# Additive and Multiplicative Multi-Grid –a Comparison

Peter Bastian[1)], Wolfgang Hackbusch[2)], Gabriel Wittum[1)]

## Summary

In the present paper, we describe and classify the numerous variants such as BPX, hierachical basis, HBMG, local multi-grid etc. in the framework of multi-grid. We compare additive and multiplicative multi-grid and investigate in particular the different behaviour with respect to smoothing. Theoretical as well as numerical results clearly show the superiority of multiplicative multi-grid over the additive variants.

## 1. Introduction

In recent years, multi-grid methods have become powerful tools for solving problems from many scientific and technical applications. As methods of optimal complexity, they contribute substantially to solving very large problems on supercomputers as well as medium-scale problems on workstations. In practical problems, features like robustness are crucial for the actual performance of the method. There are two main components of the multi-grid method, smoothing and coarse-grid correction. Particularly in view of robustness, smoothing plays a very important rôle.

In combination with adaptive grid refinement, many multi-grid variants have been developed. In the present paper, we describe and classify these numerous variants such as BPX, hierarchical basis, HBMG, local multi-grid etc. in the classical multi-grid framework. We compare additive and multiplicative multi-grid and investigate in particular their different behaviour with respect to smoothing. Theoretical as well as numerical results clearly show the superiority of multiplicative multi-grid over the additive variants.

In § 2, we introduce multi-grid methods and describe and classify different multi-grid variants like additive and multiplicative ones, local mg, BPX, hierarchical basis and HBMG. §3 introduces the notion of robustness and leads to the problem that improvement of the smoother has different effects on the different multi-grid variants. This question is discussed in the following sections. In §4, the behaviour of additive and multiplicative multi-grid for an increasing number of smoothing steps is analyzed. §5 gives experimental results in accordance with the analytic ones.

1. Institut für Computeranwendungen, Universität Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany

2. Institut für Informatik und Praktische Mathematik, CAU, Olshausenstr. 40, 24098 Kiel, Germany

## 2. Classification of Multi-Grid Methods

Let the linear boundary value problem

(2.1) $Ku = f$ in $\Omega$

(2.2) $u = u_R$ on $\delta\Omega$

with a differential operator $K:U \to F$ between some function spaces $U$ and $F$ be given on a domain $\Omega \subset \mathbb{R}^d$. Let (2.1) be discretized by some local discretization scheme on a hierarchy of admissible grids (cf. [11])

(2.3) $\Omega_l, l = 1,\ldots,l_{\max}$

with

(2.4) $\Omega_l \subset \Omega_{l+1} \subset \Omega.$

We use nested grids only for ease of presentation. Most of the methods discussed below can readily be applied to general loosely coupled grids violating (2.4). The discretized equations on $\Omega_l$ are denoted by

(2.5) $K_l u_l = f_l$ in $\Omega_l$, for $l = 1,\ldots,l_{\max}$,

(2.6) $u_l = u_{R,l}$ on $\partial\Omega_l$

with

(2.7) $K_l:U_l \to F_l$,

$U_l$, $F_l$ denoting the discrete analogues of $U$ and $F$ with finite dimension. We assume that the discretized equations are sparse. Further, let some "smoother"

(2.8) $\mathcal{S}_l : U_l \to U_l$ for $l = 1,\ldots,l_{\max}$,

and "grid transfer operators"

(2.9) $p_{l-1} : U_{l-1} \to U_l$, $r_{l-1} : F_l \to F_{l-1}$, for $l = 1,\ldots,l_{\max}$,

be given.

Multi-grid methods are fast solvers for problem (2.5). The multi-grid idea can be explained as follows. Let an $u^i \in \mathbb{R}^n$ be an iterate approximating $u \in \mathbb{R}^n$. Let $b_i$, $i = 1,\ldots,n$, be a set of basis functions for $\mathbb{R}^n$. Expanding the error $u - u^i$ in this basis, we obtain

(2.10) $$u - u^i = \sum_k \alpha_k b_k \ .$$

In the case of a Fourier basis, which makes sense for certain model problems (cf. [10]), these basis functions can be distinguished into rough and smooth ones. The basic multi-grid idea is to remove error components on grids where they are rough, i.e. the wavelength is of the order of the gridsize, which can be done easily by an averaging process like a linear iterative scheme. Correspondingly, we have to use a scale of grids and to combine the problems on the single grids appropriately. There are two basic

strategies for cycling through this grid hierarchy, additive and multiplicative multi-grid. The multiplicative method is the well-known classical multi-grid (cf. [10]) as given in Algorithm 2.1.

**Algorithm 2.1**: Multiplicative multi-grid method

*mmgm*$(l, u, f)$;

integer $l$; grid function $u$, $f$;

{ grid function $v$, $d$; integer $j$;

if $(l=0)$ $u := K_l^{-1} f$;

else {

$u := \mathcal{S}_l^{\nu}(u, f)$;

$d := r_{l-1}(K_l u - f)$;

$v := 0$;

for $j := 1$ step 1 to $\gamma$ do *mmgm*$(l-1, v, d)$;

$u := u - p v$; }

}

The additive multi-grid method is given by the following algorithm.

**Algorithm 2.2:** Additive multi-grid method

*amgm*$(l, u, f)$;

integer $l$; grid function $u, f$;

{ grid function $v, g$;

if (l=0) then $u := K_0^{-1} f$

else {

$g := r_{l-1}(K_l u[l] - f[l])$; $v := 0$;

$u := u + \vartheta_1 (\mathcal{S}_l^{\nu}(u, f) - u)$;

*amgm*$(l-1, v, g)$; $u := u - \vartheta_2 p_{l-1} v$; }

}

The standard approach uses

(2.11) $$\vartheta_1 = \vartheta_2 = 1.$$

We will further use

(2.12) $$\vartheta = \vartheta_1 = \vartheta_2.$$

The structure of both algorithms can be seen from Fig. 2.1. The main difference between these two variants is that in the multiplicative method, smoothing and restriction of the defect to the next coarser level are performed on one level after the

other sequentially, while in the additive method smoothing on the different levels can be performed in parallel. Restriction and prolongation, however, are sequential in the additive method, too. Thus, on a parallel machine, the additive method also has a logarithmic complexity.

**Remark 2.3:** *On a serial computer the work count of both methods is equal.*

Usually, the additive methods are applied as preconditioners, since acceleration methods like conjugate gradient pick an optimal damping parameter directly, the multiplicative methods are used as solvers. According to [22], these methods can be formulated as additive Schwarz methods.

