

# Multigrid Tau-Extrapolation for Nonlinear Partial Differential Equations

K. Bernert\*

M. Jung\*

U. Rde†

## Abstract

*Multigrid* methods are among the fastest solvers for partial differential equations. Using the tau-extrapolation principle of Brandt, the multilevel structure can be used *additionally* to obtain higher order approximations.

In this paper we examine three different approaches with special emphasis on nonlinear partial differential equations. In all cases, higher order is achieved by implicitly using the extrapolation principle, that is by exploiting the information between the grids with different discretization parameters.

**Key words:** finite differences, finite elements, extrapolation, multigrid, Navier-Stokes equations.

**AMS subject classifications:** 35Q30, 65B99, 65N06, 65N55, 76D05.

## 1 Introduction

High accuracy solutions of differential equations can only be obtained efficiently when the smoothness of the solution is exploited by high order discretization. High order methods may provide sufficient accuracy with a much smaller discrete system. On the other hand, in practical applications, the discrete systems are often still large, and good overall efficiency requires fast solvers. For problems in two or three spatial dimensions multilevel and multigrid techniques are often used. Since these algorithms use a hierarchy of successively refined meshes, it is obvious that methods using such a mesh structure for both constructing a

*fast solver* and a *high order discretization* are particularly attractive. In this paper we will study three different but related approaches to such algorithms.

The starting point of our consideration is a hierarchy of meshes, on each of which the differential equation

$$(1) \quad Au = f$$

is discretized by a basic low order discretization. In the first part of the paper we will consider finite difference techniques; however, as shown later, the basic principle also applies to finite element discretizations. In the simplest case, we study just two mesh levels and denote the corresponding discrete systems by  $A_h u_h = f_h$  and  $A_H u_H = f_H$ , respectively. On uniform meshes, typically  $H = 2h$ , and thus  $u_h$  involves about four times as many unknowns as  $u_H$  in a 2D example.

A multigrid algorithm uses the  $H$ -grid approximation to accelerate the convergence of iterative methods for the  $h$ -grid problem by using *coarse grid corrections* to a current fine grid iterate  $u_h$  of the form

$$(2) \quad \begin{aligned} \tilde{f}_H &= f_H + \tau_h^H(u_h) \\ \text{with } \tau_h^H(u_h) &= A_H \hat{R}_h^H u_h - R_h^H A_h u_h \end{aligned}$$

$$(3) \quad \text{Solve } A_H u_H = \tilde{f}_H$$

$$(4) \quad u_h := u_h + P_H^h(u_H - \hat{R}_h^H u_h).$$

The algorithm (2 – 4) is the core of a so-called multigrid full approximation scheme (FAS) (see Brandt [2]). The operators  $\hat{R}_h^H$ ,  $R_h^H$  are restrictions from the fine to the coarse grid and  $P_H^h$  is a prolongation (interpolation) from the coarse to the fine grid. The  $\tau$ -term can be interpreted as a correction to the level  $H$  equation to make its solution coincide with the equation on level  $h$ .

$\tau$ -extrapolation now uses this basic algorithm with a slight modification. In (2),  $\tau_h^H(u_h)$  is multiplied by an extrapolation parameter  $\omega$  to become  $\omega \tau_h^H(u_h)$ , thus the name  $\tau$ -extrapolation. In a finite difference setting, this modification is justified, when  $\tau_h^H(u_h)$  has a suitable asymptotic expansion.

\*Fakultät für Mathematik, TU Chemnitz-Zwickau, D-09107 Chemnitz

†Fakultät für Informatik, TU München, Arcisstr. 21, D-80290 München

With this assumption and the trivial modification, a straightforward multigrid algorithm can be shown to converge to a higher order approximation of the solution (see [4]), provided the components of the algorithm satisfy certain compatibility conditions. These conditions will be explained in section 2 of this paper, where a number of experiments for finite difference discretizations will also be presented.

In section 3 and 4 we will then study alternative approaches to  $\tau$ -extrapolation. The method of section 3 is derived from the formulation of a differential equation as a minimization problem. The resulting algorithm turns out to be a special case of the algorithms in section 2; however, the new derivation allows an extension to more general mesh structures, in particular to unstructured finite element meshes. This is similar to the approach developed in section 4, where the starting point is the observation that  $\tau$ -extrapolation can be interpreted as an implicit way to construct higher order finite element stiffness matrices. These two interpretations permit an application of the  $\tau$ -extrapolation principle to a much wider class of problems; including those where sufficient smoothness is only present locally, and where a higher order discretization must be augmented with adaptive techniques to resolve local singularities.

In all these cases the  $\tau$ -extrapolation algorithm maintains its structural simplicity. Higher order is obtained implicitly, without ever constructing complicated difference stencils or high order finite elements.  $\tau$ -extrapolation is also naturally combined with the multigrid principle so that it automatically provides a very efficient solver for the discrete systems. The combination of these features potentially makes  $\tau$ -extrapolation one of the most efficient approaches to the high accuracy solution of differential equations.

## 2 Classical $\tau$ -extrapolation

In this section we will study the classical  $\tau$ -extrapolation algorithm introduced by Brandt [2]. Our presentation is based on the multigrid terminology and assumes a basic knowledge of multigrid principles. For a more detailed presentation see [1].

### 2.1 The basic idea for linear problems

Let  $R_H$  be a restriction operator projecting the right-hand side  $f$  of the differential equation (1) into the image space

of the discretized operator  $A_H$ , i.e.

$$(5) \quad f_H = R_H f.$$

The truncation error of the discrete problem is given by inserting a projection  $\hat{R}_H u^*$  of the exact solution in the discrete equation:

$$(6) \quad \tau_H(u^*) = A_H \hat{R}_H u^* - f_H = A_H \hat{R}_H u^* - R_H A u^*.$$

Operators  $R_H$  and  $\hat{R}_H$  may coincide, if preimage and image of  $A$  coincide.

The approximation order  $p$  of a discrete problem is defined by the relation

$$\tau_H(u) = A_H \hat{R}_H u - R_H A u = O(H^p) \quad \text{for } u \in C^{o+p},$$

where  $o$  is the order of the differential operator  $A$ .

An approximation of  $\tau_H(u^*)$  up to order  $\alpha > p$  can be used to improve the accuracy of the original discrete problem to order  $\alpha$ .

**Lemma 2.1** *For the right hand side of  $A_H u_H = \tilde{f}_H$  let  $\tilde{f}_H = f_H + \tau_H(u^*) + O(H^\alpha)$  with  $\alpha > p$  and suppose (5), (6) and  $\|A_H\|^{-1} \leq M$ ,  $M = \text{const.}$ , Then it follows that  $\|u_H - \hat{R}_H u^*\| = O(H^\alpha)$ .*

**Proof** The difference of  $A_H u_H = \tilde{f}_H$

$$\text{and } A_H \hat{R}_H u^* = f_H + \tau_H(u^*)$$

$$\text{gives } A_H(u_H - \hat{R}_H u^*) = O(H^\alpha)$$

$$\text{i.e. } \|(u_H - \hat{R}_H u^*)\| \leq M \cdot O(H^\alpha) \quad \square$$

A direct application of Lemma 2.1. presupposes  $\|\hat{R}_H u^* - I_H u^*\| \leq O(H^\alpha)$ , where  $I_H$  is an injection. This can be fulfilled most easily by  $\hat{R}_H = I_H$ . A correction of the right hand side  $f_H$ , which estimates  $\tau_H(u^*)$  with an error of order  $\alpha > p$ , improves the accuracy of the solution of  $A_H u_H = f_H$  to the same order.

