Multigrid Solvers for the Non-aligned Sonic Flow: The Constant Coefficient Case

Achi Brandt, Boris Diskin

Department of Applied Mathematics
and Computer Science
The Weizmann Institute of Science
Rehovot, Israel

Abstract

An approach for the construction of multigrid solvers for non-elliptic equations on a rectangular grid is presented. The results of both the analysis and the numerical experiments demonstrate that such an approach permits to achieve a full multigrid efficiency even in the case that the equation characteristics do not align with the grid. To serve as a model problem, the 2D and 3D linearized sonic-flow equations with a constant velocity field have been chosen. Efficient FMG solvers for the problems are demonstrated.

Contents

1 Introduction **3**

2 General description **4**
2.1 Differential problem 4
2.2 Principles of discretization 4
2.3 Strong cross-characteristic coupling 6

3 Two dimensional problem **6**
3.1 Problem statement 6
3.2 Multigrid cycles 8
3.2.1 Relaxation schemes 9
3.2.2 Switching criterion 10
3.2.3 Two-level cycles 11
3.2.4 Multilevel cycle 12
3.3 FMG solver and numerical experiments 12
3.4 Two-level Half-space Mode Analysis 14
3.4.1 Two-level FMG mode analysis 16
3.4.2 Computed results 23

4 3D problem: discretization and solver **27**
4.1 Problem statement 27
4.2 2D-Prototype: Laplacian on non-orthogonal grid 28
4.3 Discretization 29
4.4 Multigrid cycle 30
4.5 FMG solver: numerical results 31

A Tables of Numerical Results **33**

1 Introduction

The fastest solvers for discretized elliptic partial differential equations are the full-multigrid algorithms. The solution of general nonlinear elliptic systems can be obtained in just a few *minimal work units*, a *minimal* unit being defined as the number of operations required for the *simplest* discretization of the system on the target grid. However, attempts to apply the same techniques to non-elliptic equations, such as the supersonic (transonic) potential flow equation or its degenerate case - the sonic flow equation, have met with more limited success. Although such multigrid methods are usually much more efficient than comparable single-grid methods, the goal of solution in just a few minimal work units has not been attained. Indeed, many of the reported solvers require hundreds of minimal work units. Others, such as solvers based on various modifications of the ILU decomposition, being efficient in 2D, cannot be directly extended to 3D without losing most of their efficiency.

The increased amount of required work is usually contributed by a number of factors, and the first important step therefore is to separate them out. The present research addresses one basic difficulty in separation from others: the problem of non-alignment. The problem arises wherever characteristics of a differential equation do not coincide with grid lines. We study the non-alignment effect on the model tasks: the 2D and 3D sonic-flow equations linearized over a constant velocity field. These problems are reduced to solving semi-elliptic equations which are elliptic on low dimensional manifolds embedded in the original space. The 2D problem itself can also be solved by other methods, but the development of an efficient robust solver in 3D has remained an open problem up to now.

The solution approach suggested here is based on the Full Multigrid (FMG) algorithm (see, e.g., Sec. 7 in [3] or [2]). The usual goal of an FMG solver is to reduce a norm of the target-grid (grid h) *algebraic error* $\|u^h - \tilde{u}^h\|$ below the level of the *discretization error* $\|u^h - U^h\|$, where u^h and $\tilde{u}^h$ are the exact and computed solutions of the discretized equation, respectively, U^h is some target-grid representation of the true solution to the differential equation, and $\|\cdot\|$ is a given norm of interest. Sometimes it is useful to measure also *the total error* $\|\tilde{u}^h - U^h\|$.

In many cases regular FMG algorithms are sufficient to yield a final solution with an algebraic error much smaller than the discretization error. In some problems, however, the usual algorithm cannot efficiently treat certain smooth components of the target-grid solution. The sonic flow problem, for example, in which the discretization scheme introduces a numerical viscosity if the flow is not consistently aligned with the grid, falls into this category. The same trouble has been observed and treated for convection-diffusion equations (see [1]) and for high-Reynolds incompressible entering flows (see [5]). In such cases the usual multigrid cycles lose their efficiency and so does the FMG algorithm employing the cycles.

A simple explanation of the problem can be given when the characteristics of the differential equation emanate from the boundary, in which case the quality of the coarse-grid correction is determined by how well certain cross-characteristic oscillations are advected from the inflow boundary into the domain. The main trouble is the increased numerical viscosity on coarse grids, which causes the decay and phase shift of these cross-stream oscillations to differ much from their values on the fine grid.

The idea suggested in this paper to overcome the trouble is to use *semi coarsening* together with the introduction of a well-balanced *explicit numerical viscosity* on coarse grids

to control the penetration of the incoming cross-stream oscillations.

In Section 2 we give a general formulation of the problem and introduce the main ideas. This is followed by detailed descriptions of the concrete problems and the solution algorithms in 2D (Section 3) and 3D (Section 4). The half-space mode analysis given in Section 3.4 and the results of the numerical experiments (Appendix) show the high efficiency of the suggested solvers.

2 General description

2.1 Differential problem

The full potential flow equation has the quasi-linear form

$$(\bar{u} \cdot \nabla)^2 \Phi - a^2 \triangle \Phi = f, \tag{1}$$

where $\bar{u}$ is the velocity vector, a is the speed of sound and the unknown scalar function $\Phi(x, y)$ is the velocity potential of the irrotational flow. More important, the operator appearing in Eq. 1 is one of the factors of the principal part determinant of the Euler system of equations for compressible flows (see §21 in [3]). Hence, according to rules developed in [3] (see also [2], [5] and [8]), the development of fully efficient multigrid solvers for the Euler system depends on devising such solvers for the principal part of (1), in which $\bar{u}$ and a are *given* fields unrelated to Φ.

The type of this equation depends on the ratio $M = (\bar{u} \cdot \bar{u})^{1/2}/a$, which is called the Mach number. When $M = 1$, equation (1) degenerates to the *sonic flow equation*

$$\tilde{\triangle} \Phi = f, \tag{2}$$

where $\tilde{\triangle}$ is the Laplacian on a manifold orthogonal to the velocity field. We call each such manifold a *characteristic manifold.*

We study the phenomena of non-alignment in the case of *constant fields* $\bar{u}$ and a. In this case the characteristic manifold is a straight line in 2D (*characteristic line*) and an oblique plane in 3D (*characteristic plane*). Although relatively simple, the problem nevertheless contains one of the main difficulties appearing in flows not consistently aligned with the discretization grid. Moreover, the algorithms developed here for this simplified problem can be applied with some extension to more complicated problems including variable velocity fields.

2.2 Principles of discretization

The target grid on which we discretize our equation is a rectangular grid. Usually it is a uniform grid, but grids with any other aspect ratios are also acceptable. In the present research we use only grids with a fixed aspect ratio (the ratio of the meshsize in the reference axis to the meshsize in the vertical axis). For 3D, grids in the reference plane are always assumed to be uniform.

Let us consider a discretization of the differential operator (2) at a given grid point. There exists one and only one characteristic plane going throughout the point. In the

general position the characteristic plane does not contain any other grid point. In order to discretize such an operator we introduce ghost points located at the intersections of the characteristic plane with the adjacent vertical grid lines. The function value at a ghost point is interpolated from its genuine neighbors placed on the same vertical grid line. For small aspect ratios this interpolation degrades the discrete operator approximation order and to compensate for the degradation we also add a few points to the discretization stencil. These points are added on the vertical grid line crossing the central point at which the operator is defined.

The general approach to the discretization of operators defined on a characteristic manifold contains the following steps.

1) The target semi-elliptic operator is first discretized on the characteristic manifold, using the ghost points, employing an *h-elliptic discretization* (see §2.1 in [3] or [2]). We call this lower-dimensional discrete operator the *low-dimensional prototype*.

2) The real-dimension discretization is obtained from the low-dimensional prototype by using vertical linear interpolation to the ghost points together with the addition of several compensating points on the vertical grid line going through the point where the discrete operator is being defined. This discretization is h-elliptic in the full dimension.

Such a discretization possesses numerical viscosity (because of the non-alignment and the resulting interpolation) that appears to be relatively small, especially on grids with a high aspect ratio. Nevertheless, this viscosity exists on each grid. We will call it *the inherent numerical viscosity*, to distinguish it from the "explicit numerical viscosity" introduced below. Quantitatively, the inherent numerical viscosity is defined as the coefficient of the lowest pure cross-characteristic derivative arising in the *first differential approximation* (FDA) to the discrete operator (see [7]), the cross-characteristic direction being defined as the direction perpendicular to the characteristic manifold. In our model problems the cross-characteristic direction coincides with the velocity direction.

Let us introduce now some useful terms. A function defined on the space under consideration will be referred to as a *characteristic component* if it is a smooth function on the characteristic manifold. The terms *high-frequency characteristic component* and *smooth characteristic component* will refer to components that are very smooth on the characteristic manifold and are respectively highly-oscillating and smooth in the *cross-characteristic* direction (but not as smooth as in any of the characteristic directions).

Previous studies on several types of non-elliptic equations (see [5] and [1]) have shown that the basic trouble in constructing an efficient multigrid solver is the poor approximation of smooth characteristic components on coarse grids. The reason is the increased coarse-grid inherent numerical viscosity appearing in cycles with full coarsening, i.e. when the coarse grid has all the meshsizes twice as large as those of the fine grid. A general way to overcome this trouble would be to use *semi coarsening*, with meshsizes being doubled only in the reference plane. When applied in its pure form, semi coarsening also results in some difficulties, since the inherent numerical viscosity of the semi-coarsened grid will be much less than that of the fine grid. But we can supply the operator on the semi-coarsened grid with an additional term (explicit numerical viscosity term) so that the total viscosity on the semi-coarsened grid would be the same as on the fine grid.

The three- or even four-level version of such a cycle, with two pointwise relaxations on each level and an appropriate inter-grid transfers, can already be used to solve efficiently

the model problems discretized on a uniform target grid. However, the implementation of a cycle with more levels raises the following new difficulty.

2.3 Strong cross-characteristic coupling

The inherent numerical viscosity in our algorithms arises from the vertical interpolation to the ghost points. To obtain the same total viscosity, we introduce an explicit numerical viscosity on the coarse grids by adding a term which is a discrete approximation to a vertical derivative of a suitable order.

The multigrid theory of h-elliptic discrete operators (see [3], [1]) shows that a pointwise relaxation can only reduce the error components that oscillate in the strong-coupling directions. The coupling analysis of the discretization considered in Sec.3.2.2 and 4.4 below shows that the *target-grid* discrete-operator directions of stronger coupling approximately coincide with the characteristic manifold. Thus a target-grid pointwise relaxation can reduce efficiently the non-characteristic error components and also some of the high-frequency characteristic components of the error. That is all we need from the relaxation since the smooth characteristic components (and most of the high-frequency characteristic components) are well reduced on the next *semi-coarsened* grids. However, successive semi coarsening implies a fast decrease in the *inherent numerical viscosity* on the coarse grids and hence a fast increase in the weight of the compensating explicit numerical viscosity in the coarse-grid discrete operator. Thus the direction of the strongest coupling after several semi-coarsening steps tends to be vertical, hence any pointwise relaxation on such coarse grids can not efficiently reduce some *non-characteristic* components of the error. A way to eliminate this efficiency degradation is to use a vertical zebra line relaxation, in which all the points located on the same vertical grid line are relaxed simultaneously. Thus our general strategy is to use pointwise relaxation on several of the finest levels and switch to zebra relaxation on all coarser levels. Hence, although line relaxation is somewhat more expensive, the total amount of work remains nearly the same as for a solver using only pointwise relaxation.

Instead of switching to zebra relaxation on some of the coarser grids, one can avoid creating strong cross-characteristic coupling altogether by replacing part of the semi coarsening steps by *full* coarsening steps (see [6]). This *conditional coarsening* is slightly cheaper in computing time but considerably more complicated to program, especially in extensions to variable coefficients.

3 Two dimensional problem

3.1 Problem statement

In two dimensions Eq. (2) turns into the simple equation

$$\frac{\partial^2 \Phi}{\partial \xi^2} = F, \tag{3}$$

where the direction of differentiation ξ is perpendicular to the velocity direction.

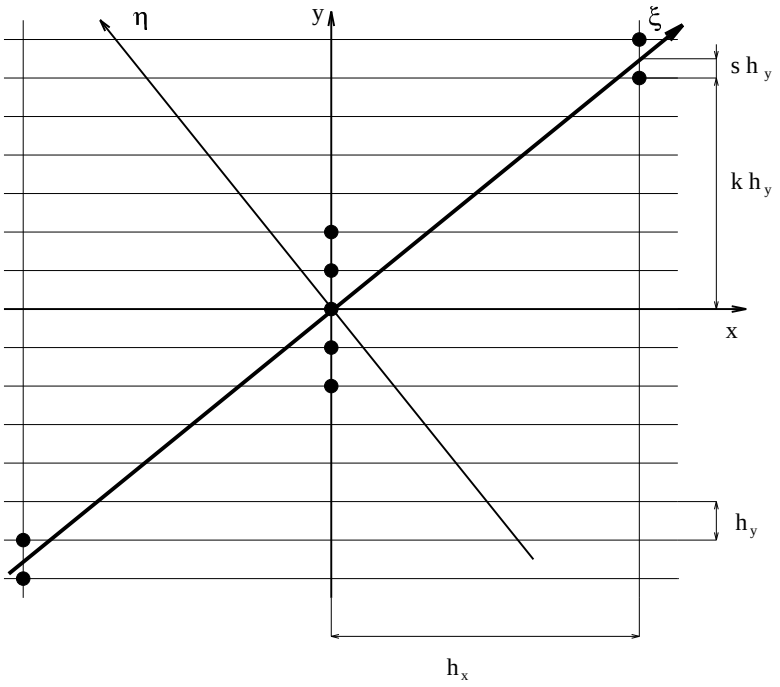

Figure 1: Anisotropic grid; nine point stencil.

In our model problem an unknown scalar function $\Phi(x,y)$ is defined on the square $(x,y) \in [0,1]\times[0,1]$, and the variable along the characteristic direction is $\xi = \Big(x+ty\Big)\Big(1+t^2\Big)^{-1/2}$, where $t = \tan\psi$ is the tangent of the *angle of non-alignment*, i.e., the angle between the characteristic direction and the reference $x-$axis. Usually $|t| \le 1$, otherwise one can obtain a better discretization than the one described below by switching the reference axis.

We supply Eq. (3) with Dirichlet boundary conditions in the x direction and periodic conditions in the y direction:

$$\Phi(0,y) = g_0(y), \quad \Phi(1,y) = g_1(y), \quad \Phi(x,y) = \Phi(x,y+1), \tag{4}$$

where $g_0(y)$ and $g_1(y)$ are given functions. The choice of periodic boundary conditions is quite important, as it precludes boundary layers, which could obscure the phenomena we would like to examine. It also facilitates implementation of any angle of non-alignment.

Let us introduce an auxiliary Cartesian coordinate system (ξ,η), where ξ is the variable along the characteristic direction defined above and $\eta = \Big(-tx+y\Big)\Big(1+t^2\Big)^{-1/2}$ is the variable along the cross-characteristic direction.