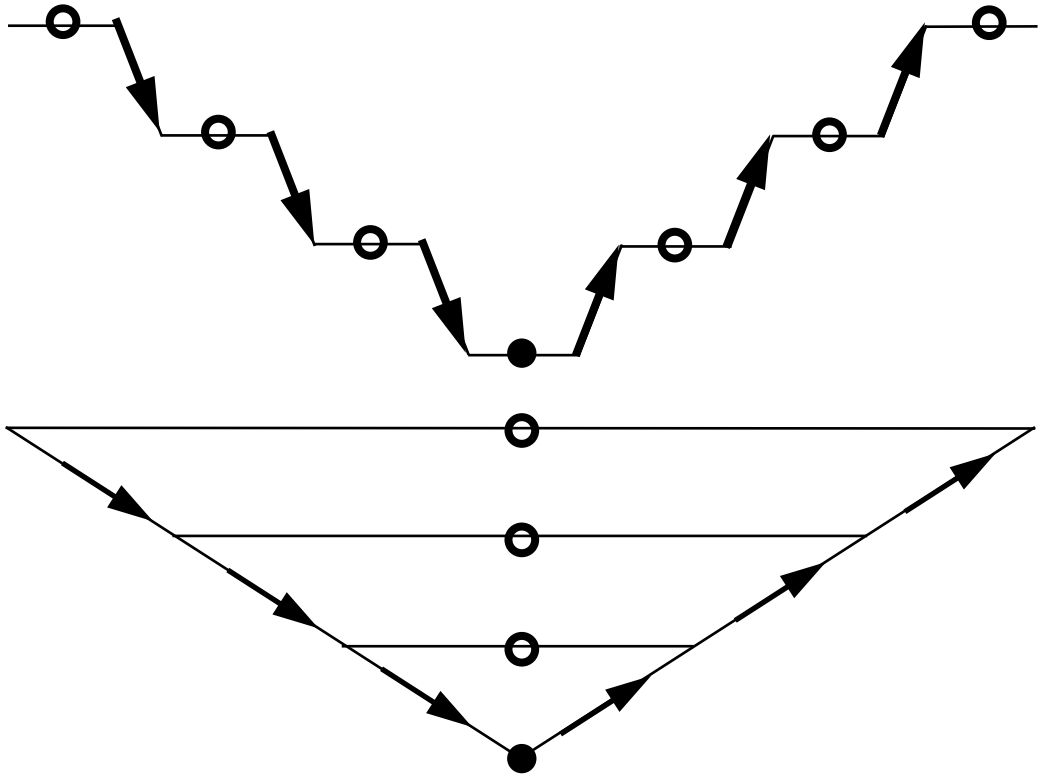


**Fig. 2.1:** Outline of multiplicative multi-grid algorithm *mmgm* (V-cycle) above and of the additive multi-grid algorithm *amgm* below. Symbols used:


Applying multi-grid methods to problems on locally refined grids one has to think about the basic question of how to associate grid-points with levels in the multi-grid hierarchy. Consider the hierarchy of grids $\{\Omega_l, l = 0,\ldots,l_{\max}\}$ from (2.3). Early multi-grid approaches smooth all points in $\Omega_l$. This may cause a non-optimal amount of work and

memory of $O(n \log n)$ per multi-grid step. This problem was the starting point for Yserentant, [23], and Bank-Dupont-Yserentant, [1], to develop the method of hierarchical bases (HB) and the hierarchical basis multi-grid method (HBMG). These were the first multi-grid methods with optimal amount of work per step for locally refined grids. This is due to the fact that on level $l$, only the unknowns belonging to points in $\Omega_l \setminus \Omega_{l-1}$ are treated by the smoother. However, the convergence rate deteriorates with $\log n$. This problem was solved by the introduction of the additive method by Bramble, Pasciak and Xu, [6], (BPX). There, on level $l$ the smoother treats all the points in $\Omega_l \setminus \Omega_{l-1}$ and their direct neighbours, i.e. in all points within the refined region. The theory from [22], however, does not say anything about an improvement of the convergence with a better smoothing e.g. by performing more than one smoothing step or applying robust smoothers.

| basic structure / smoothing pattern | additive | multiplicative |
|---|---|---|
| new points only | hierarchical basis, [23] | hierarchical basis multi-grid, [1] |
| whole refined region | BPX, [6] | local multi-grid, [16], [7], [2] |
| all points | mgmp, [9] | standard mg |

**Table 2.2**: Multi-grid methods for locally refined grids.

Table 2.2 gives an overview of the multi-grid methods used for the treatment of locally refined grids. The methods mentioned above differ in the smoothing pattern, i. e. the choice of grid points treated by the smoother. The methods in the first two lines are of optimal complexity for such problems. The amount of work for one step is proportional to the number of unknowns on the finest grid. However, only the methods in the second line, BPX and local multi-grid, converge independently of $h$ for scalar elliptic problems. The basic advantage of the multiplicative methods is that they do not need cg-acceleration and can thus be applied directly to unsymmetric problems; further, they show a better convergence rate and, on a serial computer, the additive process does not have any advantage. The local multi-grid scheme is the natural generalisation of the classical multi-grid method to locally refined grids, since, in the case of global refinement, it is identical with the standard classical multi-grid method.

The local multi-grid method was first introduced by Rivara in 1984, [16] recognized again and analyzed in 1991 by Bramble, Pasciak, Wang and Xu, [7]. They considered it

a multiplicative variant of their so-called BPX-method, [6]. Without knowledge of this, one of the authors developed this method as a variant of standard multi-grid based on the idea of robustness (cf. [2]). The main advantage of this approach is that the application to unsymmetric and non-linear problems is straightforward (cf. [2]).

Robustness for singularly perturbed problems is achieved by combining local multi-grid with robust smoothers, as explained in the next section.

## 3. Robust Multi-Grid

When we apply standard multi-grid to singularly perturbed problems, the convergence depends on the perturbation parameter; in particular, it deteriorates strongly when the parameter approaches the type change of the discrete equation. To overcome this problem, Wesseling suggested the first robust multi-grid method for singularly perturbed problems discretized on structured grids [18]. The main idea is to apply a smoother which solves the limit case exactly. This is possible, for example, for a convection-diffusion equation using a Gauß-Seidel smoother and numbering the unknowns in convection direction, cf [5]. Often, it is preferable to use an incomplete LU-smoother, since this handles the convection-dominated case as well as the anisotropic diffusion (cf. [3], [4], [19]).

We denote the singularly perturbed operator $K(\varepsilon)$ by

(3.1) $K(\varepsilon) = K_{\mathrm{I}} + \varepsilon K_{\mathrm{II}}$ with $\varepsilon > 0$ and $K_{\mathrm{I}}$ and $K_{\mathrm{II}}$ of different type

and the smoother by

(3.2) $\mathcal{S}(u, f, \varepsilon) = u - N_l(\varepsilon)\,(K_l(\varepsilon)\,u - f)$.

The iteration matrix of the smoother is given by

(3.3) $S(\varepsilon) = I - N(\varepsilon)K(\varepsilon)$

which corresponds to the splitting

(3.4) $K(\varepsilon) = N(\varepsilon)^{-1} + R(\varepsilon)$.

In this terminology, robustness requires

(3.5) $\|R(\varepsilon)\| \to 0$, for $\varepsilon \to 0$,

i.e. the smoother becomes an exact solver in the limit $\varepsilon \to 0$.

Under this condition, the method will be fast even in the limit case $\varepsilon \to 0$, provided the outer multi-grid algorithm does not spoil the accuracy. To this end, we investigate to

what extent an improvement of the smoother affects the overall convergence of all multi-grid variants. This is the key to robustness. In §4, we consider improving the smoother by increasing the number of smoothing steps, which has a similar effect to (3.5).

