)} - \beta a h^{2s-1} \sum_{j\in S} (\partial^h_{jj})^s + \sum a_j \, \partial^{(p)}_j \, , \qquad (3.9)$$

where $\Delta^{(p)}$ and $\partial^{(p)}_j$ are p-order approximations to Δ and to $\partial/\partial x_j$, respectively, $s = [\frac{p}{2}] + 1$, ∂^h_{jj} is as in (3.6), and as before $a = \max|a_j|$, $0 < \beta = O(1)$. The choice of S is still governed by the same considerations as above (Sec. 3.3). Note that (3.9) is not R-elliptic (cf. [14], Sec. 3.6). It would be R-elliptic if the sign of β were $(-1)^s$, but for even s this would not be physical: Discontinuities will be admitted at the wrong end of characteristic curves.

High-order approximations are no better than lower order ones near discontinuities, or generally when the solution has O(1) change over a meshsize. They may cause less smearring of the discontinuity but may on the other hand cause overshoots and oscillations leading sometimes to nonlinear instabilities (see [2]) or to nonphysical solutions (see [23]). See more on the treatment of discontinuities in [22] and in Sec. 4 below.

3.5 Streamwise and optimized artificial viscosity

Streamwise artificial viscosity has been suggested in [24] in terms of finite element formulation. It can similarly be introduced via finite differences. For example in case of (3.1) this can be done by adding, to a central approximation, an artificial-viscosity term of the form

$$-\frac{\beta h}{a}\left(\sum_{1}^{d} a_j\,\partial_j^F\right)\left(\sum_{1}^{d} a_j\,\partial_j^B\right) U^h , \tag{3.10}$$

where $\partial_j^F U^h(\underline{x}) = [U^h(\underline{x}+\underline{h}_j) - U^h(\underline{x})]/h_j$ and $\partial_j^B U^h(\underline{x}) = [U^h(\underline{x}) - U^h(\underline{x}-\underline{h}_j)]/h_j$. This term is an approximation to a second derivative in the stream direction. It is advantageous when the solution is smooth in all directions and has a special smoothness in the characteristic direction; e.g., a smooth solution which is constant along characteristic lines. (The successful examples in [24] are of this kind.) In such cases the performance of (3.10) is comparable to a higher-order scheme, since the cross-stream viscosity it creates is of third order. But (3.10) is simpler than a regular higher-order scheme.

Near discontinuities, however, the scheme is not better than other schemes, since its cross-stream viscosity does not really vanish. In fact, in case $d=2$ and $a_1=a_2=1$ for example, the artificial viscosity created by (3.10) for the highest streamwise oscillations $(-1)^{(x+y)/h}$ exactly equals that for the highest cross-stream oscillations $(-1)^{(x-y)/h}$. A similar equality holds in the first-order scheme (3.2) - (3.4). Hence, in order to control the streamwise oscillation (which is the purpose of artificial viscosity), cross-stream dissipation must be created by (3.10) which is the same (near discontinuities) as the one created by the first-order scheme. Also, even for smooth solutions, (3.10) is not much better than the first-order scheme in case the physical cross-stream viscosity is comparable to the physical streamwise viscosity, because in that case the artificial cross-wind viscosity avoided by using (3.10) is comparable to the artificial viscosity introduced anyway in the stream direction. Moreover, (3.10) has the usual disadvantage of upstream differencing: The need, at least for the purpose of algebraic smoothing, for a multi-direction hyper-plane relaxation in case the coefficients vector $\underline{a}$ accidentally aligns itself with different grid directions (see Sec. 3.3).

Another possible approach is to construct difference operators based exactly on the requirement that, for a given stencil and a given amount of streamwise stability, the cross-stream viscosity will be minimal. In the above example ((3.1) with $d=2$, $a_1=a_2=1$ and $\varepsilon=0+$) the optimal nine-point differencing on a square grid will use only the three points (x_1-h, x_2-h), (x_1, x_2) and (x_1+h, x_2+h). The cross-stream artificial viscosity will truely vanish, unlike the one produced in this case by

(3.10). Such will not however be the case for general values of a_1 and a_2. Also, in case of variable $\underline{a}(\underline{x})$, the optimal L^h at each point may require too laborious calculations. The usual disadvantage of upstream differencing may be enhanced: Line relaxation sweeps may be required not only in the two coordinate directions but also in the two diagonal directions.

4. DISCONTINUITIES

4.1 Unidentified or untraced discontinuities

Not all discontinuities are identified as such by the numerical solution process. Some shocks, for instance, cannot be identified because they are too weak, just comparable to normal increments of the solution on the given grid; or too numerous. Some turn out to be clear strong discontinuities, but at the initial stage of calculation we may not know them, or may not know their location.

A basic rule in treating untraced discontinuities is to use difference equations in conservation form (conservative schemes), corresponding to the conservation laws described by the differential equations. See [26]. The physical h-elliptic and semi-h-elliptic operators in Sec. 4 can all easily be constructed in conservation form. This form implies that the jump conditions across discontinuities (derived from integration of the conservation laws) are still satisfied by the numerical scheme, even where the differential equation is not well approximated. As a result, convergence of the discrete solution to the differential one is guaranteed. It may, however, be a slow convergence.

Unidentified or untraced discontinuities usually have $O(h)$ errors in their discretization, reflecting for instance the $O(h)$ errors that must enter in determining the location of the discontinuities. More precisely: if the difference equations straddle the discontinuity, an error equivalent to $O(h)$ artificial viscosity must enter, because the discontinuity is smeared at least along one meshsize. More generally, an $O(h)$ error must enter since along an $O(h)$ part of the domain there enters an $O(1)$ error in the equations. The error may seem even larger if measured without care (see Sec. 4.5). Hence, the O(h) physical perturbations discussed above should perhaps be preferred near untraced discontinuities. Higher-order approximations near a discontinuity would be wasteful. They would not provide there a better approximation, and they may even introduce undesired oscillations.

One way to reduce the O(h) artificial viscosity at such discontinuities is by reducing h; i.e., by adaptive discretization schemes that automatically decide how far to go with local refinements, without getting into the task of deciding whether and where a clear discontinuity is involved (see Sec. 4.4).

Another way is to try to use difference equations that will automatically avoid straddling discontinuities. Upstream differencing tend to do just that. Not always, though. In higher-dimensional $(d \geq 2)$ problems, a discontinuity will often be straddled by upstream differencing, although with less likelihood then when all points participating in central differencing are involved. A similar tendency not to straddle discontinuities can also be obtained in full h-elliptic operators, if the size of the artificial terms is suitably chosen. For example, if $\beta = \frac{1}{2}$ is used in (3.4), then the coefficient in (3.2) of the most forward point of the stencil will vanish, and the chance of not straddling a discontinuity will decrease, except when there is a consistent (non-accidental) alignment between grid lines and streamlines. In the latter case full h-ellipticity is not recommended anyway (cf. Sec. 3.3).

The chance of straddling a discontinuity by such partly upstream h-elliptic operators is still higher than by usual upstream differencing. One has to weigh this disadvantage against the considerations of Sec. 3.3.

When the discontinuity is weak enough, the error caused by straddling it may be tolerable. If a strong discontinuity emerges, and is identified, one can then switch to the (superior) type of treatment described in the next section.

4.2 Known and traced discontinuities

The exact location of a discontinuity or a potential discontinuity is often known. For example, boundary discontinuities (boundary layers not resolved by the grid) are expected along boundaries. A strong internal discontinuity, shock, interface, etc., may emerge and be identified at some previous stage of the calculations. Even when its exact location is not given, it may be traced, i.e. special variables (unknowns) may be added to describe its location.

Special equations, corresponding to these special variables, are then added to the system. They describe the jump conditions. In our simple example (3.1) the jump condition is simply that an interior discontinuity rides along a characteristic; i.e., its direction at the point $\underline{x}$ is $(a_1(\underline{x}),\dots,a_d(\underline{x}))$. In fluid dynamics the jump conditions are known as Rankine-Hugomiot equations, and discontinuity tracing is called "shock-fitting" (see e.g. [29], Sec. 12.9).

When the exact location is known, or traced, superior discretization schemes near the discontinuity are possible, even without local refinements (which in some particular cases may still be needed - see Sec. 4.4).

Discretization schemes which use the unknown location of a discontinuity as an explicit variable should be superior basically because that variable is well-posed, whereas the solution function itself is an ill-posed parameter at the discontinuity. Some authors fail to recognize this: They develop sophisticated numerical schemes for treating discontinuities, which implicitly or explicitly involve their identification, without recognizing that when this has been done the most effective and perhaps simplest next step is to assign special variables to their location.

The basic rule for optimal differencing near a known or traced discontinuity is obvious, but often overlooked:

BASIC RULE. The difference equations should not straddle a discontinuity whose location is known or traced. Only physically small terms (like those with coefficient ε in (3.6)) may violate this requirement.

What is often overlooked is that this rule should override upstream differencing and even h-ellipticity considerations. Although upstream schemes tend to obey this rule, they quite often violate it, too. When they do, there is no reason to use them.

Owing to tracing, h-ellipticity can partly be ignored along the discontinuity. More precisely, needed is only h-ellipticity in directions parallel to the discontinuity, not perpendicular to it. This is easy to do in case the discontinuity runs along a grid hyperplane (see Sec. 3.3). But it can also be done for more general discontinuities.

Consider for example equation (3.1) in two dimensions $(d=2)$, at a point (A) with a traced contact discontinuity near it, and assume for simplicity that ε is infinetesimally small ($\varepsilon = 0+$; only its sign matters), and that $a_1^2 + a_2^2 = 1$. Assume also for definiteness that $a_2 > a_1 > 0$, giving the stream direction as shown by the arrow in Fig. 1. Upstream differencing at the point A would then include points B and D, violating the basic rule. We should instead base our differencing at A on the stencil (A,D,E), with the operator $L^h = a_1 \partial_1^F + a_2 \partial_2^B$, where ∂_j^F and ∂_j^B are as defined for (3.10). Comparing this operator to central differencing,

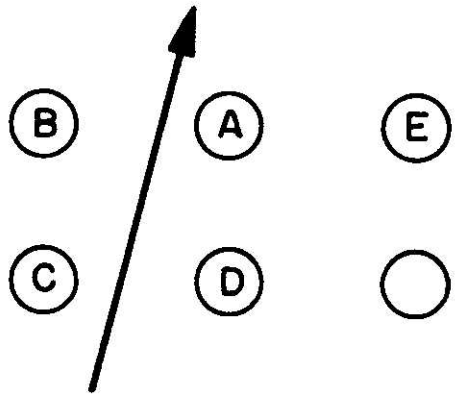

FIGURE 1. Oblique traced discontinuity cuts between grid-points. The arrow on the discontinuity shows the stream direction.

we see that the added term is $.5h(a_1 \partial^h_{11} - a_2 \partial^h_{22})$, which clearly does not correspond to an artificial viscosity term; the sign on ∂^h_{11} is reversed. However, when this term is rewritten in terms of the characteristic direction ξ and the perpendicular direction η, it reads

$$.5h[a_1(a_1\partial_\xi + a_2\partial\eta)^2 - a_2(a_2\partial_\xi - a_1\partial_\eta)^2]$$

$$= .5h[(a_1^3 - a_2^3)\partial_{\xi\xi} + 2a_1a_2(a_1 + a_2)\,\partial_{\xi\eta} + a_1a_2(a_2 - a_1)\,\partial_{\eta\eta}]$$

The coefficient of $\partial_{\xi\xi}$ is negative. (If $a_1 > a_2$, the discontinuity in Fig. 1 must cut on the other side of point C, and we would then base L^h on the stencil (A,C,D)). Thus, in the characteristic direction we still have the correct viscosity sign. Near the traced discontinuity this is the only direction that matters, since only at that direction instabilities may grow and non-physical discontinuities may enter.

When double discretization is used (see Sec. 2), the basic rule should be satisfied by both L^h_0 and L^h_1. This is even true when simple defect corrections, rather than multigrid corrections, are used: If L^h_0 fails to satisfy the rule, and we make defect corrections with a better approximation L^h_1 that does satisfy the rule, then we do at convergence get a good solution (corresponding to the operator L^h_1), but near the discontinuity this convergence will be painfully slow.

4.3 Corrections interpolation near discontinuities

Usually, when the multigrid algorithm switches from a fine grid to the next coarser one, the error on the fine grid is smooth, hence the correction calculated on the coarser grid is smooth, and its interpolation is therefore easy. However, near discontinuity, especially in a nonlinear problem near a discontinuity whose location is unknown (whether traced or not), the coarse-grid correction may change the location of the discontinuity. A smooth change in the characteristic directions, for example, will change the location of the shock created by their collision. This is allowed for by the FAS scheme, which is the scheme one would use anyway because of the nonlinearity (cf. Sec. 2.1). Now, when this happens, the coarse-grid correction is no longer smooth; near the discontinuity it looks like a pulse function. Interpolating it by any usual interpolation would create new errors with large oscillation near the discontinuity.

There is a natural way to avoid this trouble, exactly due to the Full Approximation Scheme (FAS). Usually in FAS, the coarse-grid (approximate) solution u^{2h} is not directly interpolated to the fine-grid. It is the correction $u^{2h} - I_h^{2h} u^h$ which is the smooth function approximated by the FAS equations, where $I_h^{2h} u^h$ is the same interpolated function as used in generating the FAS equations (cf. (2.1) - (2.2) for example). Hence it is that correction which should be interpolated. So the FAS correction interpolation is usually given by

$$u^h_{NEW} = u^h_{OLD} + I^h_{2h}(u^{2h} - I_h^{uh} u^h) , \qquad (4.1)$$

rather then by the simpler procedure

$$u^h_{NEW} = I^h_{2h} u^{2h} \qquad (4.2)$$

The usual advantage of (4.1) is that it preserves the high-frequency content of u^h_{OLD} (earned by previous relaxation sweeps on grid h), while (4.2) destroys this content (producing high-frequencies depending only on the interpolation). But this advantage disappears when the correction is not small compared with u^h_{OLD} itself, which may exactly be the situation when the discontinuity moves. By looking at simple one-dimensional examples is easy to see that in such a situation (4.2) gives better results. For example, (4.1) does not preserve monotonicity, even if I^h_{2h} is a linear interpolation.

Thus, the suggested rule is to generally use (4.1), except at (or near) points where $u^{2h} - I^{2h}_h u^h_{OLD}$ is comparable to u^h_{OLD}. In the later case (4.2) should be preferred.