Consider the nine-point discretization of (3) on a grid with aspect ratio $m = h_x/h_y$, where h_x and h_y are the meshsizes in the x and y directions respectively. For integers (i_1,i_2) the discrete approximation to $\Phi(i_1h_x, i_2h_y)$ is denoted ϕ_{i_1,i_2}, and the discrete analog of the differential operator (3) is defined by

$$L^{(h_x,h_y)}\phi_{i_1,i_2} \equiv \frac{1}{h_x^2+(k+s)^2h_y^2}\Big[(1-s)\left(\phi_{i_1-1,i_2-k}+\phi_{i_1+1,i_2+k}\right)+s\left(\phi_{i_1-1,i_2-(k+1)}+\phi_{i_1+1,i_2+(k+1)}\right)$$

$$-2\phi_{i_1,i_2} - s(1-s)\left(\phi_{i_1,i_2-1} - 2\phi_{i_1,i_2} + \phi_{i_1,i_2+1}\right)] \qquad (5)$$
$$-A\frac{1}{h_y^2}\left[\phi_{i_1,i_2+2} - 4\phi_{i_1,i_2+1} + 6\phi_{i_1,i_2} - 4\phi_{i_1,i_2-1} + \phi_{i_1,i_2-2}\right],$$

where $k+s = mt$, k is integer and $0 \le s < 1$ (see Figure 1); A is the *explicit* numerical viscosity coefficient.Thus the differential problem (3)–(4) is discretized on the grid as

$$\begin{aligned} L^{(h_x,h_y)}\phi_{i_1,i_2} &= f_{i_1,i_2}, \quad i_1 = 1,\ldots,n_1-1 \\ \phi_{0,i_2} &= g_0(i_2 h_y), \\ \phi_{n_1,i_2} &= g_1(i_2 h_y), \\ \phi_{i_1,i_2+n_2} &= \phi_{i_1,i_2}, \quad i_1 = 0,1,\ldots,n_1 \end{aligned} \qquad (6)$$

where $n_1 = 1/h_x$, $i_2 \in \mathbf{Z}$, $n_2 = 1/h_y$ and $f_{i_1,i_2} = F(i_1 h_x, i_2 h_y)$.

The one dimensional prototype for the 2D problem (3) is the simple one dimensional Dirichlet problem for the second derivative operator. This operator is discretized on the grid induced on the characteristic line, as follows

$$\frac{\Phi\left((i_1-1)h_x,(i_2-(k+s))h_y\right) - 2\Phi(i_1 h_x, i_2 h_y) + \Phi((i_1+1)h_x,(i_2+(k+s))h_y)}{h_x^2 + (k+s)^2 h_y^2}. \qquad (7)$$

It is well-known that one V-cycle with two red-black relaxations per level, full-weighting residual transfer and linear interpolation of the coarse-grid correction exactly solves the problem related to this one dimensional prototype. This cycle is chosen to be simulated in 2D.

The first differential approximation (cf. [7], [1]) to the operator (5) is

$$\phi^h_{\xi\xi} - h_y^2\left[A + \left(\frac{(1-s)s\cos(\psi)}{2m}\right)^2\right]\phi_{yyyy}, \qquad (8)$$

where $\phi^h_{\xi\xi}$ is the first differential approximation to the one-dimensional prototype (7). For characteristic components $\phi^h_{\xi\xi} \approx \phi_{\xi\xi}$. On the target grid $A = 0$. On coarser grids we choose A so that the *total cross-characteristic viscosity*, i.e., the coefficient of ϕ_{yyyy} in (8), would remain the same as on the target grid. (Note that upon each semi coarsening step the values of h_y and ψ remain unchanged while $s(1-s)/m$ decreases. Note also that the true cross-characteristic viscosity could be define as the coefficient of the fourth derivative with respect to η, but for characteristic components that is just proportional to the fourth derivative with respect to y.)

3.2 Multigrid cycles

In this section the basic parts of the multigrid cycles, such as relaxation, residual transfer and correction interpolation are described. Numerical two-level tests together with mode analysis which are discussed here allow us to construct efficient multilevel cycles, which are examined at the end of this section.

3.2.1 Relaxation schemes

Two types of relaxation are considered here: pointwise and "zebra".

The elementary step of the *pointwise relaxation* is to change the solution approximation at the point (i_1, i_2) so as to satisfy Eq. (6). The order of performance of the elementary steps obeys the following rules:

1) odd vertical lines (the vertical lines with odd i_1 coordinate) are relaxed before even ones.

2) the relaxation in each vertical line consists of four sweeps. Each sweep performs the elementary step for every fourth point in the line. The first sweep starts from the point with vertical coordinate $i_2 = 0$, the second — from the point with $i_2 = 2$, the third — from the point with $i_2 = 1$ and the last — from the point with $i_2 = 3$.

This order of relaxation is not necessary for efficient smoothing. It is chosen to enable full parallelization and precludes the appearance of relaxation "boundary layers". To be sure, the usual red-black relaxation order would be efficient as well, but then the results would slightly depend on *where* the sweeps start and end, which we wanted to avoid.

The elementary step of the *"zebra" line relaxation* is to solve (or solve approximately) the system of all the discrete equations centered at the same vertical grid line. The step results in simultaneous replacement of the solution approximation at all the grid points belonging to that line. All the residuals on this line are thereby reduced to zero (or near zero). The order of line relaxation remains the same as above: all the odd lines are relaxed before all the even ones (hence the name "zebra").

Inter-grid transfers

In any cycle there are two types of inter-grid communication. The fine-to-coarse transfer produces a coarse-grid approximation to the fine-grid residual function

$$r_{i_1,i_2} = f_{i_1,i_2} - L^{(h_x,h_y)}\phi_{i_1,i_2}.$$

The coarse-to-fine transfer is the interpolation of the coarse-grid correction. In the present algorithm both inter-grid transfers are anisotropic. They roughly simulate the corresponding transfers of the one-dimensional prototype solver.

Residual transfer to the semi-coarsened grid is given by

$$\begin{aligned} R_{i_1,i_2} = \left(I_h^H r\right)_{i_1,i_2} = \; & .5\, r_{2i_1,i_2} + .25\Big[(1-s)\Big(r_{2i_1-1,i_2-k} + r_{2i_1+1,i_2+k}\Big) \\ & + s\Big(r_{2i_1-1,i_2-k-1} + r_{2i_1+1,i_2+k+1}\Big)\Big], \end{aligned} \tag{9}$$

where R and r denote the coarse and fine grid residual functions respectively.

Notice that the weighted average $(1-s)r_{2i_1-1,i_2-k} + sr_{2i_1-1,i_2-k-1}$ defines the residual value at the ghost point r_{2i_1-1,i_2-k-s}, hence (9) corresponds to the standard one-dimensional full-weighting residual transfer.

The scheme is described in Fig. 2. The solid lines show where a fine-grid point residual is sent to. The dashed arrows exhibit all the fine-grid points sending their residuals to a given coarse-grid point.

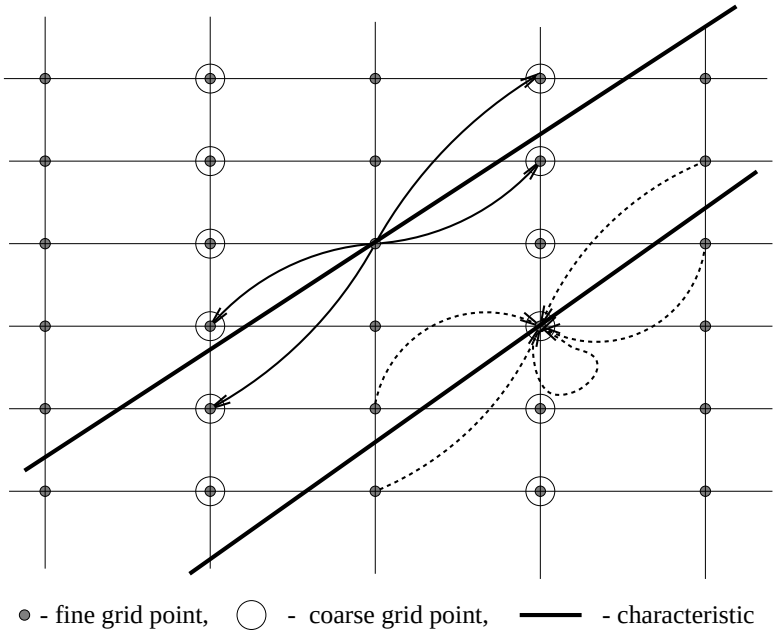

Figure 2: Residual transfer to semi-coarsened grid.

Interpolation of the coarse-grid correction simulates the one-dimensional linear interpolation, giving the operation adjoint to (9).

$$\left\{\begin{array}{lcl} v_{2i_1,i_2} & = & V_{i_1,i_2}, \\ v_{2i_1+1,i_2} & = & .5\Big[\big(1-S_1\big)\big(V_{i_1,i_2-K_1}+V_{i_1+1,i_2+K_1}\big) \\ & & +S_1\big(V_{i_1,i_2-K_1-1}+V_{i_1+1,i_2+K_1+1}\big)\Big], \end{array}\right. \tag{10}$$

where V denotes the solution of the coarse-grid problem, v denotes the correction to the fine-grid solution approximation and K_1 is an integer such that $(K+S)/2 = K_1 + S_1$, $0 \le S_1 < 1$, K and S being the parameters of the coarse-grid discretization (defined like k and s in Fig. 1).

3.2.2 Switching criterion

The condition of switching from pointwise to zebra relaxation can be derived from the coupling analysis of the FDA approximation (8). The term of the FDA responsible for the "characteristic" coupling is $\phi^h_{\xi\xi}$, and a quantitative measure of this coupling is

$$h_\xi^{-2} = \Big(\big(m^2+(k+s)^2\big)h_y^2\Big)^{-1}.$$

The "viscous" coupling is maintained by the second term in (8), in which ϕ_{yyyy} has the coupling strength h_y^{-4}. We switch to the zebra scheme when this "viscous" coupling becomes larger than the "characteristic" one, i.e., when the ratio between them, which we call the

Relative Coupling (RC), becomes larger than one. The derived criterion completely agrees with the experimental one obtained from comparing the asymptotic convergence rates of two-level cycles.

3.2.3 Two-level cycles

A two-level cycle $V_2(\nu_1, \nu_2)$ can be defined as the following six steps.

(i) *Pre-relaxation sweeps.* Improve the initial fine-grid approximation by ν_1 relaxation sweeps.

(ii) *Residual transfer.* Build the coarse-grid approximation R to the fine-grid residual function $r_{i_1,i_2} = f_{i_1,i_2} - L^{(h_x,h_y)}\phi_{i_1,i_2}$, namely, $R = I_h^H r$, where I_h^H is the fine-to-coarse residual transfer defined by (9).

(iii) *Coarse-grid equation.* Form the coarse-grid equation

$$L^{(H_x,H_y)}V = R.$$

At this stage we choose the precise dose of the coarse-grid explicit numerical viscosity A and recalculate the new values of the discretization parameters S and K depending on the coarse-grid aspect ratio.

(iv) *Exact solution.* Solve the coarse-grid equation by whatever method.

(v) *Coarse-grid correction.* Interpolate the obtained coarse-grid solution V to the fine grid (using (10)); the result v is used to correct the current fine-grid approximation.

(vi) *Post-relaxation sweeps.* Improve the corrected fine-grid approximation by ν_2 relaxation sweeps.

We have run two-level cycles with either pointwise or zebra relaxations on grids with different aspect ratios and with either full or semi coarsening. Using zero right hand sides and zero boundary conditions, the function $U(x, y) \equiv 0$ is the exact solution of the differential problem (3-4). This choice of data together with a random choice of the initial approximation facilitate the observation of the cycle asymptotic behavior.

$V(1,1)$ cycles have been performed on fine grids with aspect ratios $m = 1, 2, 4, 8, 16$. In these experiments we have chosen the explicit numerical viscosity factor A on the fine grid assuming that that grid itself has been obtained by ($\log_2 m$ steps of) semi coarsening, starting with a uniform target grid. In other words, the total viscosity TV of the algorithm is equal to the inherent numerical viscosity of a uniform grid with meshsize h_y.

Each experiment has included three different runs, each starting from a random initial error. Run I uses pointwise relaxation and semi coarsening; Run II — pointwise relaxation and full coarsening; Run III employs zebra relaxation and semi-coarsened coarse grid. Each run consists of at least 12 cycles, stopping further cycling when the convergence factor has been stabilized, in the sense that the the largest difference between the convergence factors of the last three cycles does not exceed 0.01.

The results of these experiments are collected in Table 1. The notation used in the table is the following: m is the aspect ratio; h_y is the vertical meshsize. The meshsize in the reference axis always remains the same $h_x = .03125$, except for the last group of experiments for $m = 16$, where $h_x = .125$; $t = \tan\psi$, where ψ is the slope, i.e., the angle between the reference axis and the characteristic line; RC is the relative coupling (cf. Sec. 3.2.2). The column "cycles" shows the number of cycles performed until the convergence factor has

been stabilized. In the column "final" the convergence factor of this last cycle (the L_2 error norm before the cycle divided by that norm after the cycle) is printed. The column "aver." exhibits the convergence factor averaged over *all* the cycles performed in the experiment.

One can see some very large *average* convergence factors in Table 1. This is due to very large factors (sometimes several thousands per cycle!) in the first few cycles. Our main concern is the asymptotic ("final") convergence, which always shows the worst-converging components. Concerning these asymptotic factors the first obvious result is the superiority of the present algorithm over the algorithm with full coarsening. The two semi-coarsening algorithms show similar asymptotic convergence factors on grids with small RC ($m = 1, 2, 4$); but when $RC > 1$, only the algorithm with zebra relaxation provides good asymptotic convergence.

3.2.4 Multilevel cycle

We have also performed experiments with a *multilevel* V(1,1) cycle, using the switching criterion introduced above. The multilevel cycle can be defined similarly to the two-level cycle, but the step (iv) is replaced with the recursive call for the same cycle applied to the coarse grid problem. The experiments showed a stable asymptotic convergence with nearly the same rate as in the two-level experiments. The results are collected in Table 2. In each experiment the cycle with conditional switching between the two types of relaxation is compared with the cycle employing only zebra relaxation on all the levels. The main conclusion is that the asymptotic convergence factors of these cycles are the same, while the average factors of the cycle with zebra relaxation are substantially better.

3.3 FMG solver and numerical experiments

The FMG algorithm (see, e.g., [3]) based on the $V(1,1)$ cycle proves to be quite a powerful solver for our model problem. Its *setup* work can be described by the following steps.

1. *Target-grid problem.* We formulate the discrete equation (5) on the chosen target grid. The parameter A for this grid is set to zero. The total viscosity value for the entire algorithm is defined as this target-grid inherent numerical viscosity. A proper discretization of the right-hand side f and boundary condition functions g_0 and g_1 is also performed. In our implementation these discrete functions are simply injected from the corresponding continuous ones.

2. *Next coarse-grid construction.* The next coarse grid is constructed by semi coarsening, as in the cycles described above.

3. *Coarse-grid problem.* The discretization parameters such as the aspect ratio, the new K, S parameters and the artificial viscosity coefficient A are calculated for the new grid. The general form of the coarse-grid operator remains the same. The coarse-grid right-hand side function F is formed by the same averaging procedure that is used for the residual transfer inside the cycles, i.e.,

$$F_{i_1,i_2} = .5\, f_{2i_1,i_2} + .25\Big[(1-s)\Big(f_{2i_1-1,i_2-k} + f_{2i_1+1,i_2+k}\Big) + s\Big(f_{2i_1-1,i_2-k-1} + f_{2i_1+1,i_2+k+1}\Big)\Big].$$

The coarse-grid boundary conditions are *injected* from the previous fine grid (averaging could as well be used).

Steps 2 and 3 are repeated until the coarsest possible grid is reached and its problem is defined.

The *execution* of the FMG algorithm then involves the following steps.

a) The coarsest-grid problem is solved by whatever method.

b) The solution obtained on the current grid is interpolated to the next fine grid to serve as an initial approximation to the fine-grid solution. The "FMG interpolation" used in this step is fourth order in the characteristic direction and second order in the vertical direction. (The experiments show that even with this lower order vertical interpolation the algorithm successfully reduces all the algebraic errors well below the level of the discretization errors. Nevertheless, in the 3D case below we do use the fourth order interpolation throughout).

c) The obtained initial approximation is improved by one V(1,1) cycle.

We repeat the steps *b*) and *c*) up to the target grid. There we perform one additional improving cycle (mainly for checking purposes).