In the case of the hierarchical basis methods (HB, HBMG), we can directly see that the smoothing pattern is too poor to allow robust smoothing. Thus, those methods do not benefit from improving the smoother in the sense of (3.5).

**Remark 3.1:** *In the hierarchical basis method and the hierarchical basis multi-grid method, only a part of the grid points on the finest grid is treated by the smoother. Thus, (3.5) is not satisfied and these methods do not allow robust smoothing. This holds for uniformly as well as for locally refined grids.*

Based on this observation, we extend the smoothing pattern, incorporating all points in the refined region. With the usual refinement strategies, this is equivalent to extending the smoothing to all neighbours of points in $\Omega_l / \Omega_{l-1}$. This allows the smoother to solve the limit case exactly, provided the grid refinement is appropriate. This is confirmed by numerical evidence given in Chapter 5.

Up to now, some theory is contained in [19], [20], [21] for uniformly refined grids. This theory shows that the basic requirement for using a smoother which is an exact solver in the limit case is not sufficient to obtain robustness. Additionally, it must be guaranteed that the spectrum of the smoother is contained in $[-\vartheta, 1]$ for $0 \leq \vartheta < 1$. This can be achieved by modification (cf. [19], [21]).

There are also other ways of improving the coarse-grid correction for singularly perturbed problems which are applied in practice, e.g. algebraic multi-grid like [17]. In [14], however, Neuß showed that there are problems which cannot be reasonably approximated on coarse grids, even with algebraic multi-grid algorithms. Thus, robust smoothing is essential to construct parameter-independent multi-grid methods. An overview of robust multi-grid can be found in [3], [4].

In the following chapter, we investigate the effect of increasing the number of smoothing steps on the convergence of the multi-grid variant.

## 4. Convergence Depending on the Number of Smoothing Steps

In the following, we investigate the dependence of the convergence of the different multi-grid variants on the number $\nu$ of smoothing steps performed. There, additive and multiplicative methods behave quite differently.

## 4.1 Iteration Matrices

The iteration matrix of the smoothing iteration $\mathcal{S}_l(u,f) = u - N_l(K_l u - f)$ involved in Algorithms 2.1 and 2.2 is

(4.1a) $\quad S_l = I - N_l K_l .$

The ν- fold smoothing leads to $\mathcal{S}_l^{\nu}(u,f) = u - N_l^{(\nu)}(K_l u - f)$ with the iteration matrix

(4.1b) $\quad S_l^{\nu} = I - N_l^{(\nu)} K_l$ with $N_l^{(\nu)} = (I - S_l^{\nu}) K_l^{-1}$ .

Simple examples of smoothing iterations are the Jacobi iteration damped by 1/2,

(4.2a) $\quad N_l = \frac{1}{2} D^{-1}$ ($D$: diagonal of $K_l$),

or the Cimmino iteration

(4.2b) $\quad N_l = \omega K_l^T \qquad \omega \le \frac{1}{\left\| K_l^T K_l \right\|}.$

In the following, we investigate the two-grid method corresponding to Algorithms 2.1 and 2.2 with $l$=1. In this case, the choice of $\gamma$ is arbitrary (e.g. set $\gamma$= 1). The superscripts „mtgm“ and „atgm“ abbreviate the multiplicative and additive two-grid method. The iteration matrices of both two-grid versions are

(4.3a) $\quad M_l^{\mathrm{mtgm}}(\nu) = (I - p_{l-1} K_{l-1}^{-1} r_{l-1} K_l) S_l^{\nu} \qquad \nu \ge 1 ,$

(4.3b) $\quad M_l^{\mathrm{atgm}}(\vartheta_1, \vartheta_2, \nu) = I - \left( \vartheta_2 P_{l-1} K_{l-1}^{-1} r_{l-1} + \vartheta_1 N_l^{(\nu)} \right) K_l , \qquad \nu \ge 1 .$

## 4.2 Analysis of Smoothing and Convergence in the Multiplicative Case

The convergence analysis of the multiplicative variant is able to explain that, with increasing $\nu$, the smoothing effect becomes stronger and that the convergence improves. This is expressed by the *smoothing property*

(4.4a) $\quad \left\| K_l S_l^{\nu} \right\| \le \eta(\nu) \left\| K_l \right\| ,$

where $\eta(\nu)$ is independent of the dimension (i.e. of the grid size and of $\mathrm{cond}(K_l)$ ) and satisfies

(4.4b) $\quad \eta(\nu) \to 0$ as $\nu \to \infty$ .

For a generalisation including divergent smoothing iterations, compare [10] and [12]. A classification of the smoothing property in the symmetric case can be found in [20] and estimates w.r.t. other norms in [15], [13] and [8] .

If the *approximation property* $\left\| K_l^{-1} - p_{l-1} K_{l-1}^{-1} r_{l-1} \right\| \le C_A / \left\| K_l \right\|$ holds, we obtain the convergence estimate

(4.5) $\left\| M_l^{\mathrm{mtgm}}(\nu) \right\| \le C_{\mathrm{A}}\, \eta(\nu)$ .

Because of (4.4b), it is plain that convergence improves with increasing number $\nu$ of smoothing steps. In the positive definite case, the analysis can even show that all $\nu \ge 1$ lead to convergence, i.e. , $\left\| M_l^{\mathrm{mtgm}}(\nu) \right\| < 1$ (cf. [12], § 10.7).

The standard behaviour of $\eta(\nu)$ is

(4.6) $\eta(\nu) = C/(1+\nu) \qquad (\nu \ge 0)$

as we shall confirm below. Concerning the optimal choice of $\nu$, one has to minimize the effective costs as discussed in [10],§4.3. We recall that usually $\nu = 2$ or $\nu = 3$ are more effective than $\nu = 1$:

(4.7) The choice $\nu = 1$ is not optimal for *mmgm*.

## 4.3 The Additive Two -Grid Method

If *amgm* is used within a (conjugate) gradient method, the choice of the damping factor $\vartheta$ from (2.12) is irrelevant. We recall the following facts (cf. [12]). If we denote $p_{l-1} K_{l-1}^{-1} r_{l-1} + N_l^{(\nu)}$ by $W_l^{-1}$ (inverse of the preconditioner), the best constants $\gamma$, $\Gamma$ in

(4.8) $\gamma K_l \le W_l \le \Gamma K_l$

define the condition number $\kappa = \Gamma/\gamma$, provided that $K_l$ and $W_l$ are positive definite. The gradient method as well as the iteration with the optimal scaling

(4.9) $\vartheta_{\mathrm{opt}} = 2/(\gamma + \Gamma)$

has the convergence rate

(4.10a) $\rho = (\kappa - 1)/(\kappa + 1)$ ,

whereas the conjugate gradient method converges at least with the rate

(4.10b) $\rho = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$

(cf. [12]). However, it is questionable whether (2.12) is the best realisation of the additive method. If $S$ is convergent (i.e., $\rho(S) < 1$), $S_l^{\nu}$ becomes the exact solver as $\nu \to \infty$, i.e., $\lim_{\nu \to \infty} N_l^{(\nu)} = K_l^{-1}$, but the iteration *amgm* (or *atgm*) is not exact when $N_l^{(\nu)}$ is replaced by $K_l^{-1}$. For this purpose, one has to introduce individual weighting factors for each correction in Algorithm 2.2 ( $\vartheta_1$ for the correction from the smoothing process, $\vartheta_2$ for the coarse-grid correction). In the case of exact smoothing , $\vartheta_1 = 1$ and $\vartheta_2 = 0$ ensure that *amgm i*s exact too.