In the full approximation scheme (FAS, see also (2-4)) the coarse grid problems can be written in the form

$$(7) \quad A_H u_H = f_H + \tau_h^H(u_h), \quad \tau_h^H(u_h) = A_H \hat{R}_h^H u_h - R_h^H A_h u_h.$$

The correction term on the right-hand side can be considered as an estimate of the approximation error based on the solution on the finer grid. If

$$(8) \quad \hat{R}_h^H \hat{R}_h = \hat{R}_H \quad \text{and} \quad R_h^H R_h = R_H$$

one can show that the accuracy of the solution  $u_H$  on the coarse grid is the same as that of  $u_h$  on the fine grid (see [1]). However, a higher order of accuracy can not be obtained in this way. Taking into consideration Lemma 2.1. we need a correction term which is a better approximation to  $\tau_h^H(u^*)$ . Such an approximation is given by

**Lemma 2.2** Assume (7),  $H = 2h$ , and

$$(A1) \quad \tilde{u}_h = \hat{R}_h(u^* + \xi), \quad \xi = O(H^q), \quad \xi \in C^{(o+p)}$$

$$(A2) \quad \tau_h(u) = O(H^p) \text{ for } u \in C^{(o+p)}$$

$$(A3) \quad R_h^H \tau_h(u^*) = \frac{1}{2^p} \tau_H(u^*) + O(H^\beta), \quad \beta > p.$$

Then it follows

$$\frac{2^p}{2^p-1} \tau_h^H(\tilde{u}_h) = \tau_H(u^*) + O(H^\alpha) \text{ with } \alpha = \min(p+q, \beta).$$

**Proof** Based on

$$\begin{aligned} \tau_h^H(\hat{R}_h u^*) &= A_H \hat{R}_h^H \hat{R}_h u^* - R_h^H A_h \hat{R}_h u^* \\ &= (A_H \hat{R}_H u^* - R_H A u^*) - (R_h^H A_h \hat{R}_h u^* - R_h^H R_h A u^*) \\ &= \tau_H(u^*) - R_h^H \tau_h(u^*) = \tau_H(u^*) - \frac{1}{2^p} \tau_H(u^*) + O(H^\beta) \\ &= \frac{2^p-1}{2^p} \tau_H(u^*) + O(H^\beta) \end{aligned}$$

we obtain

$$\begin{aligned} \tau_h^H(\tilde{u}_h) &= \tau_h^H(\hat{R}_h(u^* + \xi)) = \tau_h^H(\hat{R}_h u^*) + \tau_h^H(\hat{R}_h \xi) \\ &= \frac{2^p-1}{2^p} \tau_H(u^*) + O(H^\beta) + O(H^{p+q}). \quad \square \end{aligned}$$

Lemma 2.1. and Lemma 2.2. together lead to the following improved formulation of the problem for the coarse grid

$$(9) \quad A_H u_H = f_H + \frac{2^p}{2^p-1} \tau_h^H(\tilde{u}_h).$$

Equation (9) gives a higher order approximation on the grid with the discretization parameter  $H = 2h$ :

$$u_H = \hat{R}_H u^* + O(H^\alpha) \text{ with } \alpha > p.$$

Using the usual formulation (7) of the problems for coarser grids, the improved accuracy can be carried over up to the coarsest grid.

In the correction step the solution on grid  $H$  can be transferred to grid  $h$  in such a way that the order  $\alpha$  for the low-frequency part remains unchanged.

Equation (9) implemented in a multigrid method is the classical form of  $\tau$ -extrapolation. It can be completed by a post-smoothing correction, and by a fine grid correction for the first smoothing step on a new finest grid in the full approximation scheme.

These corrections, however, are not essential for the higher convergence order and can be omitted.

**Remark** For the sake of simplicity, the idea of  $\tau$ -extrapolation was explained for linear problems only. The very same algorithm is applicable for nonlinear problems too, however.

## 2.2 Conditions for the grid transfer operators with respect to $\tau$ -extrapolation

A successful application of  $\tau$ -extrapolation requires a careful tuning of all multigrid components. Special attention must be paid to the grid transfer operators in the  $\tau$ -extrapolation step of the multigrid algorithm.

### 2.2.1 Prolongation

In the FMG-algorithm a prolongation is needed in two different situations. In the correction step of the multigrid iteration the correction of a fine grid solution  $u_h$  must be interpolated from the coarse grid, and the initial guess for the iteration on a new finest grid must be interpolated with a possibly different operator (FMG-prolongation). In both situations it is necessary to preserve the accuracy reached on the coarser grid and to avoid introducing large high-frequency errors on the finer grid.

First, we consider the correction step (4). If the restriction operator is not an injection operator ( $\hat{R}_h^H \neq I_h^H$ ), i.e.  $(\hat{R}_h^H - I_h^H)u_h = O(H^s)$  the term  $\hat{R}_h^H u_h$  causes an error; which, after the correction step, can be found in the low-frequency part of the error of  $u_h$ . Operator  $P_H^h$  primarily produces high-frequency errors. These errors depend on the order of magnitude of the function to be interpolated and on the interpolation formula. The interpolation error of a smooth function by an interpolation polynomial of order  $(n-1)$  is of order  $O(H^n)$  (the proof is given in [20]). Table 1 summarizes the order of prolongation errors and conditions for  $s$  and  $n$  which must be fulfilled.

Multigrid-method	Low-frequency error of $\hat{R}_h^H u_h$ , Restriction by		High-frequency error of $P_H^h(u_H - \hat{R}_h^H u_h)$
	Averaging ( $s < \infty$ )	Injection ( $s = \infty$ )	
without $\tau$ -extrapolation	$p + s$	$\infty$	$p + n$
Conditions	-	-	$n \geq o$
with $\tau$ -extrapolation	$p + s$	$\infty$	$p + n$
Conditions	$s \geq \alpha - p$	-	$n \geq o$

Parameters:  $p$  – approximation order of the discrete operator,  $\alpha$  – order of accuracy of the MG-method with  $\tau$ -extrapolation  $s$  – order of accuracy of  $\hat{R}_h^H$ ,  $n$  – order of the error for polynomial prolongation with degree  $(n-1)$ ,  $o$  – order of the differential equation to be solved

Table 1: Order of errors caused by prolongation and conditions for  $s$  and  $n$

The FMG-prolongation  $\tilde{P}_h^{\frac{h}{2}}$  produces the initial solution for a MG-cycle on a new grid. The quality of this interpolation has a great influence on the accuracy of the whole method. An essential difference to the prolongation  $P_H^h$  is that we have to interpolate the solution, i.e. a function with an order of magnitude  $O(1)$ , and not a correction to the solution.

If  $o$  is the order of the differential operator, the errors caused by the interpolation of the solution  $u_h$  should be at least  $o$  orders smaller than the defects. This can be obtained by

$$n - o \geq p, \text{ i.e. } n \geq p + o$$

in the case of the FMG-algorithm, without  $\tau$ -extrapolation; and in the case of  $\tau$ -extrapolation, without fine grid correction

and by  $n - o \geq \alpha$ , i.e.  $n \geq \alpha + o$

in the case of  $\tau$ -extrapolation, with post-smoothing- and fine-grid correction. The conditions do not guarantee smooth defects after prolongation; but oscillations decrease with the same order as the defects, if the grids become increasingly finer.

With the usual number of pre-smoothing steps oscillations in the defects can not be smoothed completely. If the restriction  $A_h u_h$  in (7) is performed with an averaging operator the remaining wiggles can be tolerated because  $R_h^H A_h u_h$  operates like a filter which removes them. However, if  $R_h^H$  is an injection operator, the conditions  $n \geq p + o$  and  $n \geq \alpha + o$  can be insufficient. In this case the interpolation should be taken one or two orders higher.

### 2.2.2 Combination of the restriction operators $\hat{R}_h^H$ and $R_h^H$ in the context of $\tau$ -extrapolation

In the MG-algorithm without  $\tau$ -extrapolation the two restriction operators  $\hat{R}_h^H$  and  $R_h^H$  can be chosen independently. For the restriction of  $u_h$ , injection  $\hat{R}_h^H = I_h^H$  is sufficient, because  $u_h$  is a smooth function. For the restriction of  $A_h u_h$ , however, an averaging operator is a better choice, because the defects are often less smooth.

In the case of  $\tau$ -extrapolation the two restriction operators have an effect on the estimation of the truncation error (cf. Lemma 2.2). An inappropriate choice of  $\hat{R}_h^H$  and  $R_h^H$  may destroy any effect of  $\tau$ -extrapolation.