In our experiments, the right-hand side f and the boundary conditions g_0 and g_1 have been chosen so that the pure component $\sin(\theta x + \omega y)$ be the exact solution of the differential problem (3)-(4). Six-level experiments have been performed for all kinds of such components. The target finest grid throughout our experiments has been a uniform grid with meshsizes $h_x = h_y = 2^{-7}$. For each component we check five different characteristic inclinations $t = \tan\psi$. The results are collected in Table 3, where the target-grid discretization error is compared with the algebraic errors after the FMG interpolation of the coarse-grid solution and at the end of the first and the second improving cycles. The results show that for nearly all components the algebraic error after the first cycle is much less than the discretization error (pathological exceptions are discussed below). In fact, in the case of characteristic components the algebraic error is less than the discretization error already after the solution interpolation from the coarse grid. This is due to the artificial viscosity introduced at the coarse level, ensuring nearly the same characteristic-component discretization error on all the grids (the differences are proportional to h_y^4). The situation is different for non-characteristic components since they cannot be well approximated on coarse grids. However, it is exactly these components that are remarkably reduced in the target-grid cycle. Thus we can conclude that the FMG algorithm requires only one $V(1,1)$ cycle per FMG level, or a total of about 13 "minimal work units" to reach the discretization accuracy for the target-grid approximation. (The work-unit count is about twice larger than usual in uniformly elliptic problem, due to the somewhat more expensive coarser levels, using semi coarsening instead of full coarsening. This increased expense will disappear in 3D.)

Finally we discuss the phenomenon of "pathological" non-characteristic components which exhibit unusually small discretization errors. Non-characteristic components usually possess relatively large discretization errors (compared with characteristic components). But a very special choice of parameters (solution component U and slope angle ψ) can result in vanishing discretization errors, as the analysis in the next section will show. It is clear that in such special situations we cannot expect the algebraic error to be smaller than or comparable to the discretization error at any stage of the algorithm. In the numerical experiments this quite delicate effect is visible only for precisely adjusted problem parameters, disappearing at small perturbations. A few examples are shown in Table 4. In these experiments we investigate six sets of cases. The first experiment in each set is performed for a special slope-component pair having (nearly) zero discretization errors. The two following experiments are for the

same component but with a rounded slope angle parameter. The last two experiments in each set are with the original slope but the component frequency in the characteristic direction (β_ξ) is slightly changed. It is evident from the table that in spite of the fact that the algorithm for the special pairs fails to reach the discretization accuracy after the first improving target-grid cycle, the *total* (algebraic plus discretization) error in these special cases is much smaller than in neighboring regular cases. These experiments also show that upon any reasonable perturbation the behavior becomes normal: the algebraic error after the first improving cycle is already substantially below the level of the discretization error. It is thus clear in any case that the statement that the algebraic error after one cycle is much less than the discretization error will most likely hold in any *real* calculations (where many non-pathological components and slope values are most likely present).

3.4 Two-level Half-space Mode Analysis

Since in non-elliptic problems high-frequency components far in the interior can still be strongly affected by the boundary, it is quite important for such problems to use half-space mode analysis instead of the traditional full-space analysis (see [1] and Sec. 7.5 in [3]). This analysis considers a half-space problem for the same equation, with boundary conditions being represented by one Fourier component at a time. To regularize the half-space problem for the second order differential operators the solution is not allowed to grow faster than a polynomial function. Then the problem is well-posed. In this way the original multidimensional problem is translated into a one-dimensional problem, where the frequencies of the boundary Fourier component are considered as parameters. The grid direction is perpendicular to the boundary. The subject of the analysis is to compare the exact solutions of the continuous and the discrete problems with the approximate solutions obtained at different stages of the solver.

The half-space mode analysis has usually been used to estimate the quality of characteristic component approximations in a two-level FMG algorithm (see [1], [5]). That analysis usually consider homogeneous equations. In our case a good approximation of the characteristic components is easily guaranteed by the choice of the coarse grid discretization: we have built the coarse grid operator so that it approximates the characteristic components with fourth order accuracy, whereas all other components are approximated only with second order. But there still exists another interesting problem to be investigated: right-hand side interaction with incoming frequencies. The right-hand side influence on a solution approximation can, usually, be described in the framework of full-space mode analysis. But studying right-hand-side-boundary interactions must combine both kinds of analyses.

Choosing the domain to be $\{(x,y) : x \geq 0\}$, for each Fourier frequency ω_2, the differential problem (3)-(4) can be reformulated in the following way: find a function Φ obeying the restriction to polynomial growth, such that

$$\frac{\partial^2 \Phi}{\partial \xi^2} = -\left(\beta_\xi^2\right) e^{i(\omega_1 x + \omega_2 y)}, \qquad \Phi(0,y) = e^{i\omega_2 y}, \tag{11}$$

where $\beta_\xi = \left(\omega_1 + t\omega_2\right)\left(1+t^2\right)^{-1/2}$ is the characteristic frequency. The exact solution of this problem is $\Phi(x,y) = e^{i(\omega_1 x + \omega_2 y)}$.

The discretization of (11) looks like

$$L^{(h_x,h_y)}\phi_{i_1,i_2} = -\Big(\beta_\xi^2\Big)e^{i(\Omega_1 i_1+\Omega_2 i_2)}, \qquad \phi_{0,i_2} = e^{i\Omega_2 i_2},$$

where $L^{(h_x,h_y)}$ is defined in (5), and $\Omega_1 = \omega_1 h_x = \omega_1 m h_y$ and $\Omega_2 = \omega_2 h_y$ are normalized frequencies.

We seek a solution of the discrete problem in the form

$$\phi_{i_1,i_2} = \phi_{i_1} e^{i\Omega_2 i_2}. \tag{12}$$

Then the problem reformulated for ϕ_{i_1} is

$$\frac{a_0\phi_{i_1-1} + a_1\phi_{i_1} + a_2\phi_{i_1+1}}{\Big(m^2+(k+s)^2\Big)h_y^2} = -\Big(\beta_\xi^2\Big)e^{i\Omega_1 i_1}, \qquad \phi_0 = 1. \tag{13}$$

$$a_0 = e^{-ik\Omega_2}\Big(1 + se^{-i\Omega_2} - s\Big), \tag{14}$$

$$a_1 = -2\Big(1 - s(1-s)\Big(1-\cos(\Omega_2)\Big) + a\Big(m^2+(k+s)^2\Big)\Big(\cos(2\Omega_2) - 4\cos(\Omega_2) + 3\Big)\Big), \tag{15}$$

$$a_2 = e^{ik\Omega_2}\Big(1 + se^{i\Omega_2} - s\Big), \tag{16}$$

where s, k and m are the discretization parameters and a is the *fine-grid* explicit numerical viscosity coefficient (reserving the former notation A for the coarse-grid value).

The solution $\phi(i_1)$ can be represented as

$$\phi_{i_1} = de^{i(\Omega_1 i_1)} + (1-d)X^{i_1},$$

where $d = -\Big(\beta_\xi^2\,(m^2+(k+s)^2)\,h_y^2\Big)\Big(a_0e^{-i\Omega_1} + a_1 + a_2e^{i\Omega_1}\Big)^{-1}$ and X is the root of the quadratic equation

$$a_0 + a_1x + a_2x^2 = 0$$

satisfying $|x| \le 1$ (so that ϕ_{i_1,i_2} indeed obeys the regularization requirement). The discretization error function is

$$\Phi(i_1h_x, i_2h_y) - \phi_{i_1,i_2} = \Big[e^{i(\Omega_1 i_1)} - \phi(i_1)\Big]\,e^{i\Omega_2 i_2} = \Big[e^{i(\Omega_1 i_1)} - X^{i_1}\Big]\,(1-d)e^{i\Omega_2 i_2}. \tag{17}$$

The amplitude of this function depends on two factors. The first is the difference between X^{i_1} and $e^{i\Omega_1 i_1}$. The closeness of X and $e^{i\Omega_1}$ means that the function $\Phi(i_1h_x, i_2h_y)$ is a smooth characteristic function and, therefore, the small discretization error in this case is quite natural. The second factor $(1-d)$ depends on several problem parameters (not only on frequencies) and may become zero for non-characteristic frequencies as well. Generally speaking, for any given component $e^{i(\omega_1 x+\omega_2 y)}$ we can find parameter values (k, s, a) for which $1-d$ vanishes. This explains the aforementioned phenomenon of vanishing discretization errors for some "pathological" non-characteristic components.

3.4.1 Two-level FMG mode analysis

We will analyze the solution approximation obtained in various stages of the two-level FMG algorithm described in the previous section. We assume that on the fine level the zebra line relaxation is used. The analysis of pointwise relaxation and conditional coarsening can be found in [6].

Let the problem (13) be stated on a fine grid.

Coarse-grid exact solution. The FMG coarse-grid problem derived from the fine-grid formulation, is the following:

$$\frac{A_0\Phi_{i_1-1} + A_1\Phi_{i_1} + A_2\Phi_{i_1+1}}{\Big(M^2 + (K+S)^2\Big)h_y^2} = -\left(\beta_\xi^2\right)\frac{1+W_2}{2}e^{i2\Omega_1 i_1}, \tag{18}$$

$$\Phi_0 = 1, \tag{19}$$

where

$$W_2 = (1-s)\cos\Big(\Omega_1 + k\Omega_2\Big) + s\cos\Big(\Omega_1 + (k+1)\Omega_2\Big),$$

$$A_0 = e^{-iK\Omega_2}\left(1 + Se^{-i\Omega_2} - S\right), \tag{20}$$

$$A_1 = -2\Big[1 - S(1-S)\Big(1-\cos(\Omega_2)\Big) + A\Big(M^2 + (K+S)^2\Big)\Big(\cos(2\Omega_2) - 4\cos(\Omega_2) + 3\Big)\Big], \tag{21}$$

$$A_2 = e^{iK\Omega_2}\left(1 + Se^{i\Omega_2} - S\right); \tag{22}$$

$M = 2m$, K is an integer such that $K + S = 2(k+s)$ and $0 \le S < 1$. The factor $(1+W_2)/2$ arises due to the (second order) averaging of the right-hand side function in constructing the coarse-grid equation.

The exact solution of the problem (18) in the regular case is

$$\Phi_{i_1} = D_0\left(w^2\right)^{i_1} + (1-D_0)Z^{i_1},$$

where

$$w = e^{i\Omega_1},\ Z = -\frac{A_1 + \sqrt{A_1^2 - 4A_0A_2}}{2A_2},\ \Big(|Z| \le 1\Big),$$

$$D_0 = -\beta_\xi^2\frac{1+W_2}{2}\frac{\Big(M^2 + (K+S)^2\Big)h_y^2}{A_0w^{-2} + A_1 + A_2w^2}.$$

In non-degenerate cases $A_0w^{-2} + A_1 + A_2w^2 \neq 0$. The first order degenerate case ($A_0w^{-2} + A_1 + A_2w^2 = 0$, but $-A_0w^{-2} + A_2w^2 \neq 0$) implies that $Z = w^2 = e^{i2\Omega_1}$ and

$$\Phi_{i_1} = (D_1 i_1 + 1)\,Z^{i_1},$$

$$D_1 = -\beta_\xi^2\frac{1+W_2}{2}\frac{\Big(M^2 + (K+S)^2\Big)h_y^2}{-A_0w^{-2} + A_2w^2}.$$

The higher order degeneration ($-A_0 w^{-2} + A_2 w^2 = 0$) can appear only in inviscid problems ($S = 0;\ A = 0$). In this case the discrete operator is the central three-point second-derivative discretization. Hence, $e^{i2\Omega_1} = e^{-iK\Omega_2} \Rightarrow K\Omega_2 + 2\Omega_1 = 2\pi n$, $n = 0, \pm 1, \pm 2, \ldots$ and the exact solution of (18) is

$$\Phi_{i_1} = \left(D_2 i_1^2 + 1\right) Z^{i_1},$$
$$D_2 = -\frac{1}{2}\beta_\xi^2 \frac{1+W_2}{2}\left(M^2 + (K+S)^2\right) h_y^2.$$

Note that in both degenerate cases the obtained solution is unbounded. This is a resonance type phenomenon where the characteristic direction plays the role of time.

The general form of the exact solution of Eq. (18) is

$$\Phi_{i_1,i_2} = \Phi_{i_1} e^{i\Omega_2 i_2}, \qquad \Phi_{i_1} = D\left(w^2\right)^{i_1} + P^1(i_1) Z^{i_1}.$$

In the non-degenerate case $D = D_0$ and $P^1(i_1) = 1 - D$ is a constant independent of i_1. In irregular cases $P^1(i_1)$ is a polynomial of either first or second order and $D = 0$.

FMG solution interpolation. The calculated coarse-grid solution is interpolated by cubic interpolation to the fine grid. The obtained fine-grid initial approximation is

$$\tilde{\phi}_{i_1} = D\left(\frac{1+W_4(w)}{2} w^{i_1} + \frac{1-W_4(w)}{2}(-w)^{i_1}\right)$$
$$+\left(\frac{P^1\left(\frac{i_1}{2}\right) + W_4(z,P^1)}{2} z^{i_1} + \frac{P^1\left(\frac{i_1}{2}\right) - W_4(z,P^1)}{2}(-z)^{i_1}\right),$$

where

$$W_4(w) = \tfrac{1}{16}\Bigg(-\frac{\hat{s}e^{-i(\hat{k}+1)\Omega_2} + (1-\hat{s})e^{-i\hat{k}\Omega_2}}{w^3} + 9\frac{se^{-i(k+1)\Omega_2} + (1-s)e^{-ik\Omega_2}}{w}$$
$$+9\left(se^{i(k+1)\Omega_2} + (1-s)e^{ik\Omega_2}\right) w - \left(\hat{s}e^{i(\hat{k}+1)\Omega_2} + (1-\hat{s})e^{i\hat{k}\Omega_2}\right) w^3\Bigg),$$

$$W_4(z,P^1) = \tfrac{1}{16}\Bigg(-P^1\left(\frac{i_1-3}{2}\right)\frac{\hat{s}e^{-i(\hat{k}+1)\Omega_2} + (1-\hat{s})e^{-i\hat{k}\Omega_2}}{z^3}$$
$$+9P^1\left(\frac{i_1-1}{2}\right)\frac{se^{-i(k+1)\Omega_2} + (1-s)e^{-ik\Omega_2}}{z}$$
$$+9P^1\left(\frac{i_1+1}{2}\right)\left(se^{i(k+1)\Omega_2} + (1-s)e^{ik\Omega_2}\right) z$$
$$-P^1\left(\frac{i_1+3}{2}\right)\left(\hat{s}e^{i(\hat{k}+1)\Omega_2} + (1-\hat{s})e^{i\hat{k}\Omega_2}\right) z^3\Bigg),$$

$z = \sqrt{Z}$, $\hat{k}$ is an integer, $\hat{k} + \hat{s} = 3(k+s)$, $0 \le \hat{s} < 1$. This representation is correct for all points except $i_1 = 1$. At that point

$$W_4(w) = \tfrac{1}{16}\Bigg(5\frac{se^{-i(k+1)\Omega_2} + (1-s)e^{-ik\Omega_2}}{w} + 15\left(se^{i(k+1)\Omega_2} + (1-s)e^{ik\Omega_2}\right) w$$

$$-5\left(\hat{s}e^{i(\hat{k}+1)\Omega_2}+(1-\hat{s})e^{i\hat{k}\Omega_2}\right)w^3+\left(\hat{\hat{s}}e^{i(\hat{\hat{k}}+1)\Omega_2}+(1-\hat{\hat{s}})e^{i\hat{\hat{k}}\Omega_2}\right)w^5\Bigg),$$

$$W_4(z,P^1)=\tfrac{1}{16}\Bigg(\quad 5P^1\left(\frac{i_1-1}{2}\right)\frac{se^{-i(k+1)\Omega_2}+(1-s)e^{-ik\Omega_2}}{z}$$
$$+15P^1\left(\frac{i_1+1}{2}\right)\left(se^{i(k+1)\Omega_2}+(1-s)e^{ik\Omega_2}\right)z$$
$$-5P^1\left(\frac{i_1+3}{2}\right)\left(\hat{s}e^{i(\hat{k}+1)\Omega_2}+(1-\hat{s})e^{i\hat{k}\Omega_2}\right)z^3$$
$$+P^1\left(\frac{i_1+5}{2}\right)\left(\hat{\hat{s}}e^{i(\hat{\hat{k}}+1)\Omega_2}+(1-\hat{\hat{s}})e^{i\hat{\hat{k}}\Omega_2}\right)z^5\Bigg),$$

where $\hat{\hat{k}}$ is an integer, $\hat{\hat{k}}+\hat{\hat{s}}=5(k+s)$, $0\le\hat{\hat{s}}<1$.