When the coarse-grid correction does not change the location of the jump, and when this location is known or traced, the important rule is not to straddle that discontinuity by the interpolation operator I^h_{2h}.

4.4 Local refinements

One can indefinitely reduce all truncation errors, including for example total artificial viscosity, by reducing the effective meshsize (the meshsize on the finest grid) . This however costs rapidly increasing computational work. The reduced meshsize is in fact much more needed where the truncation errors are large; e.g., near discontinuities, especially near untraced discontinuities. A general scheme to reduce the errors should therefore be to adapt the meshsize locally so as to (nearly) minimize the total error (suitably measured), in a given amount of computational work, and without spending too much computational work on the grid adaptation and optimization processes themselves.

A FAS multigrid algorithm that does just that has been developed, using increasingly finer grids which are confined to increasingly smaller neighborhoods of the discontinuity, and which assume increasingly more accurate orientations, tending to coincide with the orientation of the discontinuity (even when the latter is not explicitly identified). The refinement criteria are based on local truncation errors, which are approximately known anyway due to the FAS processing. See [5] or [6], with additional details in [7]. This adaptive algorithm is a natural extension of the Full Multi-Grid (FMG) algorithm, in which a solution of a differential equation starts on the coarsest possible grid and proceeds to finer ones. (On each grid an approximation is first obtained by interpolation from the coarser solution, and then it is improved by a multigrid cycle. See Sec. 6.2.) The only added feature is the adaptation. Namely, before proceeding to a new finer grid the algorithm decides to which subdomain this new grid should be confined.

Observe that when the algorithm proceeds to finer grids (whether globally or just near discontinuities), the total amount of artificial viscosity steadily decreases. (The multigrid double discretization scheme does not itself eliminate the artificial viscosity at discontinuities straddled by the difference operator.) In this respect the FMG algorithm, especially with local refinements, is equivalent to a continuation method which starts with large viscosity and gradually takes it out.

Does the tracing of a discontinuity (or the exact knowledge of its location) eliminate the need for local grid refinements? Often it does; the truncation errors at traced discontinuities need not be larger than anywhere else. Exceptions are cases where the details of the boundary layers (or other thin viscous layers) need to be determined numerically because of their complexity and because they are essential for getting the desired results. For example, in driven cavity problems, the whole flow is driven by the cross-stream viscosity acting through the boundary layer. Hence any misrepresentation of the boundary layer entails large errors. In other problems the boundary-layer representation may be less important for the external flow, but it may be important for calculating the pressure distribution at the boundary.

Local refinements are of course very useful wherever tracing is impossible or inaccurate, such as regions of interaction of *two* discontinuities.

A remark concerning h-ellipticity in locally refined grids at a thin viscous layer, such as a boundary layer. The techniques described in [7] employ a sequence of grids each of which refines its predecessor only in one direction, perpendicular to the layer. If the correct rate of refinement is used then there is no need to use semi h-ellipticity. Full h-ellipticity, hence simple pointwise relaxation, can be used. (An example for h-ellipticity on a grid with different meshsizes in different directions is given by (3.6). Full h-ellipticity is the case $S=\{1,\ldots,d\}$). It is interesting to note that with this full h-ellipticity, the artificial cross-stream viscosity becomes comparable to the physical one exactly when the cross-stream meshsize becomes fine enough to resolve the boundary layer. If one refines further than that in this direction (perpendicular to the layer) for some reason, the physical viscosity (ε in (3.7)) becomes dominant in this direction k and dictates the artificial ellipticity $\overline{\varepsilon}_j^h = \overline{\beta}\,\varepsilon\, h_j^2/h_k^2$ in other directions j. A better procedure in this situation is not to use the last term in (3.7), thus avoiding excessive viscosity in directions parallel to the layer. This would then create $\{k\}$-h-elliptic difference operator. The point is that in this situation simple pointwise relaxation is still all we need, since direction k is the only direction of refinement and coarsening (cf. Theorem 3.2).

4.5 Measuring errors near discontinuities

In our numerical experiment very large errors, or changes in the solution, have sometimes been measured locally. The use of wrong error norms has then caused distorted pictures and wrong algorithmic decisions. These large changes occur near discontinuities whose location, whether traced or not, changes numerically. Indeed, suppose our solution has $O(h^p)$ errors. This involves $O(h^p)$ errors in characteristic lines, hence $O(h^p)$ errors in the location of discontinuities. That is, over a region whose area (volume) is $O(h^p)$ the solution is switched and $O(1)$ errors are measured in its values or its derivatives, depending on the nature of the discontinuity. If we measure these errors by L_q norm we will get $O(h^{p/q})$ error norms.

CONCLUSION: In the presence of discontinuities, only the L_1 norm gives the correct picture.

This is also true near a discontinuity whose location is fixed but straddled by the difference or interpolation operators.

For a traced discontinuity a better measure is one which separately measures the changes in the discontinuity location (which is the well-posed unknown near the discontinuity) and avoid measuring changes of function values at points switched from one side of the discontinuity to its other side.

5. LOCAL MODE ANALYSIS

An important tool in developing multigrid programs, deciding between various options, and precisely predicting their performance, is the "local mode analysis". For general nonlinear regular elliptic problems the analysis proceeds as follows: The difference equations are linearized around some approximate solution, and the coefficients of the linearized equations are then frozen at local values. (or, more generally, the coefficients may assume some typical mode of oscillations.) The resulting constant-coefficient (or simple-mode-coefficient) problem is then assumed to hold in a grid covering the entire space, and its convergence properties under various processes can be studied in terms of the Fourier components of the error. This local analysis is a very good approximation to the true behavior of high-frequency components away from boundaries, since such components interact at short distances (comparable to the wavelength) and are therefore not influenced by distant boundaries and slow changes of coefficients. The analysis is inaccurate for low-frequency components, but those may be ignored in the multigrid work estimates, since low-frequency convergence is obtained on coarser grids, where the computational work is negligible. The analysis is also inaccurate for high-frequency components near boundaries, but this is a secondary problem because the boundary is a lower dimensional manifold, hence we can invest more work to get better high-frequency convergence there. In fact, it is an important advantage of the local mode analysis that it separates the study of the main (interior) multigrid processes from the study (and mistakes) of processes at the boundaries. (See details in Sec. 4 of [14] and additional remarks in Secs. 2 and 3 of [10]).

This separation between the interior and the boundary is not quite possible in non-elliptic and singular perturbation problems. In such problems some high-frequency modes far from the boundary are still strongly affected by the boundary conditions. That is, high-frequency oscillations on the boundary can travel along characteristics into the domain with negligible decay. This can also happen in the discrete solutions, in case grid lines coincide with characteristic lines and the corresponding semi h-elliptic discretization is used. Full h-elliptic solutions, on the other hand, behave like elliptic ones, all high-frequencies being attenuated within few meshsizes.

Thus, in analyzing semi-h-elliptic operators, or in analyzing convergence to the differential solutions (even by fully h-elliptic solutions), or even in analyzing smooth components of full h-elliptic solutions, the infinite-space mode analysis must be complemented with a mode analysis which includes the boundary. This is most conveniently modelled by considering the semi-infinite domain, $\{x_1 \geqslant 0\}$ say, with boundary conditions prescribed on (and/or near) $\{x_1 = 0\}$. The mode analysis is therefore done in terms of components $\exp(i\,\underline{\theta}\cdot\underline{x}/\underline{h})$, where only $\theta_2,\ldots,\theta_d$ are required to be real, while $Im\ \theta_1 \geqslant 0$. More precisely, for each real vector $(\theta_2,\ldots,\theta_d)$ the number of modes for which $Im\ \theta_1 < 0$ should coincide with the number of boundary conditions supposedly given at the far $(x_1 \to \infty)$ boundary, so that these modes could be determined by those conditions.

The semi-infinite analysis is enough by itself, but we can make it simpler by considering separately the infinite-space case and subtracting it off, so that at the semi-infinite analysis we can consider homogeneous equations only (with non-homogeneous boundary conditions). In other words the semi-infinite analysis is done for characteristic modes only.

Another significant departure from mode analysis of regular elliptic cases is the existence of different components with markedly different convergence properties. As we will see (Sec. 5.1), this situation calls for another type of two-level analysis, the FMG mode analysis (Secs. 5.2, 5.3). Also, in non-elliptic cases interesting convergence features are related to low-frequency characteristic components, the analysis of which is simplified by using first-differential-approximation mode analysis (Sec. 5.1).

Various local-mode-analysis calculations have been made for studying various features of multigrid solutions to non-elliptic and singular perturbation problems, and more are planned. Generally they are more complicated then in the regular elliptic case. Sometimes more levels should be brought into the picture (see end of Sec. 5.7). Here we will bring some of the simpler analyses, which do not require computerized calculations but exhibit some of the main conclusions.

5.1 Slow convergence in characteristic smooth components

One fact that was noticed both in numerical experiments and in mode analyses (by N. Dinar and separately by K. Börgers) is that non-elliptic and singular-perturbation problems exhibits rather disappointing convergence factors, not much better than .5, per miltigrid cycle, and this cannot be improved by putting more relaxation sweeps into the cycle. See for example Table 1.

a_1	$\beta = .5$		$\beta = .75$		large β	
	r = 1	r = 2	r = 1	r = 2	r = 1	r = 2
1.0	1	1	.52	.50	.50	.50
0.5	.60	.50	.50	.50	.50	.50
0.0	.51	.50	.50	.50	.50	.50

TABLE 1.

Asymptotic two-level convergence factors per multigrid cycle for the difference operator $-\beta h\Delta^h + a_1\partial_1^h + \partial_2^h$. The cycle includes r sweeps of pointwise Gauss-Seidel relaxation with red-black ordering (RB relaxation), a fine-to-coarse residual transfer with full weighting, a coarse grid solution with the operator $-2\beta h\,\Delta^{2h} + a_1\partial_1^{2h} + \partial_2^{2h}$, and a linear coarse-to-fine interpolation of corrections. (Calculated by N. Dinar.)

This result could not be understood by one-level analysis, since it applied to low-frequency ($|\underline{\theta}| << \pi$) components.

A simple way to see clearly the behavior of lower-frequency components is to apply first-differential-approximation (FDA) analysis. In this analysis we replace the difference operators by their first-differential approximations (see [32],[33],[34]), ignore inter-grid transfer operators (i.e., since the function is smooth we ignore the small-amplitude high-frequency components created by its interpolation), and ignore relaxation (since it is ineffective for smooth components). What is left are just some simple intergrid relations.

Consider, in this fashion, a two-level cycle for solving (3.2), where the physical ε is infinitesimally small and hence replaced by the artificial $\varepsilon(h)$ given by (3.4). The FDA representation of the difference operator is then

$$L^h = -\beta ah\,\Delta + \sum a_j \frac{\partial}{\partial x_j} \;, \tag{5.1}$$

where $a = \max|a_j|$. Consider now in the infinite space a characteristic smooth error component V^h, i.e., $\sum a_j\, \partial V^h/\partial x_j = 0$. Its residual is therefore given by $-\beta ah\,\Delta V^h$. Hence the coarse-grid equation is

$$L^{2h}V^{2h} \equiv -\beta a(2h)\,\Delta V^{2h} + \sum a_j \frac{\partial V^{2h}}{\partial x_j} = -\beta ah\,\Delta V^h \;, \tag{5.2}$$

and its solution is $V^{2h} = .5V^h$. Thus the new error on the fine grid after a full multigrid cycle will by $V^h_{NEW} = V^h - V^{2h} = .5V^h$.

CONCLUSION: The asymptotic convergence factor .5 per cycle is attained for all characteristic smooth components and it results from the meshsize dependence of the artificial viscosity.

The asymptotic factors observed in practice were sometimes even much worse than .5 . That happened when a V-cycle algorithm was used (Sec. 6.1). In such a cycle we solve the coarse-grid equations again by just one cycle, whose convergence rate is again no better than .5 . We thus come far from solving the grid 2h equations, hence the above analysis is not a good approximation. Indeed, if in the coarse of a cycle a much coarser grid, grid kh say, is visited just once, an FDA infinite-space analysis similar to the above would show a convergence factor no better than $1-k^{-1}$ per such cycle. This, the experiments however show, is too pessimistic. Moreover, when a W cycle is used (i.e., switching twice from each grid to the next coarser one before returning to the next finer one), convergence factors better than .5 per cycle were obtained. The reason is the boundedness of the domain and the influence of the boundary far into the domain, especially for the smooth characteristic components considered here.

But before rushing to an improved analysis, taking the boundary into account, we pause to consider the practical meaning of the above analysis. One conclusion is of course to use W cycles (Sec. 6.1). Another conclusion is that we could improve the convergence rate by using less artificial viscosity on the coarser grid. This is effectively obtained by applying the double discretization scheme (Sec. 2) to the coarse grid. A more important point, however, somewhat surprising but typical to similar multigrid situations, is that we need not really be troubled by the slower asymptotic rates.

Indeed, the slower convergence is obtained for very particular components, namely, smooth characteristic components far from boundaries. For such components L^{2h} is not a very good approximation to L^h, hence the slowness. But, for exactly the same components and the same reason, L^h is not a good approximation to L. Hence, exactly for these components, we do not need much algebraic convergence (convergence to the discrete solution), since that solution itself is far from the differential solution.

We will in fact show in the next section that one-cycle FMG algorithm brings the algebraic errors well below truncation errors for all components, even if the coarse-grid double-discretization scheme mentioned above is not used. The latter is actually needed only if we want a one-cycle FMG algorithm to produce algebraic errors well below truncation errors for a scheme where double-discretization is used on the fine grid (thus producing much smaller truncation errors). This is an example of a general principle of correspondence between discretization techniques and coarsening techniques (See [12]).

5.2 FMG Mode analysis: infinite space

As explained in the previous section, because of marked differences between the behavior of different modes, the usual two-level local mode analysis used for predicting the asymptotic multigrid convergence factors (Sec. 4.6 in [14]) may be inadequate for predicting the power of Full Multi-Grid (FMG) algorithms (Sec. 6.2); i.e., predicting how many multigrid cycles will be needed to reduce the grid-h algebraic errors below truncation errors, when the first approximation is obtained by interpolation from a solution on grid 2h. Thus we will replace that analysis by a new type of two-level mode analysis, the FMG mode analysis, which analyzes the two-level stages of the FMG algorithm. As usual, it can be generalized to a three-level analysis, etc., but the two-level case provides accurate enough predictions.

