Astron. Astrophys. 324, 161–176 (1997)

ASTRONOMY
AND
ASTROPHYSICS

Multidimensional radiative transfer with multilevel atoms

II. The non-linear multigrid method

P. Fabiani Bendicho[1,2], **J. Trujillo Bueno**[2], **and L. Auer**[3]

[1] Kiepenheuer Institut für Sonnenphysik, Schöneckstr. 6, D-79104 Freiburg, Germany
[2] Instituto de Astrofísica de Canarias, E-38200, La Laguna (Tenerife), Spain
[3] Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 2 September 1996 / Accepted 2 January 1997

Abstract. A new iterative method for solving non-LTE *multilevel* radiative transfer (RT) problems in 1D, 2D or 3D geometries is presented. The scheme obtains the self-consistent solution of the kinetic and RT equations at the cost of only a few (< 10) formal solutions of the RT equation. It combines, for the first time, *non-linear* multigrid iteration (Brandt, 1977; Hackbush, 1985), an efficient multilevel RT scheme based on Gauss-Seidel iterations (cf. Trujillo Bueno & Fabiani Bendicho, 1995), and accurate *short-characteristics* formal solution techniques. By combining a valid stopping criterion with a nested-grid strategy a converged solution with the desired true error is automatically guaranteed. Contrary to the current operator splitting methods the very high convergence speed of the new RT method does not deteriorate when the grid spatial resolution is increased. With this non-linear multigrid method non-LTE problems discretized on N grid points are solved in O(N) operations. The nested multigrid RT method presented here is, thus, particularly attractive in complicated multilevel transfer problems where small grid-sizes are required. The properties of the method are analyzed both analytically and with illustrative multilevel calculations for Ca II in 1D and 2D schematic model atmospheres.

Key words: radiative transfer – stars: atmospheres – methods: numerical

1. Introduction

In astrophysics and other branches of physics the problems needing investigation often exceed the capacity of even the most powerful computers using the available methods. One is, thus, confronted with the need to develop and implement new numerical methods capable of offering much higher efficiency. That is the goal of this paper for the solution of complicated non-LTE transfer problems. We show how multidimensional, multilevel problems can be accurately and rapidly solved by means of *non-linear* multigrid iteration. By "accurately" we mean a *true* accuracy better than, say, 0.1% at all points, while by "rapidly" we mean that the CPU-time required to obtain the converged solution (either for 1D, 2D or 3D multilevel problems) is roughly the time needed to perform very few (< 10) formal solutions of the radiative transfer equation.

The development of powerful numerical methods for solving multilevel transfer problems is relevant not only in the stellar atmospheres context where there is strong interest in investigating 2D and 3D non-LTE effects using complex atomic models, but also in many other astrophysical research fields where the radiation-matter interaction must be considered in detail. Such problems are found, for instance, in the subjects of astronomical masers, spectral line formation in accretion disks, and the interpretation of spectro-polarimetric observations to determine cosmic magnetic fields. Further, there are a variety of astrophysical problems which require *extreme* grid-fineness for their correct solution. For example, magneto-hydrodynamic simulations of stellar atmospheric processes show the existence of abrupt changes in the physical state of the stellar plasma because of wave propagation and shock formation. In order to obtain the spectral signatures corresponding to selected snapshots of the time-dependent simulations, it is often necessary to use very fine spatial grids to resolve fully the abrupt changes in the excitation and ionization balance of the chemical species. The combination of multigrid and Gauss-Seidel iteration which we present here for the first time, is a remarkably effective general tool for meeting these needs.

Multigrid methods have their roots in the following observation about the convergence properties of "classical" operator-splitting methods like the Jacobi scheme: the *coarser* the grid the faster the convergence of the *low-frequency* spatial components of the error, while the *finer* the grid the faster the convergence of the *high-frequency* components (see, e.g., Hackbush, 1985). Therefore, in order to maximize the overall convergence, the multigrid iteration should be composed of two essential parts:

Send offprint requests to: J. Trujillo Bueno a *smoothing* one where a small number of iterations on the desired *fine* grid damp the high-frequency spatial components of the error in the current estimate, and a *coarser* grid correction to suppress the low frequency components. The advantages of such multigrid iteration are its very high convergence rate and that this high convergence speed does not deteriorate when the discretization is refined. This is in sharp contrast with the fastest iterative methods currently in use for the solution of radiative transfer problems (see, e.g., Rybicki & Hummer, 1991,1992; Auer, Fabiani Bendicho & Trujillo Bueno, 1994), for which the convergence speed decreases for increasing grid resolution.

The convergence speed of the recently-developed RT methods based on Gauss-Seidel (GS) and successive over-relaxation (SOR) iteration (see Trujillo Bueno & Fabiani Bendicho, 1995; hereafter TF-1), which have dramatically better convergence properties and are much faster than the above-mentioned ones, is also sensitive to the spatial resolution of the grid. Although these GS-based iterative schemes can rapidly solve complicated non-LTE problems in very fine grids, the critical point is that they provide extremely good smoothing properties and, therefore, make a multigrid approach exceptionally effective. The combination can treat fine spatial discretization with great effectiveness, as the time for solution scales only *linearly* with the number of grid points.

The mathematical foundations of multigrid methods are described in the monograph by Hackbusch (1985), where references to original work are also given, while numerical recipes for the application of multigrid methods can be found in Press *et. al.* (1994). The first application of the *linear* multigrid method to RT was done by Steiner (1991) for the particular case of coherent scattering. For the *smoothing* part he tried the *local* and *non-local* operator splitting methods described, respectively, by Olson, Auer and Buchler (1986; hereafter OAB) and by Olson & Kunasz (1987), with the result that the *local* approximate operator method has a much worse smoothing capability than a *non-local* one, often leading to failure of the multigrid iteration. For *one-dimensional* applications there is no problem in using a *non-local* operator splitting method which takes *nearest* depth couplings into account; however, in 2D and 3D applications the currently-used *non-local* approximate operator methods are not suitable because they require prohibitively large computing time and memory storage (see the discussion on this point in TF-1). The method of choice for the smoothing part of multigrid RT codes is the new RT method developed by TF-1, which is based on Gauss-Seidel (GS) iterations. With this new radiative transfer method there is no need for constructing and inverting any *non-local* operator and the computing time per iteration is minimal (i.e. virtually identical to that of a *local* operator splitting method). Further, the method has excellent *smoothing* capabilities. As we shall show, its efficiency in smoothing the high-frequency error components is so great, that only one or two smoothing iterations normally suffice to obtain the very high convergence rate which is obtainable with the multigrid method.

The above remarks also imply that 2D and 3D *two-level* atom RT codes based on the same *linear* multigrid scheme applied by Steiner (1991) and Väth (1994) should be implemented by making use of the GS-based method of TF-1 for their *smoothing* part. In this paper, however, we shall concentrate on the much more complicated problem of applying the *non-linear* multigrid method of Brandt (1977) to multilevel radiative transfer. This *non-linear* multigrid method is characterized by two essential parts: a *smoothing* one and a *coarse-grid* correction. For the *smoothing* part one needs a *non-linear* iterative relaxation method suitable for the multilevel atom problem, which, of course, should also be extremely efficient in its smoothing capability. For this *smoothing* purpose we shall correspondingly apply the method described by Trujillo Bueno & Fabiani Bendicho (1996; hereafter TF-2), which is nothing but the generalization to the multilevel non-linear case of the GS-based method of TF-1.

The outline of this paper is as follows. Sect. 2 discusses the application to multilevel RT of the non-linear multigrid method of Brandt (1977), showing examples that illustrate the complementary action of the coarse grid correction (CGC) and the *smoothing* steps. It also quantifies and compares the smoothing ability of two non-linear iterative schemes for multilevel RT calculations (one based on Jacobi's method and the other on Gauss-Seidel iterations), showing that the method of choice for the *smoothing* part of a non-linear RT multigrid code is the GS-based method of TF-2. Sect. 3 studies in detail the convergence rate of the multigrid iteration. Sect. 4 derives and discusses analytical formulae concerning the computational work and efficiency of our non-linear multigrid RT code. Sect. 5 introduces an improved multigrid RT technique (called the "nested multigrid method"), which is nothing but the combination of our grid-doubling strategy of Paper I (cf. Auer, Fabiani Bendicho and Trujillo Bueno, 1994) with the *standard* multigrid method. Here we show examples of 2D line-transfer calculations, comparing the efficiency of this nested multigrid RT method with the one of the grid-doubling strategy of Paper I. Finally, in Sect. 6 we present our conclusions.

Before entering into details we would like to clarify from the outset the two following points. First, in this paper we put the emphasis on describing in great detail the basic ingredients of the new RT method and the properties of our non-linear multigrid RT code. To this end, we derive some general analytical formulae, and concentrate on showing demonstrative numerical results for the same 5-level Ca II 1D and 2D transfer problems described in Paper I. Future papers will show results for other atoms (like, e.g., K, Fe and H) and inhomogeneous model atmospheres, demonstrating that a similarly good behaviour is obtained in these cases. Second, we should stress that the methods of this paper require only scalar-class computers, and that our nested multigrid RT method permits the modeling of 2D and 3D multilevel problems with no more than a workstation.

2. The non-linear multigrid method

The aim of this section is to describe in detail the standard version of the *non-linear* multigrid method. To this end we, first, formulate in Sect. 2.1 the multilevel transfer problem pointing out the basic features of two "classical" iterative schemes which have been developed for solving this problem: one called MALI (Rybicki & Hummer, 1991,1992; see also Paper I) which is based on Jacobi iterations, and other called MUGA (cf. TF-1; TF-2) which is based on Gauss-Seidel iterations. We then study in Sect. 2.2 the *smoothing* properties of these two iterative methods, demonstrating that the method of choice for the smoothing part of a multigrid RT code is the GS-based method of TF-2. Sect. 2.3 derives the equation which is at the basis of the multigrid method: the *coarse-grid* equation. The *standard* multigrid method itself is considered in Sect. 2.4, while Sect. 2.5 describes a good stopping criterion for deciding when the standard multigrid iterative process can be terminated.

2.1. The multilevel transfer problem

The multilevel non-LTE transfer problem is the simultaneous solution of the transfer equation for the radiation field and the rate and conservation equations for the level populations (cf. Mihalas, 1978 and Paper I). Because the opacities and emissivities depend on the populations, which, in turn, depend on the radiative rates, the problem is both strongly non-linear and non-local.

The first step in its solution is the discretization of the spatial dependence. In fact, in this paper we use a set of grids, whose spatial resolution is indicated by the level index, l, and the convention that the larger the positive integer number l the *finer* the grid. We replace the continuous variation of quantities by a discrete set of values at NP_l grid points, and all equations by their discrete approximations.

There are two sets of equations in the resultant system: those for the radiation field, and those for the populations. The population equations are deceptively simple in form,

$$\mathscr{L}_l \cdot \mathbf{n}_l = \mathbf{f}_l, \tag{1}$$

where $\mathscr{L}_l$ is a block-diagonal matrix formed by NP_l submatrices, each NL×NL in size. NL is the number of atomic levels being treated in the vector $\mathbf{n}_l$, which like the *known* vector $\mathbf{f}_l$ is of length $\mathrm{NL}\times\mathrm{NP}_l$. The coefficients in Eq.(1) depend on the collision rates and thermodynamic variables, which are assumed to be known *a priori*, and the radiative rates, whose self-consistent values are being sought. Because the operator $\mathscr{L}_l$ of Eq. (1) is *block*-diagonal it would appear that populations at each point are independent. This, of course, is fundamentally incorrect. The values at all points in the grid are coupled non-linearly by the radiation field. Because of this inherent non-linearity, the solution of the multilevel non-LTE problem requires an iterative method capable of finding the populations $\mathbf{n}_l$ such that Eq. (1) is satisfied when the radiative rates, which appear in the *block*-diagonal matrix $\mathscr{L}_l$, are calculated from those populations via the solution of the RT equation.