The general form of components appearing at this stage of the algorithm is $P(i_1)v^{i_1}$, where v is either $\pm w$ or $\pm z$; $P(i_1)$ is a polynomial of i_1 of the second order at most. We will henceforth consider the function $P(i_1)$ as a polynomial of an arbitrary order. Such a generalization of the FMG analysis makes it less dependent of the exact shape of the analyzed components. Moreover this representation of components $(P(i_1)v^{i_1})$ is invariant under all the transformations of the analysis and it allows us to analyze each building block of the algorithm, such as relaxation or coarse-grid correction, in separation from others. The underlying idea is to use computer capabilities already at the step of deriving analytic representation for the current solution approximation. We will actually analyze the response of each building block to an input component of the form

$$C(i_1)v^{i_1}, \tag{23}$$

where

$$C(i_1)=\begin{cases} P(i_1) & \text{if} \quad i_1>N \\ B(i_1) & \text{if} \quad i_1=0,...,N, \end{cases}$$

$P(i_1)$ being a given complex-coefficient polynomial and $B(i_1)$ being a complex number vector of length N.The domain $i_1>N$ will be referred to as the *analytic representation region*, while the segment $i_1=0,...,N$ is the *pointwise representation region*. In analyzing a two-level FMG algorithm N is a small integer. As a matter of fact $N=4$ at the final stage of our algorithm. Thus the pointwise representation region plays an essential role only in presenting fast decreasing functions ($|v|<.9$).

The output of any building block is formulated in the same form (23), except that each block usually produces several output components, differing in their bases v. In our algorithm these are the four interacting components with $v=w,-w,z,-z$.

Relaxation. Let $U_{i_1}=C(i_1)v^{i_1}$ be a particular component of the initial fine-grid solution approximation. Each elementary step of line relaxation is replacing U_{i_1} by $\bar{U}_{i_1}$ so as to satisfy the following equation

$$a_0U_{i_1-1}+a_1\bar{U}_{i_1}+a_2U_{i_1+1}=\delta, \tag{24}$$

where a_0, a_1 and a_2 are given in (14)-(16). If the initial component U_{i_1} is the one appearing in the right-hand side, i.e., $v = w$, then $\delta = \beta_\xi^2\Big(m^2 + (k+s)^2\Big)h^2$, otherwise $\delta = 0$. If the old solution U is given by (23), then (24) is satisfied by

$$\bar{U}_{i_1} = \mathcal{R}_C(i_1)v^{i_1},$$

where

$$\mathcal{R}_C(i_1) = \frac{\delta + C(i_1-1)e^{-ik\Omega_2}v^{-1}\Big(se^{-i\Omega_2} + (1-s)\Big) + C(i_1+1)e^{ik\Omega_2}v\Big(se^{i\Omega_2} + (1-s)\Big)}{a_1}.$$

The order of performing the elementary steps defines the type and properties of the relaxation scheme. The scheme we use in the analyzed algorithm is zebra relaxation (all the lines with i_1 odd are relaxed first; then all the even lines). A full such relaxation sweep produces an approximation that is the following collection of components:

$$R : U_{i_1} \rightarrow \left\{ \begin{array}{llll} b_1(i_1)v^{i_1} & \text{for} & i_1 & \text{odd} \\ b_2(i_1)v^{i_1} & \text{for} & i_1 & \text{even} \end{array} \right\} = C_1(i_1)v^{i_1} + C_2(i_1)(-v)^{i_1};$$

where $b_1(i_1) = \mathcal{R}_C(i_1), b_2(i_1) = \mathcal{R}_{\mathcal{R}_C}(i_1)$ and $C_1(i_1) = \Big(b_1(i_1) + b_2(i_1)\Big)/2, C_2(i_1) = \Big(b_1(i_1) - b_2(i_1)\Big)/2$. Note also that the new upper boundary N_{new} of the region of pointwise representation is moved by the relaxation sweep to the smallest even number exceeding the maximal N_{old} of the two interacting components (v and $-v$).

Coarse-grid correction. Let us consider the evolution of a pair of fine-grid components aliasing on the coarse grid. These are

$$U_1 = C_1(i_1)v^{i_1} \quad \text{and} \quad U_2 = C_2(i_1)(-v)^{i_1}.$$

Their residuals are calculated as

$$\mathcal{R}_1(i_1)v^{i_1} = \left[\Lambda_1 - \frac{C_1(i_1-1)\frac{a_0}{v} + C_1(i_1)a_1 + C_2(i_1+1)a_2 v}{\left(m^2 + (k+s)^2\right)h^2}\right]v^{i_1},$$

$$\mathcal{R}_2(i_1)(-v)^{i_1} = \left[\Lambda_2 - \frac{C_2(i_1-1)\frac{a_0}{(-v)} + C_2(i_1)a_1 + C_2(i_1+1)a_2(-v)}{\left(m^2 + (k+s)^2\right)h^2}\right](-v)^{i_1},$$

where $\Lambda_j = -\beta_\xi^2$ if U_j is the right-hand side component, otherwise $\Lambda_j = 0$.

The fine-to-coarse residual transfer produces the coarse-grid right-hand side component $\mathcal{R}(i_1)V^{i_1}$, where $V = v^2$ and

$$\mathcal{R}(i_1) = \frac{1}{4}\Big[\Big(\mathcal{R}_1(2i_1-1) - \mathcal{R}_2(2i_1-1)\Big)e^{-ik\Omega_2}\Big(se^{-i\Omega_2} + (1-s)\Big)v^{-1}$$
$$+2\Big(\mathcal{R}_1(2i_1) + \mathcal{R}_2(2i_1)\Big) + \Big(\mathcal{R}_1(2i_1+1) - \mathcal{R}_2(2i_1+1)\Big)e^{ik\Omega_2}\Big(se^{i\Omega_2} + (1-s)\Big)v\Big].$$

The upper bound of the region of pointwise representation for this coarse-grid residual function is $N_{\text{coarse}} = N_{\text{fine}}/2 + 1$, where $N_{\text{fine}} = \max\Big(N_1, N_2\Big)$ and N_1 and N_2 are the fine-grid

upper bounds of the input components. In fact, if the coarse-grid correction follows the zebra relaxation then $N_{\text{fine}} = N_1 = N_2$.

The coarse-grid correction itself can again be represented as a function $\mathcal{P}(i_1)V^{i_1}$, it satisfying

$$\begin{aligned} &\frac{A_0}{V}\mathcal{P}(i_1-1) + A_1\mathcal{P}(i_1) + A_2V\mathcal{P}(i_1+1) = \mathcal{R}(i_1)\left(M^2+(K+S)^2\right)H^2, \qquad (25)\\ &\mathcal{P}(0) = 0, \end{aligned}$$

where the coarse-grid coefficients A_0, A_1 and A_2 are given by (20) - (22). The function $\mathcal{P}(i_1)$ is a polynomial for $i_1 \geq N_{\text{coarse}}$. In the non-degenerate case $\left(A_0/V + A_1 + A_2V \neq 0\right)$ the polynomial $\mathcal{P}(i_1)$ has the same order as $\mathcal{R}(i_1)$ (in most cases they are constants independent of i_1). For polynomials of an arbitrary order

$$\begin{aligned} \text{if} \quad & \mathcal{R}(i_1) = r_n i_1^n + r_{n-1} i_1^{n-1} + ... + r_1 i_1 + r_0, \\ \text{then} \quad & \mathcal{P}(i_1) = p_n i_1^n + p_{n-1} i_1^{n-1} + ... + p_1 i_1 + p_0, \end{aligned}$$

where (using superscript T to denote vector transposition)

$$\begin{aligned} &(p_n, p_{n-1}, ..., p_1, p_0)^T = \left(M^2+(K+S)^2\right)H^2 L_n^{-1}(r_n, r_{n-1}, ..., r_1, r_0)^T, \qquad (26)\\ &L_n \equiv \left[\frac{A_0}{V}T_n^{-1} + A_1 I_n + A_2 V T_n\right], \end{aligned}$$

where I_n is the unit $n \times n$ matrix and T_n is the matrix corresponding to the shift $\left(\mathcal{P}(i_1) \rightarrow \mathcal{P}(i_1+1)\right)$.

$$T_n = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ \binom{n}{1} & 1 & 0 & 0 & \cdots & 0 \\ \binom{n}{2} & \binom{n-1}{1} & 1 & 0 & \cdots & 0 \\ \binom{n}{3} & \binom{n-1}{2} & \binom{n-2}{1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \binom{n}{n-1} & \binom{n-1}{n-2} & \binom{n-2}{n-3} & \binom{n-3}{n-4} & \cdots & 0 \\ 1 & 1 & 1 & 1 & \cdots & 1 \end{pmatrix}$$

$$T_n^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -\binom{n}{1} & 1 & 0 & 0 & \cdots & 0 \\ \binom{n}{2} & -\binom{n-1}{1} & 1 & 0 & \cdots & 0 \\ -\binom{n}{3} & \binom{n-1}{2} & -\binom{n-2}{1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{n-1}\binom{n}{n-1} & (-1)^n\binom{n-1}{n-2} & (-1)^{n-1}\binom{n-2}{n-3} & (-1)^n\binom{n-3}{n-4} & \cdots & 0 \\ (-1)^n & -(-1)^n & (-1)^n & -(-1)^n & \cdots & 1 \end{pmatrix}.$$

Expression (26) gives us a solution of Eq. (25) in the analytic representation region. For connecting this region with that of pointwise representation one must add to it a function μZ^{i_1} from the kernel of the operator L_n. As before, Z is the root of the equation $A_0 + A_1 x + A_2 x^2 = 0$. In order to define the values of μ and $\mathcal{P}(i_1)$ in the region of pointwise representation one has to solve the following system of linear equations.

$$\begin{cases} A_1\mathcal{P}(1)V + A_2\mathcal{P}(2)V^2 = \mathcal{R}(1)V\left(M^2 + (K+S)^2\right)H^2, \\ A_0\mathcal{P}(1)V + A_1\mathcal{P}(2)V^2 + A_2\mathcal{P}(3)V^3 = \mathcal{R}(2)V^2\left(M^2 + (K+S)^2\right)H^2, \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ A_0\mathcal{P}(N_{\text{coarse}}-2)V^{N_{\text{coarse}}-2} + A_1\mathcal{P}(N_{\text{coarse}}-1)V^{N_{\text{coarse}}-1} + A_2\left(\mathcal{P}(N_{\text{coarse}})V^{N_{\text{coarse}}} + \mu Z^{N_{\text{coarse}}}\right) \\ \quad = \mathcal{R}(N_{\text{coarse}-1})V^{N_{\text{coarse}}-1}\left(M^2 + (K+S)^2\right)H^2, \\ A_0\mathcal{P}(N_{\text{coarse}}-1)V^{N_{\text{coarse}}-1} + A_1\left(\mathcal{P}(N_{\text{coarse}})V^{N_{\text{coarse}}} + \mu Z^{N_{\text{coarse}}}\right) \\ \quad + A_2\left(\mathcal{P}(N_{\text{coarse}}+1)V^{N_{\text{coarse}}+1} + \mu Z^{N_{\text{coarse}}+1}\right) = \mathcal{R}(N_{\text{coarse}})V^{N_{\text{coarse}}}\left(M^2 + (K+S)^2\right)H^2, \end{cases} \quad (27)$$

where the values of $\mathcal{P}(N_{\text{coarse}})$ and $\mathcal{P}(N_{\text{coarse}}+1)$ are computed by means of the analytic representation of $\mathcal{P}(i_1)$ calculated above. Thus, the coarse-grid correction is $\mathcal{P}(i_1)V^{i_1} + \mu Z^{i_1}$.

In the first-order degenerate case $\left(A_0/V + A_1 + A_2 V = 0;\ Z = V\right)$ the order of $\mathcal{P}(i_1)$ is greater than that of $\mathcal{R}(i_1)$. If it is the first order degeneration then the polynomial order increase by one and $\bar{r} = (0, r_n, r_{n-1}, ..., r_1, r_0)'$, $\bar{p} = (p_{n+1}, p_n, p_{n-1}, ..., p_1, p_0)'$.

$$(p_{n+1}; p_n; p_{n-1}; ...; p_1; p_0)^T = \left(M^2 + (K+S)^2\right)H^2 L_{n+1}^{-1}(0, r_n, r_{n-1}, ..., r_1, r_0)^T,$$

$$L_{n+1} \equiv \left[\frac{A_0}{V}T_{n+1}^{-1} + A_1 I_{n+1} + A_2 V T_{n+1}\right]$$

Generally speaking the operator L_{n+1} is not invertible as operator in space $\mathbb{P}_{n+1}$ of $n+1$ order polynomials $\left(Det_{n+1}(L) = 0\right)$ but it is invertible as $L : \mathbb{P}_{n+1}\Big|_{p_0=0} \to \mathbb{P}_n$, where $\mathbb{P}_{n+1}\Big|_{p_0=0}$ is the factor subspace of $\mathbb{P}_{n+1}$ with respect to constants. In fact L_{n+1} is no longer represented as a square matrix and therefore the record L_{n+1}^{-1} makes a sense just as a symbol to denote the inverse operator, but not an inverse matrix. Square matrix representation could be suggested for the operator L restricted on the factor subspace. It is

$$\mathcal{L}_{n+1} \equiv \left[\frac{A_0}{V}\mathcal{T}_{n+1}^{-1} + A_1\mathcal{I}_{n+1} + A_2 V\mathcal{T}_{n+1}\right],$$

$$(p_{n+1}, p_n, ..., p_2, p_1)^T = \left(M^2 + (K+S)^2\right) H^2 \mathcal{L}_{n+1}^{-1} (r_n, r_{n-1}, ..., r_1, r_0)^T,$$

where $\mathcal{T}_{n+1}$ is the $(n+1) \times (n+1)$
(like T_n) matrix

$$\mathcal{T}_{n+1} = \begin{pmatrix} \binom{n+1}{1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ \binom{n+1}{2} & \binom{n}{1} & 1 & 0 & \cdots & 0 & 0 \\ \binom{n+1}{3} & \binom{n}{2} & \binom{n-1}{1} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \binom{n+1}{n} & \binom{n}{n-1} & \binom{n-1}{n-2} & \binom{n-2}{n-3} & \cdots & \binom{2}{1} & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix}$$

$$\mathcal{I}_{n+1} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix},$$

$$\mathcal{T}_{n+1}^{-1} = \begin{pmatrix} -\binom{n+1}{1} & 1 & 0 & \cdots & 0 & 0 \\ \binom{n+1}{2} & -\binom{n}{1} & 1 & \cdots & 0 & 0 \\ -\binom{n+1}{3} & \binom{n}{2} & -\binom{n-1}{1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^n \binom{n+1}{n} & -(-1)^n \binom{n}{n-1} & (-1)^n \binom{n-1}{n-2} & \cdots & -\binom{2}{1} & 1 \\ (-1)^{n+1} & (-1)^n & (-1)^{n+1} & \cdots & 1 & -1 \end{pmatrix}.$$

In fact all these matrices are simply restricted matrices T_{n+1}, I_{n+1}, and T_{n+1}^{-1} where the first row and the last column are absent. Constant p_0 remains free and for definiteness we suppose $p_0 = 0$. In case of the second order degradation the polynomial order of $\mathcal{P}(i_1)$ is $n+2$. We can repeat now the same calculations, but now two values p_0 and p_1 are left free.