Since the main trouble (slow algebraic convergence) occurs for smooth components, we will greatly simplify our example here of the FMG mode analysis by applying it in the framework of the first-differential approximations (Sec. 5.1). On the other hand, since the troublesome components carry information from the boundary far into the domain, we will complement our infinite-space analysis with a half-space analysis (Sec. 5.3).

Thus, we consider our standard example (3.1) with vanishingly small but positive $\varepsilon = 0+$, and with the difference approximation L^h represented by (5.1). In the infinite space we consider a solution Fourier component

$$U(\underline{x}) = \exp(i\underline{\omega}\cdot\underline{x}) = \exp i\,(\sum_{1}^{d} \omega_j x_j) .$$

(In analyzing only smooth components the Fourier variable $\underline{\omega} = (\omega_1,\ldots,\omega_d)$ is more convenient than $\underline{\theta} = h\underline{\omega}$).

The corresponding right-hand side is therefore

$$F(\underline{x}) = \gamma\, U(\underline{x}) \;, \qquad \gamma = i\,\underline{a}\cdot\underline{\omega} \;,$$

and the exact discrete solution is

$$U^h(\underline{x}) = (\gamma+\delta)^{-1} F(\underline{x}) \;, \qquad \delta = \beta h a \omega^2 \;.$$

The first approximation, obtained by interpolation from grid 2h, is

$$u^h = U^{2h} = (\gamma+2\delta)^{-1} F \;,$$

giving the residual

$$R^h = F - L^h u^h = \delta(\gamma+2\delta)^{-1} F \;.$$

The coarse-grid correction is

$$V^{2h} = (\gamma+2\delta)^{-1} R^h$$

giving the new approximation

$$\bar{u}^h = u^h + V^{2h} = (\gamma+3\delta)(\gamma+2\delta)^{-2} F \;. \qquad (5.3)$$

This is then the solution of the two-level one-cycle FMG algorithm. We need to compare the algebraic error $\bar{u}^h - U^h$ with the truncation error $U^h - U$. From the above we easily get (for any imaginary γ, real δ)

$$\left|\frac{\bar{u}^h - U^h}{U^h - U}\right| = \left|\frac{\gamma\delta}{(\gamma+2\delta)^2}\right| = \frac{|\gamma|\delta}{|\gamma|^2 + 4\delta^2}$$

$$\leq \tfrac{1}{4} \;.$$

CONCLUSION: For all components, the algebraic error after one cycle is well below (less than a quarter of) the truncation error. There is indeed no reason to be concerned about the rather slow asymptotic algebraic convergence exhibited above for characteristic components.

5.3 Half-space FMG mode analysis

Here we can restrict ourselves to the homogeneous equation. Let the boundary condition at $x_1 = 0$ be the Fourier component

$$U(0,x_2,\ldots,x_d) = U_0 = \exp(i \sum_{j=2}^{d} \omega_j x_j) \ .$$

We will consider the case of a non-characteristic boundary; i.e., $|a_1|$ is comparable to $a = \max |a_j|$. Hence the differential solution is

$$U(\underline{x}) = U_0 \, e^{i\alpha x_1}, \quad \alpha = -\frac{1}{a_1} \sum_{2}^{d} \omega_j a_j \ .$$

The exact solution of the difference equation (5.1) is given by

$$U^h(x) = e^{i\alpha_h x_1} \, U_0 \ ,$$

where α_h satisfies

$$\beta h a(\alpha_h^2 + \overline{\omega}^2) + i \, a_1 \, (\alpha_h - \alpha) = 0, \ \overline{\omega}^2 = \sum_{2}^{d} \omega_j^2 \ . \qquad (5.4)$$

Assuming $h\overline{\omega} << 1$, the two solutions of (5.4) are

$$\alpha_h^{(1)} = \alpha + i\eta \ , \qquad \eta = \beta h \frac{a}{a_1} \, (\alpha^2 + \overline{\omega}^2), \qquad (5.5)$$

and

$$\alpha_h^{(2)} \approx - \frac{i a_1}{\beta h a} \ ,$$

where $O(h^2 \bar{\omega}^2)$ terms in α_h are neglected, consistently with our first-differential approximation. The second solution is significant only on the far boundary (at some large x_1, tending to ∞). Since it is $O(h^{-1})$, it actually affects the solution near that boundary only in a layer of width $O(h)$. Outside this numerical boundary layer the solution is given by $\alpha_h = \alpha_h^{(1)}$ and is only affected by the boundary condition at $x_1 = 0$. This is the solution we analyze here.

The solution on grid $2h$ is similar, α_{2h} replacing α_h. Hence the first approximation on grid h is

$$u^h = U^{2h} = U_0 \, e^{i(\alpha+2i\eta)x_1} ,$$

giving the residuals

$$R^h = -L^h u^h = a_1 \eta \, u^h \; . \tag{5.6}$$

The coarse-grid correction V^{2h} satisfies $L^{2h}V^{2h} = R^h$ with homogeneous boundary conditions at $x_1 = 0$, hence (still assuming $h\bar{\omega} << 1$)

$$V^{2h} = a_1\eta(a_1 - 4i\beta h a\alpha)^{-1} \, x_1 \, u^h \approx \eta \, x_1 \, u^h \, . \tag{5.7}$$

The one-cycle FMG solution is given by $\bar{u}^h = u^h + V^{2h}$, from which we calculate the algebraic-error-to-truncation-error ratio

$$\left|\frac{\bar{u}^h - U^h}{U^h - U}\right| = \left|\frac{(1+\xi)\,e^{-2\xi} - e^{-\xi}}{e^{-\xi} - 1}\right| \lesssim .15623 \; , \tag{5.8}$$

where $\xi = \eta x_1$ and the last inequality is calculated numerically.

We thus see again that algebraic errors at the end of one-cycle FMG algorithm are far smaller then truncation errors. Note that this result is obtained uniformly in x_1: For large x_1 both algebraic and truncation errors are large. Both errors grow linearly in x_1 and become $O(1)$ for $\omega^2 h x_1 \geqslant O(1)$; but their ratio always satisfies (5.8).

A similar FMG mode analysis can be done without first-differential approximations, thus analyzing also high-frequency components and their interaction with lower ones, taking relaxation and interpolation processes into account, and optimizing β. Our present analysis shows that if enough smoothing is provided by relaxation, one-cycle FMG algorithms are enough to solve the algebraic problem to well below truncation errors. This is also amply confirmed by numerical experiments, such as in Sec. 7.

5.4 Analysis of multigrid double discretization

In terms of this simplified, FDA two-level FMG analysis, let us now examine the effect of the double discretization scheme of Sec. 2; i.e., the effect of dropping the artificial viscosity in calculating the residuals transferred to the coarse grid. Applying this modification to the half-space analysis of Sec. 5.3, instead of (5.6) we get

$$R^h = L_1^h u^h = [2a_1\eta + O(a\bar{\omega}^3 h^2)]\, u^h$$

where L_1^h is a second-order approximation to L^h, hence the $O(a\bar{\omega}^3 h^2)$ term. Continuing as before we easily find that the corrected fine-grid solution is now

$$\bar{u}^h = [1 + 2\eta x_1 + O(\bar{\omega}^3 h^2 x_1)]\, u^h \ .$$

Hence

$$\begin{aligned} |\bar{u}^h - U| &= |(1 + 2\eta x_1)\, e^{-2\eta x_1} - 1| + O(\bar{\omega}^3 h^2 x_1) \ . \\ &= O(\bar{\omega}^3 h^2 x_1 + \bar{\omega}^4 h^2 x_1^2) \ . \qquad (5.9) \end{aligned}$$

whereas the previous solution (Sec. 5.3) gave $|\bar{u}^h - U| = O(\bar{\omega}^2 h x_1)$.

The constants in those estimates cannot of course be obtained by this simple FDA analysis, since they depend on terms ignored by FDA.

5.5 Higher-order method for time-dependent problems

The example discussed above (equation (3.1) with $\varepsilon = 0+$, and Dirichlet conditions on $\{x_1 = 0\}$), can also serve as an example for a time-dependent problem, x_1 being the time coordinate. The double discretization (or coarse-grid defect-correction) scheme roughly analyzed above, indicates an attractive method for obtaining higher-order approximations for time-dependent problems.

Namely, let

$$\begin{aligned} &L_0^h U_0^h = 0 , \qquad\qquad (x_1 > 0) \\ &U_0^h \text{ is given for } x_1 = 0 , \end{aligned} \tag{5.10}$$

be a stable discretization of the problem, regarded as time-dependent problem. That is, (5.10) can stably be integrated by marching in time. Typically L_0^h will be a simple low-order implicit operator. And let $L_1^h U^h = 0$ be a higher-order discretization of the same problem, where L_1^h is simple (e.g., it may be <u>central in time</u>) but unstable. In our example, L_0^h may be given by (3.2) with $\varepsilon = ha_1/2$, and L_1^h by (3.2) with $\varepsilon = 0$.

One can then improve the lower-order solution U_0^h by integrating

$$L_0^H V^H = -I_h^H L_1^h U_0^h \tag{5.11a}$$

and correcting

$$U_1^h = U_0^h + I_H^h V^H \,, \qquad (5.11b)$$

where $H = 2h$ (coarser-grid defect correction) or $H = h$ (defect correction on the same grid). The new solution U_1^h is both stable and of higher approximation order. Indeed, if error without the correction is $|U_0^h - U^h| = O(\omega^{p_0+1} h^{p_0} x_1)$, then (5.11) will give

$$|U_1^h - U| = O(\omega^{p_1+1} h^{p_1} x_1 + \omega^{2p_0+2} h^{2p_0} x_1^2) \,, \qquad (5.12)$$

where p_j is the order of approximation of L_j^h .

The scheme (5.10) - (5.11) is always stable, since we integrate only with the stable discretization. Notice that this is true only if U_0^h and V^H are integrated independently. The seemingly similar scheme, in which after each time step we reinitialize U_0^h by replacing it with U_1^h , may well be unstable. On the other hand it does pay to reinitialize every $O(1)$ time interval. This would prevent the $O(x_1^2)$ growth of error (for large x_1) indicated by (5.12). (This remark is due to a discussion with J.M. Hyman.)

The independent integration of U_0^h and V^H requires extra storage. By taking $H = 2h$ (coarse-grid defect correction) this extra storage is only a fraction of the basic storage (one time level of U_0^h). The extra computational work is not very significant either. This two-level scheme can be extended to more levels.

5.6 Principal finite-difference terms

Mode analysis of complex systems will be very much simplified, and more easily yield the desired insights, if non-essential terms are discarded. The only terms significant in local mode analysis are the "scaled principal terms" as defined in Sec. 3.8 of [14]. In our simple example (3.2), if the artificial ε is given by (3.4), then all terms are scaled-principal. Generally, in singular perturbation problems the scaled-principal terms are all those corresponding to the principal and sub-principal terms in the differential system. The principal terms in the differential operator are all those which contribute to the highest-order terms in the determinant of the operator. The sub-principal terms are those which become principal when the singular-perturbation terms are omitted. See an example in [13], showing how important this simplification is in designing relaxation schemes.

5.7 Some smoothing factors

The above discussion (except for Table 1) has dealt with the two-level behavior of smooth components only. To give basic ideas about the behavior of high-frequency components, we now describe some elementary smoothing-rate analyses, especially for Gauss-Seidel schemes in various orderings. See Sec. 6.3 below for our motivation in considering such schemes, in particular red-black (RB) schemes.

The smoothing factor $\bar{\mu}$ and the smoothing rate $1/|\log \bar{\mu}|$ measure the efficiency of relaxation in converging high-frequency components (see Sec. 3.1 in [5], Sec. 4.1 in [14] and Sec. 2 in [10]). Their definition is extended to RB schemes (and to other schemes which couple several Fourier components) in [11]. For such schemes $\bar{\mu}$ depends on the number r of finest-grid relaxation sweeps per multigrid cycle, so we denote it $\bar{\mu}_r$. For the general $(2d+1)$-point operator

$$L^h U^h(x) = \sum_{|j_1|+\ldots+|j_d|\leq 1} b_{\underline{j}}\, U^h(\underline{x}+\underline{j}h), \qquad (5.13)$$

the smoothing factor of RB Gauss-Seidel relaxation turns out to be

$$\bar{\mu}_r = \max[\bar{\mu}^{(1)},\ \bar{\mu}_r^{(2)}]\ , \qquad (5.14)$$

where

$$\bar{\mu}^{(1)} = \max_{|\theta_1|\leq .5\pi\leq|\underline{\theta}|} |g(\underline{\theta})|^2$$

$$\bar{\mu}_r^{(2)} = \max_{|\underline{\theta}|\leq .5\pi} |.5\, g(\underline{\theta})^{2r-1}[1-g(\theta)]|^{1/r}$$

$$g(\underline{\theta}) = b_{\underline{0}}^{-1} \sum_{\underline{j}\neq 0} b_{\underline{j}} \exp(i\,\underline{j}\cdot\underline{\theta})$$

and $|\underline{\theta}| = \max(|\theta_1|,\ldots,|\theta_d|)$. Cf Sec. 2.3.1 in [27].

For any operator of the form (5.13), all Gauss-Seidel schemes badly diverge, unless $|a_{\underline{0}}|$ is sufficiently large compared with $\sum_{\underline{j}\neq\underline{0}}|a_{\underline{j}}|$. If it is not, distributed relaxation (e.g., Kaczmarz relaxation) must be used. The distributed schemes are considerably more expensive, hence enough artificial viscosity should be used to obtain enough diagonal dominance. (The diagonal dominance is important only in this limited sense. The equations are still stable and still "physical" when less artificial viscosity is used.)

Smoothing factors for various discretizations of our model problem ((3.1) with $\varepsilon = 0+$) have been calculated. For upstream differencing in two dimensions $(d=2)$ it has been found that pointwise lexicographic GS (Gauss Seidel relaxation), symmetric pointwise GS (relaxing point-by-point in lexicographic order and then in the reversed order), lexicographic line GS, red-black GS and Zebra (odd-even line GS) are all inadequate in general; i.e., for each of these schemes there is a coefficient vector $\underline{a}$ such that $\bar{\mu} = 1$. For symmetric lexicographic line GS, $\mu \leq 5^{-1/2}$. (Cf. Table 7.1 in [18]). To get better factors, schemes with more than two marching directions should be used. Generally, in d dimensions, 2^d marching directions of pointwise GS or 2^{d-1} marching directions of line GS are required to get $\bar{\mu} < 1$ for all $\underline{a}$.