The basic difference among the various strategies that have been proposed lies in the way one constructs, at each iterative step, a linear set of equations whose solution leads to correction of the current estimates. The simplest procedure one might think of is $\Lambda-$iteration: Using the current estimate $\mathbf{n}^{old}$, evaluate the opacities and source functions, solve the transfer equation and compute the radiation field in all transitions; then, with the radiative rates obtained, the resulting linear system would be

$$\mathscr{L}_l{}^{old} \cdot \mathbf{n}_l{}^{new} = \mathbf{f}_l, \tag{2}$$

where $\mathscr{L}_l{}^{old}$ is a $\mathrm{NP}_l\times\mathrm{NP}_l$ block-*diagonal* matrix. Unfortunately, as is well-known (cf. Mihalas, 1978), this simple $\Lambda-$iteration scheme has very poor convergence properties, and is useless for solving optically thick non-LTE problems.

A successful multilevel iterative scheme for the self-consistent solution of Eqs. (1) is the one we applied in Paper I. That method is based on the preconditioning strategy of Rybicki & Hummer (1991, 1992), on a local *approximate* operator given by the *diagonal* of the full $\Lambda-$operator, and on the short-characteristics formal solution technique (cf. Kunasz & Auer, 1988; Auer & Paletou, 1994; Paper I). In this method, the linear system one solves in order to obtain the "improved" estimate of the atomic level populations is

$$\mathscr{J}_l \cdot \mathbf{n}_l{}^{new} = \mathbf{f}_l, \tag{3}$$

where $\mathscr{J}_l$ like $\mathscr{L}_l{}^{old}$ is a block-*diagonal* matrix whose elements are obtained via the formal solutions of the transfer equation using opacities and source functions calculated directly from the population numbers $\mathbf{n}^{old}$ of the previous iteration. This particular multilevel accelerated $\Lambda-$iteration scheme (Rybicki & Hummer 1991; 1992; see also Paper I) may be considered a generalization to the *non-linear* multilevel atom case of the Jacobi-based method which OAB developed for the *linear* two-level atom case. As we shall show below the *smoothing* capabilities of this multilevel Jacobi scheme (hereafter "the MALI scheme") are *not* sufficient to produce a powerful multigrid RT method.

A superior multilevel iterative scheme was developed by TF-2, and may be considered a generalization to the *non-linear* multilevel atom case of the Gauss-Seidel RT method which TF-1 presented for the *linear* two-level atom case. At each iterative step the linear system one has to solve to obtain a "new" estimate of the atomic level populations is as simple as above:

$$\mathscr{G}_l \cdot \mathbf{n}_l{}^{new} = \mathbf{f}_l, \tag{4}$$

since $\mathscr{G}_l$ is also a block-*diagonal* matrix, but where the elements of the *block* $\mathscr{G}_l^{\,j}$ -corresponding to the spatial grid point j- are obtained via formal solutions using opacities and source functions calculated from the "new" population numbers (already available at the points $j+1$, $j+2$, ...,NP) and from the "old" populations associated to the remaining points for which the corrections have not yet been computed (see TF-1 and TF-2 for details). As we shall illustrate below, the *smoothing* capabilities of this multilevel Gauss-Seidel method (hereafter "the MUGA scheme") are *very* good and yield a powerful multigrid RT method.

We point out that, in the three methods summarized above, the calculation at the current iterative step of the estimate, $\mathbf{n}^{new}$, requires only the inversion of NP_l *independent* blocks, i.e. one point after the other. The computing time per iteration is indeed virtually the same for these three methods, but the number of iterations required to reach convergence is *at least* a factor 2 *larger* for MALI than for MUGA (see TF-2). Therefore, since the MUGA scheme of TF-2 is faster than the MALI method of Paper I, we shall always compare the performance of the non-linear multigrid RT method developed in this paper with that of the MUGA scheme.

Before examining the smoothing properties of the MALI and MUGA methods we should mention that our formal solvers for 1D, 2D and 3D multilevel transfer applications are based on the short-characteristics (SC) technique (cf. Kunasz & Auer, 1988; Auer & Paletou, 1994; Paper I). The TF-1 and TF-2 papers show the basic features of the formal solution solvers that we had to develop for performing the GS-based MUGA iterations. Our multidimensional formal solvers use periodic boundary conditions, which allow us to treat plasma structures that repeat themselves periodically in the horizontal direction. In this paper we shall show examples of 1D and 2D multilevel calculations with the aim of illustrating the multigrid performance. We shall leave the topic of 3D multilevel transfer for a future publication.

2.2. The smoothing ability of the MALI and MUGA schemes

As with other operator splitting techniques, the convergence rate of the MALI and MUGA methods deteriorates as the resolution level l of the grid is increased; i.e., the spectral radius ρ_l of the amplification matrix for the scheme tends to unity as the grid size is decreased (see Paper I, TF-1, and TF-2). What actually happens in fine grids is that the amplitude of the *low-frequency* Fourier spatial-components of the error are only slightly reduced in each iteration, despite the very effective reduction of the amplitude of the *high-frequency* components.

In order to illustrate this behaviour we shall now show some results of multilevel calculations performed with the MALI and MUGA schemes. We use the same standard 5-level Ca II atomic model and the same 1D atmospheric model described in Paper I. The initialization of the atomic level populations in a given grid of level l was a rapidly-fluctuating function that we created by applying to the LTE population levels a sinusoidal fluctuation with a wavelength of four grid points. This was done in order to simulate an initial error characterized by high spatial-frequency components. The solid line of Fig. 1 shows, for the fourth level of Ca II, the error in the departure coefficient for this initialization, and the dashed line shows the corresponding *low-frequency* error. That *low-frequency* component of the error, $\mathbf{F}[\mathbf{e}]$, is calculated by applying to the total error $\mathbf{e}$ a spatial smoothing filter $\mathbf{F}$. Likewise, at each iterative step "itr", we find the *high-frequency* error component by

$$\mathbf{h}_l(itr) = \mathbf{e}_l(itr) - \mathbf{F}[\mathbf{e}_l(itr)] \ , \tag{5}$$

where

$$\mathbf{e}_l(itr) = \mathbf{n}_l(itr) - \mathbf{n}_l(itr = \infty) \tag{6}$$

is the full error of the current population numbers with respect to the fully converged solution.

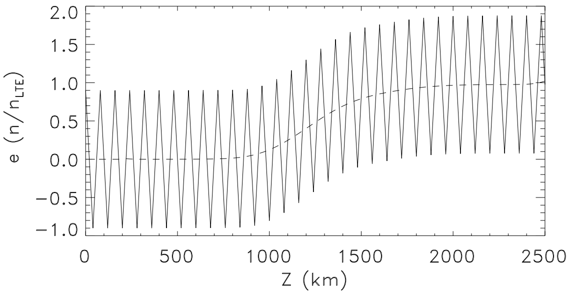

Fig. 1. The solid line gives the total error (Eq. 6) in the departure coefficient of the fourth atomic level of Ca II corresponding to our highly-fluctuating initialization. The dashed-line shows the low-frequency error component. Grid Δz_l=20 km.

In order to study the convergence properties of all these iterative methods we will use the same quantities introduced in Paper I: the relative *change* R_c, the relative *convergence error* C_e, and the relative *true error* T_e. For example, the relative convergence error is defined as

$$\mathrm{C}_e(itr, l) = \left|\left| \frac{\mathbf{e}_l(itr)}{\mathbf{n}_l(itr = \infty)} \right|\right|_\alpha , \tag{7}$$

where $||.||_\alpha$ is the $\alpha-$norm, and the expression "$\frac{\mathbf{e}_l(itr)}{\mathbf{n}_l(itr=\infty)}$" is to be understood as the *vector* whose j-component is equal to the j-component of vector $\mathbf{e}_l(itr)$ divided by the j-component of the population vector $\mathbf{n}_l(itr = \infty)$. For instance, the expression for the convergence error C_e used in Paper I is nothing but Eq. (7) with $\alpha = \infty$, since the $\infty-$norm is defined as the maximum absolute value of such *vector* components. Thus, in Paper I, our choices for R_c, C_e and T_e were the ∞-norm of the bracketed quantities there. However, one is free to make a study of the convergence properties of an iterative method by choosing a different norm, like e.g. the Euclidean norm, which is also called the 2-norm. The 2-norm of a general vector $\mathbf{x}$ of dimension D is given by

$$||\mathbf{x}||_2 = \frac{1}{D}\sqrt{\sum_{j=1}^{D} |x_j|^2}. \tag{8}$$

Note that, in our multilevel RT context, the dimension D=NP$_l\times$NL. The 2-norm seems to be a more standard choice among mathematicians, and we have used it throughout the main body of this paper for investigating the mathematical properties of our non-linear multigrid RT methods. We point out, however, that all the conclusions of this paper remained basically the same when using the ∞-norm instead of the 2-norm.

Fig. 2 shows the convergence error $\mathrm{C}_e(itr, l)$ corresponding to both the full error $\mathbf{e}$ and to the *high-frequency* error component $\mathbf{h}$. First, note that the MUGA scheme of TF-2 has, in general, better convergence properties than the MALI scheme of Paper I. Note also that while both methods are very efficient in reducing the *high-frequency* error components in the *fine* grid, only the MUGA scheme also has this behaviour on the *coarse* grid. The

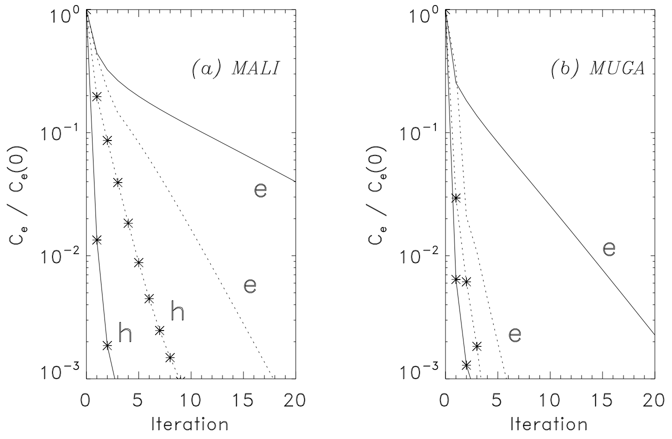

Fig. 2a and b. The variation with the iteration number of the total convergence error (cf. Eq. 6) and of the high-frequency component of the error (cf. Eq. 5). The solid lines refer to a 1D grid with Δz = 20 km, while the dotted ones to Δz = 70 km. The results of **a** where obtained with the MALI method of Paper I, while those of **b** with the MUGA method of TF-2.

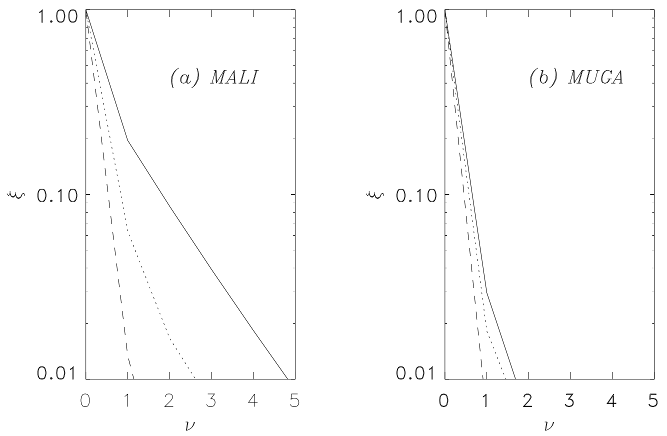

Fig. 3a and b. The variation of the smoothing number against the number (ν) of smoothing iterations: **a** for the MALI scheme and **b** for the MUGA method. Solid lines refer to Δz = 70 km, dotted lines to Δz = 50 km, while dashed-lines to Δz = 20 km.

number of iterations one has to perform with MALI to reduce the high-frequency error components is generally larger than with MUGA.