In the literature, the great majority of the articles use injection operators both for  $\hat{R}_h^H$  and  $R_h^H$  in the  $\tau$ -extrapolation step. Only in [17], [2], and [10] are some hints at other possibilities given. In the case of staggered grids, injection for  $\hat{R}_h^H$  and  $R_h^H$  is excluded, for points of the coarser grid are not collocated with points of the finer

Case of restriction	Conditions for		
	$\alpha \geq p+1$	$\alpha \geq p+2$	$\alpha \geq 4$ if $p=2$
<b>A:</b> (Injection-Injection) $\hat{R}_h^H = I_h^H, R_h^H = I_h^H$		$q \geq 2$ $r \geq p+2$	$q \geq 2$ $r \geq 4$
<b>B:</b> (Injection-Averaging) $\hat{R}_h^H = I_h^H, R_h^H = M_h^H$	$s \geq p+1$	$q \geq 2$ $r \geq p+2$ $s \geq p+2$	$q \geq 2$ $r \geq 4$ $s \geq 4$
if $f^{(s)} \equiv 0$ :		$q \geq 2$ $r \geq p+2$ $s \geq 2$	$q \geq 2$ $r \geq 4$ $s \geq 2$
<b>C:</b> (Equal averaging) $\hat{R}_h^H = M_h^H, R_h^H = M_h^H$ linear Problems:	$t \geq p$	$q \geq 2, n \geq 2$ $r \geq p+2$ $s \geq 2$ $t \geq p+2$	$q \geq 2, n \geq 2$ $r \geq 4$ $s \geq 2$ $t \geq 4$
nonlinear Problems:		see case <b>B</b> , upper part	
<b>D:</b> (Non-equal averaging) $\hat{R}_h^H = \hat{M}_h^H, R_h^H = M_h^H$	in general as case <b>B</b> , upper part		

Parameters:

$\alpha$  - approximation order of the multigrid method with  $\tau$ -extrapolation,  $q$  - approximation order of  $u_h$  before application of  $\tau$ -extrapolation,  $p$  - approximation order of the discrete operator  $A_h$ ,  $r$  - order of the second term of the approximation error  $\tau_H(\xi) = c(x)H^p + O(h^r)$ ,  $n$  - order of accuracy of prolongation with a polynomial of degree  $(n-1)$ ,  $s$  - order of accuracy of the restriction operator  $R_H = M_H$  or minimal order if two different restriction operators are used,  $t$  - order of the second error term of the restriction,  $(M_H - I_H)(\xi) = d(x)H^s + O(H^t)$

Table 2: Conditions in coherence with restriction and  $\tau$ -extrapolation

grid. In this situation it is necessary to work with averaging operators. Such operators can also be favorable for non-staggered grids because they have a stabilizing effect on the  $\tau$ -extrapolation.

Table 2 contains four combinations for the two restriction operators. The given conditions result from a careful analysis of assumption (A3) of Lemma 2.2, see [1].

The first choice for the restriction operators (case A) can be used for both linear and nonlinear problems without essential restraints. If the possible order of accuracy for the solution is not reached, the cause may be non-smooth defects in the  $\tau$ -extrapolation step. A higher order of the FMG-prolongation, or a higher number of smoothing steps, will give better results in this situation.

Because of the averaging  $M_h^H$  in the restriction of  $A_h u_h$ ,

Multigrid algorithm	Type of Cycle	$R_h^H$ is averaging operator		$R_h^H$ is injection operator	
		$P_h^H$ linear ( $n = o$ )	$P_h^H$ cubic	$P_h^H$ linear ( $n = o$ )	$P_h^H$ cubic
MG-algorithm without $\tau$ -extrapolation	V(1, 1)	$\rho = 0.31$	$\rho = 0.10$	$\rho = 0.30$	$\rho = 0.10$
	V(2, 2)	$\rho = 0.11$	$\rho = 0.04$	$\rho = 0.11$	$\rho = 0.04$
FMG-algorithm with $\tau$ -extrapolation	V(1, 1)	$0.56E-09$	<b>0.41E-11</b>	$0.58E-09$	$0.28E-10$
	V(2, 2)	$0.73E-11$	$0.59E-11$	<b>0.16E-11</b>	<b>0.37E-11</b>
	F(1, 1)	<b>0.36E-11</b>	<b>0.34E-11</b>	$0.27E-10$	$0.27E-10$
	F(2, 2)	$0.59E-11$	$0.59E-11$	<b>0.16E-11</b>	<b>0.32E-11</b>

Table 3: Convergence rates and accuracy in dependence on restriction and prolongation

$\tau$ -extrapolation combined with the second choice of restriction operators (case B) is less sensible to the quality of smoothing and FMG-prolongation.

The increased order of accuracy of the restriction  $R_h^H$  ( $s \geq p + 1$  or  $s \geq p + 2$ ) in the case  $f^{(s)} \neq 0$  leads to some additional work.

Restriction operators, according to case C for linear problems, give the advantage of case B without an increased accuracy of the restriction operators  $R_h^H$  and  $\hat{R}_h^H$ . For nonlinear problems, higher order restriction operators are needed.

Case D is the most obvious generalization of case A. The restriction operators are chosen independently as averaging operators with high accuracy (close to injection operators).

### 2.3 One-dimensional test calculations

Many properties of the multigrid method are independent of the dimension of the problem to be solved. Therefore, one-dimensional test problems are a useful tool to check theoretical results about multigrid algorithms; even though multigrid is usually used for higher dimensional problems. In [1] many tests are documented. At this place only a small selection can be given.

We consider the linear boundary value problem

$$(10) \quad -u'' = f(x) = \pi^2 \cos(\pi x), \quad u(-1) = u(1) = 0$$

with the solution  $u(x) = \cos x$

and the nonlinear Burger's equation

$$(11) \quad uu_x - \nu u_{xx} = 0, \quad u(-1) = \tanh\left(\frac{1}{2\nu}\right), \quad u(1) = -\tanh\left(\frac{1}{2\nu}\right)$$

with the solution  $u(x) = -\tanh\left(\frac{x}{2\nu}\right)$  for  $\nu = 0.1, 0.005$ .

The two problems are discretized by standard central differences of second order. We use a sequence of non-staggered, equally spaced grids.

#### Test 1 Influence of restriction and prolongation

Table 3 refers to the linear problem with 1025 points on the finest grid. It shows convergence rates  $\rho$  for the normal MG-algorithm with eight grids and the largest errors for the FMG-method. In the notation  $T(i, j)$  for the type of the cycle we have  $T = \begin{cases} V, & \text{V-cycle} \\ F, & \text{F-cycle} \end{cases}$ ;  $i, j$  are the numbers of pre- and post-smoothing iterations. To avoid any negative influence on the FMG-results, quintic FMG-prolongation was used.

Remark: For the FMG-algorithm the headline " $R_h^H$  is injection operator" in Table 3 is meant for the current finest grid only. On coarser grids the defects were restricted, as usual, by an averaging operator. To perform the  $\tau$ -extrapolation, the restriction operator  $\hat{R}_h^H$  on the current finest level was chosen in agreement with  $R_h^H$  (see Table 2, cases A and C). On coarser grids, and in the MG-algorithm without  $\tau$ -extrapolation,  $\hat{R}_h^H = I_h^H$  was used.

The numbers in Table 3 can be explained in the following way:

1. Taking into account that the discretization error of the original second order scheme on the finest grid is  $0.31E-05$  all results with  $\tau$ -extrapolation in the table are much better than second order. Indeed, it can be shown that most of them are equal or close to fourth order.
2. Table 3 shows that the condition  $n \geq o$  (see Table 1) for the minimal accuracy of prolongation is correct. In fact, linear prolongation in the case  $o = 2$  is possible; but in some situations the potential of the numerical

algorithms is not treated fully in this way. The convergence rates of the MG-algorithm with cubic prolongation are better. In the case of  $\tau$ -extrapolation for the  $V(1, 1)$ -cycle the fourth order of accuracy occurs only on the first grids; on finer grids the order reduces to a value between two and three. However, this effect can easily be removed by increasing the number of smoothing steps.

3. For the accuracy of the FMG-algorithm with  $\tau$ -extrapolation, Table 3 shows the following behavior: If the amount for smoothing is small, restriction with averaging operators gives the better results. In the case of more smoothing iterations, restriction by injection leads to more accurate solutions. An explanation of the observed effect can be given by the accuracy of the error estimation and by the low-frequency error, which is proceeded from a restriction with  $s = \alpha - p$  (see Table 1).

### Test 2 $\tau$ -extrapolation for nonlinear problems

Apart from the fact that a combination of restriction operators according to case C is not favorable, the FMG-algorithm with  $\tau$ -extrapolation works in the nonlinear case too. The only problem is to find out the optimal structure of the multigrid cycle.

For our test problem we have to pay attention to some specialties: With  $\nu \gg 1$  the transition from the left boundary value  $u(-1) \approx 1$  to the right boundary value  $u(1) \approx -1$  takes place in a very narrow region. This excludes grids, which do not have enough grid-points in this region. As an orientation we can take the stability constraint for the difference scheme  $Re_h = \frac{uh}{\nu} < 2$ . For the number of grid-points this means  $n > \nu^{-1}$ , i.e.  $n_{min} = 17$  for  $\nu = 0.1$  and  $n_{min} = 257$  for  $\nu = 0.005$ .

**Results for  $\nu = 0.1$**  The numbers in Table 4 are the quotients of consecutive maximal errors. A  $V(1, 1)$ -cycle in the case of 6 grids (from 33 to 1025 points) has fourth order of convergence. If a 7th grid with 17 points is added, the order of convergence is not much larger than two. Even with a  $V(3, 3)$ -cycle the order of convergence is below three. A  $F(1, 1)$ -cycle is clearly superior for nonlinear problems. In the case of seven grids, it gives fourth order of convergence beginning with the third grid (65 points). If eight grids are used (the coarsest has 9 points only), fourth order is reached on the last four grids.