Better algebraic smoothing factors are more easily obtained if the set S of S-h-ellipticity directions is enlarged (cf. Sec. 3.2 and 3.3). For the fully h-elliptic operator

$$L^h = -\beta ha\Delta^h + \sum_{j=1}^{d} a_j \hat{\partial}_j^h \ , \qquad (5.15)$$

symmetric pointwise Gauss Seidel (SGS) in three dimension $(d = 3)$ gives $\bar{\mu} \leq (2/5)^{1/4} = .795$ for $\beta = .5$, and $(13/109)^{1/4} = .59$ for $\beta = 1$. In two dimensions SGS yields $\bar{\mu} \leq 5^{-1/4} = .67$ for $\beta = .5$, and $\mu \leq .527$ for $\beta = 1$, for any coefficient vector $\underline{a}$. SGS and one-pass Gauss-Seidel schemes in two dimensions are summarized in Table 2.

Numerical experiments and full two-level FMG analysis show the best β for FMG algorithms to be close to the minimal β for which reasonable smoothing rates are still obtained. Much larger β cause some components to be damped too much in relaxation, while significantly smaller β already cause divergence. This minimal β, as shown by the Table 2, has some dependence on the vector $\underline{a}$. Some numerical experiments (see Sec. 7.1) show that the best (or a very good) β can even be such that gives inefficient smoothing. For example $\beta = .5$ in case $d = 2$, $a_1 = a_2 = 1$ and RB relaxation. The exact reason can be understood only by a higher-level analysis, as follows:

Relaxation	r	a_1	β=2	β=1	β=.75	β=.5	β=.4	β=.3	β=.25
Lex+		1	.375	.249	.167	0	∞	∞	∞
		0	.456	.423	.415	.470	.581	.91	1.36
Lex+, Lex-		-1	.544	.631	.721	1.00	1.67	5.00	∞
Lex-		1	.625	.749	.833	1.00	1.29	2.00	3.00
		0	.580	.664	.721	.835	.919	1.05	1.223
SGS		1	.480	.420	.353	0	∞	∞	∞
		0	.496	.487	.480	.492	.546	.703	.889
		-1	.505	.527	.558	.669	.801	1.51	∞
RB	1	±1	.266	.333	.45	1.00	1.56	2.78	4.00
		±.5	.266	.316	.374	.583	.879	1.56	2.25
		0	.266	.313	.361	.500	.641	.944	1.25
	2	±1	.386	.499	.595	1.0	1.56	2.78	4.00
		±.5	.372	.429	.499	.644	.879	1.56	2.25
		0	.364	.394	.422	.500	.641	.944	1.25
	3	±1	.36	.438	.503	1.00	1.56	2.78	4.00
		±.5	.345	.402	.444	.583	.879	1.56	2.25
		0	.341	.381	.423	.505	.641	.944	1.25

TABLE 2.

Smoothing factors $\bar{\mu}_r$ for the operator $-\beta h\Delta^h + a_1\partial_1^h + \partial_2^h$. The relaxation is pointwise Gauss-Seidel in lexicographic ordering (Lex+) or reverse lexicographic ordering (Lex-) or in red-black ordering (RB) or SGS (Lex+ followed by Lex-). r is the number of sweeps per multigrid cycle. $\bar{\mu}_r$ for Lex schemes is independent of r. For Lex schemes some components may be amplified infinitely. Although these are sometimes low-frequency components (not normally taken into account in smoothing factors) we designate this catastrophic behavior by ∞ in the table.

The bad smoothing is attained only for very particular components -- those which are smooth in the cross-stream direction and highly oscillating in the streamwise direction. Exactly these components will usually have negligible amplitude. Moreover, each component which is stream-wise high-frequency on grid 2h but cross-stream smooth converges efficiently by the relaxation on the finer grid h, since interpolation (if it is symmetric bipolynomial) converts half of it into a cross-stream high-frequency component on grid h. Only components which are smooth in the cross-stream direction and are high-frequency components on the finest grid will not converge, but they are ill approximated anyway.

6. THE MULTIGRID ALGORITHMS

In this section we will describe the multigrid algorithms used in our numerical experiments. They are motivated by the discussions in the previous sections as well as by some supplementary remarks below. The algorithms are all "fixed" algorithms; i.e., their complete flow is preassigned, with no dependence on internal checks. Such algorithms are more convenient at this research stage, enabling us to have more precise comparisons between various options, and also to avoid the problem of devising internal checks (if at all desired).

We assign ordinal numbers 1,2,...,M to our sequence of grids, grid 1 being the coarsest, grid M the finest. We usually use uniform square grids, the meshsize of grid k being $h_k = 2h_{k-1}$. The approximate solution on grid k at any stage is denoted u^k, where the meshsize superscripts used before are here replaced by the corresponding ordinal number. Note that in case Correction Scheme (CS) is used (rather than FAS), at a stage when u^{k+1} is defined, u^k will stand for an approximate *correction* to u^{k+1}, not an approximate solution of the original problem itself. On each grid k we will have (or we will construct) two discretization schemes: $L_i^k U^k = F_i^k$, $(i = 0, 1)$. The first $(i = 0)$ is a stable discretization to be used in relaxation, the second $(i = 1)$ is an accurate one to be used in computing residuals to be transferred to grid $k-1$ (see Sec. 2). The two schemes may of course coincide, as they always do in "classical" codes. As usual, F_i^k is not fixed in the process, but depends on u^{k+1} when the latter has been defined.

6.1 Multigrid cycles C, V, W and F

We denote by $C(r_1,r_2)^{\nu}(\ell)$ the multigrid cycle for grid ℓ which relaxes on each grid r_1 sweeps before switching to the next coarser grid, and r_2 sweeps after returning from it, and which switches ν times from each grid to the next coarser grid before returning to the next finer grid. Formally, $C(r_1,r_2)^{\nu}(\ell)$ is flowcharted in Fig. 2, with the following explanations.

Start. When the cycle starts, an approximate solution u^{ℓ} on grid ℓ is assumed to be given. The purpose of the cycle is to improve the approximation. For this purpose the two discrete equations for this grid $L_i^{\ell} U^{\ell} = F_i^{\ell}$, $(i=0,1)$, are assumed to be given.

Sweeps on grid k are relaxation sweeps with the equation $L_0^k U^k = F_0^k$, where for $k<\ell$ the right-hand side F_0^k is defined by the coarsening process. Each sweep introduces new (improved) values for the approximate solution u^k, based on its old values. Various relaxation schemes have been used (see Sec. 6.3).

Coarsening. This process defines the right-hand sides F_i^{k-1} on the coarser grid $k-1$, and also sets on it the first approximate solution u^{k-1}. In Correction Schemes we set $u^{k-1}=0$ and define

$$F_i^{k-1} = {}_iI_k^{k-1}(F_1^k - L_1^k u^k)\ , \ (i=1,2) \qquad (6.1)$$

where ${}_iI_k^{k-1}$ are transfer operators from grid k to grid $k-1$ (discussed in Sec. 6.4). In Full Approximation Schemes (FAS) we set $u^{k-1} = I_k^{k-1} u^k$ and

$$F_i^{k-1} = L_i^{k-1}(I_k^{k-1} u^k) + {}_iI_k^{k-1}(F_1^k - L_1^k u^k),\ (i=1,2) \qquad (6.2)$$

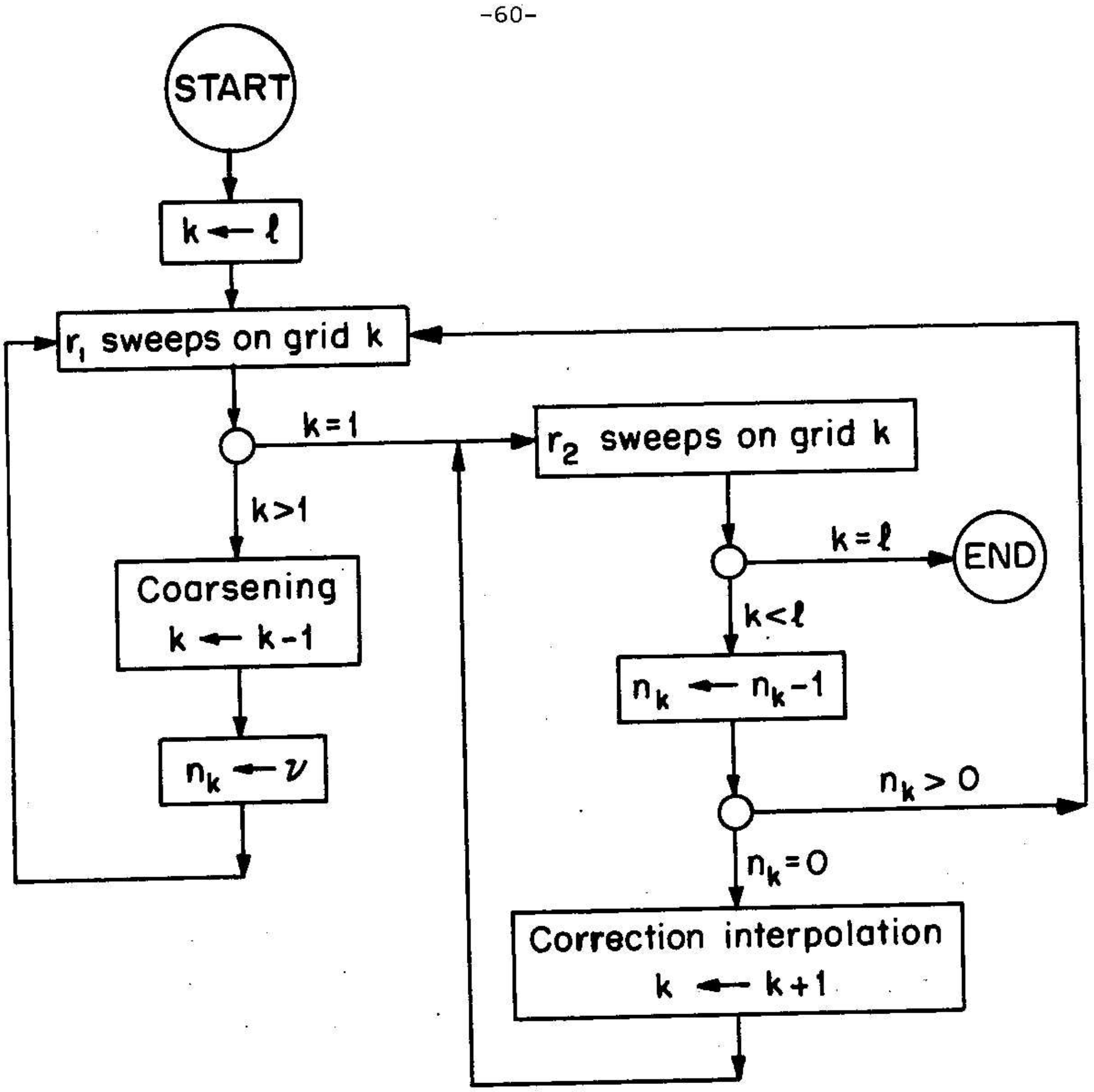

FIGURE 2. Multigrid cycle $C(r_1,r_2)^{\nu}(\ell)$. The index k is used to denote the grid number we currently work with, and n_k is used to denote the number of switches remaining to be made from grid k to grid $k-1$ before returning to grid $k+1$.

(cf.(2.2)). See remarks in Sec. 2.1 concerning the calculation of F_0^{k-1} and F_1^{k-1}. Sometimes L_i^{k-1} are also calculated at the coarsening stage based on L_i^k, ${}_iI_k^{k-1}$ and I_{k-1}^k (see Sec. 6.4). Having so defined, one way or other, the coarser-grid problem, we then set $k \leftarrow k-1$, to start working with that coarser grid.

Correction interpolation. The approximation u^k is used to correct the finer grid solution u^{k+1}. In CS

$$u^{k+1} \leftarrow u^{k+1} + I_k^{k+1} u^k \qquad (6.3)$$

where $I_k^{k+1} u^k$ is an interpolation of u^k to the finer grid $k+1$. (See remarks in Secs. 4.3 and 6.4). In FAS

$$u^{k+1} \leftarrow u^{k+1} + I_k^{k+1}(u^k - I_{k+1}^k u^{k+1}), \qquad (6.4)$$

where I_{k+1}^k should be identical with the one used in (6.2, $k \leftarrow k+1$). We then set $k \leftarrow k+1$, to return working on the finer grid.

V and W cycles. Cycles of the type $C(r_1,r_2)^1(\ell)$ are called V cycles and are also denoted $V(r_1,r_2)(\ell)$. Cycles $C(r_1,r_2)^2(\ell)$ are called W cycles and denoted $W(r_1,r_2)(\ell)$.

Total work. In complicated problems (which are our main objective) most of the work in these cycles is spent on calculating $L_j^k u^k$ on the various levels. If we consider the calculation of $L_j^\ell u^\ell$ over the finest grid as our work-unit, then a relaxation sweep on grid k costs $2^{d(k-\ell)}$ work units, where d is the dimension, and a similar work is involved in computing the residuals to be transferred from grid k to grid $k-1$. Since the cycle involves $\nu^{\ell-k}r$ sweeps on grid k, where $r = r_1 + r_2$, its total work roughly is

$$\sum_{k=1}^{\ell} 2^{d(k-\ell)} \nu^{\ell-k} (r+1) \le \sum_{j=0}^{\infty} 2^{-dj} \nu^j (r+1) = \frac{r+1}{1-\nu 2^{-d}} \qquad (6.5)$$

work units, assuming $\nu < 2^d$.

F cycles. Another cycle we have sometimes used is the similar $F(r_1,r_2)^{\nu}(\ell)$ cycle described in Fig. 3. It has similarity to the algorithm $FMG(\nu-1, 0, V(r_1,r_2),\ell)$ explained in Sec. 6.2. The number of relaxation sweeps on grid k is $[(\ell-k)(\nu-1)+1]r$, hence the total work is bounded by

$$\sum_{k=1}^{\ell} 2^{d(k-\ell)}[(\ell-k)(\nu-1)+1](r+1) \leq \sum_{j=0}^{\infty} 2^{-dj}[j(\nu-1)+1](r+1) \tag{6.6}$$

$$= \frac{(r+1)[1+(\nu-2)2^{-d}]}{(1-2^{-d})^2}$$

work units. The performance of $F(r_1,r_2)^2(\ell)$ is very much like that of $W(r_1,r_2)(\ell)$, but its total work is theoretically nicer in the one-dimensional $(d=1)$ case, since it is uniformly bounded for all ℓ, while the work associated with the W cycle grows linearly with ℓ.