A more appropriate demonstration of the *smoothing* ability of these two multilevel schemes is given in Fig. 3. This figure shows, for three grids of different resolution levels, the variation with the iteration number of a *smoothing number* (ξ_l) which quantifies the *smoothness* of the error vector $\mathbf{e}_l$ (Hackbush, 1985). This *smoothness* is determined by the relative size of the total error and its high-frequency component. Our choice for the *smoothing number* is

$$\xi_l(\nu) = \frac{D_l[\,\bar{\mathbf{e}}_l\,]}{D_l[\,\mathbf{e}_l\,]} \qquad \text{with} \quad D_l[\,\mathbf{e}_l\,] = \left|\left| \frac{d^2\mathbf{e}_l}{dz^2} \right|\right|_2 , \tag{9}$$

where $\bar{\mathbf{e}}_l = \mathscr{S}_l^{\,\nu}[n_l(0)] - n_l(\infty)$ and $\mathscr{S}_l^{\,\nu}[n_l(0)] = n_l(itr=\nu)$, i.e. the symbol $\mathscr{S}_l^{\,\nu}$ means that ν *smoothing* iterations have been applied to the initialization using either the MALI or the MUGA schemes.

As indicated by Eq. (9), the operator D_l is the 2-norm of the second-order spatial derivative. The smoothing number given by Eq. (9) thus provides a measure of the *smoothness* of the error vector $\mathbf{e}_l$ after having carried out ν iterations with the MALI or MUGA schemes. No smoothing (e.g. for ν = 0) is indicated by ξ_l=1, while very low values of ξ_l imply a high degree of smoothing. Therefore, Fig. 3 shows that the smoothing capabilities of the MALI scheme drastically deteriorate when going from fine to coarse grids, while the MUGA scheme of TF-2 does indeed perform an excellent smoothing job for all the grid resolution levels.

Having compared the smoothing capabilities of these two methods, we conclude that the MUGA scheme (cf. TF-2) is the method of choice for the required *smoothing* iterations in the multigrid non-LTE method.

2.3. The coarse-grid equation

The above presentation was aimed not only to compare the *smoothing* capabilities of the MALI and MUGA schemes, but also to emphasize that the coarser the grid the faster the convergence of the *low-frequency* components of the error. Multigrid methods are based precisely on this last observation. One seeks a coarse-grid equation whose solution permits the calculation of a coarse-grid correction which, once it is interpolated to the *fine* grid, provides the fine-grid correction to the current fine-grid estimate. By doing this one aims at improving the convergence of the low-frequency components of the fine-grid error.

Suppose one has a fine-grid estimate ${\mathbf{n}_l}^{old}$ such that its residual

$$\mathbf{r}_l = \mathbf{f}_l - \mathscr{L}_l^{old} \cdot {\mathbf{n}_l}^{old}, \tag{10}$$

is *smooth*. (We have used this particular notation in order to stress that the calculation of the residual just requires the evaluation of the same $\Lambda-$iteration *block*-diagonal matrix as Eq. 2). One would like to obtain a fine-grid correction $\Delta\mathbf{n}_l$ such that the new estimate

$$\mathbf{n}_l^{new} = {\mathbf{n}_l}^{old} + \Delta\mathbf{n}_l \tag{11}$$

satisfies Eq. (1), i.e. such that

$$\mathscr{L}_l \cdot \mathbf{n}_l^{new} \;=\; \mathbf{f}_l \tag{12}$$

Since the current residual is given by Eq. (10) we also have that

$$\mathscr{L}_l \cdot \mathbf{n}_l^{new} - \mathscr{L}_l^{old} \cdot \mathbf{n}_l^{old} \;=\; \mathbf{f}_l - \mathscr{L}_l^{old} \cdot \mathbf{n}_l^{old} \;=\; \mathbf{r}_l \tag{13}$$

The crucial point is the following: Since the residual $\mathbf{r}_l$ is assumed to be *smooth*, we can map the left-hand side of Eq. (13) to a coarser grid of level $l-1$. This leads directly to the following coarse-grid equation:

$$\mathscr{L}_{l-1} \cdot \tilde{\mathbf{n}}_{l-1} \;=\; \mathscr{L}_{l-1}^{old} \cdot \mathscr{R}(\mathbf{n}_l^{old}) + \mathscr{R}(\mathbf{r}_l), \tag{14}$$

where the linear operator $\mathscr{R}$ is a *fine-to-coarse* or *restriction* operator (we point out that if $\mathbf{g}_l$ is a *fine-grid* vector having fine-grid accuracy, then $\mathscr{R}(\mathbf{g}_l)$ is a *coarse-grid* vector whose values still have such fine-grid accuracy).

Note that the *rhs* of the system of Eqs. (14) can be obtained directly and that, therefore, Eqs. (14) are formally identical to the original system of Eqs. (1), but formulated for a grid of level $l-1$. Once one has solved this coarse-grid equation to obtain $\tilde{\mathbf{n}}_{l-1}$ (e.g. by using the fast MUGA scheme of TF-2 or any other multilevel method), the coarse-grid correction (CGC) is simply given by

$$\Delta\mathbf{n}_{l-1} = \tilde{\mathbf{n}}_{l-1} - \mathscr{R}(\mathbf{n}_l^{old}), \tag{15}$$

and

$$\mathbf{n}_l^{new} = \mathbf{n}_l^{old} + \mathscr{P}\left[\tilde{\mathbf{n}}_{l-1} - \mathscr{R}(\mathbf{n}_l^{old})\right], \tag{16}$$

where $\mathscr{P}$ is a *coarse-to-fine* or *prolongation* operator (we point out that if $\mathbf{g}_{l-1}$ is a *coarse-grid* vector having the accuracy provided by a given coarse-grid equation, then $\mathscr{P}(\mathbf{g}_{l-1})$ is a *fine-grid* vector whose values still have the accuracy provided by such a coarse-grid equation). We use cubic centered interpolation for the prolongation and a restriction $\mathscr{R}$ given by the *adjoint* of the linear interpolation (see details in Hackbush, 1985). At the upper and lower boundaries we simply use for $\mathscr{R}$ the trivial injection along the vertical direction, while we take the adjoint of the linear interpolation along the horizontal direction (cf. Hackbush, 1985). We have found optimal behaviour using these choices.

Eq. (14) is the desired *coarse-grid* equation (Brandt, 1977; see also Hackbush, 1985). It is crucial to note that one is solving this equation for $\tilde{\mathbf{n}}_{l-1}$, but that this solution for the populations in the *coarse* grid does not have the accuracy of the coarse-grid solution $\mathbf{n}_{l-1}$, which would be obtained by solving Eq. (1) but in a grid of level $l-1$. What accuracy then is the coarse-grid Eq. (14) for the population numbers $\tilde{\mathbf{n}}_{l-1}$ giving us? To answer this question we note again that the residual is given by Eq. (10) and, thus, rewritten Eq. (14) as follows

$$\mathscr{L}_{l-1}\cdot\tilde{\mathbf{n}}_{l-1} \;=\; \mathscr{R}(\mathbf{f}_l) + \tau_{l-1} \tag{17}$$

where

$$\tau_{l-1} = \mathscr{L}_{l-1}^{old}\cdot\mathscr{R}({\mathbf{n}_l}^{old}) - \mathscr{R}\left[{\mathscr{L}_l}^{old}\cdot{\mathbf{n}_l}^{old}\right] \;. \tag{18}$$

As pointed out nicely by Press *et. al.* (1994) τ_{l-1} is an approximation to the *truncation residual* defined in the *coarse*-grid relative to the *fine* grid, since the exact coarse-grid *truncation residual* would be just that given by Eq. (18) but using the fully converged solution $\mathbf{n}_l$ instead of the *current* estimate $\mathbf{n}_l^{old}$. Therefore, in Eq. (17) one can think of τ_{l-1} as the correction to $\mathscr{R}(\mathbf{f}_l)$ that "forces" the solution of the coarse-grid equation to have the accuracy of the *fine-grid* solution.

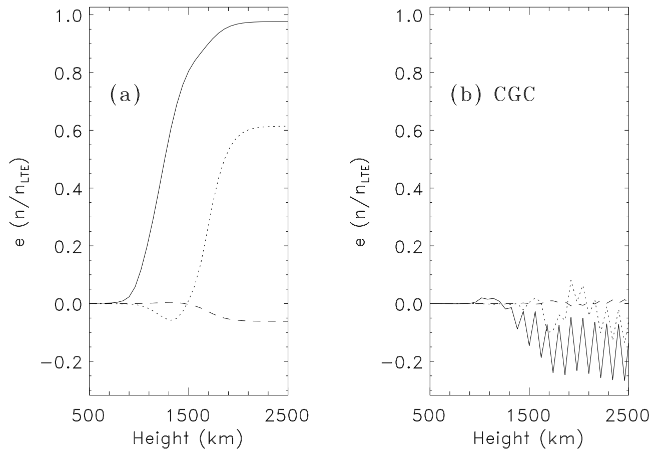

Fig. 4a and b. The error in the departure coefficient before **a** and after **b** a pure CGC (cf. Eq. 16). The dashed-line corresponds to the Ca II ground level, the dotted line to the second level, and the solid line to the fourth one. The error has been calculated with respect to the fully converged solution in the fine grid which has Δz_l = 60 km. The coarse-grid spacing chosen was $\Delta z_{l-1} = 2\Delta z_l$ = 120 km.

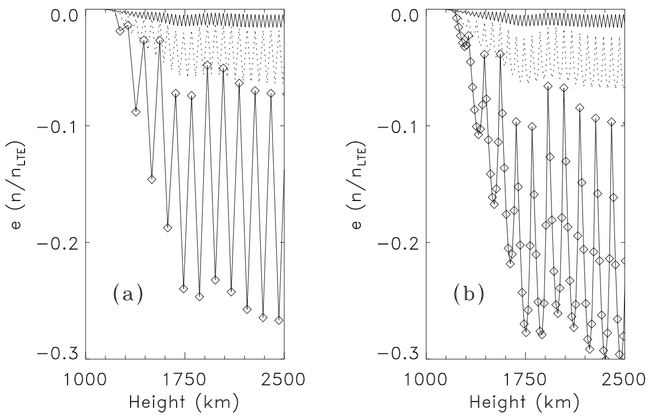

Fig. 5a and b. The error in the departure coefficient of the fourth level of Ca II after a pure CGC (cf. Eq. 16). **a** For cases where the coarse-grid $\Delta z_{l-1} = 2\Delta z_l$, with a fine-grid Δz_l of 15 km (solid line), of 30 km (dotted line) and of 60 km (solid line with squares). **b** For a fixed fine-grid spacing of 15 km, but with coarse-grid Δz_{l-1} equal to 30 km (solid line), 60 km (dotted line) and 120 km (solid line with squares).

Fig. 4 aims at illustrating the efficiency of *one* CGC for an example where the *fine* grid spacing Δz_l = 60 km and the *coarse* one is Δz_{l-1}=120 km. Fig. 4a shows the error in the departure coefficient associated with an LTE initialization for ${\mathbf{n}_l}^{old}$. Fig. 4b gives the error corresponding to the "new" estimate obtained by means of *one* CGC (cf. Eq. 16). Note that only one CGC strongly reduces the total error of the initial estimate, but that the new estimate still has a high-frequency component that is not effectively removed by only a coarse-grid correction.

A further illustration of the role played by the CGC is given in Fig. 5. Here we show the height variation of the error of the departure coefficient in the fourth Ca II atomic level after *one* CGC, but for various two-grid combinations. Fig. 5a shows CGC examples for cases where $\Delta z_{l-1} = 2\Delta z_l$, and we use various fine-grid spacings; note that the smaller the grid-spacing of the *fine* grid the higher the CGC (smaller the remaining error). Fig. 5b shows examples for a fixed *fine*-grid spacing but different coarse-grid spacings; note that the larger the grid spacing of the *coarse* grid the smaller the CGC (the larger the remaining error). By comparing these two figures one can see that it is the coarse grid resolution level that determines the *low-frequency* error component remaining after one CGC.

