Applications of Mathematics

Petr Vaněk

Fast multigrid solver

Applications of Mathematics, Vol. 40 (1995), No. 1, 1–20

Persistent URL: http://dml.cz/dmlcz/134274

40 (1995) APPLICATIONS OF MATHEMATICS No. 1, 1–20

FAST MULTIGRID SOLVER

PETR VANĚK, Plzeň

(Received February 5, 1993)

Summary. In this paper a black-box solver based on combining the unknowns aggregation with smoothing is suggested. Convergence is improved by overcorrection. Numerical experiments demonstrate the efficiency.

Keywords: Algebraic multigrid method, smoothing, unknowns aggregation, black-box solver

AMS classification: 65F10

1. INTRODUCTION

The usual iterative methods for solving systems of linear algebraic equations—for example the damped Jacobi method or the Gauss-Seidel method—eliminate some types of errors very effectively and some of them very slowly. As a rule oscillating errors are effectively eliminated therefore we speak about the so called smoothing effect. A multilevel method is based on a combination of a smoothing iteration with the so called correction on a coarse level. Components of an error which are not effectively removable by smoothing are tried to be represented in a space the dimension of which is smaller than the dimension of the original system. The standard correction on a coarse level is an orthogonal projection of the error along components of the error represented in the space with the smaller dimension.

The aim of this paper is to describe the construction of coarse spaces based on combining the unknowns aggregation with smoothing. The advantage of unknowns aggregation is that it can be easily done using a graph algorithm without explicit knowledge of the geometry of a grid. However, piecewise constant coarse spaces (generated by unknowns aggregation) contain high energy functions and therefore the multigrid algorithm based on unknowns aggregation is not too efficient. In order to improve convergence properties we will suppress the energy of coarse space functions using a smoother. Numerical experiments (see Section 7) have shown that the

convergence factor of the suggested algorithm is approximately 10–20 times smaller than that one of a black-box solver suggested by Ruge and Stüben in [8], Chap. 4, p. 108. The presented algorithm is based on tricks described in [2], [3], [4]. Acceleration by overcorrection was suggested by Blaheta [1]. A modification of this technique used in this paper was analyzed in [2]. The algorithm (generation of a disjoint covering of a set of degrees of freedom) is a modification of the algorithm 2.2 in [3]. The presented modification enables to solve systems of PDEs and respects the coupling of degrees of freedom. Improvement of transfer operators by smoothing was suggested and analyzed in [4].

Let us consider a system of linear algebraic equations

$$\boldsymbol{A}u = f, \tag{1.1}$$

$\boldsymbol{A}$ is a positive definite matrix of order n, $f \in \mathbb{R}^n$ a given right-hand side. Further let

$$\boldsymbol{p}\colon \mathbb{R}^m \to \mathbb{R}^n, \quad m < n \tag{1.2}$$

be an injective operator. This operator will be called a prolongation (for technical details see Section 5). The smoothing iterative method is supposed to be of the form

$$S(x) = \boldsymbol{M}x + \boldsymbol{N}f, \tag{1.3}$$

$x \in \mathbb{R}^n$, $\boldsymbol{M}$, $\boldsymbol{N}$ are nonsingular square matrices satisfying the consistence condition

$$\boldsymbol{I} = \boldsymbol{M} + \boldsymbol{N}\boldsymbol{A}. \tag{1.4}$$

Let us note that (1.4) implies

$$S(\hat{x}) = \hat{x}, \tag{1.5}$$

where $\hat{x} = \boldsymbol{A}^{-1}f$. An error of the vector $x \in \mathbb{R}^n$ will be defined as

$$e(x) = x - \hat{x}. \tag{1.6}$$

Components of the error which are not effectively removable by smoothing, i.e.

$$\boldsymbol{M}e \approx e \tag{1.7}$$

will be called smooth components. We try to represent these smooth components in $\mathbb{R}^m$ using the prolongation operator. The error $e \in \mathbb{R}^n$ can be represented in $\mathbb{R}^m$ if there is a $v \in \mathbb{R}^m$ such that

$$e = \boldsymbol{p}v. \tag{1.8}$$

In the ideal case the range of $\boldsymbol{p}$ always contains all smooth components of an arbitrary error $e \in \mathbb{R}^n$. In practice this requirement seems to be too strong. Usually we will be satisfied with the fulfilment of a weaker condition

$$\boldsymbol{M}\boldsymbol{p}v \approx \boldsymbol{p}v. \tag{1.9}$$

The construction of the prolongation operator contained in this paper follows the scheme:

1. An auxiliary prolongation operator

$$\hat{\boldsymbol{p}}\colon \mathbb{R}^m \to \mathbb{R}^n \tag{1.10}$$

will be constructed by the unknowns aggregation technique. This process is little time-consuming but the properties of the operator $\hat{\boldsymbol{p}}$ are very bad from the point of view of the condition (1.9).

2. A prolongation operator $\boldsymbol{p}$ will be constructed by the formula

$$\boldsymbol{p} = \boldsymbol{M}\hat{\boldsymbol{p}}. \tag{1.11}$$

The range of $\boldsymbol{p}$ is expected to contain vectors with smooth components mostly because non-smooth vectors from $\mathrm{Im}(\hat{\boldsymbol{p}})$ are suppressed by the smoothing operator $\boldsymbol{M}$.

2. Standard two-level method

Let $\langle\cdot,\cdot\rangle_1$ and $\langle\cdot,\cdot\rangle_2$ denote the usual scalar products in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, and $\|\cdot\|_1 = \langle\cdot,\cdot\rangle_1^{1/2}$, $\|\cdot\|_2 = \langle\cdot,\cdot\rangle_2^{1/2}$ the corresponding euclidean norm. Let us define the energetic scalar product by

$$(\cdot,\cdot)_1 = \langle \boldsymbol{A}\cdot,\cdot\rangle_1 \tag{2.1}$$

and the corresponding energy norm by

$$\|\cdot\|_1 = (\cdot,\cdot)_1^{1/2}. \tag{2.2}$$

A multilevel algorithm consists of a correction on a coarse level and of smoothing (see Introduction). Let $x \in \mathbb{R}^n$ be an approximation of the solution $\hat{x} = A^{-1}f$ of the problem (1.1). The correction on a coarse level consists in subtracting $\boldsymbol{p}v$, $v \in \mathbb{R}^n$ fulfils the minimalization condition

$$\|e(x) - \boldsymbol{p}v\|_1 = \min_{w\in\mathbb{R}^m} \|e(x) - \boldsymbol{p}w\|_1. \tag{2.3}$$

The orthogonal projection theorem implies that $e(x) - \boldsymbol{p}v$ is the A-orthogonal projection of $e(x)$ into $\mathrm{Im}(\boldsymbol{p})$, i.e.

$$(e(x) - \boldsymbol{p}v, \boldsymbol{p}w)_1 = 0 \quad \text{for all } w \in \mathbb{R}^m. \tag{2.4}$$

Now we can easily define the algorithm of the correction on a coarse level. The restriction operator

$$\boldsymbol{r}\colon \mathbb{R}^n \to \mathbb{R}^m \tag{2.5}$$

will be the adjoint operator to $\boldsymbol{p}$ with respect to the scalar products, i.e.

$$\langle \boldsymbol{r}x, y\rangle_2 = \langle x, \boldsymbol{p}y\rangle_1 \quad x \in \mathbb{R}^n, y \in \mathbb{R}^m. \tag{2.6}$$

Let us mention that matrix representations of $\boldsymbol{r}$, $\boldsymbol{p}$ will be mutually transposed. Using (2.1) and (2.6) we get the equivalent form of (2.4)

$$\langle \boldsymbol{r}\boldsymbol{A}\big(e(x) - \boldsymbol{p}v\big), w\rangle_2 = 0 \quad \text{for all } w \in \mathbb{R}^m. \tag{2.7}$$

It is clear that (2.7) is valid if and only if

$$\boldsymbol{r}\boldsymbol{A}e(x) = \boldsymbol{r}\boldsymbol{A}\boldsymbol{p}v. \tag{2.8}$$