6.2 Full Multigrid Algorithms FMG

Let C be any multigrid cycle; e.g. $C = C(r_1,r_2)^{\nu}$ or $C = F(r_1,r_2)^{\nu}$. Then $FMG(N_1,\ldots,N_M;C)$ is the Full Multi-Grid algorithm with N_ℓ cycles $C(\ell)$ performed for each grid ℓ, starting with an initial approximation u^ℓ interpolated from the final result of the cycles for level $\ell-1$. It is flowcharted in Fig. 4, with the following explanation.

Start. When we start an FMG algorithm we assume the basic discretization schemes $L_i^\ell U^\ell = F_i^\ell$ $(i=1,2;\ \ell=1,\ldots,M)$ are given, together with some first approximation u^1 on the coarsest grid. In linear problems and also in many nonlinear problems, the trivial approximation $u^1 \equiv 0$ can be used.

Solution interpolation is the interpolation

$$u^{\ell+1} \leftarrow \underline{\Pi}_{\ell}^{\ell+1} u^\ell , \tag{6.7}$$

interpolating the current approximation u^ℓ to a new (the next finer) grid $\ell+1$, to serve there as the first approximation. We then increase

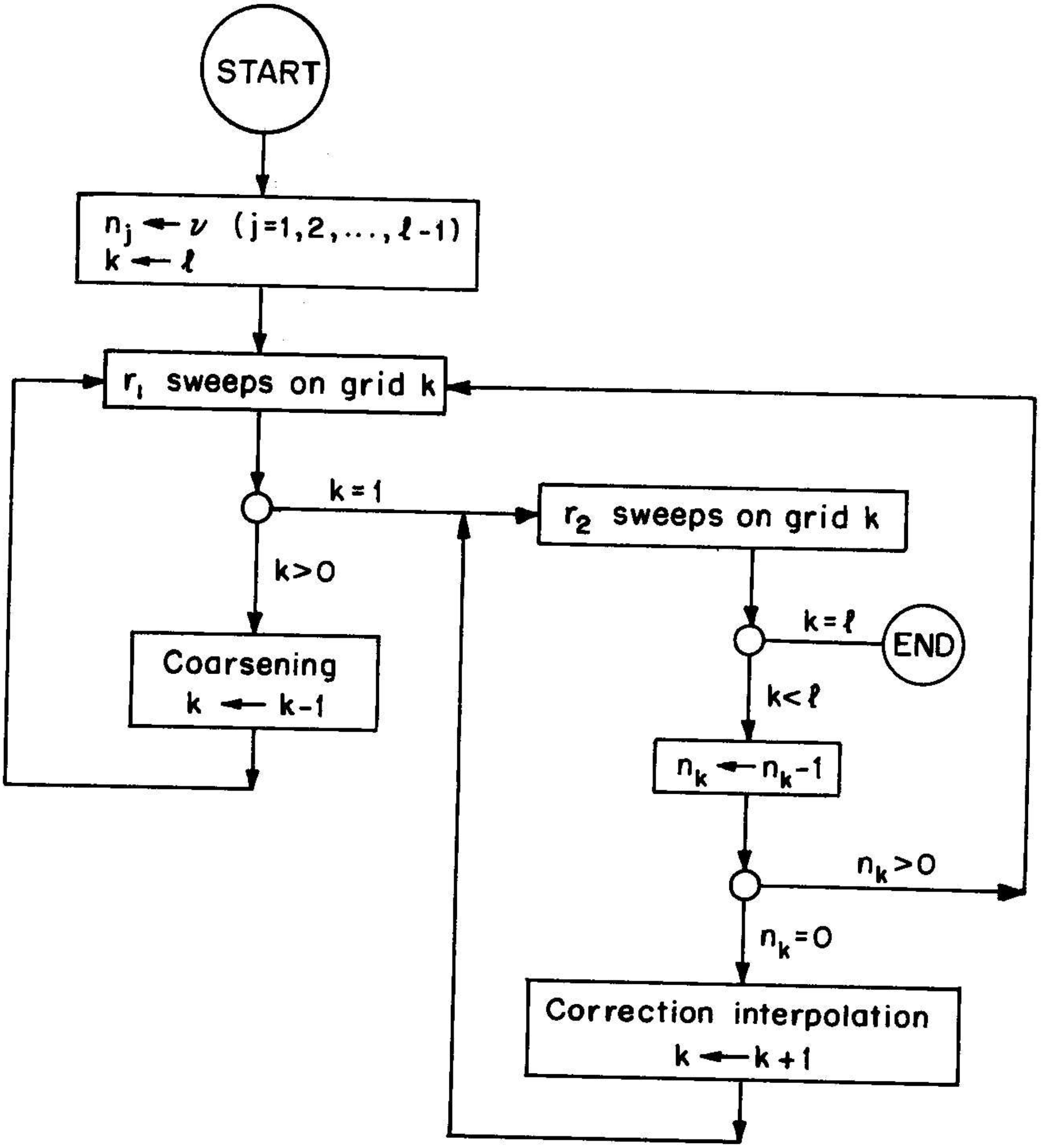

FIGURE 3. Cycle $F(r_1,r_2)^{\nu}(\ell)$.

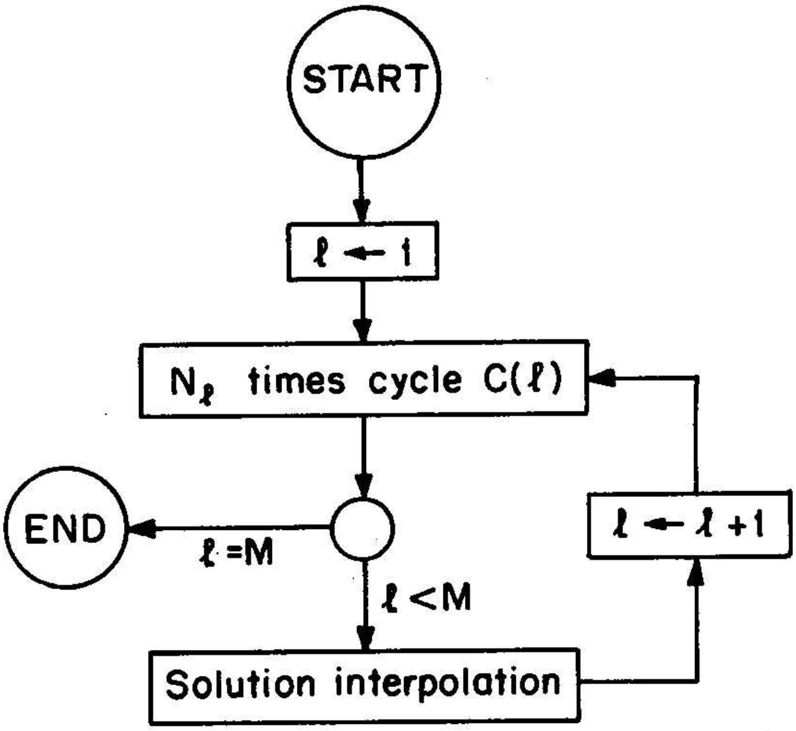

Figure 4. Algorithm FMG$(N_1, \ldots, N_M; C)$.

ℓ by one to start improving that approximation by C cycles.

The order of interpolation in (6.7) is usually higher than that of the correction interpolation I_k^{k+1} in (6.3) or (6.4). (See Sec. 5.1 in [10].)

Various choices of $(N_1,\ldots,N_M)$ are used, depending to some extent on the order of interpolation $\Pi_{\ell}^{\ell+1}$. Often this order is such that on some of the coarsest grids, grids $1,2,\ldots,\ell_1-1$ say, not enough points are available from which to interpolate at that order. The usual procedure then is to choose N_{ℓ_1} high enough to ensure we get below the truncation errors of grid ℓ_1 before proceeding to grid ℓ_1+1. Then we continue with a fixed number of cycles on intermediate grids: $N_{\ell_1}+1 = \ldots = N_{M-1} = N$, say, where most often $N=1$, sometimes $N=2$. Such an algorithm we denote FMG(N, N_M, C, M). In production codes one would usually use $N_M = N$. We call FMG(N, N, C, M) an N-cycle FMG algorithm, and denote it also by FMG(N, C, M). In our experiments we used larger values of N_M in order to verify that the smaller N_M could do just as well. See Secs. 5.2 and 5.3 in [10] concerning the choice of N and N_M.

Total Work. From (6.5) we can easily calculate that FMG(N, $C(r_1,r_2)^{\nu}$, M) roughly costs

$$N(r+1)(1-\nu 2^{-d})^{-1}(1-2^{-d})^{-1} \qquad (6.8)$$

grid-M work units.

6.3 Remarks on relaxation. RB schemes

General aspects of relaxation for non-elliptic and singular perturbation problems were extensively discussed in previous sections. More specific remarks follow.

The first approach to singular-perturbation and non-elliptic problems ([4],[30],[5],[7]) was to employ downstream Gauss-Seidel Relaxation, where a point A is relaxed before a point B if the stream goes from A to B; e.g., lexicographic order in case $a_j > 0$ $(j = 1,\dots d)$ in (3.1). This provides very good smoothing factors. In case the stream direction changes over the domain, however, this may require successive relaxation sweeps in different directions, each being effective in only part of the domain. Since one or two (efficient) sweeps are usually all that is needed at each stage at each part, the multi-direction procedure is not fully efficient. Also, it requires more complicated programs. Hence direction-free schemes were developed. They include distributed relaxation [7],[8] and relaxation with increased artificial viscosity ([8], Sec. 4.3).

Since they are direction-free, those schemes may also be implemented in red-black (RB) ordering. In each RB sweep one first relaxes all the red points (grid points $\underline{x}$ where $\sum x_j/h_j$ is even), then all the black ones (the others). This ordering has been found more efficient than others for regular elliptic problems [19]. It also has the advantage of being suitable for parallel and vector processing. For systems (non-scalar) problems similar advantages are obtained by distributed-red-black (DRB) relaxation [11] . In case of non-elliptic and singular perturbation problems RB schemes have also the important advantage of slower convergence rates (cf. Sec. 2.2): In RB schemes information propagates at most two meshsizes per sweep, hence the order of convergence (r_0) cannot vanish, whereas it does sometimes vanish for downstream Gauss-Seidel schemes (e.g., in case of upstream differencing; or in case of (5.15) with $\beta = .5$ and $a_1 = \dots = a_d$) . For the same reason, in RB schemes no catastrophic error growth can occur (cf. Table 2). On the other hand RB schemes also have a certain disadvantage when used in double-discretization schemes: No matter how smooth the solution is, red-black ordering introduces high-frequency error components whose amplitude is $O(h^{p_0+m})$, where p_0 is the approximation order of L_0^h and m is the order of the differential equation being relaxed (or the reduced equation, in case of singular perturbations. In fact, the order here should be the "scaled" order, denoted by $2\overset{o}{m}$ in Sec. 3.8 of [14].) These errors cannot be corrected by the coarse-grid correction, so they may degrade the accuracy in case $p_0 + m \leq p - q$, p being the approximation order of L_1^h and q being the order of derivatives taken into our error norms.

6.4 Remarks on coarsening.

In coarsening the grid-k residual problem, we have to choose the fine-to-coarse residuals-transfer operators ${}_iI_k^{k-1}$, the coarse grid operators L_i^{k-1}, and also the correction-interpolation operator I_{k-1}^k which will transfer the coarse-grid solution back to the fine grid. Choices here are different than in regular elliptic problems.

Regular elliptic problems have symmetric principal parts, i.e., the matrix corresponding to L^h is nearly symmetric. For such problems variational discretizations are very natural, and similarly, variational coarsening techniques are straightforward and efficient, even in highly discontinuous problems (see [1]): Having chosen a suitable interpolation I_{k-1}^k (suitably reflecting the smoothness properties of solutions to the homogeneous equations) one can simply take

$$ {}_iI_k^{k-1} = I_{k-1}^{k*} \qquad (6.9a)$$

$$ L_i^{k-1} = {}_iI_k^{k-1} \, L^k \, I_{k-1}^k \ , \qquad (6.9b)$$

where I_{k-1}^{k*} denotes the adjoint of I_{k-1}^{k}. That is, the matrix corresponding to I_{k-1}^{k*} is the transpose of the one corresponding to I_{k-1}^{k}. Non-elliptic and singular perturbation problems, on the other hand, are highly asymmetric, sometimes even antisymmetric. The choice (6.9) then leads to bad approximations on coarser grids. Relation (6.9b), to be sure, can still be used, but with a different ${}_iI_k^{k-1}$.

Generally, any method of coarsening implies a certain method of discretization (see [12]), which show that the task of coarsening is at least as difficult as that of discretization. For non-elliptic and singular perturbation problems, discretization techniques are mainly based on physical considerations, such as upstream differencing or artificial viscosity (Sec. 3). Hence we have used similar physical considerations in coarsening. In fact, we use the same operator L_i^{k-1} as when grid $k-1$ is the finest grid. Thus, in case double discretization is used on the finest grid ($L_1^{\ell} \neq L_0^{\ell}$), it is also used on coarser grids. (Except for some particular experiments designed to measure algebraic convergence factors.)

The residual-transfer operators ${}_iI_k^{k-1}$ should be of the full-weighting type (cf., e.g., Sec. 3 in [10]),since the transferred residuals are not likely to be smooth enough for injection. One reason for their non-smoothness is that they are calculated by a different operator than the one used in relaxation ($L_1^k \neq L_0^k$). Other potential reasons are the RB relaxation (for which "half-injection", as in [19], cannot be made if $L_1^k \neq L_0^k$), and the presence of discontinuities.