The above examples show that the CGC is indeed very efficient in reducing the amplitude of the *low-frequency* components of the error. On the contrary, as is seen in Figs. 4 and 5, components of the error with wavelengths smaller than or similar to twice the distance between the coarse grid points remain, because such high-frequency components are not resolved by the coarse grid. For this reason the CGC, by itself, is a *non-convergent* iteration (see Hackbush, 1985 for a formal proof of this observation). What is needed to achieve a true *two-grid* convergent iterative scheme is to apply a number of suitable iterations in the *fine* grid to remove the *high-frequency* components of the error. As advanced earlier the MUGA scheme is the multilevel method of choice for this purpose. We note that, besides being very efficient in reducing the amplitude of the high-frequency error components, it also makes a contribution to the reduction of the low-frequency error components. Thus, the convergence rate of the non-linear *two-grid* iteration is not determined by the CGC alone. As we shall show below, it is actually the combination of smoothing iterations in the fine grid and the CGC that sets the very high convergence rate which is characteristic of the *two-grid* method.

2.4. The standard multigrid method

We may now summarize the steps of our non-linear *two-grid* method for multilevel RT applications as follows:

1. Pre-smoothing: Given the current estimate, perform ν_1 smoothing iterations in the grid l using our MUGA method in order to obtain $\mathbf{n}_l{}^{old}$.
2. Coarse-grid correction step: Obtain $\mathbf{n}_l{}^{new}$ according to Eq. (16). This is done by the following operations:
   - Compute the residual on the fine grid from Eq. (10). (Note that this requires formal solutions of the RT equation in the fine grid of level l in order to find the radiation field corresponding to the populations $\mathbf{n}_l{}^{old}$).
   - Apply the restriction operator to the fine-grid residual to obtain $\mathscr{R}(\mathbf{r}_l)$.
   - Apply the restriction operator to the current estimate to obtain $\mathscr{R}(\mathbf{n}_l{}^{old})$.
   - Compute the first term in the *rhs* of Eq. (14). (Note that this requires formal solutions of the RT equation on the coarse grid in order to calculate the radiation field corresponding to the populations $\mathscr{R}(\mathbf{n}_l{}^{old})$).
   - Use the MUGA method to solve the coarse-grid Eq. (14), and then obtain the correction $\Delta\mathbf{n}_{l-1}$ according to Eq. (15).
   - Apply the prolongation operator to this coarse-grid correction in order to obtain the fine-grid correction and a new estimate as indicated by Eq. (16). (It is very important to note that, as the prolongation to the fine grid of the coarse-grid correction given by Eq. (15) only provides an *approximate* correction, one does not actually need to solve the coarse-grid Eq. (14) exactly. As we shall emphasize below, it is sufficient to iterate using the MUGA method *only* till achieving a reduction of an order of magnitude in the error of the initial estimate used for the solution of Eq. (14)).
3. Post-smoothing: Perform ν_2 smoothing iterations in the grid l using the MUGA scheme. Terminate the multigrid iterative process once a rigorous stopping criterion is satisfied (see Sect. 2.5); otherwise return to point (1).

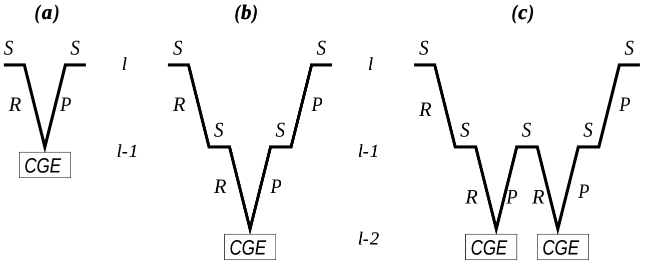

Fig. 6a–c. Schematic visualization of the standard multigrid iterative scheme for some cases: **a** two grids, **b** three grids with $\gamma = 1$, i.e. the V-cycle, **c** three grids with $\gamma = 2$, i.e. the W-cycle. The symbol $\mathscr{S}$ indicates smoothing, $\mathscr{R}$ restriction, and $\mathscr{P}$ prolongation. CGE means "coarse-grid equation".

These steps define a *two-grid* (TG) cycle. In the *non-linear* problem, one is solving the coarse-grid equation for $\tilde{\mathbf{n}}_{l-1}$, and not just for the correction $\Delta\mathbf{n}_{l-1}$ as in the *linear* case (cf. Hackbush, 1985; Steiner, 1991). Further, in the *non-linear* case, the restriction operator $\mathscr{R}$ has to be applied not only to the residual $\mathbf{r}_l$, but also to the current estimate $\mathbf{n}_l{}^{old}$. Besides the additional complexity of having to develop a suitable *non-linear* multilevel scheme (see TF-2) for the smoothing part and for solving rapidly the coarse-grid equation, these are the main differences between a linear and non-linear TG cycle.

A *three-grid* method is obtained if the coarse-grid Eq. (14) is solved by applying a number (γ) of *two-grid* iterations involving grids with resolution levels $l-1$ and $l-2$. A *multigrid* (MG) method is obtained if one applies this idea recursively down to some coarsest grid where the solution can be found easily by using any of the multilevel methods currently in use. Fig. 6 gives the usual pictorial representation of a multigrid *cycle* for three and four grids and for two values of γ.

Concerning the resolutions of the spatial grids, we have found that an excellent choice to get the successively coarser grids is simply to double the vertical (Δz) and horizontal (Δx and Δy) step sizes, i.e. halve the number of grid-points in each direction. This implies that, in 1D, the coarse-grid reduction factor is $\delta = \mathrm{NP}_{l-1}/\mathrm{NP}_l = 1/2$, in 2D $\delta = 1/4$, and in 3D $\delta = 1/8$. As we shall see in more detail below, with the TG method the computational effort required to solve the CGE increases as the spatial resolution of the fine-grid improves; however, the MG method does not suffer from this limitation because as the fine-grid resolution level "l" increases we correspondingly increase

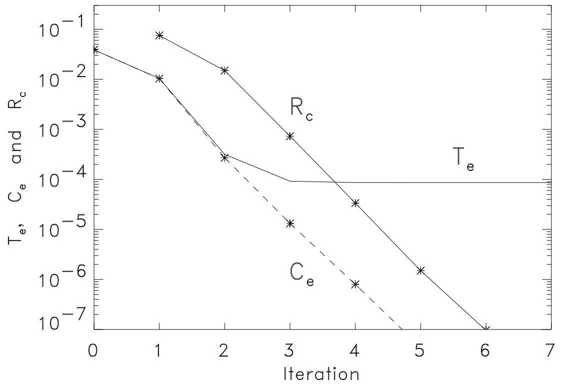

Fig. 7. Convergence properties of the multigrid RT method are demonstrated in this 2D Ca II calculation with a horizontal periodicity of 750 km in the temperature inhomogeneities.

the number (N) of coarse grids, so that the CGE is always solved in the same coarsest grid.

Before studying in detail the convergence rate of the multigrid RT code it is instructive to show an example of its performance. Fig. 7 gives the variation with the iteration number of the *true error* (T$_e$), of the *convergence error* (C$_e$), and of the *relative change* (R$_c$) corresponding to the same 5-level Ca II *two-dimensional* model problem of Paper I, which is characterized by sinusoidal temperature fluctuations along the horizontal direction with an amplitude of 500 K and a horizontal wavelength of 750 km. First, note that our non-linear multigrid method for multilevel RT applications gives the converged solution (i.e. a C$_e$-value smaller than the truncation error T$_e$) in only three iterations. Second, note that the *relative changes* R$_c$ are *smaller* than the convergence error C$_e$. This last feature can be easily understood after recalling, from the appendix of our Paper I, that these quantities are related asymptotically by

$$C_e \approx R_c \frac{\rho}{1-\rho}, \tag{19}$$

where ρ is given by Eq. (22) below. In Fig. 7 C$_e$ < R$_c$ because the multigrid ρ-values are well below 0.5. This is important because other methods currently in use (e.g. with MALI) have C$_e$ >R$_c$. With them, a small value of R$_c$ (which is all that one has available from the iterations themselves) does not necessarily imply a small value of C$_e$ (which is the quantity that has to be sufficiently small to guarantee that convergence has been achieved). The convergence rate of our multigrid RT method, however, is so high that a small value of R$_c$ directly implies an even smaller value of C$_e$.

2.5. A stopping criterion for the standard MG method

We have not yet discussed when the *standard* multigrid process should be terminated. As mentioned in Sect. 2.4 the check for convergence should be made after a number (ν_2) of post-smoothing iterations in the finest grid. In Paper I we emphasized that one should actually terminate the iterations once the convergence error is dominated by the truncation error of the finest-grid selected. A suitable stopping criterion to achieve this goal within the framework of the *standard multigrid method* consists in terminating the iterative process once the α-norm of the current fine-grid *residual* is smaller than the same norm of the fine-grid *truncation residual*, i.e. to stop iterating once

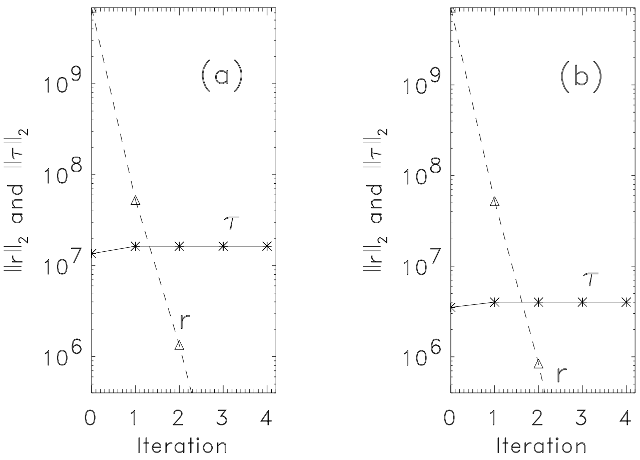

Fig. 8a and b. Illustration of a stopping criterion for the standard multigrid RT method working in practice.

$$||\mathbf{r}_l||_\alpha \;<\; ||\tau_l||_\alpha. \tag{20}$$

The relevant question now is: how can we estimate the fine-grid *truncation residual* τ_l ? Eq. (18) allows one to estimate the *truncation residual* (τ_{l-1}) corresponding to the *coarser* grid of level "$l-1$", but not τ_l since one does not know $\mathbf{n}_{l+1}^{old}$, simply because the *finest* grid-level is "l" and not "$l+1$" To estimate the fine-grid residual τ_l we follow Press *et al.* (1994) and note that for a solution method like ours (which has second-order accuracy and which uses a coarse-grid spacing twice larger than the fine-grid one) one has

$$||\tau_l||_\alpha \sim \frac{1}{3}||\tau_{l-1}||_\alpha. \tag{21}$$

Fig. 8 shows this stopping criterion working in practice. The figures show the variation of the 2-norm of the fine-grid residuals versus the iteration number for two cases: a finest grid with (a) Δz_l = 15 km and (b) Δz_l = 7.5 km. The solid lines give the estimate of the finest-grid truncation residual ($||\tau_l||_\alpha$) at each iteration. The dashed-lines give the variation of the finest grid residual ($||\mathbf{r}_l||_\alpha$) (cf. Eq. 10). By comparing the truncation residual values of Fig. 8a and Fig. 8b one can see that the relation given by Eq. (21) is satisfied. The asymptotic value for $||\tau_l||_\alpha$ is rapidly reached, i.e. the values for $||\tau_{l-1}||_\alpha/3$ from Eq. (18) do not significanlty change after the first iteration. This demonstrates that this stopping criterion is easily applied, and that only 2 *standard* multigrid iterations suffice to reach the true accuracy the chosen grids can provide.