It is easy to see that

$$\boldsymbol{A}e(x) = \boldsymbol{A}x - f. \tag{2.9}$$

Setting

$$\boldsymbol{A}_2 = \boldsymbol{r}\boldsymbol{A}\boldsymbol{p}, \tag{2.10}$$

(2.8) becomes the system of m linear algebraic equations in m variables

$$\boldsymbol{A}_2 v = \boldsymbol{r}(\boldsymbol{A}x - f). \tag{2.11}$$

The correction on the coarse level can therefore follow this algorithm:

$$d := \boldsymbol{A}x - f, \qquad d \in \mathbb{R}^n, \tag{2.12a}$$

$$d_2 := \boldsymbol{r}d, \qquad d_2 \in \mathbb{R}^m, \tag{2.12b}$$

$$v := \boldsymbol{A}_2^{-1} d_2, \qquad v \in \mathbb{R}^m, \tag{2.12c}$$

$$x := x - \boldsymbol{p}v, \qquad x \in \mathbb{R}^n. \tag{2.12d}$$

A two-level algorithm contains usually two smoothing steps. The first (so called pre-smoothing) is applied before and the second (so called post-smoothing) is applied after the correction on the coarse level. Smoothing steps represent several (mostly 2–5) iterations of the smoothing iterative method (1.3). The complete algorithm of a two-level method can be, for example:

1. Setup phase
   - generation of the operator $\boldsymbol{p}$ (in dependence on $\boldsymbol{A}$).
   - calculation of $\boldsymbol{A_2 = rAp}$
2. Iterative phase: (x^i denotes the i-th iteration, x^0 being chosen arbitrarily)

(2.13a) $\tilde{x} := S^{(\nu_1)}(x^i)$ (ν_1 times iterating (1.3)), $\tilde{x} \in \mathbb{R}^n$,

(2.13b) $d := A\tilde{x} - f$ $d \in \mathbb{R}^n$,

(2.13c) $d_2 := rd$ $d_2 \in \mathbb{R}^m$,

(2.13d) $v := A_2^{-1} d_2$ $v \in \mathbb{R}^m$,

(2.13e) $\tilde{\tilde{x}} := \tilde{x} - pv$ $\tilde{\tilde{x}} \in \mathbb{R}^n$,

(2.13f) $x^{i+1} := S^{(\nu_2)}(\tilde{\tilde{x}})$ $x^{i+1} \in \mathbb{R}^n$.

The only step which has not yet been precisely defined is the generation of the operator $\boldsymbol{p}$. This step will be described in detail later.

3. Standard multi-level method

Generalization of a two-level method to a multi-level one is very natural. We do not solve the system $A_2 v = d_2$ exactly as in (2.13d), but approximately by the recursive application of the two-level method. In the setup phase we generate matrices A_l, $l = 2, \ldots, k$, with decreasing order n_l. Generation is stopped when the order of the matrix is sufficiently small for the effective direct solution of the corresponding system of linear algebraic equations.

Let c denote the maximal order of a matrix on "the coarsest level". Then the setup phase consists of the following steps:

(3.1a) setting $A_1 = A$, $l := 1$,

(3.1b) p_l is generated depending on A_l,

(3.1c) $A_{l+1} := r_l A_l p_l$,

(3.1d) $l := l + 1$,

if $n_l > c$ then

go to (3.1b)

else $k := l$.

Let us denote by H^l the space $\mathbb{R}^{n_l}$, $l = 1, 2, \ldots, k$. Let ν_1 resp. ν_2 be the number of the pre-smoothing (resp. post-smoothing) iterations. The iterative phase $(x^i \to x^{i+1})$ of the standard multi-level method can be written as

(3.2a) $l := 1, f_1 := f, x_1 := x^i,$ $f_1, x_1 \in H^1,$

(3.2b) $\tilde{x}_l := S_l^{(\nu_1)}(x_l),$ $\tilde{x}_l \in H^l,$

(3.2c) $d_l := \boldsymbol{A}_l \tilde{x}_l - f_l,$ $d_l \in H^l,$

(3.2d) $f_{l+1} := \boldsymbol{r}_l d_l,$ $f_{l+1} \in H^{l+1},$

(3.2e) if $l + 1 = k$ then

$x_{l+1} := (\boldsymbol{A}_{l+1})^{-1} f_{l+1},$ $x_{l+1} \in H^{l+1}$

else

$x_{l+1} := 0,$

$l := l + 1,$

for $k = 1, \ldots, \gamma$ do (3.2b–3.2g),

$l := l - 1.$

(3.2f) $\tilde{\tilde{x}}_l := \tilde{x}_l - \boldsymbol{p}_l x_{l+1},$ $\tilde{\tilde{x}}_l \in H^l$

(3.2g) $x_l := S_l^{(\nu_2)}(\tilde{\tilde{x}}_l),$ $x_l \in H^l$

(3.2h) $x^{i+1} := x_l.$

$\gamma \in \mathbb{N}$ in (3.2e) determines how many iterations of the multi-level method are used to solve the problem on the "coarse level". Usually we choose $\gamma = 1$ or $\gamma = 2$.