The two operators ${}_0I_k^{k-1}$ and ${}_1I_k^{k-1}$ need not be different. It seems however that somewhat different considerations enter in choosing them. Since L_0^{k-1} has an upstream bias, it seems advantageous to use such a bias also in ${}_0I_k^{k-1}$. We have for example experimented with the transfer

$$({}_iI_k^{k-1}\,R^k)(\underline{x}) = \tfrac{1}{4}[R^k(\underline{x}) + R^k(\underline{x}-\underline{h}_1) + (R^k(\underline{x}-\underline{h}_2) + (R^k(\underline{x}-\underline{h}_1-\underline{h}_2)] \qquad (6.10)$$

applied for residuals of approximations to (3.1) in case $d=2$, $a_1>0$, $a_2>0$. We call it <u>downstream residual transfer</u>, since each residual on grid k is added at a point $\underline{x}$ on the coarser grid which roughly lies downstream from the point of the residual. This is also physically reasonable to do: Under strong convection, influence of forcing terms (such as residuals) is convected downstream.

On the other hand, ${}_1I_k^{k-1}$ should be such that higher order approximation is obtained in transferring residuals from grid $k-1$ to grid $k-2$. Central interpolation is then natural. So for $i=1$ we may prefer the usual full-weighting operator

$$({}_iI_k^{k-1}R^k)(\underline{x}) = \sum_{|\underline{\nu}|\le 1} 2^{-d-|\underline{\nu}|}\, R^k(\underline{x}+\underline{\nu}h)\ , \qquad (6.11)$$

where $\underline{\nu} = (\nu_1,\ldots,\nu_d)$, ν_i are integers and $|\underline{\nu}| = |\nu_1| + \ldots + |\nu_d|$.

The residual transfer is of course simpler and somewhat less expensive if ${}_1I_k^{k-1} = {}_0I_k^{k-1}$. We have therefore sometimes used either (6.10) or (6.11) for both transfers.

<u>The interpolation</u> I_{k-1}^k should reflect the smoothness properties of solutions with relatively small residuals, which are like the smoothness properties of solutions to the homogeneous differential equations.

In case of non-elliptic or singular-perturbation problems this implies symmetric streamwise interpolation. Because of the grid geometry, however, this is not possible to do (except approximately, and by using values at far coarse-grid points). We have therefore mostly used the usual symmetric bilinear interpolation from nearest coarse-grid neighbors. We have also examined the efficiency of using first-order interpolation from upstream, meaning

$$(I^k_{k-1} V^{k-1})(\underline{x}) = V^{k-1}(\underline{x}^{US}), \tag{6.12}$$

where $\underline{x}^{US}$ is the nearest upstream coarse-grid point. For example, in case (3.1), $\underline{x}^{US}$ is the point nearest to $\underline{x}$ among all coarse-grid points $\underline{y}$ such that $(x_j - y_j)\, a_j \geq 0$, $(j = 1,\dots,d)$.

7. NUMERICAL EXPERIMENTS

Results of preliminary numerical experiments are described in [37], a further systematic study is reported here, and a fuller account will be given elsewhere. The experiments reported here were all performed in collaboration with D. Sidilkover. They all deal with the multigrid solution of our model differential equation in two dimensions

$$LU \equiv -(0+)\Delta U + a_1\frac{\partial U}{\partial x_1} + \frac{\partial U}{\partial x_2} = F\,, \qquad (7.1)$$

where $0+$ means an infinitesimally small positive number (much smaller than any meshsize used). The sign of this singular perturbation coefficient is important, since it determines the position of boundary layers and the sign of the $O(h)$ numerical viscosity. Note that we normalize our equation, taking here $a_2 = 1$. With no loss of generality we examine only the case $0 \leq a_1 \leq 1$, so that $a = \max|a_j| = 1$. With these signs of a_1, a_2 and the singular-perturbation coefficient, the downstream direction is always in positive-x_1-positive-x_2 direction.

For coding simplicity we chose our domain to be the rectangle

$$\Omega = \{(x_1,x_2):\ \ 0 \leq x_1 \leq 3\,,\ \ \ 0 \leq x_2 \leq 2\}\,, \qquad (7.2)$$

and we placed Dirichlet boundary conditions all around its boundary. With the above signs, only the conditions on the $\{x_1 = 0\}$ and $\{x_2 = 0\}$ boundaries actually affect the solution, while the Dirichlet conditions on the other boundaries will usually create boundary-layer discontinuities.

This problem can of course be solved as a time-dependent problem, x_2 for example taken as the time coordinate. But for us it just serves as a simple example for a non-elliptic BVP, hence the processes we apply are only such that can be applied to general BVPs.

We used five grids (levels). Each of them is placed so that the four segments of the boundary $\partial\Omega$ all coincide with gridlines. The meshsize of grid k is $h_k = 2^{1-k}$, hence it has $(3 \times 2^{k-1} - 1) \times (2^k - 1)$ interior grid points and 5×2^k boundary points, $(1 \leq k \leq 5)$.

7.1 FMG algorithms for smooth solutions

The purpose of our evperiments is to separate as far as possible the various algorithmic questions, so that each can be studied on its own. First we study how effective are different algorithms in case of smooth solutions, avoiding any influence of discontinuities. For this purpose we solve (7.1) with

$$\begin{aligned} F &= a_1 \cos x_1 \sin x_2 + \sin x_1 \cos x_2 && \text{in } \Omega , \\ U &= \sin x_1 \sin x_2 && \text{on } \partial\Omega , \end{aligned} \qquad (7.3)$$

so that $U = \sin x_1 \sin x_2$ is the exact solution. With these particular boundary conditions there are no boundary layers. In other experiments we chose the boundary values so that boundary layers did appear, but this affected our solution only near those boundaries, and, on the other hand, the large errors in the boundary layers confused our study of the smooth regime, hence we show here the clearer results without boundary layers. To avoid influence of even small numerical boundary layers, we have measured our errors only in a subdomain bounded away from the potential layers. Thus, all the error measurements shown in Table 3 are L_1 norms of the difference $u^h - U$ in the subdomain

$$\Omega' = \{(x_1,x_2): \quad 0 \leq x_1 \leq 2 , \quad 0 \leq x_2 \leq 4/3\} , \qquad (7.4)$$

where u^h is the numerical solution produced by the specified algorithm.

The algorithms are all of the FMG(N, N_M, C, M) type (Sec. 6.2) with $M = 5$, using the two basic discretizations of (7.1)

$$L_i^\ell U^\ell \equiv (-\beta_i h_\ell \Delta^\ell + a_1 \hat{\partial}_1^\ell + \hat{\partial}_2^\ell) U^\ell = F^\ell , \quad (i=0,1), \qquad (7.5)$$

where Δ^ℓ and $\hat{\partial}_j^\ell$ are as Δ^{h_ℓ} and $\hat{\partial}_j^{h_\ell}$ in (3.2), and F^ℓ is a certain representation (injection, unless otherwise specified) of $F(x,y)$ on grid ℓ. The first interpolation to a new grid $\overline{\Pi}_\ell^{\ell+1}$ is always cubic interpolation. Since the coarsest level $(\ell = 1)$ is too coarse to cubically interpolate from, we have used five C(2) cycles to safely reduce the error on grid 2 below truncation errors.

Each column in Table 3 describes one numerical experiment. Above the double line the parameters of the experiment are given, and below it some output results are shown, all as indicated by the left-most column. The cycles C shown in the so-designated row are explained in Sec. 6.1. "N" is the number of cycles at intermediate levels (levels 3 and 4). In the row "Relax" the relaxation scheme is specified: It is always a point-by-point Gauss-Seidel scheme; RB indicates red-black ordering, Lex+ lexicographic ordering, Lex- reversed lexicogrpahic ordering, and Lex-+ indicates ordering reversed in x_1 and forward in x_2. "β_0" and "β_1" are the artificial-viscosity coefficients used in relaxation and in residual transfers, respectively (see (7.5)). In some particular experiments anisotropic artificial viscosity was used, in the form

$$L_i^\ell = -\beta_i h_\ell \partial_{jj}^\ell + a_1 \hat{\partial}_1^\ell + \hat{\partial}_2^\ell , \qquad (7.6)$$

where ∂_{jj}^ℓ is as $\partial_{jj}^{h_\ell}$ in (3.6). This is indicated by (j) being attached to the value of β_i. In the row designated "${}_iI_k^{k-1}$", the type of residual transfer is specified: "Cen" means that the central full weighting (6.11) was used for both ${}_0I_k^{k-1}$ and ${}_1I_k^{k-1}$, "Dwn" denotes the downstream transfer (6.10), "Dwn2" indicates the modified downstream transfer

$$({}_iI_k^{k-1}R^k)(\underline{x}) = \frac{1}{8}[R^k(\underline{x}-\underline{h}_1) + R^k(\underline{x}+\underline{h}_1) + R^k(\underline{x}-\underline{h}_1-\underline{h}_2) + R^k(\underline{x}+\underline{h}_1-\underline{h}_2)]$$
$$+\frac{1}{4}[R^k(\underline{x}) + R^k(\underline{x}-\underline{h}_2)] , \qquad (7.7)$$

and "DwCn" means that (6.10) was used for ${}_0I_k^{k-1}$ and (6.11) for ${}_1I_k^{k-1}$. In the row titled "I_{k-1}^k" the type of the coarse-to-fine correction interpolation is given: "Lin" stands for the usual bilinear interpolation; "Ups" for the first-order interpolation from upstream (6.12); and "SSL" for Streamwise Symmetric Linear interpolation, which is the same as Lin except for points $\underline{x}$ of grid k neither on rows nor on columns of grid k-1, where the linear interpolation

$$(I_{k-1}^k \; V^{k-1})(\underline{x}) = \tfrac{1}{2}[V^{k-1}(\underline{x}+\underline{h}_1+\underline{h}_2) + V^{k-1}(\underline{x}-\underline{h}_1-\underline{h}_2)] \tag{7.8}$$

is used (for the case $a_1 > 0$).

In the row marked "Other" we indicate deviations from the above standards. "U_1" indicates that the solution $U_1 = \sin(2\pi x_1)\,\sin(2\pi x_2)$ is used instead of (7.3). Also, whereas usually we use injected right-hand sides $F^\ell(\underline{x}) = F(\underline{x})$, with U_1 it is necessary to use proper weighting, lest the results on coarser grids will be grossly distorted. So with "U_1" we use $F^\ell(\underline{x}) = {}_iI_{\ell+1}^\ell\, F^{\ell+1}(\underline{x})$ for $\ell < M$, where $I_{\ell+1}^\ell$ is the central full weighting (6.11) and

$$F^M(\underline{x}) = F(\underline{x}) = 2\pi a_1 \cos(2\pi x_1)\sin(2\pi x_2) + 2\pi\sin(2\pi x_1)\cos(2\pi x_2) .$$

"Ω_1'" in the row "Other" indicates that

$$\Omega_1' = \{(x_1,x_2):\ .5 \leqslant x_1 \leqslant 2.5 \ ,\quad 0 \leqslant x_2 \leqslant 4/3\}$$

is used instead of (7.4).

The output rows show the error $\|u^h - U\|_{L_1(\Omega')}$ at various stages, the stage being specified in the left-most column: In the row marked "$\ell(n)$", u^h is the result of FMG(N, n, C, ℓ) where C is the cycle $V(r_1,r_2)$ or $W(r_1,r_2)$ or $F(r_1,r_2)$, as indicated in the row "Cycle". In case $r_2 \neq 0$ we show a pair of numbers in row "$\ell(n)$" (for $n > 0$), the first being the error before the last relaxation sweep, the second being the error after that sweep.

a_1	1														$2/\pi$	
Cycle	W(2,1)				W(2,0)				W(1,0)				W(1,0)		W(1,0)	
N	2				2	2	1	1	2	2	1	1	2		2	
Relax	RB				RB				RB				RB		RB	
β_0	1	1	1	.5	1	.5	1	.5	1	.5	1	.5	1	.5	1	.5
β_1	1	0	0	0	0				0				1	.5	0	0
${}_iI_k^{k-1}$	Cen	Cen	Dwn	Dwn	Dwn				Dwn				Cen		Dwn	
I_{k-1}^k	Lin				Lin				Lin				Lin		Lin	
Other																
2(5)	.907	.624 .741	.569 .708	.303 .371	.530	.287	.530	.287	.439	.241	.439	.241	.903	.467	.465	.221
3(N)	.570	.0516 .146	.0575 .170	.0258 .115	.0416	.0254	.0896	.0258	.0548	.0347	.148	.0410	.558	.262	.0725	.0346
4(N)	.239	.0105 .0261	.0081 .0332	.0062 .0298	.0085	.0071	.0124	.0076	.0089	.0095	.0187	.0153	.231	.113	.0098	.0100
5(0)	.238	.0328	.0399	.0349	.0064	.0057	.0146	.0075	.0060	.0053	.0175	.0128	.232	.110	.0068	.0052
5(1)	.114	.0031 .0067	.0026 .0089	.0018 .0084	.0026	.0022	.0026	.0025	.0024	.0038	.0051	.0058	.114	.0566	.0024	.0035
5(2)	.120	.0025 .0071	.0023 .0086	.0019 .0081	.0023	.0021	.0025	.0022	.0024	.0025	.0027	.0035	.121	.0579	.0024	.0025
5(6)	.120	.0025 .0072	.0023 .0084	.0019 .0081	.0023	.0020	.0023	.0020	.0024	.0023	.0024	.0025	.120	.0580	.0024	.0023
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

TABLE 3

a_1	1											
Cycle	W(1,0)											
N	2											
Relax	RB											
β_0	.2	.3	.4	.5	.6	.7	.8	.9	1.0	2.0	4.0	8.0
β_1	0											
${}_iI_k^{k-1}$	Dwn											
I_{k-1}^k	Lin											
Other												
2(5)	7.75	.195	.183	.241	.290	.336	.376	.410	.439	.591	.683	.732
3(N)	56.5	.124	.0421	.0347	.0319	.0330	.0382	.0454	.0548	.184	.312	.395
4(N)	9580	.177	.0150	.0095	.0085	.0084	.0085	.0087	.0089	.0234	.0911	.176
5(0)	$.114\times10^5$	.181	.0123	.0053	.0044	.0045	.0049	.0055	.0060	.0226	.0884	.171
5(1)	$.163\times10^8$	.251	.0114	.0038	.0028	.0026	.0025	.0025	.0024	.0053	.0374	.113
5(2)	$.749\times10^{10}$	1.12	.0050	.0025	.0024	.0024	.0024	.0024	.0024	.0030	.0171	.0765
5(6)	$.123\times10^{22}$	679[illegible]	.0078	.0023	.0023	.0023	.0024	.0024	.0024	.0025	.0034	.0176
	17	18	19	20	21	22	23	24	25	26	27	28