3. The convergence rate

A number of important mathematical theorems that demonstrate the convergence of the *two-grid* and *multigrid* iterative schemes can be found in Hackbush (1985). Our objective here is, first, to establish a practical and accurate way to measure the convergence rate and, then, to investigate how this quantity depends on the number of smoothing iterations, and resolution levels used. The results of this section refer to 1D multilevel calculations for Ca II. We point out that similar results are obtained for 2D grids with both Δx and Δz scaling proportionately when the resolution is changed.

3.1. Measuring the convergence speed

For a calculation performed in a grid of level l, a useful quantity to measure the rate of convergence is (cf. Paper I, TF-1):

$$\rho_l = \lim_{itr\to\infty} \frac{R_c(itr+1)}{R_c(itr)} = \lim_{itr\to\infty} \frac{C_e(itr+1)}{C_e(itr)} \ . \qquad (22)$$

The smaller this number (ρ_l) the faster the method converges, and the smaller the number ($\mathscr{N}$) of iterations required to reduce the error by one order of magnitude. This number, $\mathscr{N}$, is given by (cf. Tf-1):

$$\mathscr{N} = \frac{-1}{\log_{10}(\rho_l)} \ . \qquad (23)$$

When we say that the convergence rate is very high we actually mean that $-\log_{10}(\rho_l)$ is very large. Since the multigrid method converges in very few iterations, it turns out that a more practical and accurate way of estimating the convergence rate is

$$\rho_l = \left[\frac{C_l(iter)}{C_l(2)} \right]^{1/(iter-2)} , \qquad (24)$$

with $iter$ chosen such that $C_l(iter) < 10^{-8}$. Both definitions (cf. Eqs. 22 and 24) give the same result when applied to more slowly convergent methods (such as the MALI and MUGA schemes).

The curves in Fig. 9 were obtained using Eq. (24). Here we show, for the one-dimensional 5-level Ca II reference case of Paper I, the variation of the convergence rate ρ_l with the grid spacing. This figure demonstrates the two attractive properties of multigrid techniques. First, it is clearly seen that, while the convergence of the MALI and MUGA schemes decreases significantly as the spatial resolution level of the grid is increased, the convergence rate of the *two-grid* RT iteration method remains roughly constant. Second, the ρ_l-values of our RT two-grid scheme are always very low (e.g., for the Ca II problem $\rho_l < 0.1$ for all resolution levels l), which indicates (cf. Eq. 23) that only one iteration suffices to reduce the error by one order of magnitude, i.e. that convergence is reached in very few iterations. Of course, as the grid is refined, the effort needed to solve for the coarse-grid correction increases dramatically. This is not reflected in the value, ρ, which only inform us about the number of two-grid iterations required to reach convergence, and not about the computational work per iteration. In the next section we will consider in detail the true efficiencies of the two-grid and multigrid iterations comparing them with the one of the MUGA method.

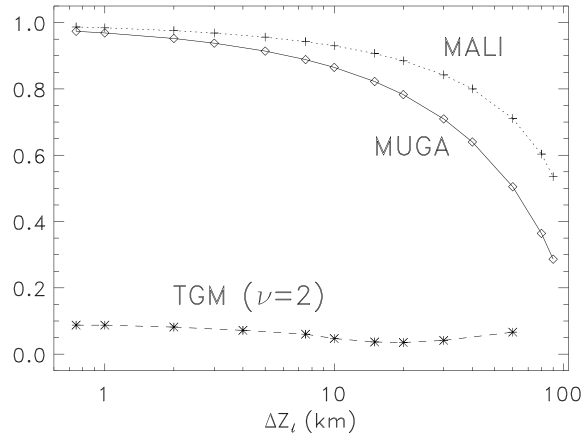

Fig. 9. The variation of the convergence rate (cf. Eq. 24) with the spatial resolution level of the grid. The methods used to obtain the solution to the Ca II multilevel 1D problem referred to in the text are the MALI scheme of Paper I, the MUGA method of TF-2 and the two-grid non-linear method with $\nu = 2$ and $\delta = \Delta z_l/\Delta z_{l-1} = 1/2$.

It is important to point out that in 1D we always choose $\delta = \frac{\Delta z_l}{\Delta z_{l-1}} = 1/2$. Although it is true that a smaller δ-factor saves some computational work when solving the coarse-grid equation (14), the number of smoothing iterations required at the fine-grid turns out to be larger and the total computational work is not improved. This is due to the fact that, as seen in Fig. 5b, the wavelength of the high-frequency error in the fine grid is inversely proportional to δ; therefore, the smaller δ the larger the wavelength of the high-frequency error, and the larger the number of smoothing iterations required to remove it.

3.2. The influence of the smoothing number

As shown in Fig. 3b the larger the number, ν, of MUGA iterations in the fine grid the *smaller* the amplitude of the high-frequency spatial components of the current error and residual. Although the smoothing capabilities of our MUGA scheme are very good for all grid resolution levels, the finer the grid the easier it is for MUGA to achieve a given level of error reduction (cf. Fig. 3b). Therefore, one expects that the ρ_l-values decrease as the smoothing number (ν) increases, and in a way that depends on the finest-grid resolution level.

Fig. 10 gives useful information on this point. Besides showing that the convergence rate improves when ν is increased, it demonstrates that the sensitivity of ρ_l to the smoothing number ν is very small for fine grids (i.e. for $\Delta z_l < 10$ km), but significant in coarser grids. For coarser grid cases, we get a sizable decrease of ρ_l with ν. In fact, $\rho_l \propto 1/(1+\nu)$, but note that $\nu = 2$ already yields $\rho_l < 0.1$ for all grid levels. In the next section we shall show that with $\nu = 2$ one achieves optimal efficiency in the multigrid method.

The above examples are for two grids. When more than two grids are used, the smoothing steps are divided among smooth-

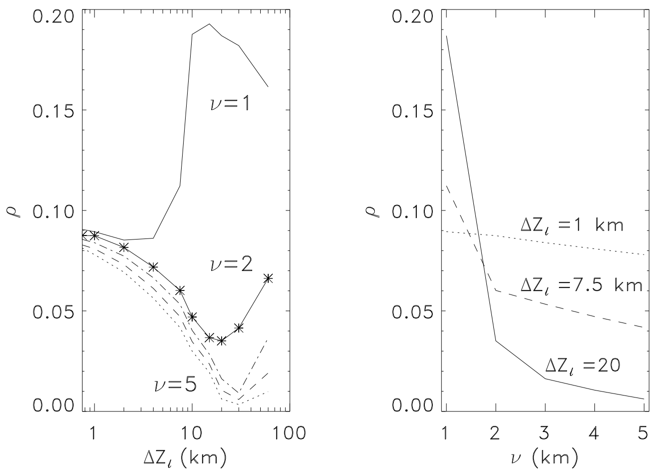

Fig. 10. a The convergence rate of the TG method versus the fine-grid resolution level for various values of the number (ν) of smoothing steps. **b** The TG convergence rate versus the number (ν) of smoothing iterations for three values of the fine-grid spacing Δz_l.

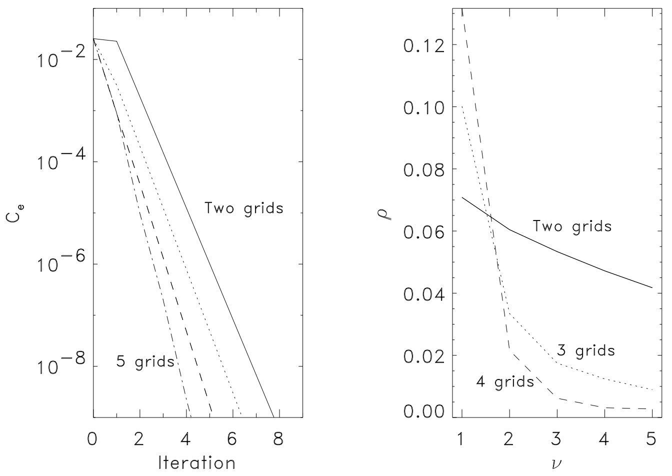

Fig. 11. a The convergence error of the MG method using 2, 3, 4 and 5 grids (Δz_l=2 km and ν=2) **b** The convergence rate ρ_l versus the number of smoothing iterations for three choices of the number of grids, but using always Δz_l=7.5 km.

ings before (ν_1) and after (ν_2) each CGC. We have found $\nu_2 = 0$ can impair convergence if many coarse grids are used, while $\nu_1 = 0$ can lead to a bad behaviour in the first iterations. The best choice is to use $\nu_2 \simeq \nu_1$ and, if $\nu = \nu_1 + \nu_2 = 1$, then to make it after the prolongation (i.e. $\nu_1 = 0$ and $\nu_2 = 1$). In what follows, when we say that $\nu = 2$ we will be actually indicating that $\nu_1 = \nu_2 = 1$.

3.3. The influence of the number of coarse grids

Finally, the convergence rate also depends on the number of coarse grids used. In Fig. 11a we show the variation of the convergence error with the iteration number for multigrid V-cycle iterations, for Δz_l=2 km and using two, three or more grids. Fig. 11b shows the variation of ρ_l with ν for a case with Δz_l = 7.5 km. Note also that the sensitivity of ρ to ν for a given finest grid resolution level depends on the number of coarse grids, i.e. it increases when we use more coarse grids. Note that when $\nu > 2$, the convergence speed depends much more strongly on the number of grids used than on ν.

Finally, it is worth mentioning that the multigrid convergence rate is slightly better for W-cycles than for V-cycles when one uses a large number of coarse grids. However, performing W-cycles requires more computational work, and we have not found any significant improvement from the use of W-cycles instead of V-cycles.

4. Efficiency

The analysis of the previous section has shown that the convergence rate of the multigrid method is very high and does not deteriorate when the discretization is refined; however, it says nothing about the efficiency of the multigrid method compared to the one of the MALI or MUGA schemes. Each multigrid cycle requires more computation than a single MUGA iteration. As we shall now show the rapid convergence of the multigrid scheme more than compensates for this increased cost per iteration.

We can define the "*cost*" of a numerical method as the CPU time needed to reduce the error by one order of magnitude. The corresponding "*efficiency*" is inversely proportional to this "*cost*". Since the number of iterations required to achieve a factor of ten improvement is given by Eq. (23), the cost measured in arbitrary units is just

$$\Re_l = \frac{time}{-\log_{10}(\rho_l)}, \tag{25}$$

where $time$ is the CPU-time required by one iteration.

A relatively small value of $\Re_l$ makes the efficiency of the method high. The ratio of the costs of two different iterative methods measures their relative merit. In what follows we will first consider the 1D case, and afterwards the multidimensional one. In each case we will analyze the TG and MG efficiencies.

4.1. The MUGA method

As with the Jacobi-based MALI method of Paper I, each MUGA iteration requires the formal solution of the RT equations and the inversion of the NP$_l$ *blocks* of dimension NL×NL. For not too large values of NL the formal solution time dominates and we can neglect the computational effort required for performing such inversions. Since our formal solvers are based on the short-characteristics technique (cf. Paper I) the computing time per MUGA iteration will be, either in 1D, 2D or 3D, directly proportional to the number of spatial grid points, NP$_l$. Thus, the cost of the GS-based MUGA method of TF-2 (measured in arbitrary units) is approximately given by

$$\Re_l(\mathrm{MUGA}) \approx \frac{\mathrm{NP}_l}{-\log_{10}(\rho_l^{\mathrm{MUGA}})} \tag{26}$$

From Fig. 9 we know that as the spatial resolution of the grid is improved $\rho_l^{\rm MUGA} \rightarrow 1$. It is this deterioration of the convergence rate, rather than the increased formal solution time that dominates the cost of MUGA iterations on fine grids. In fact, this same drawback is characteristic of all operator splitting methods based on approximate $\Lambda-$operators which are constructed as a direct *simplification* of the full $\Lambda-$operator corresponding to a single grid.