**Remark 4.1** The fact that *amgm* becomes the exact solver for $\nu \to \infty$ does not imply any statement concerning the behaviour for small $\nu$ (as was possible in (4.5)). The reason is that $S_l$ is very slowly convergent. In the case of (4.2a), $\nu = O(|\log \varepsilon| \mathrm{cond}(K_l))$

steps are necessary to ensure $\|S^\nu\| \le \varepsilon$.

We add that the standard analysis of the additive Schwarz method does not discuss different choices of $\nu$ (cf. [12],§11.3).

## 4.4. The Model Problem

To obtain a precise statement about the dependence of the convergence rate on $\nu$, we consider the simplest possible system. The one-dimensional Poisson equation

$$-u'' = f \text{ in } (0,1), \qquad u(0) = u(1) = 0,$$

discretised by the grid size $h = 1/n$ ($n$ even) leads to the system

$$Ku = f, \qquad K = h^{-2}\,\mathrm{tridiag}\{-1, 2, -1\},\ f_j := f(jh). \tag{4.11}$$

The matrices $K, D = \mathrm{diag}(K) = 2h^{-2}I$, and $S = I - 1/2\, D^{-1}K$ (in the case of (4.2a)) or $S = I - K^2/16$ (in the case of (4.2b)) can be diagonalised simultaneously by means of the sine functions $\sin(\nu\mu x)$, μ=1.....$n$-1. The matrix $p_{l-1}K_{l-1}^{-1}r_{l-1}K_l$ arising from the coarse-grid correction is transformed into a block-diagonal matrix consisting of ($n$-2)/2 2x2 blocks and one 1x1 block. Pairing the frequencies μ and $n-\mu$ $(\mu < n/2)$, the 2x2 blocks take the form

$$\begin{bmatrix} s^2 & -s^2 \\ -c^2 & c^2 \end{bmatrix}, \qquad c^2 = 1 - s^2,\ s^2 = \sin((\mu\pi h)/2). \tag{4.12}$$

For a proof, compare [10] or [12], §10.3. In total, the iteration matrix $M_l(\nu)$ has a spectral radius $\rho(M_l(\nu)) = \max\{\rho(M_l(\sin^2(\mu\pi h/2), \nu)): 1 \le \mu \le n/2\}$, where $M_l(\xi,\nu)$ are 2x2 or 1x1 blocks. For $1 \le \mu < n/2$, the 2x2 blocks are

$$\begin{bmatrix} s^2 & c^{2\nu} & c^2 & s^{2\nu} \\ s^2 & c^{2\nu} & c^2 & s^{2\nu} \end{bmatrix}, \ c^2 = 1 - s^2,\ s^2 = \sin^2((\mu\pi h)/2).$$

To obtain a bound which is independent of the discretisation parameter $h$, one uses the fact that $\{\sin^2(\mu\pi h/2): 1 \le \mu \le n/2\}$ is contained in [0, 0.5]. Therefore, a uniform bound of $\rho(M_l(\nu))$ is given by

$$\rho^{\mathrm{mtgm}}(\nu) := \max\{\rho(M_l(\xi, \nu)): 0 \le \xi \le 1/2\} \tag{4.13}$$

(cf. [10], (4.4.10)). Note that the rates $\rho(M_l(\nu))$ converge to $\rho^{\mathrm{mtgm}}(\nu)$, i.e. the bounds $\rho^{\mathrm{mtgm}}(\nu)$ are *uniform and sharp*.

## 4.5 Multiplicative Case

The maxima in (4.13) can easily be determined numerically. As predicted in (4.5 - 6), the values of $\rho(\nu)$ shown in Table 4.1 behave like const/ $(1 + \nu)$ . Since the additive variants will be used with optimal damping, it might be fair to compare these results with the optimally damped multiplicative version as well. The iteration matrix of the damped version is

(4.14) $$M_l^{\mathrm{mtgm}}(\nu, \vartheta) = (1 - \vartheta) I + \vartheta M_l^{\mathrm{mtgm}}(\nu) \quad ,$$

where $M_l^{\mathrm{mtgm}}(\nu)$ is defined in (2.3a). This gives rise to 2x2 blocks $M_l^{\mathrm{m}}(\xi, \nu, \vartheta)$ . The optimal *h*-independent convergence rate equals

(4.15) $$\rho_{\mathrm{damped}}^{\mathrm{mtgm}}(\nu) := \min\{\max\{\rho(M_l^{\mathrm{m}}(\xi, \nu, \vartheta)) : 0 \le \xi \le 1/2\} : \vartheta \in \mathbb{R}\} \quad .$$

Solving these min-max problems numerically, we obtain the values in the right part of Table 4.1. Asymptotically, $\rho_{\mathrm{damped}}^{\mathrm{mtgm}}(\nu) \approx \rho^{\mathrm{mtgm}}(\nu)/2$ holds. The optimal values of $\vartheta$, which are also shown in the table, behave like $\vartheta \approx 1 + \rho^{\mathrm{mtgm}}(\nu)/2$ .

**Table 4.1.** Convergence rates of the multiplicative two-grid iteration *mtgm*

| $\nu$ | $\rho^{\mathrm{mtgm}}(\nu)$ | $\vartheta$ | $\rho_{\mathrm{damped}}^{\mathrm{mtgm}}$ |
|---|---|---|---|
| 1 | 0.500 | 1.333 | 0.333 |
| 2 | 0.250 | 1.143 | 0.143 |
| 3 | 0.125 | 1.067 | 0.0667 |
| 4 | 0.0833 | 1.043 | 0.0435 |
| 5 | 0.0671 | 1.035 | 0.0347 |
| 6 | 0.0567 | 1.029 | 0.0292 |
| 7 | 0.0491 | 1.025 | 0.0252 |
| 8 | 0.0433 | 1.022 | 0.0221 |
| 9 | 0.0387 | 1.020 | 0.0198 |
| 10 | 0.0228 | 1.018 | 0.0178 |
| 20 | 0.0179 | 1.009 | 0.00905 |
| 100 | 0.00366 | 1.002 | 0.00181 |

We add the remark that the convergence rates do not depend on whether we apply only pre-smoothing, only post-smoothing or a combination of $\nu_1$ pre- and $\nu_2$ post-smoothing steps, provided $\nu_1 + \nu_2 = \nu$ .