TABLE 3 cont'd

a_1	1									
Cycle	V(2,0)		V(1,0)		F(1,0)		F(2,0)		F(1,0)	
N	1		2		2		1		1	
Relax	RB		RB		RB		RB		RB	
β_0	1	.5	1	.5	1	.5	1	.5	1	.5
β_1	0				0					
${}_iI_k^{k-1}$	Dwn				Dwn					
I_{k-1}^k	Lin				Lin					
Other										
2(5)	.530	.287	.439	.241	.439	.241	.530	.287	.439	.241
3(N)	.167	.0253	.156	.0425	.0548	.0347	.0896	.0258	.148	.0410
4(N)	.0312	.0091	.0330	.0101	.0092	.0095	.0118	.0076	.0191	.0153
5(0)	.0341	.0101	.0354	.0090	.0070	.0054	.0136	.0074	.0173	.0133
5(1)	.0088	.0027	.0156	.0054	.0024	.0037	.0025	.0025	.0054	.0065
5(2)	.0044	.0025	.0088	.0030	.0024	.0025	.0024	.0022	.0027	.0036
5(6)	.0042	.0023	.0055	.0027	.0024	.0023	.0023	.0020	.0024	.0025
	29	30	31	32	33	34	35	36	37	38

TABLE 3 cont'd

a_1	1													
Cycle	W(1,0)		W(2,0)		W(1,0)		W(1,0)		W(2,0)		W(1,0)		W(1,0)	
N	2		1		1		2		1		1		2	
Relax	Lex+		Lex+		Lex+		Lex-		Lex-		Lex-		Lex-+	
β_0	1	.5	1	.5	1	.5	1	.5	1	.5	1	.5	1	.5
β_1	0													
${}_iI_k^{k-1}$	Dwn													
I_{k-1}^k	Lin													
Other														
2(5)	.0890	.0685	.157	.0685	.0890	.0685	.232	.0598	.218	.0452	.232	.0598	.142	.0491
3(N)	.0242	.0278	.0571	.0278	.0566	.0278	.0078	.0165	.0369	.0137	.0338	.0121	.0292	.0160
4(N)	.0080	.0083	.0107	.0083	.0126	.0083	.0053	.0052	.0086	.0095	.0073	.0083	.0070	.0047
5(0)	.0139	.0580	.0197	.0580	.0184	.0580	.0024	.0016	.0095	.0067	.0084	.0040	.0070	.0054
5(1)	.0012	.0026	.0017	.0026	.0025	.0026	.0017	.0020	.0013	.0013	.0022	.0014	.0015	.0012
5(2)	.0012	.0026	.0011	.0026	.0008	.0026	.0018	.0016	.0018	.0014	.0017	.0016	.0013	.0014
5(6)	.0012	.0026	.0011	.0026	.0012	.0026	.0015	.0016	.0014	.0014	.0015	.0016	.0015	.0013
	39	40	41	42	43	44	45	46	47	48	49	50	51	52

TABLE 3 cont'd

a_1	1													
Cycle N	W(1,0) 2													
Relax	RB													
β_0	1	.5	1	.5	1	.5	1	.5	1	.5	1	.5	1	.5
β_1	0				0				0		0			
${}_iI_k^{k-1}$	Cen	Cen	Dwn	Dwn	Cen	Cen	Dwn	Dwn	DwCn	DwCn	Dwn	Dwn	DwCn	DwCn
I_{k-1}^k	Lin				Ups				Lin		SSL			
Other														
2(5)	.504	.257	.439	.241	.632	.346	.541	.307	.496	.297	.474	.266	.524	.316
3(N)	.104	.0521	.0548	.0347	.173	.0470	.166	.0537	.0584	.0520	.0654	.0362	.0752	.0482
4(N)	.0375	.0149	.0089	.0095	.0330	.0104	.0325	.0105	.0168	.0145	.0095	.0101	.0166	.0154
5(0)	.0345	.0083	.0060	.0053	.0361	.0081	.0345	.0099	.0161	.0106	.0062	.0060	.0172	.0113
5(1)	.0075	.0068	.0024	.0038	.0144	.0042	.0151	.0041	.0054	.0071	.0025	.0047	.0073	.0087
5(2)	.0049	.0050	.0024	.0025	.0074	.0026	.0082	.0026	.0039	.0065	.0024	.0028	.0044	.0076
5(6)	.0028	.0033	.0024	.0023	.0027	.0023	.0027	.0023	.0025	.0058	.0024	.0024	.0028	.0099
	53	54	55	56	57	58	59	60	61	62	63	64	65	66

TABLE 3 cont'd

a_1	0										
Cycle	W(1,0)										
N	2										
Relax	RB				Lex+	Lex+	Lex-	Lex-	RB	Lex+	Lex-
β_0	1	.5	1	.5	1	.5	1	.5	.5(2)	.5(2)	.5(2)
β_1	1	5	0	0	0				0		
${}_iI_k^{k-1}$	Dwn2				Dwn2				Dwn2		
I_{k-1}^k	Lin				Lin				Lin		
Other	Ω_1'				Ω_1'				Ω_1'		
2(5)	1.23	.722	.977	.436	.683	.125	.745	.172	.252	.187	.0378
3(N)	.990	.427	.300	.0430	.154	.0272	.142	.0165	.0647	.0298	.0538
4(N)	.392	.179	.0279	.0112	.0078	.0078	.0182	.0109	.0152	.0040	.0133
5(0)	.402	.179	.0240	.0084	.0120	.0125	.0217	.0054	.0067	.0454	.0072
5(1)	.186	.0927	.0136	.0062	.0082	.0014	.0113	.0038	.0049	.0010	.0040
5(2)	.186	.0916	.0035	.0030	.0015	.0018	.0023	.0014	.0039	.0010	.0043
5(6)	.193	.0916	.0029	.0028	.0024	.0020	.0021	.0022	.0028	.0010	.0107
	67	68	69	70	71	72	73	74	75	76	77

TABLE 3 cont'd

a_1	0										
Cycle	W(1,0)										
N	2										
Relax	RB						RB				
β_0	.25	.3	.5	.7	1.	2.	.3(2)	.5(2)	.7(2)	1.(2)	2.(2)
β_1	0										
${}_iI_k^{k-1}$	Dwn2										
I_{k-1}^k	Lin										
Other	u_1,Ω_1'										
2(5)	.0026	.0022	.0013	.0010	.0007	.0004	.0170	.0022	.0016	.0011	.0006
3(N)	.298	.623	1.28	1.50	1.61	1.69	.561	.420	.998	1.39	1.67
4(N)	.314	.156	.657	1.06	1.41	1.96	3.61	.322	.363	.764	1.56
5(0)	.388	.231	.692	1.17	1.60	2.09	4.19	.368	.403	.850	1.74
5(1)	.241	.127	.376	.725	1.14	1.80	7.16	.182	.168	.517	1.27
5(2)	.142	.118	.283	.490	.852	1.58	27.9	.168	.165	.233	.988
5(6)	.258	.108	.166	.250	.436	.935	$.682\times10^5$	.172	.121	.189	.427
	78	79	80	81	82	83	84	85	86	87	88

TABLE 3 cont'd

7.2 Discussion of the results

The experiments clearly show the advantage of double discretization: Experiments with $\beta_1 = \beta_0$ (columns 1, 13, 14, 67, 68) show only $O(h)$ convergence and much larger errors then with $\beta_1 = 0$. (The use of ${}_iI_k^{k-1}$ = Cen in experiments 1, 13, 14 is immaterial. When $\beta_1 = \beta_0$ Cen and Dwn give essentially the same results.)

Relaxation has temporary bad effect on the solution in double discretization schemes because of its lower order (experiments 2, 3, 4), but this is only temporary (compare the errors in columns 3 and 4 before the last relaxation with the corresponding errors in columns 5 and 6).

For $\beta_1 = 0$ the errors are clearly $O(h^2)$. This second-order convergence is obtained without too much work. The tables show that best results are obtained already by FMG(2, 2, W(1,0), M) which involves 9.7 work units (6.1 in three dimensions), by FMG(1, 1, W(2,0), M) which involves 8 units (4.6 in 3 dimensions), and even by FMG(1, 1, W(1,0), M) (see columns 50-52) which involves only 5.3 units (3.0 in 3 dimensions). This work count is based on (6.8), in which each residual transfer from grid M is counted as one work unit.

If the work count is somewhat larger than in some regular elliptic problems, this is due to the use of W cycles which are 3/2 (7/6 in 3 dimensions) more expensive than V cycles. The W cycles do seem safer (compare columns 7 and 9 with 29 and 31) although with a more careful choice of β_0, V cycles may approach similar results (columns 30 and 32). F cycles (columns 33-38) give results essentially identical with the corresponding W-cycle results (columns 7-12), but their work is also essentially the same (except for very large grids in one dimension). Since a real concern about computer time usually arises only in three-dimensional problems, where W cycles are just slightly more expensive than V cycles, we preferred using mainly W cycles in our further experiments.

Notice that in case of Lex+ relaxation with upstream differencing (columns 40, 42, 44 and 76) V cycles would give results identical with those shown for W cycles, because once such a relaxation is made (on any problem or correction problem) the upstream solution is immediately introduced and the previous history (of solving that problem) is immaterial. Hence in these cases results are also independent on r_1 and N (columns 40, 42 and 44 are identical). In fact, in these cases the FMG algorithm itself is redundant: We would obtain the same results on grid M if we started there with any solution, just performing one V(1,0)(M) cycle. This would cost only 2.7 work units (2.3 in 3 dimensions)! But this is true only for purely hyperbolic schemes where the downstream direction of relaxation is the same throughout the domain.

It is interesting to compare the errors in Table 3 to the "ideal" errors associated with the residual-transfer operator L_1^h. Taylor expansions show that the discretization error $e^h = U^h - U$ approximately satisfies $L_1^h e^h = -\frac{h^2}{6}(a_1\partial_1^3 + \partial_2^3)U$. In case of our solution $U = \sin x_1 \sin x_2$ we therefore get $L_1 e^h = \frac{h^2}{6} L_1 U$, hence $e^h = \frac{h^2}{6} U$ and

$$\|e^h\|_{L_1(\Omega')} = \frac{h^2}{6}(1 - \cos 2)(1 - \cos\frac{4}{3}) = .181\,h^2 = .00075\ ,$$

$$\|e^h\|_{L_1(\Omega_1')} = \frac{h^2}{6}(\cos\frac{1}{2} - \cos\frac{5}{2})(1 - \cos\frac{4}{3}) = .214\,h^2 = .00084\ ,$$

for $h = 1/16$ (meshsize of our grid 5). In case of $U_1 = \sin 2\pi x_1 \sin 2\pi x_2$ we similarly obtain $\|e^h\|_{L_1(\Omega_1')} = \frac{22}{3}h^2 = .0286$. These ideal errors do not take into account large high-frequency errors that would be introduced if L_1^h were the only discretization used. Still, we can see that the ideal error is sometimes approached by our numerical results (columns 39, 41, 43, 45-52, 76). In case RB relaxation is used, the results are somewhat worse because of the high-frequency errors introduced by such schemes (see Sec. 6.3; in our example here $p_0 = m = 1$). In view of other RB advantages (Sec. 6.3), its use may still be recommended (see comment at the end of this section). The results are also somewhat worse for the Lex+ relaxation with $\beta = .5$ (columns 42, 44). This can be explained by the lower order of convergence (see Sec. 2.2; here $p_0 = 1$ and $r_0 = 0$). Thus surprisingly in a way, the relaxation Lex-, against the stream direction, as well as Lex-+, give best results in the shortest algorithm (column 50-52).

The optimal artificial viscosity for relaxation is studied in experiments 17-28, 78-88. We see that there exists a reasonable range of insensitivity, with the optimal β_0 obtained close to the minimal value of convergence predicted by mode analysis (compare with Table 2 and see the discussion at the end of Sec. 5.7). The minimal value itself can be used, but smaller values give fast divergence. For large β_0 the error grows linearly with β_0.

In experiments 53 to 66 we have tried various combinations of residual-weighting and correction-interpolation schemes. Downstream residual transfer (Dwn) seems to perform better than the usual (central) full weighting (Cen). On the other hand the correction interpolation from upstream (Ups) performs worse than the usual linear interpolation, simply because of its lower order. (See remark on interpolation orders in Sec. 2.2. In our example here $p = 2$ and $p_0 = 1$.). The symmetric streamwise linear interpolation (SSL) does not improve over the usual interpolation, since both reach the lowest errors possible with RB schemes. The combined residual-weighting procedure DwCn performs worse than Dwn, indicating that Dwn is a better weighting than Cen even for smooth components. This issue perhaps requires a further clarification by mode analysis.

The last set of experiments (67-88) deals with the case of a flow aligned with the x_2-gridlines ($a_1 = 0$). Note that the later half of these experiments (78-88) were performed with a different solution (U_1), so the results are not directly comparable to those in the first half. The best results are obtained for upstream differencing with downstream relaxation (column 76. Remember, furthermore, that identical results would be obtained by using the simpler cycle V(1,0) with $N = 1$.) This, however, is not a direction-free result. If we relax against the stream direction, or in red-black ordering, results (75-77) are considerable worse. By contrast, results with full h-ellipticity are direction-free (columns 71-74). They seem to be worse than the upstream-differencing results, but this is because they do not use their minimal artificial viscosity. When isotropic artificial viscosity (7.5) is used with the minimal β_0 for which RB Gauss-Seidel still converges (in case $a_1 = 0$ this minimum is $\beta_0 = 12^{-1/2} = .288$), the performance is better than with upstream differencing (compare column 79 with 85-86). Still, the performance shown here for isotropic-viscosity differencing is not as much better (than for upstream differencing without downstream relaxation) as might be expected from smoothing-rate analysis. The reason: its algebraic smoothing is much better, but not its differential smoothing. (See Sec. 2.2 and end of Sec. 3.3.) The results here indicate that for variable-direction flows upstream differencing with symmetric Gauss Seidel relaxation may give very good results (not better than isotropic-viscosity differencing, to be sure, unless the flow consistently aligns with the grid). The optimal artificial viscosity may indeed be anisotropic, although not necessarily the one yielding upstream differencing.