**Results for  $\nu = 0.005$**  Table 5 presents results for some variants of the FMG-algorithm. Even in the case of the  $V(3, 3)$ -cycle some additional MG-cycles on the finest grid reduced the error to a value of about  $0.5E-5$ . However, the nonlinearity of a problem should be treated already on

Number of grid-points	Type of MG-algorithm				
	$V(1, 1)$ 6 grids	$V(1, 1)$ 7 grids	$V(3, 3)$ 7 grids	$F(1, 1)$ 7 grids	$F(1, 1)$ 8 grids
9					—
17		—	—	—	1.5
33	—	5.2	7.6	5.1	6.7
65	19.8	4.8	6.0	26.4	7.3
129	16.1	5.1	7.0	333.5	14.8
257	20.5	5.0	6.8	5.1	22.6
513	15.1	5.1	7.0	16.2	37.0
1025	19.3	5.1	7.0	49.6	52.0
Error	$0.74E-8$	$0.40E-5$	$0.62E-6$	$0.37E-9$	$0.79E-8$

Table 4: Convergence of the FMG-method for different cycles

coarser grids. This can be tried by using the F-cycle and by the modification described in remark 2 below, as the last two calculations show.

### Remarks on Table 5:

1. Taking into consideration the structure of the solution, three pre- and post-smoothing iterations were performed only in the small range, where the solution actually changes. Outside of this region, one iteration was sufficient. A better investigated variant of such a strategy can be found in [9].
2. If nonlinearity and/or the use of relative coarse grids cause a noticeable change of the solution from one grid to the next, then it is advantageous to perform  $\gamma > 1$  MG-cycles on each grid level of the FMG-method. On the last two grids, however, it was possible to work with  $\gamma = 1$  without loss of accuracy. Moreover, it is possible to perform the first MG-cycle on a new grid without  $\tau$ -extrapolation (see [19]). Obviously in the second MG-cycle the error can be estimated more precisely than immediately after FMG-prolongation and pre-smoothing. During the two last calculations on grid 2 and grid 3, the  $\tau$ -extrapolation was done only in the second F-cycle.

## 2.4 Solution of Navier-Stokes equations with $\tau$ -extrapolation

The change from one-dimensional test problems to the case of the two-dimensional Navier-Stokes equations includes the increase of the space dimension and the change from one equation to a system of equations.

FMG-method	Maximal error on				
	grid 1	grid 2	grid 3	grid 4	grid 5, 2049 points
$F(3, 3)$ -cycle without $\tau$ -extrapolation	0.62E+0	0.67E-1 (9.2)	0.37E-1 (1.8)	0.49E-2 (7.6)	0.14E-2 (3.5)
$V(3, 3)$ -cycle	0.62E+0	0.24E+0 (2.5)	0.18E-1 (13.6)	0.17E-2 (10.7)	0.28E-3 (5.9)
$F(3, 3)$ -cycle	0.62E+0	0.24E+0 (2.5)	0.43E-1 (5.7)	0.17E-2 (25.2)	0.33E-4 (51.6)
$F(3, 3)$ -cycle, $\gamma > 1$ MG-cycles on grids 2 – 4					
$\gamma = 2$ in FMG:	0.62E+0	0.22E+0 (2.7)	0.12E-1 (19.2)	0.16E-3 (74.4)	0.63E-5 (24.6)
$\gamma = 3$ in FMG:	0.62E+0	0.53E-1 (11.5)	0.54E-2 (9.8)	0.13E-3 (42.0)	0.22E-5 (58.2)
$\gamma = 2$ in FMG, see remark 2:	0.62E+0	0.11E+0 (5.5)	0.35E-2 (31.7)	0.22E-3 (16.5)	0.16E-5 (132.1)

Table 5: Convergence of the FMG-method for Burgers equation with  $\nu = 0.005$

Examples of the use of  $\tau$ -extrapolation for scalar equations on multidimensional domains can be found in [18] for the Poisson equation over the unit square; or in [22] for the same equation on a three-dimensional domain, which is defined by three overlapping cylindrical grids.

Besides the use for scalar equations (Poisson equation, non-linear potential equation – with special respect to Neumann boundary conditions) in [19],  $\tau$ -extrapolation is applied to the solution of the shell problem for the calculation of stresses and deformations in weakly curved thin elastic shells. This problem leads to a system of four Poisson-like equations with nonlinear coupling.

In most cases it was possible to improve the convergence order from two to four, or to a value close to four, by implementing the  $\tau$ -extrapolation. For smooth solutions this should be attainable for the Navier-Stokes equations too.

### 2.4.1 Implementation of the $\tau$ -extrapolation

The Navier-Stokes equations are considered in the form

$$\begin{aligned}
 \nabla \cdot \mathbf{u} \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p &= \mathbf{f} & \text{in } \Omega \\
 \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega \\
 \mathbf{u} &= \mathbf{u}_\Gamma & \text{on } \partial\Omega
 \end{aligned}
 \tag{12}$$

over a rectangular domain  $\Omega$ . In these equations  $\mathbf{u}$  stands for the velocity with components  $u$  and  $v$ ,  $p$  denotes the pressure,  $\nu$  is the kinematic viscosity and  $\mathbf{f}$  is an external force with components  $f_x$  and  $f_y$ . Equation (12) is discretized by a second order difference approximation on staggered grids.

A detailed presentation of the MG-method, which was

used for the Navier-Stokes equations is not intended at this place. Only components, which are related to the  $\tau$ -extrapolation are discussed in the following. At first we consider the calculation of the  $\tau$ -extrapolation terms. The system of discretized equations can be written in the form

$$\begin{aligned}
 A_h(\mathbf{u}_h)\mathbf{u}_h + GRAD_h p_h &= \mathbf{f}_h \\
 DIV_h \mathbf{u}_h &= g_h
 \end{aligned}$$

where the first equation is a vector equation with two components. Right hand sides  $g_h \neq 0$  are introduced by the MG-method. On the finest grid we have  $g_h = 0$ . Using  $\tau$ -extrapolation the problem on a coarser grid is

$$\begin{aligned}
 A_H(\mathbf{u}_H)\mathbf{u}_H + GRAD_H p_H &= R_h^H \mathbf{f}_h + \frac{4}{3} \tau_h^H(\mathbf{u}_h, p_h) \\
 DIV_H \mathbf{u}_H &= R_h^H g_h + \frac{4}{3} \sigma_h^H(\mathbf{u}_h)
 \end{aligned}$$

with

$$\begin{aligned}
 \tau_h^H(\mathbf{u}_h, p_h) &= A_H(\hat{R}_h^H \mathbf{u}_h) \hat{R}_h^H \mathbf{u}_h + GRAD_H \hat{R}_h^H p_h \\
 &\quad - R_h^H A_h(\mathbf{u}_h) \mathbf{u}_h - R_h^H GRAD_h p_h \\
 \sigma_h^H(\mathbf{u}_h) &= DIV_H \hat{R}_h^H \mathbf{u}_h - R_h^H DIV_h \mathbf{u}_h .
 \end{aligned}$$

### 2.4.2 Remarks on restriction and prolongation

Outside the  $\tau$ -extrapolation step, linear restriction operators can be used. In connection with the  $\tau$ -extrapolation the situation is more complicated: Because of the staggered grids for the velocity components only choices C and D from Table 2 are possible.

Grid	FMG – method with tauextrapolation						without tau-	
	$F(1,1)$ -cycle		– two times		– three times		extrapolation	
8 * 8	0.28E-02		0.28E-02		0.28E-02		0.28E-02	
16 * 16	0.81E-02	0.34	0.83E-03	3.35	0.97E-03	2.68	0.97E-03	2.68
32 * 32	0.17E-03	48.06	0.19E-04	44.36	0.74E-05	131.29	0.27E-03	3.62
64 * 64	0.15E-04	11.35	0.11E-05	16.69	0.84E-06	8.79	0.68E-04	3.96
128 * 128	0.40E-05	3.70	0.33E-06	3.43	0.51E-07	16.42	0.17E-04	3.99
256 * 256	0.41E-06	9.68	0.13E-07	24.68	0.19E-08	26.48	0.42E-05	4.00
additional	0.12E-06	33.71	0.46E-08	70.72	0.18E-08	28.31		
single cycles	0.28E-07	143.33	0.15E-08	216.93				

Table 6: Convergence for different FMG-cycles for the Navier-Stokes equations

The restriction of  $\mathbf{u}_h$  and  $A_h \mathbf{u}_h$  for the momentum equation can be performed by the same averaging operator  $\hat{R}_h^H = R_h^H$  with fourth order accuracy.