## 4.6 Additive Case

The iteration matrix of *atgm* can be computed in the same way. The sine transform ( see §4.4) lead us to the 2x2 blocks

$$M_l^{\text{add}}(\nu,\vartheta_1,\vartheta_2) = \begin{bmatrix} 1+\vartheta_1\,(c^{2\nu}-1)-\vartheta_2 s^2 & \vartheta_2 s^2 c^{2\nu} \\ \vartheta_2 c^2 & 1+\vartheta_1\,(s^{2\nu}-1)-\vartheta_2 c^2 s^{2\nu} \end{bmatrix}$$

with $c^2 = 1 - s^2$, $s^2 = \sin^2(\mu\pi h/2)$, involving the damping parameters $\vartheta_1$, $\vartheta_2$, from Algorithm 2.1. Setting $\vartheta_1 = \vartheta_2 = \vartheta$ as in (2.12) and $s^2 = \xi$, $c^2 = 1 - \xi$, we define $M_1^{\text{add}}(\xi,\nu,\vartheta)$. The arising min-max problem

$$\rho^{\text{atgm}}(\nu) := \min\,\{\max\,\{\rho(M_l^{\text{add}}(\xi,\nu,\vartheta)) : 0 \le \xi \le 1/2\} : \vartheta \in \mathbb{R}\} \tag{4.16}$$

is again solved numerically. Table 4.2 shows the optimal damping factors $\vartheta$ and the rates $\rho^{\text{atgm}}(\nu)$. Except for $\nu = 1$, the rates remain between 0.4 and 0.33. Obviously, they do not improve as in the multiplicative case. The reason for the exceptional behaviour of $\nu = 1$ is discussed below.

**Table 4.2.** Additive two-grid variant (one damping factor (2.12))

| $\nu$ | $\vartheta$ | $\rho^{\text{atgm}}(\nu)$ |
|---|---|---|
| 1 | 1.000 | 0.500 |
| 2 | 0.800 | 0.400 |
| 3 | 0.739 | 0.386 |
| 4 | 0.714 | 0.384 |
| 5 | 0.701 | 0.380 |
| 6 | 0.693 | 0.376 |
| 7 | 0.689 | 0.372 |
| 8 | 0.686 | 0.369 |
| 9 | 0.684 | 0.366 |
| 10 | 0.682 | 0.364 |
| 20 | 0.675 | 0.350 |
| 100 | 0.668 | 0.336 |

We try to improve the additive results by using two different damping parameters $\vartheta_1$ and $\vartheta_2$ in *amgm*. The corresponding optimal $h$-independent rates are given in Table 4.3. Obviously, they are almost the same for the whole range of $\nu$ values. For larger $\nu$, they coincide with those of Table 4.2. since the optimal values of $\vartheta_1$ and $\vartheta_2$ from Table 4.3 approach the same $\vartheta$ from Table 4.2. Only for $\nu \leq 3$ are the optimal damping factors clearly different. This explains why the choice $\vartheta = 1$ in Table 4.2 yields a worse rate than in Table 4.3.

**Table 4.3.** Additive two-grid method (two-parameter variant)

| $\nu$ | $\vartheta_1$ | $\vartheta_2$ | $\rho^{atgm}(\nu)$ | $\rho^{atgm}{}_{1/64}(\nu)$ |
|---|---|---|---|---|
| 1 | 1.333 | 0.666 | 0.333 | 0.333 |
| 2 | 0.914 | 0.666 | 0.351 | 0.351 |
| 3 | 0.818 | 0.647 | 0.363 | 0.362 |
| 4 | 0.773 | 0.641 | 0.365 | 0.365 |
| 5 | 0.743 | 0.645 | 0.365 | 0.365 |
| 6 | 0.732 | 0.642 | 0.363 | 0.363 |
| 7 | 0.721 | 0.644 | 0.360 | 0.360 |
| 8 | 0.716 | 0.644 | 0.357 | 0.358 |
| 9 | 0.708 | 0.649 | 0.356 | 0.356 |
| 10 | 0.708 | 0.646 | 0.354 | 0.354 |
| 20 | 0.687 | 0.657 | 0.344 | 0.344 |
| 100 | 0.669 | 0.667 | 0.335 | 0.355 |

In §4.3, we mentioned that the two-parameter method becomes exact as $\nu \to \infty$, since $S_l^\gamma$ becomes an exact solver. This, however, is only true for a fixed step size $h$ whereas Table 4.3 contains $h$-independent rates. The right column in Table 4.3 shows the optimal results $\rho^{atgm}{}_{1/64}(\nu)$ (with possibly other $\vartheta_1$, $\vartheta_2$) for $h = 1/64$. Obviously, the values for $h = 1/64$ do not differ from $\rho^{atgm}(\nu)$ even for the completely impractically high number $\nu = 100$. The reason is that convergence of $S_l^\gamma$ is not seen before $O(h^{-2})$ iteration steps are performed.

In practice, the additive multi-grid method is used within a gradient or conjugate gradient acceleration. For completeness, we report the results for $h = 1/64$. In Table 4.4, the gradient method is applied to the multiplicative two-grid variant. In order to have

a symmetric basic iteration, we apply $\nu/2$ pre- and $\nu/2$ post-smoothing steps. Therefore, $\nu$ must be an even number. Obviously, the results coincide with the $h$-independent rates from Table 4.1. Replacing the gradient method by a conjugate gradient method (cf. Table 4.5), we expect a halving of the average rate. This is confirmed by the results in Table 4.5. Applying these accelerations to the additive version *atgm*, we obtain the results of Table 4.6. Here, $\nu$ is not restricted to even numbers. $\rho_{ADD1}$ and $\rho_{ADD2}$ are the averaged convergence rates observed for the one- and two-parameter version. The cg-rate $\rho_{CG}$ can only be determined for the one-parameter case (2.12).

**Table 4.4.** Multiplicative two-grid variant *tmgm* combined with the gradient method (averaged convergence factors for $h = 1/64$)

| $\nu$ | 2 | 4 | 6 | 8 | 10 | 20 | 100 |
|---|---|---|---|---|---|---|---|
| $\rho$ | 0.14 | 0.04 | 0.027 | 0.022 | 0.017 | 0.007 | 0.0018 |

**Table 4.5.** Multiplicative two-grid variant *tmgm* combined with the *conjugate* gradient method *($h = 1/64$)*

| $\nu$ | 2 | 4 | 6 | 8 | 10 | 20 | 100 |
|---|---|---|---|---|---|---|---|
| $\rho$ | 0.07 | 0.024 | 0.017 | 0.010 | 0.009 | 0.005 | 0.001 |

**Table 4.6.** Rates for the one- and two-parameter gradient method (ADD1, ADD2) and cg method (CG). (averaged convergence factors for $h = 1/64$*)*

| $\nu$ | $\rho_{ADD1}$ | $\rho_{ADD2}$ | $\rho_{CG}$ |
|---|---|---|---|
| 1 | 0.46 | 0.30 | 0.29 |
| 2 | 0.38 | 0.32 | 0.21 |
| 3 | 0.35 | 0.34 | 0.20 |
| 4 | 0.37 | 0.36 | 0.21 |
| 5 | 0.37 | 0.36 | 0.20 |
| 6 | 0.36 | 0.35 | 0.20 |
| 7 | 0.36 | 0.34 | 0.20 |
| 8 | 0.36 | 0.34 | 0.19 |
| 9 | 0.35 | 0.34 | 0.19 |

| $\nu$ | $\rho_{ADD1}$ | $\rho_{ADD2}$ | $\rho_{CG}$ |
|---|---|---|---|
| 10 | 0.35 | 0.34 | 0.19 |
| 20 | 0.33 | 0.34 | 0.19 |
| 100 | 0.32 | 0.31 | 0.15 |

The rates of the columns $\rho_{ADD1}$ and $\rho_{ADD2}$ correspond to those from Table 4.2 and 4.3. The cg rates approximate the halved ADD1 rates.