Addition of some function from the kernel L_n is still necessary to connect the regions of analytical and pointwise representations. Since $Z = V$ this addition adjusts p_0 and p_1. The system (28) must be solved to compute the precise values $\mathcal{P}(i_1)$, $i_1 < N_{\text{coarse}}$.

The corrected form of the fine-grid components U_1 and U_2 looks as

$$\tilde{U}_1(i_1) = \left[C_1(i_1) + \frac{1}{2}\left(\mathcal{P}\left(\frac{i_1}{2}\right) + \frac{1}{2}\left(\mathcal{P}\left(\frac{i_1-1}{2}\right)\left(se^{-i(k+1)\Omega_2} + (1-s)e^{-ik\Omega_2}\right)v^{-1}\right.\right.\right.$$
$$\left.\left.\left. + \mathcal{P}\left(\frac{i_1+1}{2}\right)\left(se^{i(k+1)\Omega_2} + (1-s)e^{ik\Omega_2}\right)v\right)\right)\right]v^{i_1},$$

$$\tilde{U}_2(i_1) = \left[C_2(i_1) + \frac{1}{2}\left(\mathcal{P}\left(\frac{i_1}{2}\right) - \frac{1}{2}\left(\mathcal{P}\left(\frac{i_1-1}{2}\right)\left(se^{-i(k+1)\Omega_2} + (1-s)e^{-ik\Omega_2}\right)v^{-1}\right.\right.\right.$$
$$\left.\left.\left. + \mathcal{P}\left(\frac{i_1+1}{2}\right)\left(se^{i(k+1)\Omega_2} + (1-s)e^{ik\Omega_2}\right)v\right)\right)\right](-v)^{i_1},$$

Because of linearity the precise dose of correction distribution between the coupled fine-grid components at the region of pointwise representation does not play any role. In the formulas above we divide the correction in the equalled proportions. Note also that after this coarse-grid correction $N_{\text{fine}}^{new} = N_{\text{fine}}^{old} + 1$.

Besides correcting the components U_1 and U_2 the coarse-grid correction contributes also into components with bases $(\pm z)^{i_1}$.

At any stage of the analysis the current solution approximation can be explicitly written out as a function (depending on a set of parameters) of the distance from the boundary. Thus we can always compare this approximation with the known exact solution (12) to get the algebraic error and then, in turn, compare the algebraic error with the discretization error (17) to check the quality of the obtained approximation. Also we can compare these mode-analysis calculations with actual two-level numerical experiments, thereby very precisely debugging both the computer-aided analysis and the solver program.

3.4.2 Computed results

All the parts of the analysis discussed above have been programmed for quite general range of problems. The main goal of the resulting analyzer is to detect and explain various features arising in FMG algorithms. As an example of its use we show the analysis of the two-level FMG solver for the problem (13), for different initial data and on different grids.

The results of the analysis are illustrated in Figs. 3 – 8: the graphs of the discretization error and algebraic errors at different stages of the algorithm are presented. The horizontal axis shows the distance (in meshsizes) from the boundary; the vertical axis depicts absolute error values. $\Omega_2 = \omega_2 h$ is the normalized incoming frequency; $\zeta_\xi = \beta_\xi\sqrt{m^2 + (k+s)^2}h$ is the normalized characteristic frequency. In all the figures the same notation for error graphs is used: the solid line denotes the discretization error; the dashed line is the algebraic error after the FMG solution interpolation; the circles denote the algebraic error after the pre-relaxation sweep; the dash-dotted line is the algebraic error after the coarse-grid correction; the stars show the final algebraic error obtained after the post-relaxation sweep.

All the cases exhibited here, and all other multitude of cases we have run, confirm the top efficiency of the analyzed algorithm: after the first correction cycle the algebraic errors are far smaller than the discretization errors.

Applications of this analysis for obtaining precise rigorous estimates for different quantitative aspects of an algorithm (analogous to the rigorous quantitative analysis in [4]) may be

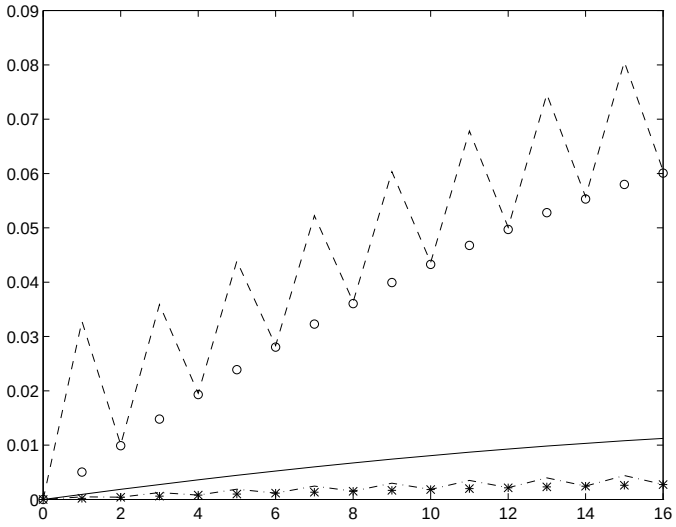

Figure 3: $\Omega_2 = .5$; $\zeta_\xi = 0.1$; $m = 1, k = 0, s = .3$; Smooth characteristic component.

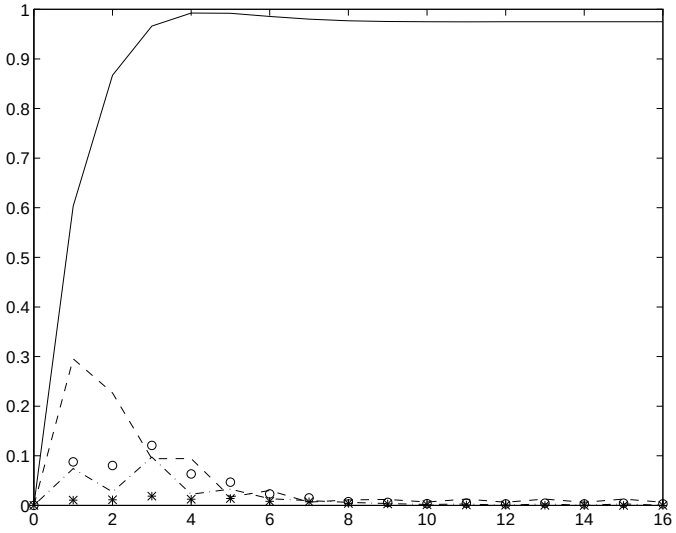

Figure 4: $\Omega_2 = 2.5$; $\zeta_\xi = 0.1$; $m = 1, k = 0, s = .7$; Non-smooth characteristic component.

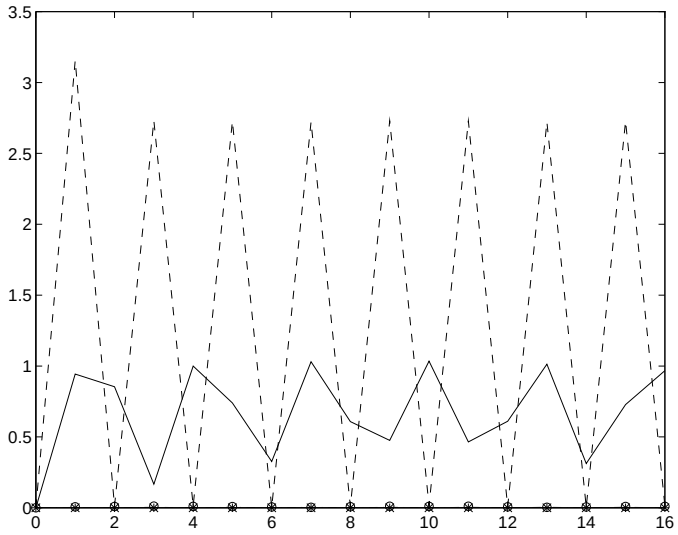

Figure 5: $\Omega_2 = .2$; $\zeta_\xi = 2.2$; $m = 1, k = 0, s = .6$; Smooth non-characteristic component.

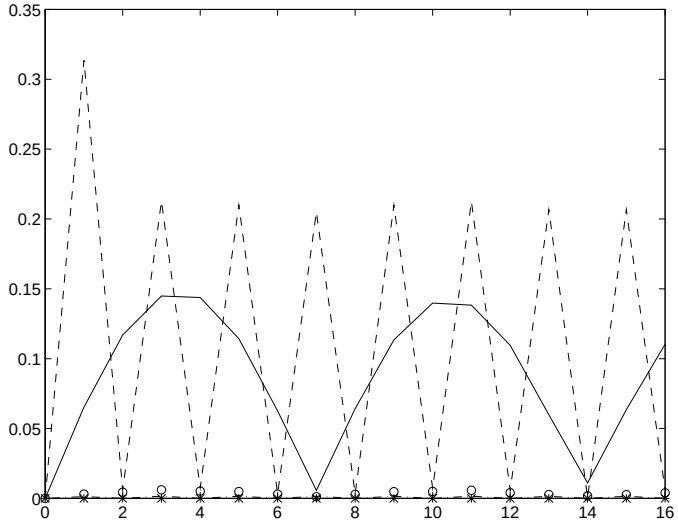

Figure 6: $\Omega_2 = .2$; $\zeta_\xi = .9$; $m = 4, k = 3, s = .4$; Coarse-grid smooth intermediate component.

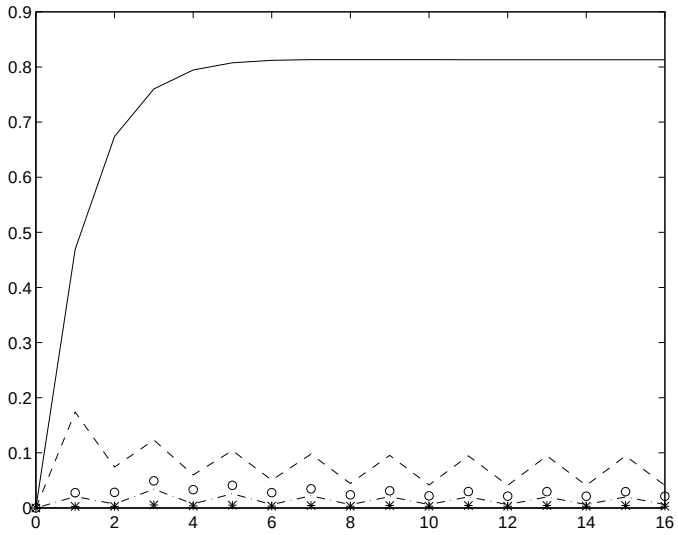

Figure 7: $\Omega_2 = 1.8$; $\zeta_\xi = .3$; $m = 4, k = 2, s = .4$; Coarse-grid non-smooth characteristic component.

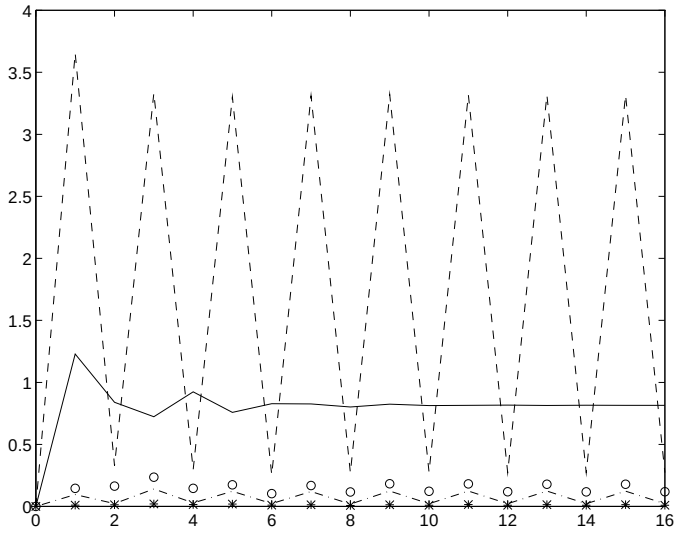

Figure 8: $\Omega_2 = 1.8$; $\zeta_\xi = 2.3$; $m = 4, k = 3, s = .6$; Coarse-grid non-smooth non-characteristic component.

possible, but would necessarily be very cumbersome. This analysis can however serve as an excellent tool for explaining delicate features recognized in numerical experiments. In particular the pathological effect of discretization errors vanishing for certain non-characteristic has been explained by means of such an analysis.

4 3D problem: discretization and solver

4.1 Problem statement

Let μ and β be arbitrary orthonormal coordinates in $\mathbb{R}^3$ and the scalar function $\Phi(x,y,z)$ be defined in the unit cube $(x,y,z) \in [0,1] \times [0,1] \times [0,1]$. Then the 3D differential equation we consider is equivalent to

$$L\Phi \equiv \frac{\partial^2 \Phi}{\partial \mu^2} + \frac{\partial^2 \Phi}{\partial \beta^2} = f. \tag{28}$$

The "characteristic plane" defined by μ and β is assumed *horizontally-inclined*; i.e., its normal is closer to the vertical than to any of the horizontal axes. For convenience we assume vertical periodicity: $\Phi(x,y,z) = \Phi(x,y,z+1)$. On the vertical faces of the cube we assume Dirichlet boundary condition: $\Phi(x,y,z)$ is prescribed for $x = 0,1$ and $y = 0,1$.

Natural (but non-orthogonal) coordinates in the characteristic plane are determined by the intersections of this plane with the x-z and y-z planes; they are

$$\xi = \frac{x + t_x z}{\sqrt{1+t_x^2}} \text{ and } \eta = \frac{y + t_y z}{\sqrt{1+t_y^2}},$$

where $t_x = \tan(\psi_x)$ is the tangent of the angle between the x axis and the intersection of the characteristic plane with the x-z coordinate plane; $t_y = \tan(\psi_y)$ is the same for the y-z coordinate plane. The horizontal-inclination assumption means that $0 \le |t_x|, |t_y| \le 1$. For simplicity we can assume that $0 \le t_x, t_y \le 1$. Let $t_\alpha = t_x - t_y$. We will use the auxiliary characteristic variable

$$\alpha = \frac{x - y + t_\alpha z}{\sqrt{2+t_\alpha^2}} = \left(\frac{1+t_x^2}{2+t_\alpha^2}\right)^{\frac{1}{2}} \xi - \left(\frac{1+t_y^2}{2+t_\alpha^2}\right)^{\frac{1}{2}} \eta.$$

Eq. (28) can be written as

$$C_\xi \frac{\partial^2 \Phi}{\partial \xi^2} + C_\eta \frac{\partial^2 \Phi}{\partial \eta^2} + C_\alpha \frac{\partial^2 \Phi}{\partial \alpha^2} = f, \tag{29}$$

where

$$C_\xi = \frac{(1+t_x^2)\left[(1-t_y t_\alpha)^2 + (1-t_y t_\alpha)(1+t_x t_\alpha)\right]}{2 + 2t_\alpha^2 + t_\alpha^2\left(t_x^2+t_y^2\right) + (t_x+t_y)^2};$$

$$C_\eta = \frac{\left(1+t_y^2\right)\left[(1+t_x t_\alpha)^2 + (1-t_y t_\alpha)(1+t_x t_\alpha)\right]}{2 + 2t_\alpha^2 + t_\alpha^2\left(t_x^2+t_y^2\right) + (t_x+t_y)^2};$$

$$C_\alpha = 1 - \frac{(1-t_y t_\alpha)(1+t_x t_\alpha)(2+t_\alpha^2)}{2 + 2t_\alpha^2 + t_\alpha^2\left(t_x^2+t_y^2\right) + (t_x+t_y)^2}.$$

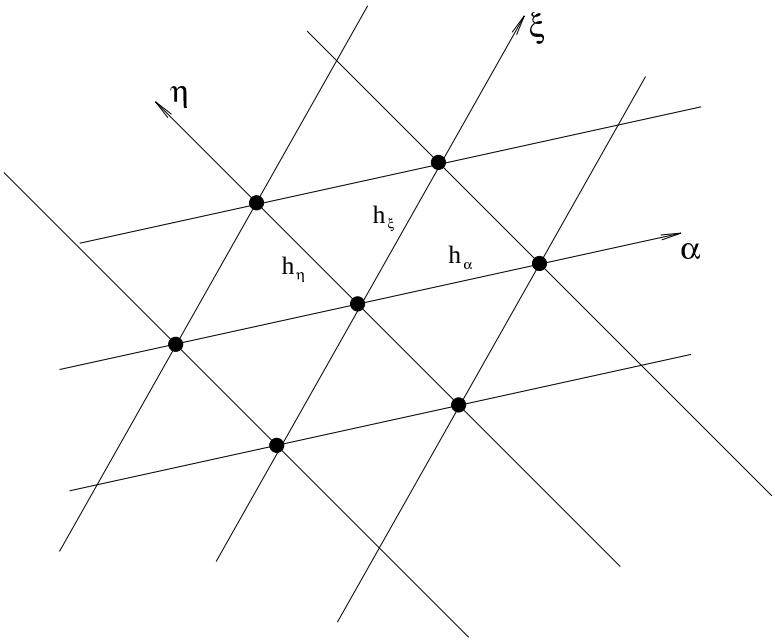

Figure 9: Non-orthogonal grid; seven point stencil for Laplace equation.