4.2. Two-grid method

The computational work done in a *two-grid* iteration is composed of the following contributions:

- On the finest grid (l) one performs "ν" smoothing iterations using the MUGA method, and one formal solution for the radiation field in order to calculate the residual for the current estimate. Each MUGA iteration requires one full formal solution and the inversion of NP_l blocks of dimension $\mathrm{NL}\times\mathrm{NL}$. For not too large values of NL we can neglect the computing time required to invert these blocks; thus, the computing time of a *two-grid* iteration will be

  $$tsm_l \approx (1+\nu)\mathrm{NP}_l. \tag{27}$$

  We shall refer to this *fine*-grid computing time as the "*smoothing time*".
- In the coarse grid, with resolution level ($l-1$), one uses MUGA iteration to solve the coarse-grid Eq. (14) plus one formal solution to calculate the *rhs* of this equation. As pointed out above, for this coarse grid we only need an approximate solution; in fact, experience has shown it is sufficient to reduce the error of the initial estimate only by an order of magnitude. The computing time corresponding to the work done in the coarse grid is the time needed to reduce the error in this grid by this factor, as defined by Eq. (26), but for a grid of level "$l-1$", i.e.

  $$tcs_l \approx \Re_{l-1}(\mathrm{MUGA}) = \frac{\delta{\cdot}\mathrm{NP}_l}{-\log_{10}(\rho_{l-1}^{\rm MUGA})}, \tag{28}$$

  where δ is the coarse-grid reduction factor ($\delta = \mathrm{NP}_{l-1}/\mathrm{NP}_l = 1/2$ in 1D). We shall refer to this time simply as the *coarse solution time*.

Accordingly (cf. Eq. 25), the cost of the TG iterative scheme is

$$\Re_l(\mathrm{TG}) \approx \frac{1}{-\log_{10}(\rho_l^{\rm TG})}\left(tsm_l + tcs\right) =$$

$$= \frac{\mathrm{NP}_l}{-\log_{10}(\rho_l^{\rm TG})}\left[(1+\nu) + \frac{\delta}{-\log_{10}(\rho_{l-1}^{\rm MUGA})}\right] \tag{29}$$

Fig. 12a compares the cost of the *two-grid* and MUGA methods (Eqs. 29 and 26) for a range of spatial grids. These results are for the Ca II multilevel 1D problem of Paper I. The TG method was applied using always two smoothing iterations (i.e. $\nu = 2$). The figure shows that TG method is more efficient than the MUGA scheme, especially on very fine grids (i.e., for $\Delta z_l < 10$ km).

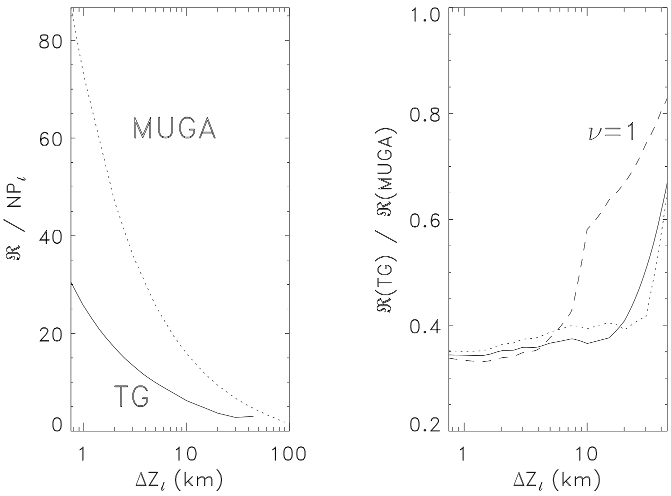

Fig. 12. a The cost of the MUGA and TG (with $\nu = 2$) methods as a function of the fine-grid spacing. **b** The ratio of the efficiencies of both methods versus the fine-grid spacing for different $\nu-$values: dashed-line ($\nu = 1$), solid-line ($\nu = 2$), and dotted-line ($\nu = 3$).

Fig. 12b shows the ratio of the TG to MUGA efficiencies as a function of the spatial resolution of the grid. In coarse grids (i.e. for $\Delta z_l > 10$ km) the TG method does not lead to a significant CPU-time saving with respect to the MUGA method. In fact, the MUGA method (cf. TF-2) is definitely superior for these 1D grids because in practice MUGA is always combined with acceleration techniques that enhance the performance of the MUGA method by *at least* a factor 2 (see TF-2). The relative efficiency of TG iteration improves as the spatial resolution is refined, but as shown in Fig. 11b that ratio tends to a constant value implying that in 1D the CPU-time saving with respect to the pure MUGA method (i.e. without combining it with acceleration techniques) is at most a factor 3.

The previous results can be understood after analyzing in more depth Eq. (29). The "smoothing time" (cf. Eq. 27) will dominate if $\Re_{l-1}(\mathrm{MUGA})$ (cf. Eq. 28) is small enough that

$$\frac{1+\nu}{\delta} >> \frac{1}{-log_{10}\rho_{l-1}^{\rm MUGA}}$$

This occurs when the resolution needed in the fine grid is low. Taking $\nu = 2$ and $\delta = 1/2$, this inequality is well satisfied for grids with Δz_l such that $\rho_{l-1}^{\rm MUGA}(\Delta z_{l-1} = 2\Delta z_l) < 0.6$. For the Ca II model problem we are considering this relation is satisfied for grids with $\Delta z_l > 10$ km (cf. Fig. 9). Noting that $\rho_l^{\rm TG} \sim 0.1$, we find that, in this limit, the cost of the TG method is $\Re(\mathrm{TG}) \sim (1+\nu)\mathrm{NP}_l$. This value is similar to the MUGA cost (cf. Fig. 12a) so one does not save a significant CPU-time by using TG instead of MUGA for these 1D grids.

In *fine* grids, however, most of the time is used in solving the error equation and the relative merit of the TG method with respect to our MUGA method is

$$\frac{\Re_l(\mathrm{TG})}{\Re_l(\mathrm{MUGA})} \simeq \frac{\delta}{-\log_{10}(\rho_l^{\rm TG})}\left[\frac{\log_{10}(\rho_l^{\rm MUGA})}{\log_{10}(\rho_{l-1}^{\rm MUGA})}\right] \tag{30}$$

Table 1. The cost of the TG method (cf. Eq. 29) and of the MUGA method (cf. Eq. 26) for different values of ν and Δz_l.

Δz_l	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	MUGA
4 km	10	8	7	8	8	13
3 km	16	11	9	10	10	22
1 km	92	58	63	67	70	159
0.4 km	259	262	276	288	299	729
0.1 km	2430	2500	2550	2610	2660	7280

From Fig. 9 it can be deduced that, as the spatial resolution of the fine grid is increased, $\log_{10}(\rho_l^{\rm MUGA}) \,/\, \log_{10}(\rho_{l-1}^{\rm MUGA})$ tends to a constant value (0.6 in this Ca II case). Therefore, the CPU time one can save in 1D calculations by using the TG method tends to a constant value equal to 0.3, which agrees with the results plotted in Fig. 12b.

We now consider the dependence of the TG efficiency on ν. Eq. (29) allows one to determine the optimum ν value. In the previous section we pointed out that ρ_l in the TG method decreases with increasing ν (see Fig. 10b), but that the variation will only be significant in coarse grids, where the smoothing capability of the GS-based MUGA method is not as great as in fine grids. In the coarse-grid regime (i.e. for $\Delta z_l > 10$ km) the dependence of $\rho_l^{\rm TG}$ on the number of smoothing iterations is proportional to $1/(1+\nu)$ (see Fig. 10b), and the TG *cost* is

$$\Re_l({\rm TG}) \propto \frac{1+\nu}{-log_{10}\left[\frac{1}{1+\nu}\right]}$$

By minimizing this function with respect to ν one finds that it has a minimum at $\nu = 3$ and strongly increases for $\nu > 3$. In Table 1 one can see that the optimum value of ν will indeed be 3 in the coarsest grids, but that it decreases to ν=1 in the finest grids where the CPU time is dominated by the *coarse grid solution time*. In this fine-grid regime, where the dependence of ρ on ν is very small, the efficiency does not depend too much on ν. We have found, in fact, the use of $\nu = 2$ gives nearly optimal behaviour for all the grid levels.

4.3. Multigrid method

As in the two-grid method the time required to perform a multigrid iteration is given by the *smoothing* time plus the *coarse* time, i.e. $time_l({\rm MG}) = tsm_l + tcs_l$. For the MG method one has more than two grids, and one must make a computational effort equivalent to $(1+\nu)$ MUGA iterations in all the grids except the coarsest. The smoothing time is, thus,

$$tsm_l \approx {\rm NP}_l \; (1+\nu) \sum_{k=0}^{N-2} \delta^k \;, \tag{31}$$

where N is the number of grids used (e.g. N=2 in the TG method).

The *coarse solution time* is the time one needs to solve Eq. (14) in the coarsest grid, of level $l - N + 1$. Using the MUGA method to reduce the error one order of magnitude in this grid, the *coarse solution time* is approximately given by

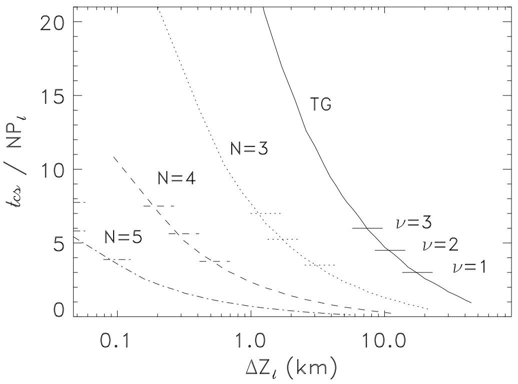

Fig. 13. The coarsest solution time (cf. Eq. 32) as a function of the spatial resolution of the finest grid for the 1D Ca II problem. Each labeled curve corresponds to a different total number of grids chosen. The horizontal lines show the smoothing time (cf. Eq. 31) associated with different choices of ν. Since NP$_l$ appears as a proportionality factor in Eqs. (31) and (32), the times given in this figure have been multiplied by $1/{\rm NP}_l$.

$$tcs_l \approx \Re_{l-N+1}({\rm MUGA}) \;=\; \frac{{\rm NP}_l \;\; \delta^{N-1}}{-\log_{10}(\rho_{l-N+1}^{\rm MUGA})} \tag{32}$$

Adding both times, the total cost of the *multigrid* scheme is

$$\Re_l \approx \frac{{\rm NP}_l}{-log_{10}(\rho_l^{\rm MG})} \cdot \left[(1+\nu) \sum_{k=0}^{N-2} \delta^k + \frac{\delta^{N-1}}{-\log_{10}(\rho_{l-N+1}^{\rm MUGA})} \right] \tag{33}$$

We can now compare the relative importance of the smoothing and coarse times in the Ca II reference 1D problem. In order to obtain the coarsest solution time tcs (Eq. 32) we use the values of $\rho^{\rm MUGA}$ shown in Fig. 9. In Fig. 13 one sees that in very fine grids the coarsest time tcs is larger than the smoothing time tsm, independently of the number of grids chosen, while in coarse grids the CPU time of one MG iteration is dominated by the smoothing time. It is also worth noting that the Δz_l values such that $tcs << tsm$ depend on the number of grids. With $N = 2$ (see the TG discussion in the previous section) the tcs time dominates for grids having $\Delta z_l < 10$ km; however, with $N = 4$ and $\nu = 2$ this only occurs for $\Delta z_l < 0.3$ km.