For the pressure, again a fourth order restriction operator (that means a bicubic interpolation) is needed; because the two restriction operators  $\hat{R}_h^H$  and  $R_h^H$  are not defined on the same grid.

In the case of the continuity equation, linear restriction for  $DIV_h \mathbf{u}_h$  is possible (operator  $R_h^H$ ), independently from the cubic restriction of  $\mathbf{u}$ . Taking into account the restriction of the right hand side of the discrete continuity equation it must be warned of any “better” interpolation for  $DIV \mathbf{u}$ . The components of  $g_h$  have to fulfill a solvability condition (their sum must be zero) and this relation must be conserved by the restriction. This is done by linear restriction, because the values from the finer grid are summed up in groups only.

As in case B this linear restriction causes no errors, because the right hand side of the original problem vanishes. The cubic restriction for  $\mathbf{u}$  leads to a fourth order error, if we compare it with injection. This has no influence on the  $\tau$ -extrapolation.

The errors of prolongation in the MG-algorithm should be no larger than that of second order for the velocity components ( $n \geq o$ ,  $o = 2$ ), and that of first order for the pressure ( $o = 1$ ). With linear prolongation for  $\mathbf{u}$  and  $p$  these conditions are fulfilled.

In the case of FMG-prolongation according to 2.2.1 (condition  $n \geq p + o$ ), fourth order for the velocity ( $p = 2$ ,  $o = 2$ ) and third order for pressure ( $p = 2$ ,  $o = 1$ ) is needed. This means cubic FMG-prolongation for  $\mathbf{u}$  and quadratic or cubic prolongation for the pressure too.

Unsymmetrical interpolation formulae at the boundaries cause larger interpolation errors than symmetrical formulae of the same order in the interior. For this reason, near the boundary, interpolation of an order higher than three was used.

#### 2.4.3 Test calculations

We consider a rotating flow

$$u(x, y) = \sin \pi x \cos \pi y, \quad v(x, y) = -\cos \pi x \sin \pi y$$

in the square  $[0, 1] \times [0, 1]$  with  $\nu = 0.01$ .

Setting this solution in (12) we get the right hand side

$$\begin{aligned} f_x(x, y) &= \pi \sin \pi x (\cos \pi x + 2\pi\nu \cos \pi y) \\ f_y(x, y) &= \pi \sin \pi y (\cos \pi y - 2\pi\nu \cos \pi x) \end{aligned}$$

At the boundary the normal components of the velocity are zero, while the tangential components are functions of  $x$  or  $y$ .

**Results** On a sequence of grids with  $8 * 8$  to  $256 * 256$  meshes on each grid, one to three F-cycles with one pre- and one post-smoothing iteration were performed. On the current finest level, one additional pre-smoothing iteration was done; on the finest grid the post-smoothing step was suppressed. Table 6 shows the maximal error for the  $v$ -component of the solution. The first use of  $\tau$ -extrapolation was done on the third grid, which caused a remarkable decrease of the error. A single F-cycle, however, can not exploit the possible increase of accuracy. To do this by additional cycles on the finest grid is inefficient; the better way is to use a larger number of F-cycles on the coarser grids. The solution on the  $64 * 64$ -grid, in this case, is



more accurate than a solution without  $\tau$ -extrapolation on a  $256 * 256$ -grid.

### 2.5 Experience with classical $\tau$ -extrapolation

The authors experience with classical  $\tau$ -extrapolation can be summarized as follows:

1. The most reliable way to implement the  $\tau$ -extrapolation algorithm is to make use of injective restriction operators in the  $\tau$ -extrapolation step combined with an increased number of smoothing steps and a high order FMG-prolongation.
2. The application of  $\tau$ -extrapolation combined with restriction operators (which are averaging operators) is more complicated. This is caused by the fact that not all combinations of restriction operators are applicable. However, with averaging operators the improved accuracy can be obtained with lower computational work because no special requirements for the MG-algorithm must be fulfilled.
3. In the case of nonlinear problems it is important to solve the problem on the coarsest grid with sufficient accuracy. The coarsest grid must not be too coarse.
4. To exhaust the full potential of the  $\tau$ -extrapolation algorithm, it can be necessary to perform  $\gamma > 1$  MG-cycles on each grid level. Before setting  $\gamma$  to a value of two or three, all other possibilities for a failure of the  $\tau$ -extrapolation should be excluded. Even in cases where  $\gamma > 1$  is convenient for the coarser grids, on the finer grids one MG-cycle can be sufficient.
5. Besides a study of the behavior of the solution, an experimental analysis of the  $\tau$ -extrapolation algorithm should include a study of the defects. Only a look at the behavior of the defects permits a deeper understanding of some properties of the method.
6. Staggered grids do not exclude the application of  $\tau$ -extrapolation. However, they make it's application more complicated and require the use of the most expensive variant for the restriction operators in the  $\tau$ -extrapolation step.

## 3 Energy extrapolation

In the previous section,  $\tau$ -extrapolation for finite difference approximations has been discussed. The theory there

is based on a one term asymptotic expansion of the truncation error ( $\tau$ -term) of the form (A3) in Lemma 2.2.

In this section we will study a different approach to extrapolation methods, which is based on the formulation of (symmetric) equations as minimization problems of the form

$$(13) \quad E(u) = \min_{u \in V}!$$

To focus ideas, we will explain the basic principle for one dimensional boundary value problems. Consider the simplest test case (10). This problem can be written in the form (13) with

$$E(u) = \int_{-1}^1 ((u'(x))^2 - 2f(x)u(x)) \, dx \text{ for } u \in V,$$

where  $V = H_0^1(-1, 1)$  denotes the Sobolev space of order 1 on the interval  $(-1, 1)$  satisfying homogeneous boundary conditions. We introduce an equidistant grid  $x_i = -1 + ih$ ,  $i = 0, 1, 2, \dots, N$ , with mesh width  $h = 2/N$ . The function  $u$  is represented by the  $N + 1$  discrete values of the vector  $u_h = (u_0, u_1, u_2, \dots, u_N)^T$ . The energy (13) can now be discretized directly by combining numerical *differentiation* and *integration* rules. For example, an approximation to  $E(u)$  may be chosen as

$$E_h(u_h) = h \sum_{i=1}^N \left( \left( \frac{u_i - u_{i-1}}{h} \right)^2 - \frac{f_i u_i + f_{i-1} u_{i-1}}{2} \right).$$

Note that here the first term involves central differences to approximate the derivatives of  $u$  on a *shifted* grid. These derivatives are then integrated by a *midpoint* quadrature rule. The second term is directly integrated by a *trapezoidal* rule applied to the product  $u(x)f(x)$ .

When the normal equations for the quadratic minimization problem

$$E_h(u_h) = \frac{1}{2} u_h^T A_h u_h - f_h^T u_h = \min_{u_h \in V_h}!$$

are constructed, where  $V_h$  is the finite dimensional vector space of grid functions, we recover the conventional discretization of  $u'' = f$  by central differences

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = f_i \text{ for } i = 1, 2, \dots, N - 1.$$

The numerical approximation of the energy  $E_h(u_h)$  can now be expressed in an asymptotic expansion of the form

$$(14) \quad E_h(u_h) = E(u) + h^2 e_2 + h^4 e_4 + \dots,$$

where the coefficient functions  $e_2, e_4, \dots$  are independent of  $h$ . For expansion (14) to be valid, we, of course; need sufficient regularity for  $u$ . However, the expansion does *not* depend on the uniformity of the mesh, and it can be generalized to higher dimensions. For the one-dimensional case the reader is referred to a classical paper by Lyness [8]. Two dimensional results for triangles first appeared in Rde [12], and the general two-dimensional case is treated in Lyness and Rde [13].

Expansion (14) is the basis to consider *extrapolated functionals*, like

$$\bar{E}_h(u_h) = 4/3E_h(u_h) - 1/3E_H(u_h).$$

Note that here  $E_H$  is applied to  $u_h$ , which means that only every second value of  $u_h$  is considered. Clearly,  $\bar{E}_h$  also defines a quadratic functional, and a further analysis shows that the corresponding normal equations take the form

$$(15) \quad (4/3A_h - 1/3(I_h^H)^T A_H I_h^H)u_h = \tilde{f}_h$$

where  $I_h^H$  is the injection operator, and  $\tilde{f}_h$  is constructed by the analogous extrapolation using  $f_h$  and  $f_H$ .