## 4.7 Cimmino Smoothing

Not only the number of smoothing steps but also the kind of smoothing can be changed. The (damped) Cimmino smoothing process is the Jacobi iteration applied to the equation preconditioned with $(h^4/4)K_l$, i.e. , $S_l(u,f) = u - (h^4/4)K_l(K_l u - f)$. It is well known that the smoothing effect of this iteration is weaker. The asymptotic behaviour is only $\eta(\nu) = C/(\sqrt{1+\nu})$ (cf. (4.6)). But also the absolute values are worse as confirmed in Table 4.7.

**Table 4.7** Multiplicative two-grid method with Cimmino smoothing

| $\nu$ | $\rho^{mtgm}(\nu)$ | $\vartheta$ | $\rho^{mtgm}_{damped}$ |
|---|---|---|---|
| 1 | 0.7500 | 1.600 | 0.600 |
| 2 | 0.5625 | 1.391 | 0.391 |
| 3 | 0.4219 | 1.267 | 0.267 |
| 4 | 0.3164 | 1.188 | 0.188 |
| 5 | 0.2373 | 1.135 | 0.135 |
| 6 | 0.1840 | 1.101 | 0.101 |
| 7 | 0.1625 | 1.088 | 0.088 |
| 8 | 0.1502 | 1.081 | 0.081 |
| 9 | 0.1413 | 1.076 | 0.076 |
| 10 | 0.1340 | 1.072 | 0.072 |
| 20 | 0.0953 | 1.050 | 0.050 |
| 100 | 0.0428 | 1.022 | 0.022 |

The additive two-grid method yields the values shown in Table 4.8 (one-parameter version (2.12)) and Table 4.9 (two-parameter version). Obviously, the same tendency is

illustrated. In particular, the rates obtained are only slightly worse than for the „better“ Jacobi smoothing.

**Table 4.8** Additive two-grid method with Cimmino smoothing (one-parameter case)

| $\nu$ | $\vartheta$ | $\rho^{atgm}(\nu)$ |
|---|---|---|
| 1 | 1.333 | 0.667 |
| 2 | 1.066 | 0.533 |
| 3 | 0.928 | 0.464 |
| 4 | 0.858 | 0.445 |
| 5 | 0.820 | 0.446 |
| 6 | 0.795 | 0.449 |
| 7 | 0.777 | 0.450 |
| 8 | 0.763 | 0.450 |
| 9 | 0.753 | 0.450 |
| 10 | 0.746 | 0.450 |
| 20 | 0.718 | 0.434 |
| 50 | 0.704 | 0.408 |
| 100 | 0.696 | 0.393 |
| 200 | 0.690 | 0.380 |

**Table 4.9** Additive two-grid method with Cimmino smoothing (two-parameter case)

| $\nu$ | $\vartheta_1$ | $\vartheta_2$ | $\rho^{atgm}(\nu)$ |
|---|---|---|---|
| 1 | 1.600 | 0.534 | 0.600 |
| 2 | 1.317 | 0.763 | 0.432 |
| 3 | 1.123 | 0.758 | 0.410 |
| 4 | 1.012 | 0.729 | 0.420 |
| 5 | 0.941 | 0.712 | 0.430 |
| 6 | 0.893 | 0.702 | 0.436 |
| 7 | 0.859 | 0.695 | 0.439 |
| 8 | 0.836 | 0.689 | 0.441 |

| $\nu$ | $\vartheta_1$ | $\vartheta_2$ | $\rho^{atgm}(\nu)$ |
|---|---|---|---|
| 9 | 0.821 | 0.683 | 0.442 |
| 10 | 0.807 | 0.680 | 0.442 |
| 20 | 0.763 | 0.668 | 0.429 |
| 50 | 0.734 | 0.670 | 0.404 |
| 100 | 0.712 | 0.678 | 0.390 |
| 200 | 0.707 | 0.671 | 0.378 |

Since an increase of the number of smoothing steps does not improve the convergence even for this simple problem, such an improvement also cannot be expected for more involved boundary values problems. The rates for the additive methods for $\nu = 1$ are similar or worse than for the multiplicative method. On the other hand, the multiplicative method is not optimal for $\nu = 1$ (cf. (4.7)), i.e. the efficiency can be increased by choosing a higher value of $\nu$. Altogether, we conclude that the multiplicative multi-grid method is more efficient than the additive version. In contrast to the additive method, the multiplicative one benefits from increasing the number of smoothing steps. In the next section, we discuss numerical results for increasing $\nu$ and with respect to robust smoothing.

# 5. Experimental Results

## 5.1 Model Problem

In order to confirm the theoretical results from the previous section, we present some numerical results for the two-dimensional case. First, we investigate the model problem

(5.1) $\quad -\Delta u = f$, in $\Omega = (0,1)^2$ with Dirichlet boundary conditions.

The unit square is covered with a structured, triangular mesh, so that the standard Galerkin method yields a five-point stencil.

Following a suggestion of R. E. Bank (personal communication), the additive multi-grid method with different damping factors has been implemented as follows. On each level, a correction $v_i$ is computed as usual and then the final correction

(5.2) $$v = \sum_{k=0}^{j} \vartheta_k v_k$$

is determined such that

$$a\left( u + \sum_{k=0}^{j} \vartheta_k v_k, w \right) = f(w) \text{ , for all } w \text{ in } W = \text{span}\{v_0, \ldots, v_j\} \text{ ,} \tag{5.3}$$

where $a(.,.)$ and $f(.)$ are the bilinear form and right-hand side of the finite-element problem. This method costs about one matrix-vector-product and some dot-products per level in addition. The system of linear equations of dimension $j+1$, which has to be solved in equation (5.3), may become singular when some of the corrections become linearly dependent. In that case, the method reverts to the standard method and chooses $\vartheta_k = 1$ for all $k$.

Table 5.1 gives average convergence rates for a $10^{-6}$ reduction of the residual measured in the Euclidian norm. Four methods are compared: multiplicative multi-grid (mmg), multiplicative multi-grid as preconditioner in cg (cg+mmg), standard additive multi-grid as preconditioner in cg (cg+amg) and the new method described above with adaptively chosen damping factors (cg+damg), which is also used as preconditioner. The mesh width is $h=1/64$ and a damped Jacobi smoother is used in Table 5.1.