Comparing results for $a_1 = 1$ and $a_1 = 0$, the best isotropic artificial viscosity seems to be as in (7.5), with $\beta_0 = (1+a_1)/3$.

Last important comment: Perhaps the small differences between many of the second-order results (those with $\beta_1 = 0$) do not matter much. The errors shown are so small that the overall error in a real problem may well be dominated by much larger errors committed near discontinuities.

7.3 Preliminary FMG experiments with contact discontinuity

Our next concern, then, is the behavior of discontinuities in multigrid processes (see Sec. 4). The numerical behavior of contact discontinuities is notorious for $O(h^{1/2})$ smearing entailed by usual artificial viscosity. So we test its behavior in our double-discretization multigrid algorithms.

The test problem is again equation (7.1) in the domain (7.2), in the special case $a_1 = 1$ and $F(x_1, x_2) \equiv 0$, with the discontinuous boundary condition

$$U(x_1, x_2) = H(x_1 - x_2 - 1) \quad \text{on } \partial\Omega , \qquad (7.9)$$

where $H(\xi) = 0$ for $\xi < 0$ and $H(\xi) = 1$ for $\xi \geq 0$ (Heaviside function). The solution is simply

$$U(x_1, x_2) \doteq H(x_1 - x_2 - 1) \quad \text{throughout } \Omega . \qquad (7.10)$$

We see that the boundary discontinuity at (1,0) travels along the characteristic line $\{x_1 - x_2 = 1\}$. This is called contact discontinuity. The boundary conditions are again chosen so that no boundary layers enter, so that we can more clearly examine the behaviour at the contact.

The numerical solution algorithms were the same as in Sec. 7.1, with the (double) discretization (7.5). The most interesting feature in the numerical solution is its profile around the discontinuity a certain distance away from the initial boundary. Such profiles are shown in Fig. 5 for the solution obtained by the algorithms $FMG(N, N, W(r_1,r_2), 5)$ with

RB relaxation, ${}_iI_k^{k-1} = \text{Dwn}$ and $I_{k-1}^k = \text{Lin}$, in the four representative cases:

(a) $r_1 = 2, \quad r_2 = 1, \quad N = 2, \quad \beta_0 = \beta_1 = 1$.

(b) $r_1 = 2, \quad r_2 = 1, \quad N = 2, \quad \beta_0 = \beta_1 = .5$.

(c) $r_1 = 1, \quad r_2 = 0, \quad N = 1, \quad \beta_0 = .5, \quad \beta_1 = 0$.

(d) $r_1 = 1, \quad r_2 = 0, \quad N = 2, \quad \beta_0 = .5, \quad \beta_1 = 0$.

The last two cases are effectively higher-order calculations that proved very good in the smooth cases. Typical to higher-order methods they exhibit less smearing but more oscillations. Various improvements, discussed in Sec. 4, are now being tested.

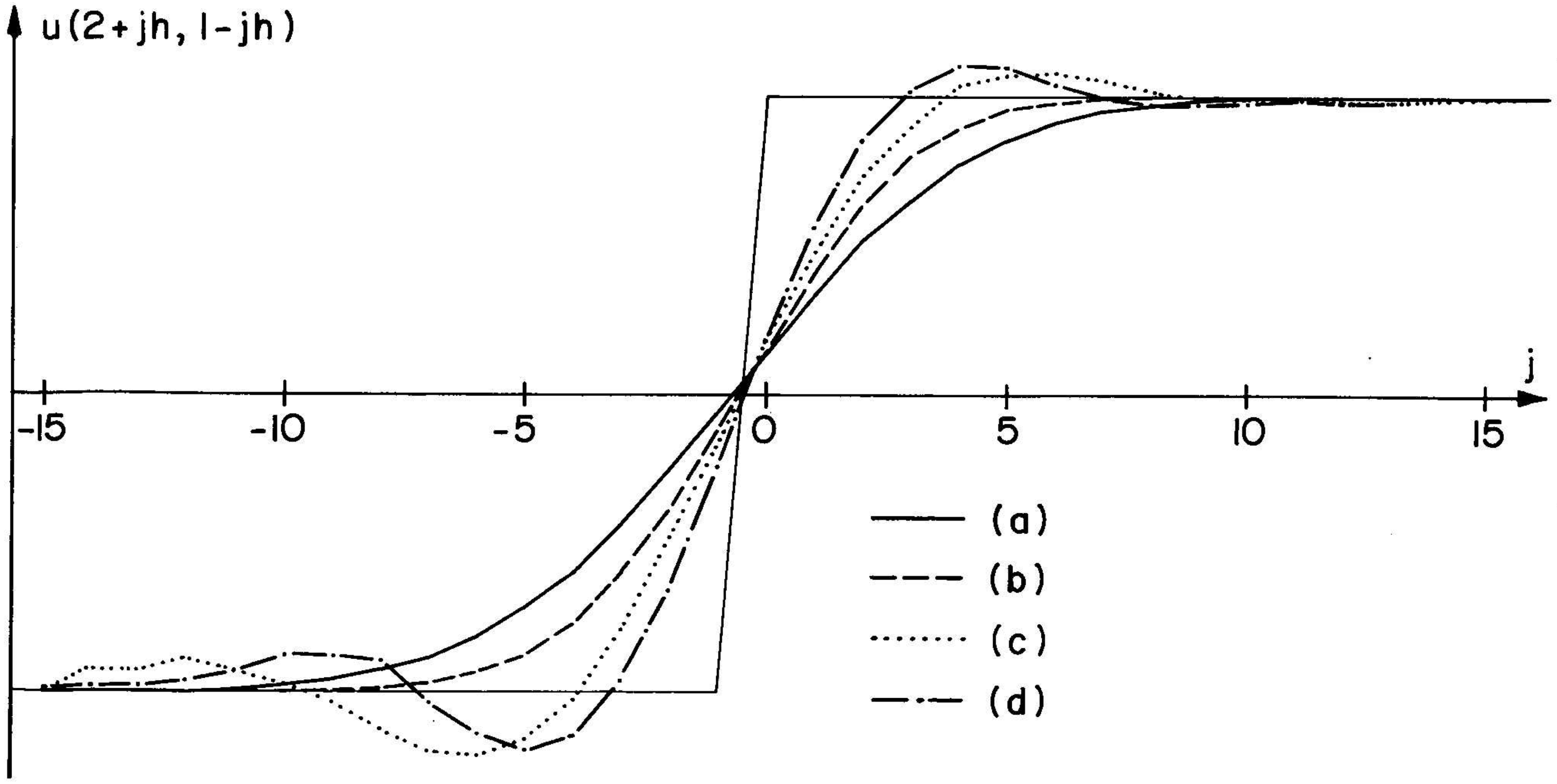

Figure 5. Smearing of contact discontinuity.

REFERENCES

[1] Alcouffe, R.E., Brandt, A., Dendy, J.E. Jr., and Painter, J.W. The multi-grid methods for the diffusion equation with strongly discontinuous coefficients. LA-UR-80-1463, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1980. SIAM J. Sci. Statistical Computing, to appear.

[2] Boris, J.P. & Book, D.L. Flux corrected transport. I. SHASTA, a fluid transport algorithm that works , J. Computational Phys. 11 (1973), 38-69.

[3] Brandt, A., Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems, in: Proceedings of the 3rd International Conference on Numerical Methods in Fluid Mechanics (Paris, 1972), Lecture Notes in Physics 18 , Springer-Verlag, Berlin and New York, 1973, 82-89.

[4] Brandt, A., Multi-level adaptive technique (MLAT). I. The multi-grid method. IBM Research Report RC-6026, IBM T.J. Watson Research Center, Yorktown Heights, New York, 1976.

[5] Brandt, A., Multi-level adaptive solutions to boundary-value problems. Math. Comp. 31 (1977), 333-390. ICASE Report 76-27.

[6] Brandt, A., Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software. Mathematical Software III (John R. Rice, Ed.), Academic Press, New York 1977, 273-314. ICASE Report 77-20.

[7] Brandt, A., Multi-level adaptive solutions to singular-perturbation problems. Numerical Analysis of Singular Perturbation Problems (P.W. Hemker and J.J.H. Miller, Eds.), Academic Press 1979, 53-142. ICASE Report 78-18.

[8] Brandt, A., Numerical stability and fast solutions of boundary-value problems. Boundary and Interior Layers - Computational and Asymptotic Methods (J.J.H. Miller, Ed.),Boole Press, Dublin 1980, 29-49.

[9] Brandt, A., Multi-level adaptive computations in fluid dynamics. Proc.

AIAA 4th Computational Fluid Dynamics Conf. (Williamsburg, Virginia, July 1979). ICASE Report No. 79-19. AIAA J. 18 (1980), 1165-1172.

[10] Brandt, A., Stages in developing multigrid solutions, Numerical Methods for Engineering (E. Absi, R. Glowinski, P. Lascaux, H. Veysseyre, Eds.), Dunod, Paris 1980, 23-44.

[11] Brandt, A., Multigrid solvers on parallel computers. Elliptic Problem Solvers. (M. Schultz, Ed.), Academic Press, New York 1981, 39-84.

[12] Brandt, A., Guide to multigrid solution development. December, 1981.

[13] Brandt, A., Multigrid solutions to steady-state compressible Navier-Stokes equations. Proc. Fifth International Symposium on Computing Methods in Applied Sciences and Engineering, Versailles, France, December 14-18, 1981.

[14] Brandt, A., and Dinar, N., Multigrid solutions to elliptic flow problems. Numerical Methods for Partial Differential Equations (S. Parter, Ed.), Academic Press 1979, 53-147. ICASE Report 79-15.

[15] Brandt, A., Dendy, J.E. Jr. and Ruppel, H., The multi-grid method for the pressure iteration in Eulerian and Lagrangian hydrodynamics. LA-UR-78-3066, Los Alamos Scientific Laboratory, Los Alamos, New Mexico. J. Computational Phys. 34 (1980), 348-370.

[16] Brandt, A. and Ta'asan, S., Multigrid methods for highly oscillatory problems. Research Report, September 1981.

[17] Brown, J.J., A Multigrid mesh-embedding technique for three-dimensional transonic potential flow analysis. Boeing Commercial Airplane Company, Seattle, Washington, April 1981.

[18] Dinar, N., Fast methods for the numerical solution of boundary-value problems, Ph.D. Thesis, The Weizmann Institute of Science, Rehovot, Israel (1979).

[19] Foerster, H., Stueben, K. and Trottenberg, U., Non-standard multigrid techniques using checkered relaxation and intermediate grids. Elliptic Problem Solvers. (M. Schultz, Ed.), Academic Press, New York, 1981, 285-300.

[20] Frank, L., Singular perturbations and finite differences, *C.R. Acad. Sci. Paris*, Ser. A 283(1976), 859-862.

[21] Frank, L., Coercive singular perturbations. I-A priori estimates. *Ann. Math Pura ed Appl*. (iv), 119 (1979), 41-113.

[22] Harten, A., The artificial compression method for computation of shocks and contact discontinuities: III. Self-adjusting hybrid schemes, *Math. Comp.* 32 (1978), 363-389.

[23] Harten, A., Hyman, J.M. and Lax, P.D., "On finite-difference approximations and entropy conditions for shocks" (with Appendix by B. Keyfitz), *Comm. Pure Appl. Math.* 29 (1976), 297-322.

[24] Hughes, T. J.R., and Brooks, A. A multi-dimensional upwind scheme with no crosswind diffusion. *Finite Element Methods for Convection Dominated Flows*, (T.J.R. Hughes, Ed.) The American Society of Mechanical Engineers, New York 1979, 19-36.

[25] Jameson, A., Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple-grid method , AIAA Paper, 79-1458, *Proceedings of AIAA Computational Fluid Dynamics Conference*, July 1979, Williamsburg, Va., 122-146.

[26] Lax, P.D. and Wendroff, B., Systems of conservation laws , *Comm. Pure Appl. Math.* 13 (1960), 217-237.

[27] Linden, J., Mehrgitterverfahren für die Poisson-Gleichung in Kreis und Ringgebiet unter Verwendung lokaler Koordinaten. Diplomarbeit, Bonn 1981.

[28] McCarthy, D.R. and Reyhner, T.A., A multi-grid code for three-dimensional transonic potential flow about axisymmetric inlets at angle of attack , AIAA Paper 80-1365, *Proceedings of AIAA 13th Fluid and Plasma Dynamics Conference*, July 1980, Snowmass, Colo.

[29] Richtmyer, R.D. and Morton, K.W., *Difference Methods for Initial-Value Problems*, 2nd ed., Interscience 1967.

[30] South, J.C. and Brandt, A., Application of a multi-level grid method to transonic flow calculations. ICASE Report No. 76-8, NASA Langley Research Center, Hampton, Virginia, 1976.

[31] Wesseling, P., Numerical solution of the stationary Navier-Stokes equations by means of a multiple grid method and Newton iteration. Report NA-18, Delft, August, 1977.

[32] Yanenko, N.N. and Shokin, Y.I., On the correctness of first differential approximations of difference schemes , *Doklady AN SSSR* 182, 4 (1968), 776-778.

[33] Yanenko, N.N. and Shokin, Y.I., On the first differential approximation of difference schemes for hyperbolic systems of equations , *Siberian Mathematical Journal* 10, 4 (1969), 1174-1188.

[34] Yanenko, N.N. and Shokin, Y.I., First differential approximation method and approximate viscosity of difference schemes . High-Speed Computing in Fluid Dynamics. *The Physics of Fluids*, Supplement II, New York, 1969.

[35] Arlinger, B., Multigrid technique applied to lifting transonic flow using full potential equation. SAAB-SCANIA Rep. L-0-1 B439, (1978).

[36] Brandt, A., Multi-level adaptive finite-elements methods: I. Variational problems. *Proc. of Conf. on Appl. Math.* (University of Bonn, 1979, Frehse, Pollaschke and Trottenberg, eds.), North Holland 1980, 91-128. ICASE Report 79-8.

[37] Börgers, C., Mehrgitterverfahren für eine Mehrstellendiskretisierung der Poisson-Gleichung und für eine Zweidimensionale singulär gestörte Aufgabe, Diplomarbeit, Bonn 1981.

[38] Mol, W.J.A., Numerical Solution of the Navier-Stokes Equations by Means of a Multigrid Method and Newton-Iteration. Afdeling Numerieke Wiskunde (Department of Numerical Mathematics) NW 92/80 Amsterdam, November 1980.