In principle it is possible to compute the system matrix in (15) explicitly, however; to construct more efficient solvers, we note that by introducing the  $\tau$ -term  $\tau_h^H(u_h)$  of (7), this equation can be written in the defect correction form (9). Therefore (15) can be solved iteratively by the the multigrid  $\tau$ -extrapolation algorithm, and our derivation has led to a special case of the algorithms considered in the previous section.

However, we have not only recovered a special case of  $\tau$ -extrapolation, but have also found a method to derive  $\tau$ -extrapolation algorithms on unstructured and possibly adaptively generated meshes. This has been discussed in Rde [9] and several more variants of this method are explored in Rde [11].

In the further analysis of the energy extrapolation method (see [12]) the above analysis is applied in a finite element framework. In this context the above expansion is applied to  $u$ , being a finite element function which is smooth in construction. In consequence, the expansion and extrapolation technique can be used even when the original problem lacks sufficient regularity. Of course the final success of  $\tau$ -extrapolation is still dependent on how well higher order finite element functions can approximate the given problem, however; the analysis of the extrapolation method remains independent of regularity constraints from the differential equation.

In the remainder of this section, we will study the application to a nonlinear problem. Since the approach is based

on the formulation as a minimization problem, it cannot be applied directly to equations with convection terms; like the Burger's equation (11). (However, generalizations in this direction are presently under study and results will be published elsewhere). Here we will consider another problem, a stationary reaction-diffusion equation in one spatial dimension

$$(16) \quad 2\nu^2 u'' + (u^3 - u) = 0 \text{ in } (-1, 1),$$

with boundary conditions as in eq. (11)

$$u(-1) = -\tanh(1/2\nu), u(1) = \tanh(1/2\nu).$$

The solution is  $u(x) = \tanh(x/2\nu)$ , just as for (11). Written as a minimization problem (16) becomes

$$E(u) = \int_{-1}^1 \frac{1}{2\nu^2} (u'(x))^2 - \left( \frac{u^4}{2} - u^2 \right) dx.$$

Fig. 1 shows the *energy surface*  $E(u, x) = u^2 - u^4/2$  and

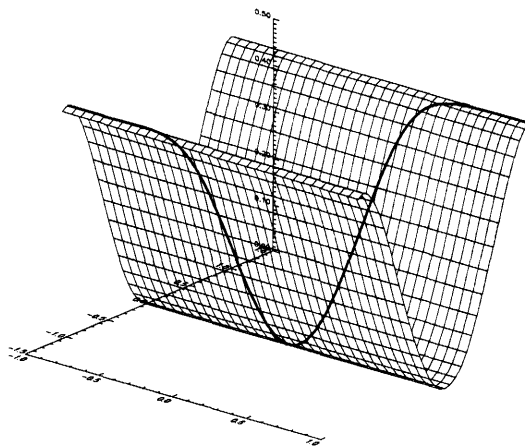


Figure 1:  $-u^4/2 + u^2$  and trajectory  $u(x)$

visualizes an optimal  $u(x)$  trajectory on this surface. If  $\nu$  is small, the transition develops to an interior layer, the conditioning of the boundary value problem becomes worse, and higher order methods become increasingly important in locating the transition region correctly.

In Fig. 2 we plot the  $L_2$ -error of three extrapolation schemes (with respect to the correct solution) versus the number of grid points in a log-log scale. The different graphs correspond to the original discretization with central differences; one extrapolation step according to (15), and the method obtained by applying two extrapolation

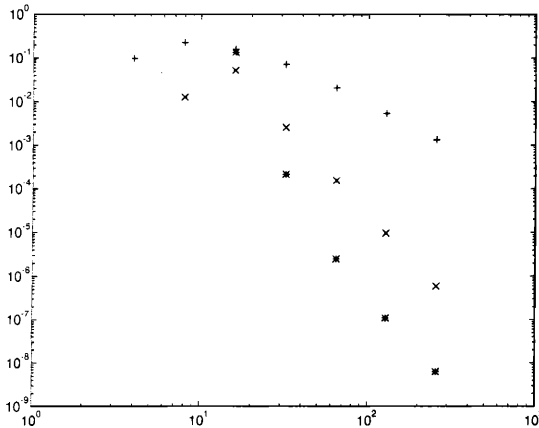


Figure 2: Error for energy extrapolation with order 2,4,6

steps to the energy, respectively. The latter approach corresponds to solving

$$\min \left( \frac{64}{45} E_h(u_h) - \frac{20}{45} E_{2h}(u_h) + \frac{1}{45} E_{4h}(u_h) \right).$$

Each methods needs a minimal number of mesh points, before the transition is correctly resolved. From then on, the asymptotic behavior of the method develops and can be read off the slope of the error graph. Clearly, the higher order methods provide superior accuracy for the same number of nodes.

While the fourth order behavior is clearly visible in these results, the method constructed with extrapolation to sixth order does not show the expected accuracy fully, though it is clearly converging much faster than the fourth order method. This is caused by stability problems. While the consistency order of the discrete system is raised to sixth order, the solution tends to have small oscillations, which start to pollute the solution for very high accuracy computations. This can be compensated by introducing stabilizing terms as introduced in [3] or [11].

In general, the energy extrapolation approach is an interesting alternative derivation of  $\tau$ -extrapolation algorithms, because it naturally defines the different algorithmic components in a compatible form. Furthermore, it permits generalization to nonuniform meshes and adaptive techniques, and it shows how to generalize  $\tau$ -extrapolation to a still higher order.

## 4 Finite element methods and extrapolation

### 4.1 Finite element discretizations of the boundary value problem

In this section, we consider two-dimensional second order elliptic boundary value problems of the form

(17) Find  $u \in V_0$  such that  $a(u, v) = \langle F, v \rangle$  for all  $v \in V_0$

holds, where  $V_0 \subset H^1(\Omega)$ ,

(18) 
$$a(u, v) = \int_{\Omega} (K(x) \nabla_x u, \nabla_x v) dx.$$

and

(19) 
$$\langle F, v \rangle = \int_{\Omega} f v dx + \int_{\Gamma_N} g_2 v ds.$$

$K(x)$  is a symmetric, positive definite  $(2 \times 2)$ -matrix,

(20) 
$$\nabla_x = \begin{pmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} \end{pmatrix}^T,$$

as well as  $(\cdot, \cdot)$  denotes the Euclidean scalar product in the space  $\mathbb{R}^2$ .

Let us first describe some finite element discretizations of problem (17). The starting point of our investigations are two triangular finite element meshes  $\mathcal{T}_H$  and  $\mathcal{T}_h$ , where we get the mesh  $\mathcal{T}_h$  by dividing all triangles of the mesh  $\mathcal{T}_H$  into four congruent sub-triangles. Later we will suppose that the mesh  $\mathcal{T}_H$  is the finest mesh of a sequence of nested triangular meshes.

Corresponding to the triangulations  $\mathcal{T}_H$  and  $\mathcal{T}_h$  we define the finite element subspaces

(21) 
$$V_H^l = \text{span}\{p_H^{(i)} : i = 1, 2, \dots, N_H\} \subset V_0,$$

and

(22) 
$$V_h^l = \text{span}\{p_h^{(i)}, i = 1, 2, \dots, N_h\} \subset V_0,$$

where the trial functions  $p_k^{(i)}$ ,  $k = H, h$ , are piecewise linear functions  $p_k^{(i)}$  which are linear in all triangles of  $\mathcal{T}_k$ , continuous, and satisfy the relations  $p_k^{(i)}(x_1^{(j)}, x_2^{(j)}) = 1$  for  $i = j$ ,  $p_k^{(i)}(x_1^{(j)}, x_2^{(j)}) = 0$  for  $i \neq j$ ,  $i, j = 1, 2, \dots, N_k$ . Here  $(x_1^{(j)}, x_2^{(j)})$  denotes the coordinates of the node  $P^{(j)}$  and  $N_k$  is the number of nodes belonging to  $\Omega \cup \Gamma_N$ , where  $\Gamma_N$  is the part of the boundary  $\partial\Omega$  on which natural boundary conditions are given. The functions  $p_k^{(i)}, i = 1, 2, \dots, N_k$  are called the *nodal basis of piecewise linear functions*.

Furthermore, we introduce piecewise quadratic functions  $q_H^{(i)}$ . These functions are polynomials of degree 2 in all triangles of  $\mathcal{T}_H$ , continuous, and satisfy the relations  $q_H^{(i)}(x_1^{(j)}, x_2^{(j)}) = 1$  for  $i = j$ ,  $q_H^{(i)}(x_1^{(j)}, x_2^{(j)}) = 0$  for  $i \neq j$ ,  $i, j = 1, 2, \dots, N_h$ . Using these functions we define the finite element subspace

$$(23) \quad V_h^q = \text{span}\{q_H^{(i)}, i = 1, 2, \dots, N_h\}.$$

with the *quadratic nodal basis*.