**Table 5.1.** Convergence rates for different methods applied to the model problem with $h=1/64$ and damped Jacobi smoother ($\omega=1/2$)**.**

| $\nu=\nu_1+\nu_2$ | mmg | cg+mmg | cg+amg | cg+damg |
|---|---|---|---|---|
| 1 | 0.75 | - | 0.49 | 0.50 |
| 2 | 0.56 | 0.21 | 0.44 | 0.43 |
| 3 | 0.44 | - | 0.41 | 0.40 |
| 4 | 0.35 | 0.11 | 0.40 | 0.37 |
| 5 | 0.30 | - | 0.39 | 0.35 |
| 6 | 0.26 | 0.075 | 0.38 | 0.33 |
| 7 | 0.23 | - | 0.38 | 0.32 |
| 8 | 0.21 | 0.058 | 0.37 | 0.31 |
| 9 | 0.19 | - | 0.37 | 0.30 |
| 10 | 0.18 | 0.047 | 0.37 | 0.30 |
| 20 | 0.099 | 0.025 | 0.35 | 0.25 |
| 100 | 0.023 | 0.0035 | 0.33 | 0.18 |

The values can be compared to those in Tables 4.1 - 4.3. Convergence rates for the standard multiplicative cycle are worse than the theoretical values for the two-grid cycle since we are using a V-cycle here. The values for the additive variant with level-

independent damping correspond nicely to the theory.

## 5.2 Anisotropic Model Problem

In this section, we apply the methods to the anisotropic model equation

(5.4) $\varepsilon u_{xx} + u_{yy} = f$, in $\Omega = (0, 1)^2$ with Dirichlet boundary conditions.

The same discretization as above is used.

Tables 5.2 -5.4 show the results for $\varepsilon=10^{-2}$, $10^{-4}$ and $10^{-6}$. The Jacobi smoother has been replaced by an $ILU_\beta$ scheme from [19] in order to have a smoother which is an exact solver for the limit case. The additive method with adaptively chosen damping factors is clearly superior to the standard additive method. When $\varepsilon$ becomes larger, fewer smoothing steps are sufficient to take advantage of the adaptively chosen damping factors.

**Table 5.2.** Anisotropic model equation with $h=1/64$, $\varepsilon=10^{-2}$, $ILU_\beta$ ($\beta=0.35$).

| $\nu=\nu_1+\nu_2$ | mmg | cg+mmg | cg+amg | cg+damg |
|---|---|---|---|---|
| 1 | 0.17 | - | 0.42 | 0.27 |
| 2 | 0.09 | 0.021 | 0.41 | 0.20 |
| 3 | 0.041 | - | 0.39 | 0.17 |
| 4 | 0.034 | 0.0049 | 0.37 | 0.13 |
| 5 | 0.023 | - | 0.34 | 0.12 |
| 6 | 0.019 | 0.0025 | 0.32 | 0.11 |
| 7 | 0.015 | - | 0.31 | 0.099 |
| 8 | 0.013 | 0.00099 | 0.30 | 0.095 |
| 9 | 0.010 | - | 0.30 | 0.086 |
| 10 | 0.0085 | 0.00066 | 0.29 | 0.078 |
| 20 | 0.0019 | 0.00014 | 0.30 | 0.15* |
| 100 | 6e-7 | 4.3e-9 | 0.096 | 0.001 |

**Table 5.3.** Anisotropic model equation with $h$=1/64, $\varepsilon$=$10^{-4}$, $ILU_\beta$ (β=0.35**).**

| $\nu=\nu_1+\nu_2$ | mmg | cg+mmg | cg+amg | cg+damg |
|---|---|---|---|---|
| 1 | 0.031 | - | 0.32 | 0.21 |
| 2 | 0.0065 | 0.00093 | 0.27 | 0.095 |
| 3 | 0.00053 | - | 0.24 | 0.052 |
| 4 | 0.00026 | 7.1e-5 | 0.18 | 0.007 |
| 5 | 6.3e-7 | - | 0.16 | 6.3e-7 |
| 6 | 2.8e-7 | 2.5e-7 | 0.14 | 2.5e-7 |
| 7 | 1.4e-7 | - | 0.12 | 1.2e-7 |
| 8 | 6.7e-8 | 6e-8 | 0.10 | 6e-8 |
| 9 | 3.3e-8 | - | 0.092 | 2.9e-8 |
| 10 | 1.6e-8 | 1.4e-8 | 0.083 | 1.4e-8 |
| 20 | 1.3e-11 | 1.2e-11 | 0.051 | 1.2e-11 |
| 100 | 9e-16 | 7e-16 | 0.051 | 1.3e-15 |

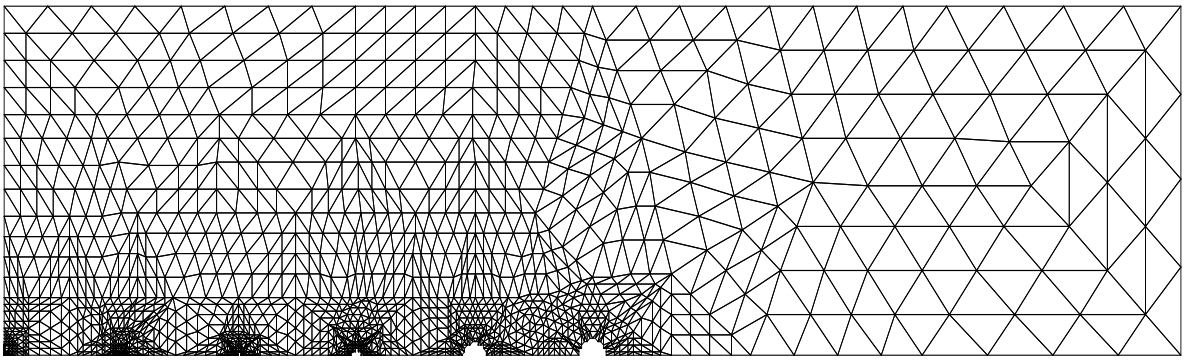
**Table 5.4.** Anisotropic model equation with $h$=1/64, $\varepsilon$=$10^{-6}$, $ILU_\beta$ (β=0.35)

| $\nu=\nu_1+\nu_2$ | mmg | cg+mmg | cg+amg | cg+damg |
|---|---|---|---|---|
| 1 | 0.00021 | - | 0.27 | 0.059 |
| 2 | 6.3e-8 | 3.9e-8 | 0.26 | 6.8e-8 |
| 3 | 2.1e-8 | - | 0.24 | 1.9e-8 |
| 4 | 1e-8 | 9.4e-9 | 0.18 | 9.4e-9 |
| 5 | 5.1e-9 | - | 0.16 | 4.6e-9 |
| 6 | 2.5e-9 | 2.3e-9 | 0.14 | 2.3e-9 |
| 7 | 1.2e-9 | - | 0.12 | 1.1e-9 |
| 8 | 6.1e-10 | 5.5e-10 | 0.10 | 5.5e-10 |
| 9 | 3e-10 | - | 0.092 | 2.7e-10 |
| 10 | 1.5e-10 | 1.3e-10 | 0.083 | 1.3e-10 |
| 20 | 1.2e-13 | 1.1e-13 | 0.051 | 1.1e-13 |
| 100 | 7.3e-16 | 6.3e-16 | 0.051 | 2.5e-15 |

## 5.3 Unstructured Mesh Example

In practice, it is often not realistic to have a smoother which is an exact solver for the limit case. In order to investigate the influence of the quality of the smoother on the convergence rate of the different methods, we solve equation (5.1) on the unstructured mesh given in Figure 5.1. The mesh shown in Figure 5.1 has been refined twice. Dirchlet and Neumann boundary conditions have been used.