Following the guiding principle formulated in Sec. 2.2, we first state a 2D discrete problem on an auxiliary grid induced on a characteristic plane and study its multigrid solver.

4.2 2D-Prototype: Laplacian on non-orthogonal grid

Let the discrete function ϕ_{i_1,i_2} defined at the nodes of the grid induced on a representative characteristic plane be a discrete approximation to the function $\Phi(i_1 h, i_2 h, \tilde{z})$, where h is the meshsize in the reference x-y plane and the value of $\tilde{z}$ is uniquely calculated from the condition that the point belongs to the characteristic plane. The seven points used for discretizing Eq. (29) in this plane are shown in Fig. 9. The discrete approximation to Eq. (29) on these points is

$$L^h \phi_{i_1,i_2} \equiv C_\xi \frac{\phi_{i_1+1,i_2} - 2\phi_{i_1,i_2} + \phi_{i_1-1,i_2}}{h_\xi^2} + C_\eta \frac{\phi_{i_1,i_2+1} - 2\phi_{i_1,i_2} + \phi_{i_1,i_2-1}}{h_\eta^2} \qquad (30)$$
$$+ C_\alpha \frac{\phi_{i_1+1,i_2-1} - 2\phi_{i_1,i_2} + \phi_{i_1-1,i_2+1}}{h_\alpha^2} = f_{i_1,i_2}.$$

The characteristic-plane horizontal-inclination assumption implies that the angle between the axes ξ and η ranges within the relatively narrow interval $[\pi/3, \pi/2]$. In particular, when the angle is $\pi/2$ the grid becomes orthogonal and the variable α disappears ($C_\alpha = 0$). The meshsizes used in (30) are $h_\xi = \sqrt{1 + t_x^2}h$; $h_\eta = \sqrt{1 + t_y^2}h$; $h_\alpha = \sqrt{2 + (t_x - t_y)^2}h$. Thus, the discretization (30) is h-elliptic and suffers no substantial anisotropy. The boundary condition for the problem is straightforwardly discretized, since the boundary strictly aligns with the grid.

The multigrid cycle $V(1,1)$ which we have tested for the discrete prototype equation consists of a pointwise 4-color Gauss-Seidel relaxation, full-weighting residual transfer and linear correction interpolation. The position of a point is defined by its ξ and η coordinates: $\xi = i_1 h_\xi$, $\eta = i_2 h_\eta$. The relaxation steps take place in the following order: the points with both i_1 and i_2 odd are relaxed first; then the points with both coordinates even; then those with i_1 odd and i_2 even; and the remaining group of points is relaxed last.

Only points with both coordinates even are present on the coarse grid. The full-weighting residual transfer and linear correction interpolation used in the cycle mimic those usually used with the five-point Laplacian solver on an orthogonal uniform grid. For example, a fine-grid point with coordinates $\big(2i_1+1, 2i_2+1\big)$ sends its residual to (and gets its correction from) the coarse-grid points $\big(i_1, i_2\big), \big(i_1+1, i_2\big), \big(i_1, i_2+1\big)$ and $\big(i_1+1, i_2+1\big)$, in equal proportions. Due to the near isotropy we need not use here line relaxation and/or semi coarsening.

This cycle for the prototype equation, with pointwise relaxation and full-coarsening, proved very efficient, always reducing the error more than an order of magnitude. Its various parts will therefore be used in constructing our 3D solver.

4.3 Discretization

Let the 3D grid have meshsizes h_x, h_y and h_z in the corresponding directions. Assuming h_z to be the smallest of those, we define the aspect ratios of the grid $m_x = h_x/h_z$ and $m_y = h_y/h_z$. Using the fact that the multigrid solver for the 2D prototype admits a system of grids with full coarsening and also assuming that the target grid is always uniform ($m_x = m_y = 1$) we can restrict our considerations to 3D grids with equal aspect ratios ($h_x = h_y = h$ hence $m_x = m_y = m$). To define the discrete operator at a given grid node, we consider the *ghost points* located at the intersections of the characteristic plane (going through the given node) with the adjacent vertical grid lines. The function values at the ghost points are defined by interpolation from the vertically-nearest genuine grid neighbors. This, and the addition of the explicit viscosity term, turns (30) into the following discretization:

$$
\begin{aligned}
L^{(h,h,h_z)}\phi_{i_1,i_2,i_3} \equiv C_\xi \tfrac{1}{m^2(1+t_x^2)h_z^2}\Big[& s_x\left(\phi_{i_1+1,i_2,i_3+(k_x+1)} + \phi_{i_1-1,i_2,i_3-(k_x+1)}\right) - 2\phi_{i_1,i_2,i_3} \\
& +(1-s_x)\Big(\phi_{i_1+1,i_2,i_3+k_x} + \phi_{i_1-1,i_2,i_3-k_x}\Big) - s_x(1-s_x)\Big(\phi_{i_1,i_2,i_3+1} - 2\phi_{i_1,i_2,i_3} + \phi_{i_1,i_2,i_3-1}\Big)\Big] \\
& +C_\eta \tfrac{1}{m^2(1+t_y^2)h_z^2}\Big[s_y\left(\phi_{i_1,i_2+1,i_3+(k_y+1)} + \phi_{i_1,i_2-1,i_3-(k_y+1)}\right) - 2\phi_{i_1,i_2,i_3} \\
& +(1-s_y)\left(\phi_{i_1,i_2+1,i_3+k_y} + \phi_{i_1,i_2-1,i_3-k_y}\right) - s_y(1-s_y)\left(\phi_{i_1,i_2,i_3+1} - 2\phi_{i_1,i_2,i_3} + \phi_{i_1,i_2,i_3-1}\right)\Big] \\
& +C_\alpha \tfrac{1}{m^2(2+t_\alpha^2)h_z^2}\Big[s_\alpha\left(\phi_{i_1+1,i_2-1,i_3+(k_\alpha+1)} + \phi_{i_1-1,i_2+1,i_3-(k_\alpha+1)}\right) - 2\phi_{i_1,i_2,i_3} \\
& +(1-s_\alpha)\Big(\phi_{i_1+1,i_2-1,i_3+k_\alpha} + \phi_{i_1-1,i_2+1,i_3-k_\alpha}\Big) - s_\alpha(1-s_\alpha)\Big(\phi_{i_1,i_2,i_3+1} - 2\phi_{i_1,i_2,i_3} + \phi_{i_1,i_2,i_3-1}\Big)\Big] \\
& -A\tfrac{1}{h_z^2}\Big[\phi_{i_1,i_2,i_3+2} - 4\phi_{i_1,i_2,i_3+1} + 6\phi_{i_1,i_2,i_3} - 4\phi_{i_1,i_2,i_3-1} + \phi_{i_1,i_2,i_3-2}\Big] = f_{i_1,i_2,i_3},
\end{aligned}
\tag{31}
$$

where the integers k_x, k_y, and k_α and the real numbers $0 \le s_x, s_y, s_\alpha < 1$ are defined by

$$mt_x = k_x + s_x,\ mt_y = k_y + s_y \text{ and } mt_\alpha = k_\alpha + s_\alpha,$$

and A is the "explicit numerical viscosity" coefficient.

This is a 17 point $h-$elliptic discretization, whose first differential approximation is

$$\tilde{\triangle}^h \phi - h_z^2 \left[A + C_\xi \frac{(1-s_x)^2 s_x^2}{4m^2\left(1+t_x^2\right)} + C_\eta \frac{(1-s_y)^2 s_y^2}{4m^2\left(1+t_y^2\right)} + C_\alpha \frac{(1-s_\alpha)^2 s_\alpha^2}{4m^2\left(1+t_\alpha^2\right)} \right] \phi_{zzzz}, \tag{32}$$

where $\tilde{\triangle}^h \phi$ is the first differential approximation to the 2D prototype discretization (30) and ϕ_{zzzz} is the fourth derivative with respect to z. The explicit numerical viscosity parameter A is chosen so as to ensure the same total numerical viscosity TV on all the grids, where TV is defined to be the coefficient of ϕ_{zzzz} in (32). The value of TV is determined by its value at the target grid, where we set $A = 0$.

4.4 Multigrid cycle

The cycle used here is again the $V(1,1)$ cycle defined in Secs. 3.2.3. (Details are omitted, since they are similar to the 2D case, and more cumbersome; they can be found in [6].)

Coarse grids used in the cycle are always obtained by *semi coarsening*, i.e. the meshsizes in the reference plane are doubled at each coarsening step, while the $z-$direction meshsize remains the same throughout the cycle. (Another approach, involving conditional coarsening, is discussed in [6].)

Relaxation. Similar to the 2D case discussed in Sec. 3.2.2 we can define "viscous" and "characteristic" couplings; the relative coupling RC, defined as their ratio, turns out to be

$$\mathrm{RC} = -\left(\frac{m}{h_z}\right)^2 \frac{TV}{\frac{C_\xi}{1+t_x^2} + \frac{C_\eta}{1+t_y^2} + \frac{C_\alpha}{1+t_\alpha^2}}. \tag{33}$$

A pure pointwise relaxation is unconditionally efficient only on grids where RC is not large. Hence, the global relaxation policy remains the same as in the 2D case: as long as $RC \le 1$ we apply a point-by-point Gauss-Seidel relaxation; on grids with $RC > 1$ we use a relaxation which simultaneously updates all the points placed on the same vertical grid line. In either case the vertical lines are taken in the red-black ordering. In the pointwise relaxation each such vertical line is relaxed by the four-color ordering described in Sec 3.2. One could of course simplify the algorithm and use the vertical line relaxation throughout.

Fine-to-coarse residual transfer is of the "full weighting" type (see Sec. 4.4 in [1] or [3]); i.e., each fine grid residual (divided by 4) is distributed to neighboring coarse grid points. (The division by 4 expresses the coarse-grid-to-fine-grid mesh volume ratio.) This distribution follows two rules:

1. A fine-grid point which is geometrically present on the next coarse grid sends its whole residual (/4) to its coarse-grid representative.

2. A fine-grid point located on a vertical grid line absent from the coarse grid sends its residual (/4) to ghost points (points on its characteristic planes) placed on neighboring vertical gridlines which do belong to the coarse grid. The transfer is the same as in the solver for the two-dimensional prototype. The values received at the ghost points are then redistributed: each ghost point sends its value to the two vertically nearest coarse grid neighbors. The ratio between the fractions received by each of the neighbors is inversely proportional to the ratio of their distances from the ghost point.

As a result each coarse-grid point receives residual fractions from 17 fine-grid points.

The coarse-grid correction interpolation is a linear interpolation in directions as close to characteristic as possible. The operator we actually used is exactly the adjoint of the residual (/4) transfer just described: to each fine grid point interpolation is made from those coarse grid points to which it distributed its residual (/4), using the same weights.

Numerical experiments with two- and five-level $V(1,1)$ cycles have shown that the asymptotic convergence factor does not really depend on the cycle depth; it is better than 3 per cycle even in the worst cases. Since more important is the behavior of the FMG algorithm, we do not present detailed tables of cycle convergence factors.

4.5 FMG solver: numerical results

The full algorithm for solving Eq. (31) with the boundary conditions mentioned above is the FMG algorithm defined in Sec. 3.3. The total cost of the algorithm in 3D is about six minimal work units.

We have performed the numerical experiments with a five-level FMG algorithm, having a uniform target grid with meshsize $h = .03125$. The continuous problem's right-hand side and boundary conditions are chosen so that the solution is $U(x,y,z) = \sin(\theta_x x + \theta_y y + \theta_z z)$. Let ξ and η be the characteristic plane directions defined in Sec 4.1 and let ζ be the coordinate perpendicular to the characteristic plane. Then the same function U can be expressed in the new variables as $U = \sin(\omega_\xi \xi + \omega_\eta \eta + \omega_\zeta \zeta)$. Remember that the ξ and η axes can be non-orthogonal. A component is considered as being "characteristic" if $h_\xi \omega_\xi$ and $h_\eta \omega_\eta$ are small. We performed our experiments for several representative components and various characteristic-plane slopes. The numerical results are collected in Table 5.

The target-grid discretization error is compared in the table with the algebraic error of the target-grid approximations obtained at different stages of the FMG algorithm. The results confirm the top efficiency of the algorithm in the sense that after just one target grid cycle the algebraic error is already *much* less than the discretization error (except for possible pathological cases; see discussion at the end of Sec. 3.3).

References

[1] A. Brandt. Multigrid solvers for non-elliptic and singular-perturbation steady-state problems. (unpublished). The Weizmann Institute of Science, Rehovot, Israel, December 1981.

[2] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods*, Lecture Notes in Math. 960. Springer-Verlag, Germany, 1982.

[3] A. Brandt. Multigrid techniques: 1984 guide with applications to fluid dynamics. Monograph (unpublished). GMD-Studie 85, GMD-FIT, Postfach 1240, D-5205, St. Augustin 1, Germany. Also available from Secretary, Department of Mathematics, University of Colorado at Denver, Colorado 80204-5300, 1985.

[4] A. Brandt. Rigorous quantitative analysis of multigrid, I: constant coefficients two-level cycle with L2-norm. *SIAM J. Num. Anal.*, 31:1695–1730, 1994.

[5] A. Brandt and I. Yavneh. On multigrid solution of high-reynolds incompressible entering flow. *J. Comput. Phys.*, 101:151–164, 1992.

[6] B. Diskin. Multigrid solver for potential-flow equation. Intermediate report on research leading to Ph.D.. degree, The Weizmann Institute of Science, November 1995.

[7] N. Yanenko and Y.I.Shokin. Correctness of first differential approximations of difference schemes. *Dokl. Akad. Nauk SSSR*, 182, 1968. (in Russian).

[8] I. Yavneh. A method for devising efficient multigrid smoothers for complicated PDE systems. *SIAM J. Sci. Comp.*, 14:1437–1463, 1993.