By means of the finite element subspaces (21), (22), and (23) we get the finite element schemes:

Find  $u_k \in V_k$  such that

$$(24) \quad a(u_k, v_k) = \langle F, v_k \rangle \quad \text{for all } v_k \in V_k,$$

where  $V_k$  stands for  $V_H^l$ ,  $V_h^l$ , or  $V_h^q$ , respectively.

The determination of the unknown function  $u_k$  is equivalent to the solution of the systems of the algebraic finite element equations

$$(25) \quad A_H^l u_H^l = f_H^l, \quad A_h^l u_h^l = f_h^l, \quad \text{and} \quad A_h^q u_h^q = f_h^q,$$

respectively.

The stiffness matrices  $A_h^l$  and  $A_h^q$  have a block structure

$$\begin{pmatrix} A_{h,vv} & A_{h,vm} \\ A_{h,mv} & A_{h,mm} \end{pmatrix},$$

where  $A_{h,vv}$  corresponds to the nodes of the triangulation  $\mathcal{T}_H$ ,  $A_{h,mm}$  corresponds to the new nodes in the triangulation  $\mathcal{T}_h$ , and  $A_{h,mv}$ ,  $A_{h,vm}$  are the coupling blocks.

Next we formulate an interesting relation between the matrices  $A_H^l$ ,  $A_h^l$ , and  $A_h^q$ , which is useful for the investigation of the convergence properties of a multigrid algorithm with extrapolation.

**Lemma 4.1** *Let  $A_H^l$ ,  $A_h^l$ , and  $A_h^q$  be defined by the bilinear form (18) using the finite element subspaces  $V_H^l$ ,  $V_h^l$ , and  $V_h^q$ , respectively. We suppose that the entries of the matrix  $K(x)$  in the bilinear form (18) are constant in each triangle  $\delta_H^{(r)} \in \mathcal{T}_H$ . Then the relation*

$$(26) \quad A_h^q = \frac{4}{3}A_h^l - \frac{1}{3}\tilde{A}_H$$

holds, where  $\tilde{A}_H = \begin{pmatrix} A_H^l & 0 \\ 0 & 0 \end{pmatrix}$ .

In Lemma 4.2 we formulate the corresponding property for the right-hand side.

**Lemma 4.2** *Let  $f_H^l$ ,  $f_h^l$ , and  $f_h^q$  be defined by the relation (19) using the finite element subspaces  $V_H^l$ ,  $V_h^l$ , and  $V_h^q$ , respectively. We suppose that  $f$  is a piecewise constant function, i.e. constant over all triangles  $\delta_H^{(r)} \in \mathcal{T}_H$ , and  $g_2$  a piecewise constant function, i.e. constant over  $\partial\delta_H^{(r)} \cap \partial\Omega$ . Then the following relation holds*

$$(27) \quad f_h^q = \frac{4}{3}f_h^l - \frac{1}{3}\tilde{f}_H, \quad \tilde{f}_H = \begin{pmatrix} f_H^l \\ 0 \end{pmatrix}.$$

A consequence of Lemma 4.1 and Lemma 4.2 is the following Theorem.

**Theorem 4.1** *Under the assumptions of Lemma 4.1 and Lemma 4.2 the FE systems of algebraic equations*

$$(28) \quad \left( \frac{4}{3}A_h^l - \frac{1}{3}\tilde{A}_H \right) u_h = \left( \frac{4}{3}f_h^l - \frac{1}{3}\tilde{f}_H \right)$$

and

$$(29) \quad A_h^q u_h = f_h^q$$

have the same solution.

The proofs of Lemma 4.1, Lemma 4.2, and Theorem 4.1 are given in [6].

In [6] an analogous theorem is proved for finite element systems based on a two-level  $h$ -hierarchical and a two-level  $p$ -hierarchical basis.

## 4.2 Multigrid algorithm with extrapolation

In the following, we discuss a multigrid algorithm using FE discretizations with piecewise *linear* functions and an implicit extrapolation step. The iterates of this algorithm converge to the solution which we get by a FE discretization of problem (17) with piecewise *quadratic* functions.

The smoothing procedures in our multigrid algorithm are defined in the following way:

- pre-smoothing  $G_h^V(u_h^{(j)}, A_h^l, f_h^l)$ :

Let the initial guess  $u_h^{(j)} = (u_{h,v}^{(j)}, u_{h,m}^{(j)})^T$  be given.

Set  $u_{h,v}^{(j+1)} = u_{h,v}^{(j)}$  and compute an approximate solution  $\tilde{z}_{h,m}$  of the system

$$(30) \quad A_{h,mm}^l z_{h,m} = f_{h,m}^l - A_{h,mv}^l u_{h,v}^{(j+1)} - A_{h,mm}^l u_{h,m}^{(j)}$$

by means of an iterative method, starting with the zero-vector.

$$\text{Set } u_h^{(j+1)} = (u_{h,v}^{(j+1)}, u_{h,m}^{(j)} + \tilde{z}_{h,m})^T.$$

We suppose that the error transmission operator of the method is of the type

$$M_{h,m} = (I_{h,m} - B_{h,mm}^{-1} A_{h,mm}^l).$$

- post-smoothing  $G_h^N(u_h^{(j)}, A_h^l, f_h^l)$ :

We use the same algorithm, however; we suppose that the error transmission operator of the iterative method for solving the system (30) is of the type  $M_{h,m} = (I_{h,m} - B_{h,mm}^{-T} A_{h,mm}^l)$ , so that the overall multigrid iteration operator becomes symmetric.

The step

$$d_H^{(k)} = R_h^H(f_h^l - A_h^l u_h^{(k,1)}),$$

i.e., the computation of the defect in a usual multigrid, we replace with the following extrapolation step

$$\begin{aligned} d_H^{(k)} &= \frac{4}{3} R_h^H(f_h^l - A_h^l u_h^{(k,1)}) \\ &\quad - \frac{1}{3} (f_H^l - A_H^l I_h^H u_h^{(k,1)}). \end{aligned}$$

Here, the operator  $R_h^H$  denotes the restriction operator (which is the transposed to the operator of the linear interpolation), and  $I_h^H$  stands for the injection operator.

The coarse grid system

$$(31) \quad A_H^l w_H^{(k)} = d_H^{(k)}$$

we solve by means of  $\mu$  iterations steps of a usual multigrid algorithm without extrapolation, which starts with the zero-vector (see, e.g. [4]).

Because of the equivalence of the matrices and right-hand sides in the systems of algebraic equations (28) and (29) as well as of the definition of the smoothing procedures we can interpret the multigrid algorithm with an extrapolation step as a multigrid algorithm without extrapolation for solving the system (29). Using a convergence theorem of Schieweck [14] for such a multigrid algorithm we get the following convergence theorem for our multigrid algorithm with extrapolation.

**Theorem 4.2** *Let the smoothing procedures, the restriction, and the interpolation operators be defined as they are at the beginning of this Section and let the assumptions of Lemma 4.1 and Lemma 4.2 be fulfilled. Then*

- (i) *The iterates of the multigrid algorithm with an extrapolation step converge to the solution which we get by a FE discretization of problem (17) with piecewise quadratic functions.*

- (ii) *The convergence estimate*

$$\|u_h^{(k+1,0)} - u_h\|_* \leq \eta \|u_h^{(k,0)} - u_h\|_*$$

holds, where  $\|\cdot\|_*^2 = ((\frac{4}{3}A_h^l - \frac{1}{3}\tilde{A}_H) \cdot, \cdot)$  and  $u_h$  is the solution of the system of algebraic FE equations

$$\left(\frac{4}{3}A_h^l - \frac{1}{3}\tilde{A}_H\right) u_h = \left(\frac{4}{3}f_h^l - \frac{1}{3}\tilde{f}_H\right).$$

The convergence rate  $\eta$  depends on the number of iteration steps for solving the systems (30), on the convergence rate of the multigrid algorithm used for solving the coarse-grid system (31), and on the constant in the strengthened Cauchy inequality

$$|a(v_h, w_H)| \leq \gamma \|v_h\| \|w_H\|$$

for all  $v_h \in T_h = \text{span}\{q_H^{(i)}, i = N_H + 1, \dots, N_h\}$ ,  
for all  $w_H \in V_H^l$ .