Figure 5.1. Mesh used in the unstructured mesh example.



The mesh contains some long and thin triangles which give a similar effect to the discretization of the anisotropic model equation. However, the ILU smoother is now only a fairly good solver. Table 5.5 shows the results of multiplicative and additive multi-grid (with and without adaptively chosen damping factors) and different mesh sizes obtained by dividing each triangle in four similar triangles. All methods are used as preconditioners in a cg-algorithm. It can be seen that multiplicative multi-grid also works very well for a small number of smoothing steps. The effect of choosing the damping factors adaptively in the additive method can be seen only for a large number of smoothing steps, i.e. one must solve the problem very accurately. Since the convergence rate of the ILU smoother is $h$-dependent, the additive methods require a larger number of smoothing steps as the grid is refined.

**Table 5.5.** Convergence rates for the unstructured mesh example, $ILU_\beta(\beta=0.35)$.

| $h$ | $\nu=\nu_1+\nu_2$ | cg+mmg | cg+amg | cg+damg |
|---|---|---|---|---|
| 2 grids, 493 unknowns | 1 | - | 0.29 | 0.29 |
| | 2 | 0.045 | 0.25 | 0.24 |
| | 4 | 0.022 | 0.2 | 0.2 |
| | 6 | 0.015 | 0.2 | 0.17 |
| | 10 | 0.0091 | 0.14 | 0.13 |
| | 20 | 0.0025 | 0.085 | 0.043 |
| | 100 | 1.4e-6 | 0.033 | 0.00022 |
| 3 grids, 1825 unknowns | 1 | - | 0.37 | 0.36 |
| | 2 | 0.058 | 0.33 | 0.31 |
| | 4 | 0.03 | 0.31 | 0.27 |
| | 6 | 0.021 | 0.29 | 0.25 |
| | 10 | 0.013 | 0.27 | 0.22 |
| | 20 | 0.006 | 0.23 | 0.18 |
| | 100 | 0.00055 | 0.12 | 0.027 |
| 4 grids, 7009 unknowns | 1 | - | 0.42 | 0.41 |
| | 2 | 0.061 | 0.38 | 0.36 |
| | 4 | 0.033 | 0.36 | 0.31 |
| | 6 | 0.022 | 0.35 | 0.29 |
| | 10 | 0.014 | 0.33 | 0.26 |
| | 20 | 0.0069 | 0.3 | 0.21 |
| | 100 | 0.00099 | 0.22 | 0.12 |
| 5 grids, 27457 unknowns | 1 | - | 0.44 | 0.44 |
| | 2 | 0.06 | 0.41 | 0.39 |
| | 4 | 0.033 | 0.38 | 0.34 |
| | 6 | 0.023 | 0.37 | 0.32 |
| | 10 | 0.014 | 0.35 | 0.29 |
| | 20 | 0.0072 | 0.33 | 0.24 |
| | 100 | 0.0011 | 0.27 | 0.17 |

## 5.4 Conclusions

The numerical results fully confirm the analysis from §4 showing that the multiplicative method is superior to the additive one. In most cases, in particular for the unstructured grid example which is the most realistic one of those considered here, multiplicative multi-grid is faster by more than one order of magnitude than the additive variants without needing additional work.

## References

[1] Bank, R.E., Dupont, T., Yserentant, H.: The hierarchical basis multigrid method. Numer. Math. 52, 427-458, 1988

[2] Bastian, P.:Locally Refined Solution of Unsymmetric and Nonlinear Problems, in: Incomplete Decompositions. In Hackbusch, W., Wittum, G.: Incomplete Decompositons - Theory, Algorithms, and Applications, NNFM, vol. 41, Vieweg, Braunschweig, 1993

[3] Bastian, P., Wittum, G: Adaptive multigrid methods: The UG concept. In: Hackbusch, W., Wittum, G. (eds.): Adaptive Methods - Algorithms, Theory and Applications, NNFM vol. 46, Vieweg, Braunschweig, 1994

[4] Bastian, P., Wittum, G: On robust and adaptive multigrid methods In: Hemker, P., Wesseling, P. (eds.): Multigrid Methods IV, ISNM, Birkhäuser, Basel, 1994

[5] Bey, J., Wittum, G.: On downwind numbered smoothing. ICA-Bericht N 3/95, submitted to Appl. Numer. Math.

[6] Bramble, J.H., Pasciak, J.E., Xu, J.: Parallel multilevel preconditioners. Math. Comput., 55, 1-22 (1990).

[7] Bramble, J.H., Pasciak, J.E., Wang, J., Xu, J.: Convergence estimates for multigrid algorithms without regularity assumptions, Math Comp 57, 23-45, 1991

[8] Ecker, A., Zulehner, W.: On the smoothing property for the non-symmetric case. Institutsbericht Nr. 489, Institut für Mathematik, Johannes Kepler Universität Linz, April 1995.

[9] Greenbaum, A.: A multigrid method for multiprocessors. Appl. Math. Comp. 19, 75-88, 1986

[10] Hackbusch, W.: Multi-grid methods and applications. Springer, Berlin, Heidelberg, 1985.

[11] Hackbusch, W.: Elliptic Differential Equations. Springer, New York, 1992.

[12] Hackbusch, W.: Iterative solution of large sparse systems of equations.

Springer, New York, 1994.

[13] Hackbusch, W.: A note on Reuskens lemma. Computing 55 (1995), 181-189,

[14] Neuß, N.: Homogenisierung und Mehrgitterverfahren. Dissertation, Heidelberg, 1995

[15] Reusken, A.: A new lemma in multigrid convergence theory. Report RANA 91-07, Eindhoven, 1991

[16] Rivara, M.C.: Algorithms for refining triangular grids for adaptive and multigrid techniques. Int. J. Numer. Meth. Engrg. 20 (1984) 745-756

[17] Ruge, J., Stüben, K.: Algebraic multigrid. In: S. McCormick (ed.): Multigrid methods, ch. 4, 73-130. SIAM, Philadelphia, 1987

[18] Wesseling, P.: Theoretical and practical aspects of a multigrid method.SIAM J. Sci. Statist. Comp. 3 (1982), 387-407.

[19] Wittum, G.: On the robustness of ILU-smoothing.SIAM J. Sci. Stat. Comput., 10, 699-717, (1989)

[20] Wittum, G.: Linear iterations as smoothers in multi-grid methods. Impact of Computing in Science and Engineering, 1, 180-215 (1989).

[21] Wittum, G.: Spektralverschobene Iterationen, ICA-Bericht N95/5, ICA, Universität Stuttgart, 1995, submitted to Numer. Math.

[22] Xu, J.: Iterative methods by space decomposition and subspace correction: A unifying approach. SIAM Review, 34(4), 581-613, 1992.

[23] Yserentant, H.:Über die Aufspaltung von Finite-Element -Räumen in Teilräume verschiedener Verfeinerungsstufen.Habilitationsschrift, RWTH Aachen, 1984.