A Tables of Numerical Results

Table 1. Multigrid solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

2 level algorithm

				asymptotic convergence factor								
				Run I			Run II			Run III		
h_y	m	t	RC	cycles	final	aver.	cycles	final	aver.	cycles	final	aver.
0.03125	1	0.100	0.002025	13	4.14	22.5	15	1.10	22.1	15	4.25	56.2
0.03125	1	0.200	0.00640	15	3.06	7.96	19	1.20	8.62	15	3.37	19.9
0.03125	1	0.300	0.011025	25	3.49	5.34	20	1.31	9.2	26	3.59	13.2
0.03125	1	0.400	0.0144	21	4.00	6.15	15	1.37	11.6	20	4.15	16.1
0.03125	1	0.500	0.015625	26	5.01	6.65	22	1.43	10.5	29	4.99	15.0
0.03125	1	0.600	0.0144	20	4.00	6.47	14	1.40	12.4	17	4.05	18.1
0.03125	1	0.700	0.011025	18	3.49	5.89	15	1.32	9.96	20	3.58	14.4
0.03125	1	0.800	0.0064	15	3.12	7.66	15	1.18	10.4	16	3.36	18.8
0.03125	1	0.900	0.002025	18	4.19	15.9	16	1.10	20.3	16	4.23	50.1
0.01562	2	0.100	0.0081	18	3.40	7.58	15	1.23	11.8	19	3.73	17.7
0.01562	2	0.200	0.0256	25	4.88	7.14	15	1.57	15.0	28	4.83	15.8
0.01562	2	0.300	0.0441	26	6.05	8.67	19	1.96	15.3	21	6.15	23.6
0.01562	2	0.400	0.0576	20	10.8	18.5	16	1.92	26.5	20	11.4	40.5
0.01562	2	0.500	0.0625	19	13.7	16.6	30	1.60	12.8	17	20.0	48.8
0.01562	2	0.600	0.0576	20	10.9	18.5	16	1.90	26.5	20	11.4	40.7
0.01562	2	0.700	0.0441	26	6.07	8.64	18	1.93	16.0	27	6.05	19.8
0.01562	2	0.800	0.0256	19	4.90	7.62	18	1.59	11.0	19	4.85	19.0
0.01562	2	0.900	0.0081	15	3.39	8.51	16	1.22	10.5	16	3.73	20.0
0.00781	4	0.100	0.0324	26	5.34	7.7	19	1.72	13.7	20	5.31	22.1
0.00781	4	0.200	0.1024	18	14.5	21.2	18	2.58	25.2	17	17.3	64.8
0.00781	4	0.300	0.1764	17	18.0	23.8	26	3.73	20.3	15	25.8	91.8
0.00781	4	0.400	0.2304	21	10.3	13.3	24	5.55	18.7	18	15.1	58.5
0.00781	4	0.500	0.2500	17	21.1	24.3	22	3.05	22.9	17	20.1	56.3
0.00781	4	0.600	0.2304	21	10.3	13.3	28	5.51	16.9	18	15.3	59.7
0.00781	4	0.700	0.1764	17	18.2	24.6	28	3.75	19.7	15	25.9	92.3
0.00781	4	0.800	0.1024	18	14.7	21.1	18	2.58	25.2	17	17.0	65.0
0.00781	4	0.900	0.0324	22	5.35	8.02	19	1.71	12.4	22	5.3	19.4

Table 1. Multigrid solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

2 level algorithms (cont.)

				asymptotic convergence factor								
				I exper.			II exper.			III exper.		
h_y	m	t	RC	cycles	final	aver.	cycles	final	aver.	cycles	final	aver.
0.00391	8	0.100	0.1296	18	16.1	22.2	22	3.00	22.4	16	20.4	80.0
0.00391	8	0.200	0.4096	22	9.92	11.9	25	7.70	19.5	16	21.6	89.5
0.00391	8	0.300	0.7056	28	6.18	6.89	27	6.26	14.1	14	31.2	113
0.00391	8	0.400	0.9216	24	4.06	4.62	37	3.11	7.45	14	36.6	89.6
0.00391	8	0.500	1.0000	14	3.69	4.60	18	3.7	7.95	17	20.0	44.3
0.00391	8	0.600	0.9216	25	4.07	4.61	37	3.75	7.54	14	36.7	89.4
0.00391	8	0.700	0.7056	28	6.19	6.91	26	6.32	14.5	14	31.3	112
0.00391	8	0.800	0.4096	22	9.92	12.0	24	7.66	20	16	21.7	88.3
0.00391	8	0.900	0.1296	17	16.2	22.8	16	2.98	28.7	16	20.4	78.9
0.00781	16	0.100	0.5184	22	9.56	10.6	22	9.57	20.8	14	31.8	113
0.00781	16	0.200	1.6384	14	2.27	3.04	13	2.26	6.4	13	48.6	109
0.00781	16	0.300	2.8224	15	1.63	2.43	15	1.64	4.95	13	42.6	135
0.00781	16	0.400	3.6864	14	1.48	2.47	14	1.48	4.99	12	48.5	208
0.00781	16	0.500	4.0000	13	1.42	2.58	14	1.42	4.86	14	31.3	113
0.00781	16	0.600	3.6864	14	1.48	2.41	13	1.48	5.15	13	46.7	195
0.00781	16	0.700	2.8224	14	1.63	2.63	13	1.64	5.46	13	41.2	136
0.00781	16	0.800	1.6384	19	2.27	2.87	17	2.26	6.1	12	49.3	121
0.00781	16	0.900	0.5184	21	9.41	10.7	22	9.69	20.5	13	32.4	119

Table 2. Multigrid solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

Multi-level V(1,1) cycles

				asymptotic convergence factor					
				conditional relax.			zebra		
h_y	m	t	RC	cycles	final	aver.	cycles	final	aver.
0.03125	1	0.100	0.00203	12	2.34	17.5	12	2.52	44.4
0.03125	1	0.200	0.00640	19	2.66	6.57	18	2.75	14.4
0.03125	1	0.300	0.01103	23	3.22	5.16	22	3.35	11.6
0.03125	1	0.400	0.01440	20	3.91	6.10	20	3.96	13.5
0.03125	1	0.500	0.01562	20	4.94	6.96	24	5.02	13.5
0.03125	1	0.600	0.01440	17	3.88	6.70	24	4.01	11.7
0.03125	1	0.700	0.01103	24	3.24	5.22	19	3.37	13.2
0.03125	1	0.800	0.00640	23	2.67	5.45	20	2.73	13.1
0.03125	1	0.900	0.00202	12	2.31	15.6	12	2.55	40.0
0.01562	2	0.100	0.00810	30	2.91	5.29	24	2.98	13.1
0.01562	2	0.200	0.02560	23	4.48	6.84	26	4.80	13.8
0.01562	2	0.300	0.04410	30	4.83	7.33	26	6.03	18.1
0.01562	2	0.400	0.05760	28	5.16	9.66	22	8.69	28.8
0.01562	2	0.500	0.06250	20	12.9	15.3	19	13.2	35.9
0.01562	2	0.600	0.05760	19	5.09	12.2	22	8.47	27.2
0.01562	2	0.700	0.04410	22	4.83	8.18	22	6.04	18.2
0.01562	2	0.800	0.02560	28	4.49	6.35	19	4.75	18.2
0.01562	2	0.900	0.00810	18	2.91	6.59	18	2.97	14.6
0.00781	4	0.100	0.03240	20	4.68	7.47	23	5.26	15.3
0.00781	4	0.200	0.10240	21	7.03	11.0	19	12.7	44.9
0.00781	4	0.300	0.17640	25	6.94	9.38	16	19.3	61.2
0.00781	4	0.400	0.23040	27	6.18	7.88	17	16.3	48.5
0.00781	4	0.500	0.25000	18	16.2	21.6	19	13.2	39.9
0.00781	4	0.600	0.23040	27	6.14	7.75	17	16.3	49.3
0.00781	4	0.700	0.17640	25	6.95	9.26	16	19.7	61.8
0.00781	4	0.800	0.10240	23	7.06	10.8	18	12.6	46.5
0.00781	4	0.900	0.03240	27	4.66	6.93	23	5.27	16.9

Table 2. Multigrid solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

Multi-level V(1,1) cycles (cont.)

				asymptotic convergence factor					
				conditional relax.			zebra		
h_y	m	t	RC	cycles	final	aver.	cycles	final	aver.
0.00391	8	0.100	0.12960	24	7.09	10.2	18	15.2	54.2
0.00391	8	0.200	0.40960	22	9.91	12.0	16	18.2	74.0
0.00391	8	0.300	0.70560	28	6.17	6.88	16	6.12	63.0
0.00391	8	0.400	0.92160	24	4.07	4.63	16	4.20	41.2
0.00391	8	0.500	1.00000	12	3.80	28.1	14	3.79	48.4
0.00391	8	0.600	0.92160	23	4.07	4.65	13	4.19	48.9
0.00391	8	0.700	0.70560	28	6.18	6.89	13	6.12	75.6
0.00391	8	0.800	0.40960	22	10.0	12.1	16	17.9	74.5
0.00391	8	0.900	0.12960	24	7.07	10.2	17	15.4	57.2
0.00391	16	0.100	0.51840	22	9.04	10.5	14	30.8	93.9
0.00391	16	0.200	1.63840	16	18.3	67.5	16	18.1	133
0.00391	16	0.300	2.82240	14	6.10	78.7	14	6.10	157
0.00391	16	0.400	3.68640	12	4.13	112	12	4.16	228
0.00391	16	0.500	4.00000	19	3.74	53.6	14	3.75	147
0.00391	16	0.600	3.68640	19	4.15	70.0	12	4.16	229
0.00391	16	0.700	2.82240	15	6.10	74.2	19	6.11	121
0.00391	16	0.800	1.63840	16	18.0	67.0	16	18.2	131
0.00391	16	0.900	0.51840	22	9.12	10.6	14	30.8	92.4

Table 3 Multigrid FMG solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

$\xi = \frac{x+ty}{\sqrt{1+t^2}}; \quad \eta = \frac{-tx+y}{\sqrt{1+t^2}}$

Basic cycle: V(1,1).

Characteristic components.

Exact continuous solution: $U = \sin(\beta_\xi \xi + \beta_\eta \eta) = \sin(\theta x + \omega y)$

t	$\beta_\xi \sqrt{1+t^2} h_y$	ωh_y	θh_x	Discretization error	Algebraic error		
					Before cycles	After 1 cycle	After 2 cycles
0.100	0.04711	0.09817	0.03706	0.00523	0.000191	$1.89 \cdot 10^{-05}$	$8.97 \cdot 10^{-06}$
0.300	0.04894	0.09817	0.01742	0.043	0.000987	$4.84 \cdot 10^{-05}$	$1.22 \cdot 10^{-05}$
0.500	0.05241	0.09817	−0.00221	0.118	0.00157	$3.88 \cdot 10^{-05}$	$2.83 \cdot 10^{-06}$
0.700	0.05722	0.09817	−0.02185	0.231	0.000628	$5.94 \cdot 10^{-05}$	$1.29 \cdot 10^{-05}$
0.900	0.06306	0.09817	−0.04148	0.383	0.000202	$4.97 \cdot 10^{-06}$	$2.14 \cdot 10^{-06}$
0.100	0.04711	0.39270	0.00761	0.0122	0.00619	0.000348	0.000148
0.300	0.04894	0.39270	−0.07093	0.0104	0.0280	0.00204	0.000327
0.500	0.05241	0.39270	−0.14947	0.0424	0.0198	0.00106	0.000127
0.700	0.05722	0.39270	−0.22801	0.151	0.0131	0.000882	0.000206
0.900	0.06306	0.39270	−0.30655	0.354	0.00554	0.000185	$2.16 \cdot 10^{-05}$
0.100	0.04711	0.88357	−0.04148	0.0549	0.0513	0.00899	0.00213
0.300	0.04894	0.88357	−0.21820	0.357	0.0555	0.00859	0.00195
0.500	0.05241	0.88357	−0.39491	0.424	0.0195	0.00169	0.000225
0.700	0.05722	0.88357	−0.57163	0.373	0.0430	0.00747	0.00143
0.900	0.06306	0.88357	−0.74834	0.0119	0.0366	0.00490	0.00103
0.100	0.04711	1.96350	−0.14947	0.523	0.0424	0.0104	0.00414
0.300	0.04894	1.96350	−0.54217	0.653	0.0487	0.00746	0.00174
0.500	0.05241	1.96350	−0.93487	0.664	0.0414	0.00630	0.000926
0.700	0.05722	1.96350	−1.32757	0.657	0.0492	0.00750	0.00173
0.900	0.06306	1.96350	−1.72027	0.581	0.0398	0.0116	0.00489
0.100	0.04711	3.53429	−0.30655	0.689	0.0441	0.0121	0.00476
0.300	0.04894	3.53429	−1.01341	0.698	0.0536	0.00667	0.00116
0.500	0.05241	3.53429	−1.72027	0.699	0.0500	0.00322	0.000149
0.700	0.05722	3.53429	−2.42713	0.699	0.0536	0.00609	0.000996
0.900	0.06306	3.53429	−3.13399	0.69	0.0442	0.0122	0.00478

Table 3 Multigrid FMG solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

$\xi = \frac{x+ty}{\sqrt{1+t^2}}; \quad \eta = \frac{-tx+y}{\sqrt{1+t^2}}$

Basic cycle: V(1,1); (cont.)

Intermediate components.

Exact continuous solution: $U = \sin(\beta_\xi \xi + \beta_\eta \eta) = \sin(\theta x + \omega y)$

t	$\beta_\xi \sqrt{1+t^2} h_y$	ωh_y	θh_x	Discretization error	Algebraic error		
					Before cycles	After 1 cycle	After 2 cycles
0.100	0.78515	0.09817	0.77143	0.0627	0.0631	0.00281	0.00135
0.300	0.81565	0.09817	0.75180	0.147	0.0688	0.00243	0.000224
0.500	0.87346	0.09817	0.73216	0.314	0.0793	0.00353	$5.82 \cdot 10^{-05}$
0.700	0.95364	0.09817	0.71253	0.564	0.0940	0.00342	0.000213
0.900	1.05106	0.09817	0.69289	0.896	0.113	0.00525	0.00251
0.100	0.78515	0.39270	0.74198	0.0708	0.0657	0.00259	0.0012
0.300	0.81565	0.39270	0.66344	0.159	0.0791	0.00331	0.000424
0.500	0.87346	0.39270	0.58490	0.319	0.107	0.00586	0.000241
0.700	0.95364	0.39270	0.50636	0.561	0.106	0.00295	0.000202
0.100	0.78515	0.88357	0.69289	0.103	0.0825	0.00724	0.00197
0.300	0.81565	0.88357	0.51618	0.201	0.137	0.00662	0.00141
0.500	0.87346	0.88357	0.33946	0.324	0.238	0.0130	0.00165
0.700	0.95364	0.88357	0.16275	0.492	0.151	0.00346	0.000774
0.900	1.05106	0.88357	−0.01397	0.806	0.147	0.0124	0.00356
0.100	0.78515	1.96350	0.58490	0.353	0.132	0.00509	0.00201
0.300	0.81565	1.96350	0.19220	0.537	0.540	0.0749	0.0131
0.500	0.87346	1.96350	−0.20050	0.326	0.592	0.0646	0.00749
0.700	0.95364	1.96350	−0.59320	0.292	0.308	0.00643	0.00113
0.900	1.05106	1.96350	−0.98590	0.545	0.229	0.0120	0.00486
0.100	0.78515	3.53429	0.42782	2.14	0.293	0.0225	0.00879
0.300	0.81565	3.53429	−0.27904	0.348	0.584	0.0981	0.0186
0.500	0.87346	3.53429	−0.98590	0.294	0.334	0.0113	0.000165
0.700	0.95364	3.53429	−1.69275	0.314	0.576	0.0204	0.00144
0.900	1.05106	3.53429	−2.39961	0.0445	0.316	0.0134	0.00465

Table 3 Multigrid FMG solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

$$\xi = \frac{x+ty}{\sqrt{1+t^2}}; \qquad \eta = \frac{-tx+y}{\sqrt{1+t^2}}$$

Basic cycle: V(1,1); (cont.)

Non-characteristic components.