Therefore, in practice, when one uses four or more grids, the time needed to solve the coarse grid Eq. (14) is always smaller that the smoothing time, and we can approach the MG cost by

$$\Re_l({\rm MG}) \simeq {\rm NP}_l \; \frac{(1+\nu) \sum_{k=0}^{N-2} \delta^k}{-\log_{10}(\rho_l^{\rm MG})} \;, \tag{34}$$

which demonstrates that the MG cost *is simply proportional to* NP$_l$. Note that, if we use a high number of grids, $\sum_k \delta^k \to 2$ in this 1D case. For the Ca II problem we have found (with $\nu = 2$)

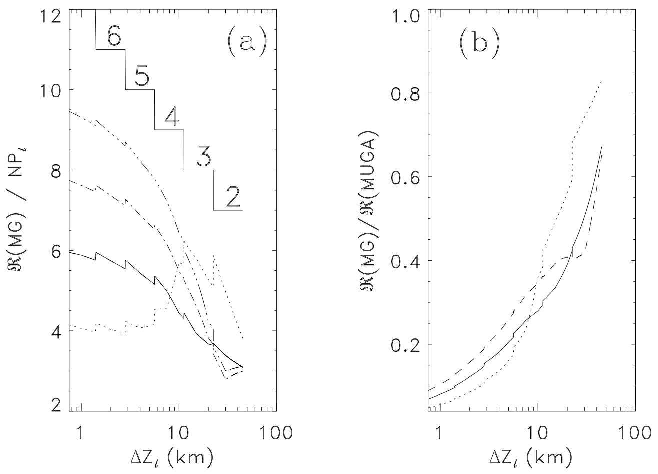

Fig. 14a and b. The cost of the multigrid (MG) method (**a**) and the ratio of the costs of the MG and MUGA methods (**b**) versus the fine-grid resolution level. Dotted line (ν = 1), solid line (ν = 2), dashed-dotted line (ν = 3) and dashed-double-dotted line (ν = 4). The numbers on the steps drawn in the upper part of **a** refer to the number (N) of grids used for each finest-grid resolution level. N was chosen such that in the coarsest grid we had at least one point per density scale height.

that $\rho_l^{\rm MG} < 0.1$ for all resolution levels "l". Therefore, we have as an upper limit

$$\Re_l({\rm MG}) \; < \; 2\,(1+\nu)\,{\rm NP}_l\,, \tag{35}$$

which is an inequality valid even for very small values of Δz_l, but only for calculations using a large number of grids.

Accordingly, the time improvement of the MG method with respect to MUGA (cf. Eqs. 34 and 26) is given by

$$\frac{\Re_l({\rm MG})}{\Re_l({\rm MUGA})} \;\simeq\; (1+\nu)\sum_{k=0}^{N-2}\delta^k \cdot \frac{-\log_{10}(\rho_l^{\rm MUGA})}{-\log_{10}(\rho_l^{\rm MG})} \tag{36}$$

Since the dependence of $\rho_l^{\rm MG}$ with the spatial resolution of the finest grid is very small, the CPU time one can save with the MG method is proportional to $log_{10}(\rho_l^{\rm MUGA})$ (i.e. inversely proportional to the number of iterations needed to solve the problem with the MUGA method). This conclusion reveals the main advantage of the MG method with respect to other iterative methods: the MG efficiency is limited only by the CPU time needed to make a single *formal solution* of the RT equation, but not at all by the number of iterations required to achieve the correct converged solution, since this number will be always very small, and basically independent of the spatial resolution required for solving a given problem. For the Ca II multilevel transfer problem considered here $\rho_l^{\rm MG}{\approx}0.1$ (see Fig. 9); therefore, when many grids are being used

$$\frac{\Re_l({\rm MG})}{\Re_l({\rm MUGA})} \;<\; -\log_{10}(\rho_l^{\rm MUGA})\cdot 2(1+\nu).$$

Fig. 14 (with the detailed description given in its caption) clearly illustrates what we have just discussed. In addition, it also shows that, at it was also the case for the TG method, an optimum behaviour is always obtained by taking ν = 2 smoothing iterations.

We end this section by pointing out that our MUGA method combined with standard acceleration techniques and/or our SOR method (see TF-1 and TF-2) are indeed capable of solving rapidly complicated multilevel problems. Therefore, we conclude that in 1D situations the use of MG is justified mainly for solving non-LTE RT problems in very fine grids ($\Delta z_l < 10$ km), since it is for these very high-resolution problems where MG turns out to be the true winner.

4.4. Results for the multidimensional case.

Unlike the 1D case, the efficiency of the multidimensional MG method is substantially better than methods like MALI (cf. Paper I) or MUGA (cf. TF-2). This follows from the fact that the convergence as a function of grid resolution in multidimensional cases is similar to the 1D case, presented in Fig. 9, but the cost of an iteration falls as a *power* of the grid resolution, which makes coarse grid computation relatively very cheap. In 3D, for example, a grid with half as many points in *each* direction requires an eighth as much computer time for a MUGA iteration. This fact plus the increase in the convergence rate as coarser grids are used makes the *multidimensional, multigrid* approach very effective.

First, consider the application of the two-grid (TG) method to grids sufficiently fine such that the TG iteration is dominated by the *coarsest solution time* (see Fig. 13). From Eq. (30) we have that the time improvement factor of the TG method with respect to the MUGA method reaches a constant value as the spatial resolution of the grid is increased. However, while this constant value was $\sim$ 0.3 in the 1D case, it is $\sim$ 0.15 in 2D and $\sim$ 0.075 for 3D situations. (Recall that we found it optimum to choose $\delta = {\rm NP}_{l-1}/{\rm NP}_l = 1/2$ for 1D, $\delta = 1/4$ for 2D and $\delta = 1/8$ for 3D applications). This means that one can save, with respect to the *pure* MUGA method, about an order of magnitude when solving multidimensional RT problems via the application of the TG method.

Second, consider the application of the MG method using a large number of grids (see Fig. 13). In this case, for most grid resolution levels the MG iteration turns out to be dominated by the *smoothing time* (cf. Eq. 31) and the improvement factor of the MG method with respect to MUGA can be approached by Eq. (36). As an upper limit we have that

$$\frac{\Re_l({\rm MG})}{\Re_l({\rm MUGA})} \;<\; -\log_{10}(\rho_l^{\rm MUGA})\,(1+\nu)\sum_{k=0}^{N-2}\delta^k, \tag{37}$$

with $\sum_{k=0}^{N-2}\delta^k \rightarrow 1.33$ in 2D, and $\sum_{k=0}^{N-2}\delta^k \rightarrow 1.14$ in 3D.

Therefore, in this limit the improvement in the efficiency by using the MG method is simply the larger the finer the grid (i.e. the closer to unity the value of $\rho_l^{\rm MUGA}$). For instance, for $\rho_l^{\rm MUGA}{\sim}0.9$ (which according to Fig. 9 it occurs for $\Delta z_l{\sim}10$ km for the Ca II problem) one already finds almost an order of

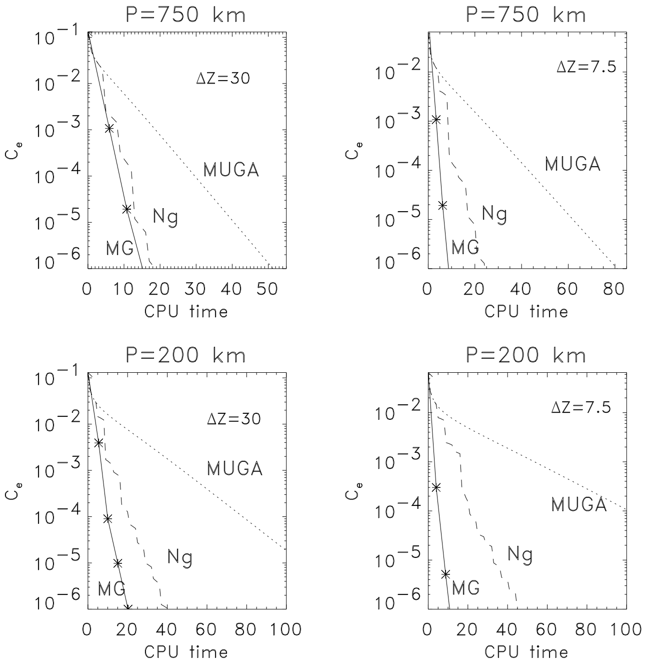

Fig. 15. The convergence error of 2D multilevel calculations for Ca II versus the CPU-time measured in units of the CPU-time required by one MUGA iteration in the finest grid. Each figure is for different values of Δz_l and P, where P is the periodicity of the horizontal temperature inhomogeneities. They show the MUGA method, MUGA combined with Ng acceleration, and the MG method. All the cases were calculated with $\nu = 2$ and with 25 horizontal points per horizontal period.

magnitude of CPU-time saving with respect to the pure MUGA method. As the spatial resolution is improved, the time one can save grows continuously, simply because $\rho_l^{\mathrm{MUGA}} \rightarrow 1$ as the discretization is refined.

In order to illustrate the performance of the MG method for multidimensional situations we show in Fig. 15 some results of 5-level Ca II calculations for a two-dimensional model atmosphere. As in Paper I, we have assumed that the temperature is fluctuating sinusoidally along the horizontal x-direction with an amplitude of 500 K and with different horizontal periodicities (P) (see Paper I for more details). For each of the two Δz_l values selected we have performed MG and MUGA calculations using two different values for the horizontal grid spacing Δx_l. The maximum number of grids was chosen according to the criterion that in the *coarsest* grid $\Delta z/\mathscr{H}$ should never become larger than unity (with $\mathscr{H}$ the density scale height), and that the *coarsest* grid must at least have three points per horizontal period. As it can be seen in Fig. 15 the improvement factor for 2D situations can indeed be very much larger than for the 1D case discussed above, especially if the grid is very fine. We conclude that the solution of complicated 2D and 3D multilevel problems with scalar-class computers can be done much more efficiently using our multigrid RT code than with our MALI or MUGA multidimensional codes, and with a saving factor which is the larger the higher the spatial resolution of the 2D and/or 3D grids required.

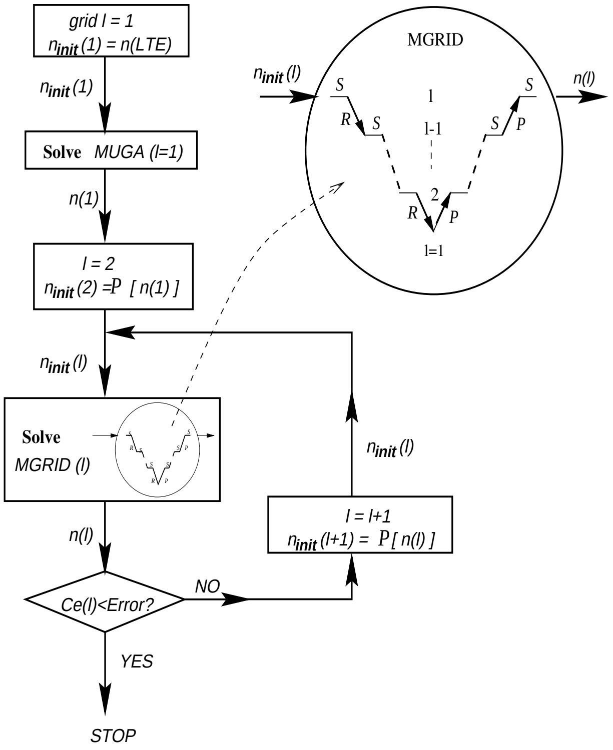

Fig. 16. Flow chart illustrating the logical structure of our nested multigrid RT code.

5. A nested multigrid RT method

We have used the term "*standard* multigrid method" to refer to the *non-linear* multigrid method described above. There is, however, a more efficient way of applying the multigrid technique, which leads to a factor 2 of CPU-time saving with respect to the *standard* multigrid method. We shall call it "the *nested* multigrid RT method".