The proof of this theorem is given in [6]

### 4.3 Numerical results

Now we want to demonstrate the iterates of the multigrid algorithm with extrapolation converge to the FE solution which we would obtain by a discretization of problem (17) with piecewise quadratic functions.

Let us consider the problem:

Find  $u \in H_0^1(\Omega)$  such that

$$(32) \quad \int_{\Omega} (K \nabla_x u, \nabla_x v) dx = \int_{\Omega} f v dx$$

for all  $v \in H_0^1(\Omega)$  holds,

where  $\Omega = (0, 1) \times (0, 1)$ ,  $K = \begin{pmatrix} 4 & 4 \\ 4 & 5 \end{pmatrix}$ , and  $f = \pi^2(9 \sin \pi x \sin \pi y - 8 \cos \pi x \cos \pi y)$ . The exact solution of this problem is  $u = \sin \pi x \sin \pi y$ .

We compare the discretization errors  $\|u - u_h^l\|_1$  and  $\|u - u_h^q\|_1$  in the  $H^1$ -norm. Here  $u_h^l$  denotes the FE solution obtained by means of the multigrid algorithm with extrapolation, and  $u_h^q$  the FE solution by a discretization with piecewise quadratic functions. We remark that in our example the right-hand side  $f$  is not constant on triangles

$\delta_H^{(\tau)}$ , which we had assumed in Theorem 4.1. Therefore, in our example the right-hand sides  $\left(\frac{4}{3}f_h^l - \frac{1}{3}\tilde{f}_H\right)$  and  $f_h^q$  are not identical. But the discretization errors are almost the same.

Level $l$	$\ u - u_h^l\ _1$	$\ u - u_h^q\ _1$
3	0.1306	0.1426
4	0.3347-01	0.3481-01
5	0.8426-02	0.8539-02
6	0.2110-02	0.2118-02
7	0.5278-03	0.5283-03

Table 7: Comparison of the discretization errors

Table 7 shows that the solution  $u_h^l$  has a discretization error of the order  $O(h^2)$  in  $H^1(\Omega)$ , which is typical for finite element solutions resulting from a discretization with piecewise quadratic functions.

## 5 Conclusions

Multigrid and multilevel techniques are generally considered as fast solvers for a given discretization of a differential equation. In this paper we have presented another aspect of the multilevel principle. Using extrapolation in the natural hierarchical mesh structure of a multigrid solver, higher order approximations can be obtained simply and efficiently by  $\tau$ -extrapolation.

In contrast to classical extrapolation for differential equations, this approach is *implicit*. Extrapolation is not applied to different approximations of the *solution* but to quantities like the *truncation error*, the *energy*, or the *stiffness matrix* in finite element computations. The higher order approximation is then obtained by an iteration similar to defect correction, which is integrated with the multilevel iteration. This algorithm avoids the explicit construction of higher order operators and can be derived easily from basic (low order) multilevel algorithms. Computationally, the modification from basic to higher order is simply a multiplication of the  $\tau$ -correction by a suitable extrapolation factor.

The method also avoids one of the main disadvantages of conventional extrapolation methods. The mathematical foundation is *not* the existence of *global* error expansions which depend on the global regularity of the solution. The implicit nature of the algorithm permits a *local* analysis

and, therefore; justifies the local application of  $\tau$ -extrapolation - and even the combination of  $\tau$ -extrapolation with adaptive mesh structures.

For all our algorithms, the basic multilevel structure automatically provides an iterative solver with the typical multigrid convergence rates. Therefore, the higher order solution can be computed at a cost which is equivalent to a few relaxation sweeps for the *basic* low order discretization on the finest mesh. As is typical for multigrid, this relation is independent of the size of the problem.

The paper has presented three different approaches to  $\tau$ -extrapolation like algorithms, giving some theoretical background and numerical examples for each of them. Any of these three different interpretations of the  $\tau$ -extrapolation principle may be useful in a particular application and together they provide a deeper understanding of the algorithm and its features.

Our results clearly show the potential of  $\tau$ -extrapolation for many practical computations, including nonlinear ones, whenever the efficient treatment of the problem requires both a high order discretization and a fast algebraic solver.

## References

- [1] Bernert, K.: *Tauextrapolation - theoretische Grundlagen, numerische Experimente und Anwendungen auf die Navier-Stokes-Gleichungen*. Preprint-Reihe der Chemnitz DFG-Forschergruppe "Scientific Parallel Computing", Juni 1994, English version submitted to SIAM J.Sci.Comp.
- [2] Brandt, A.: *Multi-level Adaptive Solutions to Boundary Value Problems*. Math.Comp. v. 31, pp. 333-390,1977
- [3] Bungartz, H.; Griebel, M.; Rude, U.: *Extrapolation, combination and sparse grid techniques for elliptic boundary value problems*. Comput. Methods Appl. Mech. Engrg., (1994), pp. 243-252.
- [4] Hackbusch, W.: *Multigrid methods and applications*. Springer-Verlag, Berlin 1985
- [5] Joppich, W.: *Mehrgitterverfahren fur Diffusionsprobleme der Prozessimulation*. Bonn, Univ., Diss., 1990
- [6] Jung, M.; Rude, U.: *Implicit extrapolation methods for multilevel finite element computations*. SIAM J. Sci. Stat. Comput., 1994, accepted for publication
- [7] Lele, S.K.: *Compact finite difference schemes with spectral-like resolution*. J. Comp. Phys. 103/1 (1993), pp. 16ff.
- [8] Lyness, J.N.: *The calculation of the Stieltjes' integral*. Num. Math. 12, pp. 252-265, 1968
- [9] Rude, U.: *Adaptive higher order multigrid methods*. in Multigrid Methods III (ed. by W. Hackbusch, U. Trottenberg), Birkhuser Verlag, Basel Boston Berlin, 1991

- [10] Rde, U.: *Multiple  $\tau$ -extrapolation for multigrid methods*. Institut fr Informatik, TU Mnchen, Preprint TUM-18701, Januar 1987
- [11] Rde, U.: *Extrapolation and related techniques for solving elliptic equations*. Bericht I-9135, Institut fr Informatik, TU Mnchen, September 1991
- [12] Rde, U.: *Extrapolation techniques for constructing higher order finite element methods*. Bericht I-9304, Institut fr Informatik, TU Mnchen, 1993
- [13] Lyness, J.N.; Rde, U.: *Two-dimensional numerical integration of functions containing derivatives*. manuscript, 1995
- [14] Schieweck, N.: *A multigrid convergence proof by a strengthened Cauchy inequality for symmetric elliptic boundary value problems*. in Second Multigrid Seminar, Garzau, November 5–8, 1985, G. Telschow, ed., Karl-Weierstrass-Institut fr Mathematik, Berlin, 1986, pp. 49–62., Report R-Math-08/86.
- [15] Schreiber, H.; Keller, H.B.: *Spurious solutions in driven cavity calculations*. J. Comp. Phys. 49 (1983), pp. 165-173
- [16] Schreiber, H., Keller, H.B.: *Driven Cavity flows by efficient numerical techniques*. J. Comp. Phys. 49 (1983), pp. 310-333
- [17] Schller, A.: *Anwendung von Mehrgittermethoden auf das Problem einer ebenen reibungsfreien kompressiblen Unterschallstrmung am Beispiel des Kreisprofils*. Diplomarbeit, Institut fr Angewandte Mechanik, Universitt Bonn, 1983
- [18] Schller, A.; Lin, Q.: *Efficient high order algorithms for Elliptic boundary value problems combining full multigrid techniques and extrapolation methods*. Arbeitspapiere der GMD 192, Sankt Augustin, 1985
- [19] Schller, A.: *Mehrgitterverfahren fr Schalenprobleme*. GMD-Bericht Nr. 171, Mnchen, Wien, Oldenburg, 1988
- [20] Schwarz, H.R.: *Numerische Mathematik*. Teubner-Ingenieurmathematik, B.G. Teubner, Stuttgart, 1988, pp. 99-100.
- [21] Stben, K.; Trottenberg, U.: *Multigrid methods: fundamental algorithms, model problem analysis and applications*. Lecture Notes in Mathematics 960, Springer-Verlag, Berlin 1982
- [22] Tu, J.Y.; Fuchs, L.; Bai, X.S.: *Finite volume and multigrid methods on 3-D zonal overlapping grids*. in: W. Hackbusch, U. Trottenberg (Hrsg.), GMD Studien Nr. 189: Multigrid methods: Special topics and applications II, 1991, pp. 337-348
- [23] Wagner, H.: *Ein Differenzenverfahren fr die Navier-Stokes-Gleichungen in mehrfach zusammenhngenden Gebieten*. Diss., Technische Universitt Berlin, 1985