Exact continuous solution: $U = \sin(\beta_\xi \xi + \beta_\eta \eta) = \sin(\theta x + \omega y)$

t	$\beta_\xi \sqrt{1+t^2} h_y$	ωh_y	θh_x	Discretization error	Algebraic error		
					Before cycles	After 1 cycle	After 2 cycles
0.100	2.04138	0.09817	2.02143	0.386	1.18	0.00227	0.000682
0.300	2.12069	0.09817	2.00180	0.486	1.27	0.0232	0.000731
0.500	2.27101	0.09817	1.98216	0.687	1.46	0.0508	0.000346
0.700	2.47946	0.09817	1.96253	0.985	1.74	0.0405	0.00121
0.900	2.73277	0.09817	1.94289	1.38	2.12	0.00340	0.000523
0.100	2.04138	0.39270	1.99198	0.392	1.19	0.00233	0.000666
0.300	2.12069	0.39270	1.91344	0.497	1.30	0.0201	0.000582
0.500	2.27101	0.39270	1.83490	0.697	1.47	0.0555	0.00179
0.700	2.47946	0.39270	1.75636	0.998	1.77	0.0440	0.00169
0.900	2.73277	0.39270	1.67782	1.4	2.13	0.00293	0.000500
0.100	2.04138	0.88357	1.94289	0.402	1.23	0.00309	0.000733
0.300	2.12069	0.88357	1.76618	0.532	1.39	0.0151	0.00174
0.500	2.27101	0.88357	1.58946	0.729	1.53	0.0532	0.00653
0.700	2.47946	0.88357	1.41275	0.989	1.88	0.0400	0.00282
0.900	2.73277	0.88357	1.23603	1.40	2.20	0.0113	0.00255
0.100	2.04138	1.96350	1.83490	0.555	1.37	0.0284	0.00900
0.300	2.12069	1.96350	1.44220	1.05	1.85	0.0204	0.00343
0.500	2.27101	1.96350	1.04950	1.32	1.84	0.0259	0.0036
0.700	2.47946	1.96350	0.65680	1.32	2.31	0.0492	0.00579
0.900	2.73277	1.96350	0.26410	1.34	2.50	0.0334	0.00857
0.100	2.04138	3.53429	1.67782	1.00	1.58	0.420	0.186
0.300	2.12069	3.53429	0.97096	5.24	3.12	0.0727	0.00571
0.500	2.27101	3.53429	0.26410	3.45	3.21	0.274	0.0175
0.700	2.47946	3.53429	−0.44275	1.41	2.69	0.349	0.0832
0.900	2.73277	3.53429	−1.14961	1.16	3.23	0.0594	0.0163

Table 4 Multigrid FMG solver for the eqn. $\frac{\partial^2 U}{\partial \xi^2} = f$ in 2D.

$$\xi = \frac{x+ty}{\sqrt{1+t^2}}; \qquad \eta = \frac{-tx+y}{\sqrt{1+t^2}}$$

Vanishing of discretization error;

Exact continuous solution: $U = \sin(\beta_\xi \xi + \beta_\eta \eta) = \sin(\theta x + \omega y)$

t	$\beta_\xi \sqrt{1+t^2} h_y$	ωh_y	θh_x	Discr. err.	Before cycles		After 1 cycle		After 2 cycles	
					Alg. Err.	Total Err.	Alg. Err.	Total Err.	Alg. Err.	Total Err.
0.073183	−1.256869	3.14159	−1.48343	$1.53 \cdot 10^{-08}$	0.527	0.527	0.0210	0.021	0.00397	0.00397
0.000000	−1.253516	3.14159	−1.25352	0.124	0.33	0.325	$6.8 \cdot 10^{-16}$	0.124	$8.89 \cdot 10^{-16}$	0.124
0.100000	−1.259768	3.14159	−1.56768	0.0201	0.666	0.686	0.00138	0.0205	0.000374	0.0201
0.073183	−1.253343	3.14159	−1.47991	0.00145	0.514	0.514	0.0217	0.0222	0.00411	0.0048
0.073183	−1.316010	3.14159	−1.54241	0.0213	0.691	0.672	0.00727	0.0213	0.00137	0.0211
0.290908	0.470074	2.35619	−0.23407	$1.91 \cdot 10^{-07}$	0.345	0.345	0.0746	0.0746	0.0196	0.0196
0.200000	0.460301	2.35619	−0.01988	0.329	0.409	0.111	0.110	0.221	0.0349	0.295
0.300000	0.471236	2.35619	−0.25550	0.0244	0.338	0.361	0.0707	0.0947	0.0180	0.0421
0.290908	0.455636	2.35619	−0.24794	0.0295	0.329	0.357	0.0700	0.0993	0.0183	0.0476
0.290908	0.520727	2.35619	−0.18544	0.105	0.402	0.302	0.0904	0.0225	0.0239	0.0815
0.284833	−0.608658	1.57080	−1.03279	$2.31 \cdot 10^{-07}$	0.118	0.118	0.00768	0.00768	0.0018	0.0018
0.200000	−0.596968	1.57080	−0.89953	0.0306	0.108	0.104	0.0106	0.0300	0.00251	0.0297
0.300000	−0.611149	1.57080	−1.05661	0.00864	0.120	0.120	0.00725	0.0136	0.00163	0.00929
0.284833	−0.584873	1.57080	−1.00991	0.0148	0.115	0.116	0.00788	0.0140	0.00192	0.0144
0.284833	−0.649859	1.57080	−1.07241	0.0245	0.124	0.126	0.0072	0.0285	0.00155	0.0252
0.511639	−0.145227	0.78540	−0.53113	$8.19 \cdot 10^{-08}$	0.0543	0.0543	0.00259	0.00259	0.000255	0.000254
0.500000	−0.144548	0.78540	−0.52199	0.00438	0.0514	0.0552	0.00226	0.00645	0.000231	0.00460
0.600000	−0.150774	0.78540	−0.60053	0.0401	0.0688	0.0366	0.00403	0.0365	0.000312	0.0398
0.511639	−0.140411	0.78540	−0.52684	0.00514	0.0525	0.0569	0.00238	0.00733	0.000229	0.00536
0.511639	−0.210616	0.78540	−0.58934	0.0782	0.0822	0.0304	0.00593	0.0726	0.000637	0.0776

Table 5 Multigrid FMG solver for the eqn. $\frac{\partial^2\phi}{\partial\mu^2} + \frac{\partial^2\phi}{\partial\beta^2} = f$ in 3D.

μ and β are orthonormal coordinates on the oblique plane,
spanned on the vectors $\bar{q} = (m_x, 0, k_x + s_x)$ and $\bar{r} = (0, m_y, k_y + s_y)$

$m_x = m_y = 1; k_x = k_y = 0$

5 levels; characteristic components;

Exact continuous solution: $U = \sin(\omega_\xi \xi + \omega_\eta \eta + \omega_\zeta \zeta) = \sin(\theta_x x + \theta_y y + \theta_z z)$
$\Omega_\xi = \omega_\xi \sqrt{m_x^2 + (k_x + s_x)^2} h_z$, $\Omega_\eta = \omega_\eta \sqrt{m_y^2 + (k_y + s_y)^2} h_z$, $\Omega_\zeta = \omega_\zeta h_z$
$\Theta_x = \theta_x m_x h_z$, $\Theta_y = \theta_y m_y h_z$, $\Theta_z = \theta_z h_z$

										Algebraic error		
h_z	s_x	s_y	Ω_ξ	Ω_η	Ω_ζ	Θ_x	Θ_y	Θ_z	Discr. err.	Before cycles	After 1 cycle	After 2 cycles
0.03125	0.100	0.200	0.01973	0.04005	0.19146	0.00010	0.00078	0.19635	0.000588	0.000323	$4.08 \cdot 10^{-05}$	$5.13 \cdot 10^{-06}$
0.03125	0.100	0.400	0.01973	0.04229	0.19492	0.00010	−0.03624	0.19635	0.000967	0.000527	$6.54 \cdot 10^{-05}$	$8.05 \cdot 10^{-06}$
0.03125	0.100	0.500	0.01973	0.04391	0.19909	0.00010	−0.05427	0.19635	0.00138	0.000444	$5.47 \cdot 10^{-05}$	$6.67 \cdot 10^{-06}$
0.03125	0.100	0.990	0.01973	0.05526	0.23681	0.00010	−0.13913	0.19635	0.00431	0.000107	$1.19 \cdot 10^{-05}$	$1.42 \cdot 10^{-06}$
0.03125	0.450	0.500	0.02153	0.04391	0.21039	−0.06683	−0.05427	0.19635	0.00473	0.000458	$4.49 \cdot 10^{-05}$	$4.52 \cdot 10^{-06}$
0.03125	0.850	0.900	0.02577	0.05283	0.26882	−0.14113	−0.12388	0.19635	0.0271	$5.57 \cdot 10^{-05}$	$3.54 \cdot 10^{-06}$	$2.85 \cdot 10^{-07}$
0.03125	0.300	0.200	0.02050	0.04005	0.61285	−0.15622	−0.07776	0.58905	0.0673	0.00879	0.00172	0.000334
0.03125	0.300	0.400	0.02050	0.04229	0.63794	−0.15622	−0.19332	0.58905	0.0744	0.00908	0.00187	0.000376
0.03125	0.300	0.990	0.02050	0.05526	0.80522	−0.15622	−0.52790	0.58905	0.0978	0.00427	0.000322	$4.84 \cdot 10^{-05}$
0.03125	0.650	0.500	0.02342	0.04391	0.73304	−0.35946	−0.25062	0.58905	0.0953	0.00286	0.000487	$8.8 \cdot 10^{-05}$
0.03125	0.750	0.900	0.02454	0.05283	0.86449	−0.41724	−0.47731	0.58905	0.117	0.00157	0.000143	$2.42 \cdot 10^{-05}$
0.03125	0.100	0.200	0.01973	0.04005	1.39865	−0.11771	−0.23484	1.37445	0.449	0.0210	0.00510	0.00127
0.03125	0.100	0.500	0.01973	0.04391	1.52150	−0.11771	−0.64332	1.37445	0.475	0.0209	0.00416	0.00118
0.03125	0.100	0.990	0.01973	0.05526	1.89877	−0.11771	−1.30544	1.37445	0.472	0.00755	0.000742	$9.66 \cdot 10^{-05}$
0.03125	0.450	0.500	0.02153	0.04391	1.63022	−0.59697	−0.64332	1.37445	0.494	0.0337	0.00536	0.00111
0.03125	0.850	0.900	0.02577	0.05283	2.14363	−1.14251	−1.18417	1.37445	0.491	0.00922	0.00127	0.000226
0.03125	0.200	0.200	0.02002	0.04005	2.43707	−0.45122	−0.43119	2.35619	0.606	0.0342	0.00521	0.000917
0.03125	0.200	0.400	0.02002	0.04229	2.56198	−0.45122	−0.90018	2.35619	0.619	0.0400	0.00501	0.000824
0.03125	0.200	0.990	0.02002	0.05526	3.30756	−0.45122	−2.27737	2.35619	0.607	0.0307	0.00299	0.000406
0.03125	0.350	0.500	0.02080	0.04391	2.73542	−0.80387	−1.13419	2.35619	0.627	0.0478	0.00294	0.000299
0.03125	0.950	0.900	0.02708	0.05283	3.83608	−2.21130	−2.06774	2.35619	0.596	0.00734	0.000462	$3.48 \cdot 10^{-05}$

Table 5 Multigrid FMG solver for the eqn. $\frac{\partial^2\phi}{\partial\mu^2}+\frac{\partial^2\phi}{\partial\beta^2}=f$ in 3D.
(continuation)

μ and β are orthonormal coordinates on the oblique plane,
spanned on the vectors $\bar{q}=(m_x,0,k_x+s_x)$ and $\bar{r}=(0,m_y,k_y+s_y)$

$m_x=m_y=1; k_x=k_y=0$

5 levels; non-characteristic components;

Exact continuous solution: $U=\sin(\omega_\xi\xi+\omega_\eta\eta+\omega_\zeta\zeta)=\sin(\theta_x x+\theta_y y+\theta_z z)$
$\Omega_\xi=\omega_\xi\sqrt{m_x^2+(k_x+s_x)^2}h_z,\ \Omega_\eta=\omega_\eta\sqrt{m_y^2+(k_y+s_y)^2}h_z,\ \Omega_\zeta=\omega_\zeta h_z$
$\Theta_x=\theta_x m_x h_z,\ \Theta_y=\theta_y m_y h_z,\ \Theta_z=\theta_z h_z$

h_z	s_x	s_y	Ω_ξ	Ω_η	Ω_ζ	Θ_x	Θ_y	Θ_z	Discr. err.	Algebraic error		
										Before cycles	After 1 cycle	After 2 cycles
0.03125	0.100	0.200	2.07195	2.46293	−0.48172	2.05232	2.42366	0.19635	0.399	1.08	0.0685	0.00326
0.03125	0.100	0.400	2.07195	2.60114	−0.94107	2.05232	2.52260	0.19635	0.432	1.11	0.0658	0.00314
0.03125	0.100	0.500	2.07195	2.70016	−1.16693	2.05232	2.60199	0.19635	0.455	1.13	0.0599	0.00267
0.03125	0.100	0.990	2.07195	3.39843	−2.25481	2.05232	3.20405	0.19635	0.633	1.32	0.0291	0.00174
0.03125	0.450	0.500	2.26080	2.70016	−1.72772	2.17244	2.60199	0.19635	0.423	1.10	0.0746	0.00552
0.03125	0.850	0.900	2.70582	3.24918	−2.97035	2.53892	3.07247	0.19635	0.142	0.87	0.0280	0.00256
0.03125	0.300	0.200	2.15245	2.46293	−0.44467	1.97573	2.34512	0.58905	0.440	1.13	0.0893	0.00552
0.03125	0.300	0.400	2.15245	2.60114	−0.84960	1.97573	2.36552	0.58905	0.467	1.15	0.0905	0.00611
0.03125	0.300	0.990	2.15245	3.39843	−1.93969	1.97573	2.81527	0.58905	0.506	1.19	0.0233	0.00245
0.03125	0.650	0.500	2.45893	2.70016	−1.51803	2.07604	2.40564	0.58905	0.439	1.14	0.136	0.0186
0.03125	0.750	0.900	2.57709	3.24918	−2.24604	2.13530	2.71904	0.58905	0.227	0.943	0.0344	0.00422
0.03125	0.100	0.200	2.07195	2.46293	0.72547	1.93451	2.18804	1.37445	0.534	1.25	0.0770	0.00331
0.03125	0.100	0.500	2.07195	2.70016	0.15548	1.93451	2.01294	1.37445	0.647	1.36	0.111	0.00806
0.03125	0.100	0.990	2.07195	3.39843	−0.59286	1.93451	2.03773	1.37445	0.694	1.37	0.0255	0.00138
0.03125	0.450	0.500	2.26080	2.70016	−0.30788	1.64230	2.01294	1.37445	0.709	1.54	0.301	0.0452
0.03125	0.850	0.900	2.70582	3.24918	−1.09554	1.53754	2.01218	1.37445	0.200	0.916	0.0715	0.00804
0.03125	0.200	0.200	2.10250	2.46293	1.57001	1.63126	1.99169	2.35619	0.989	1.73	0.0399	0.000957
0.03125	0.200	0.400	2.10250	2.60114	1.24742	1.63126	1.65866	2.35619	1.30	2.03	0.0526	0.00199
0.03125	0.200	0.990	2.10250	3.39843	0.68585	1.63126	1.06580	2.35619	0.854	1.53	0.0298	0.000817
0.03125	0.350	0.500	2.18430	2.70016	0.95540	1.35963	1.52207	2.35619	1.55	2.30	0.120	0.00356
0.03125	0.950	0.900	2.84369	3.24918	0.46474	0.60530	1.12861	2.35619	0.0845	0.769	0.0406	0.00259