A reader already familiar with the *grid-doubling* technique developed in Paper I will immediately realize its advantages. That grid-doubling strategy (cf. Paper I) consisted in (a) initialization (e.g. using LTE populations) and then iteration to convergence in the *coarsest* usable grid, i.e. in the grid of resolution level $l = 1$, (b) interpolation of these populations onto a *finer* level $l + 1 = 2$ grid, and then iteration *only* until the *convergence* error becomes smaller than the *truncation* error corresponding to that grid level, and (c) repetition of this last step "jumping" to successively finer grids until a solution with the desired truncation error is reached. An important point for our *grid-doubling* technique is, therefore, the analytical derivation of formulae (like Eq. (19)) which allow one to estimate the *convergence* error $\mathrm{C}_e(itr, l)$ from the relative changes $\mathrm{R}_c(itr, l)$ at each iterative stage "*itr*", and the *truncation* error $\mathrm{T}_e(\infty, l)$ associated with each grid used.

The method used in Paper I for solving the multilevel non-LTE equations in *each* grid was the MALI scheme; however, the same grid-doubling technique can be combined with any other multilevel transfer method, like e.g. our MUGA method,

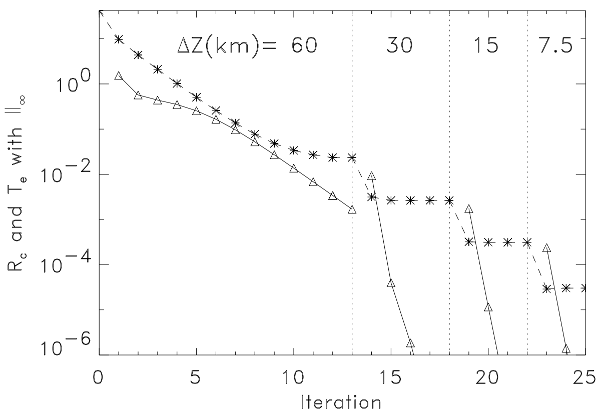

Fig. 17. The variation with the iteration number of R$_c$ (solid lines) and T_e (dashed lines) using four grids of increasing resolution (see labels). The problem treated is the 5-level Ca II 1D case. The vertical dotted lines indicate the zero iteration in each grid. In the coarsest grid of Δz=60 km we initialize with LTE populations. The $\infty-$norm was used.

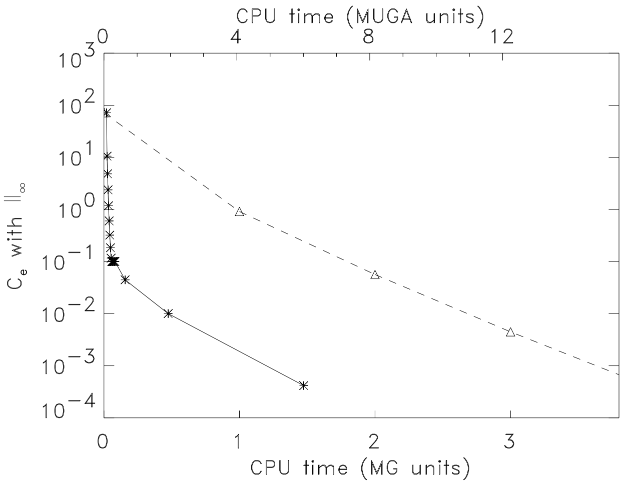

Fig. 18. Convergence error of the *standard* (dashed-line) and of the *nested* (solid-line) multigrid methods as a function of the CPU time in units of the CPU time of one standard MG iteration (bottom scale) and of one MUGA iteration (top scale) in the finest grid. These are 2D multilevel calculations for our Ca II reference problem with Δz_l=7.5 km, Δx_l=31.25 km and for a horizontal periodicity P=750 km. The $\infty-$norm was used.

or the standard multigrid scheme. What we now call "the nested multigrid RT method" consists in applying our "grid-doubling technique" (cf. Paper I), but using the *standard* multigrid RT method instead of the MALI scheme for the calculation of the atomic populations at each grid resolution level. Our *nested* multigrid method for multilevel RT applications has two loops: an *outer* loop where we sequentially refine the resolution of the grid being used, and an *inner* loop where we apply the standard multigrid method to find the run of populations on the current grid. The steps are:

(1) On the *coarsest* usable grid, level l = 1, iterate to convergence using the MUGA scheme.

(2) Interpolate the level l populations onto grid level l + 1 using cubic centered interpolation. Apply standard multigrid iterations in grids of level $l+1$ down to level 1. Iterate only until C$_e$ <T$_e$ in the current grid of level, l + 1.

(3) Continue this process until the *finest* grid has been reached or, alternatively, until the desired truncation error is obtained.

A flow chart illustrating the logical structure of our nested multigrid RT code is shown in Fig. 16. In the implementation of this nested multigrid strategy, we always use the $\infty-$norm to calculate C$_e$ and T$_e$ in order to obtain information on the *maximum* true error of the converged populations. Fig. 17 shows nested iterations with 4 grids for the Ca II reference 1D problem. We deliberately set the stopping criterion for each grid of resolution level $l > 1$ at $R_c = 10^{-8}$, instead of iterating each grid only until $C_e < T_e$. Thus, this figure demonstrates that the R_c-value of the first *multigrid* iteration is a good estimate of the truncation error of the previous grid (see also Paper I). Because a parabolic expansion is used for the formal integration of the transfer equation the truncation error decreases cubically as the grid resolution is increased (see Paper I). Therefore, the truncation error of each grid of resolution level "l" can be estimated by

$$\mathrm{T}_e(\infty, l) \approx \Big(\frac{\Delta z_l}{\Delta z_{l-1}}\Big)^3 \mathrm{T}_e(\infty, l-1) \approx \frac{1}{8}\,\mathrm{R}_c(1, l). \qquad (38)$$

At each l-grid we have to iterate only until C$_e$ <T$_e$. Therefore, from Eq. (19) we know that we have to iterate only until

$$\mathrm{R}_c(iter, l)\frac{\rho}{1-\rho} \ < \ \frac{1}{8}\,R_c(iter = 1, l) \ . \qquad (39)$$

With this "stopping criterion" we can easily estimate the number of multigrid iterations one actually needs for each grid: one iteration will be enough if $\rho \sim 0.1$, two if $\rho \sim 0.3$ and four if $\rho \sim 0.5$. This agrees perfectly with Fig. 17, where one MG iteration is always sufficient to achieve the truncation error at each grid level (note that in our Ca II case the convergence rate ρ is always smaller than 0.1 if $\nu > 1$.)

The solid line of Fig. 18 shows the convergence behaviour of the nested multigrid RT method as a function of the CPU time measured in units of one standard MG iteration (see the scale at the bottom) and in units of the CPU time required by one MUGA iteration in the finest grid (see the scale at the top). The dashed-line of Fig. 18 corresponds to a calculation performed with the standard MG method. For this four-grid example the CPU-time one can save by using the nested MG method instead of the standard one is about a factor two. Note that, as dictated by our stopping criterion, the iterations are automatically terminated once the *truncation error* of the finest grid is reached.

For a given grid resolution level the truncation error is determined by the accuracy of the formal solver used. That error for the finest grid in Fig. 18 is about 0.1 %, and the CPU time required by our nested MG method to yield the converged solution is approximately the time required to make only 1.5 *standard* MG iterations (see the scale at the bottom of Fig. 18). The scale at the top shows that the converged solution in the finest grid (where the accuracy is $\sim$0.1 %!) is reached in a time similar to that required to perform 6 MUGA iterations, which in turn indicates that the CPU time needed by one standard MG iteration is equal to the time of 4 MUGA iterations. We remind the reader that the cost of one MUGA iteration is virtually identical to the cost of one MALI iteration and that, unless NL is very large, this computational work is dominated by the cost of one formal solution (for all the atomic-model transitions).

The nested multigrid multilevel RT method presented here (either for 1D, 2D or 3D applications) is very fast, since even in very fine grids it provides the converged solution in few (< 10) formal solution times. To stress this point, we note that in Fig. 12 of Paper I we showed an example of a 2D multilevel calculation for Ca II applying our grid-doubling technique (combined with the MALI scheme) for a four-grid case. For that particular four-grid example, our nested multigrid RT technique is four times faster that the MALI-based nested-grid method of Paper I. We further note that the higher the resolution level of the finest-grid the larger the CPU-time gaining factor with respect to our MALI-based grid-doubling technique of Paper I.

6. Concluding remarks

The non-linear multigrid methods developed in this paper for multilevel RT applications are characterized by a very high convergence speed that does not deteriorate when the grid resolution is refined. For this reason they are particularly suitable for solving complicated 1D, 2D or 3D multilevel problems, where grid-sizes smaller than about 10 km are required. The computational work one can save by using the non-linear multigrid RT method instead of the operator splitting methods currently in use (like, e.g., the MALI scheme), is the larger the smaller the grid-size. This saving is larger for 3D applications than for 2D, which, in turn, is larger than for 1D.

Our nested multigrid RT code is even more effective. Its storage requirements are only a factor $1/(1-\delta)$ larger than those of the MALI method of Paper I (with $\delta = 1/2$ for 1D, $\delta = 1/4$ for 2D, and $\delta = 1/8$ for 3D applications), and it is a factor 2 faster than the *standard* multigrid RT technique. It, thus, allows the accurate solution of RT problems with a computational effort equal to the cost of only *very* few formal solutions of the RT equation. For instance, if n indicates the number of grid-points per decade, the computational work of the Jacobi-type MALI scheme of Paper I scales approximately as n^2, the MUGA method of Trujillo Bueno & Fabiani Bendicho (1995; 1996) (which is based on Gauss-Seidel iterations) as $n^2/2$, their SOR-based technique as $\sqrt{n}n/2\sqrt{2}$, while the multilevel RT *multigrid* methods presented here scale as n, because the very high convergence rate of the multigrid iteration is insensitive to the grid-size, while the cost per MG iteration scales as n. To our knowledge there is presently no other multilevel transfer code capable of offering this very high performance in scalar-class computers. Preliminary results indicate that a similarly good relative improvement is obtained in calculations for realistic atomic models and atmospheres, whose presentation we leave for future publications.

Acknowledgements. This work has been supported by the Instituto de Astrofísica de Canarias (IAC) and by the Spanish DGES through project PB 95-0028. It has also been partly supported by the European Space Agency through an external fellowship at the Kiepenheuer-Institut für Sonnenphysik, and by the U.S. Department of Energy. We are also grateful to Dr. Oskar Steiner for useful discussions during the early stages of this investigation and to the referees, Drs. Petr Heinzel and Jiri Kubat, for their careful reading of the paper.

References

Auer, L. H., Fabiani Bendicho, P., Trujillo Bueno, J. 1994, A&A 292, 599
Auer, L.H., Paletou, F. 1994, A&A 285, 675
Brandt, A. 1977, Math. Comp. 31, 333
Carlsson, M. 1986, A Computer Program for Solving Multi-Level Non-LTE Radiative Transfer Problems in Moving or Static Atmospheres, Report No. 33, Uppsala Astronomical Observatory
Hackbusch, W. 1985, Multi-Grid Methods and Applications, Springer-Verlag, Berlin
Kunasz, P., Auer, L.H. 1988, JQSRT 39, 67
Mihalas, D. 1978, Stellar Atmospheres, Freeman and Company
Olson, G.L., Auer, L.H., Buchler, J.R. 1986, JQSRT 35, 431
Olson, G. L., Kunasz, P. B. 1987, JQRST 38, 325
Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1994, Numerical Recipes, Cambridge University Press
Rybicki, G. B., Hummer, D. G. 1991, A&A 245, 171
Rybicki, G. B., Hummer, D. G. 1992, A&A 262, 209
Steiner, O. 1991, A&A, 242, 290
Trujillo Bueno, J., Fabiani Bendicho, P., 1995, ApJ 455, 646
Trujillo Bueno, J., Fabiani Bendicho, P., 1997 (in preparation)
Väth, H. M. 1994, A&A, 284, 319

This article was processed by the author using Springer-Verlag LaTeX A&A style file *L-AA* version 3.