Remark 3.1. In the case $\gamma = 1$ we speak about a V-cycle, in the case $\gamma = 2$ about a W-cycle. For more detailed information about terminology see [1].

4. Acceleration of multi-level method

In this section we will describe one technique with help of which the acceleration of the convergence of the multi-level method can be achieved. This technique is a slight modification of a technique suggested by Blaheta [1] and analyzed in [2]. The idea will be explained using the two-level algorithm, generalization to the multi-level case is trivial. Let us consider the algorithm (2.13). Roughly speaking the step (2.13e) will be replaced by the step $\tilde{\tilde{x}} := \tilde{x} - \hat{t}\boldsymbol{p}v$, $\hat{t} \in \mathbb{R}$ is chosen to minimize the ratio $\|e(x^{i+1})\|_1 / \|e(\tilde{x})\|_1$. It is easy to see that this requirement is equivalent to the condition

$$\|\boldsymbol{M}^{\nu_2}[e(\tilde{x}) - \hat{t}\boldsymbol{p}v]\|_1 \leqslant \|\boldsymbol{M}^{\nu_2}[e(\tilde{x}) - t\boldsymbol{p}v]\|_1 \quad \text{for every } t \in \mathbb{R}. \tag{4.1}$$

From (4.2) we can deduce

$$\hat{t} = \frac{(\boldsymbol{M}^{\nu_2} e(\tilde{x}), \boldsymbol{M}^{\nu_2}\boldsymbol{p}v)_1}{\|\boldsymbol{M}^{\nu_2}\boldsymbol{p}v\|_1^2} \quad \text{for } \boldsymbol{M}^{\nu_2}\boldsymbol{p}v \neq 0. \tag{4.2}$$

If $M^{\nu_2}pv = 0$, an arbitrary $\hat{t} \in \mathbb{R}$ fulfils the condition (4.2). For an arbitrary $x \in \mathbb{R}^n$ we have

$$M^{\nu_2}e(x) = e\big(S^{(\nu_2)}(x)\big), \tag{4.3}$$

$$Ae(x) = Ax - f. \tag{4.4}$$

Using (4.2)–(4.4) we get

$$\hat{t} = \frac{\langle A\bar{x} - f, \bar{v}\rangle_1}{\langle A\bar{v}, \bar{v}\rangle_1} \tag{4.5}$$

where

$$\bar{v} = M^{\nu_2}pv, \tag{4.6}$$

$$\bar{x} = S^{(\nu_2)}(\tilde{x}). \tag{4.7}$$

Now we can write down the algorithm of a modified two-level method.

$$\tilde{x} := S^{(\nu_1)}(x^i), \qquad \tilde{x} \in \mathbb{R}^n, \tag{4.8a}$$

$$d := A\tilde{x} - f, \qquad d \in \mathbb{R}^n, \tag{4.8b}$$

$$d_2 := rd, \qquad d_2 \in \mathbb{R}^m, \tag{4.8c}$$

$$v := (A_2)^{-1}d_2, \qquad v \in \mathbb{R}^m, \tag{4.8d}$$

$$\bar{v} := M^{\nu_2}pv, \qquad \bar{v} \in \mathbb{R}^n, \tag{4.8e}$$

$$\text{if } \bar{v} = 0 \text{ then } x^{i+1} := \tilde{x}, \qquad \text{else continue (4.8g)–(4.8i)} \tag{4.8f}$$

$$\bar{x} := S^{(\nu_2)}(\tilde{x}), \qquad \bar{x} \in \mathbb{R}^n, \tag{4.8g}$$

$$\hat{t} := \frac{\langle A\bar{x} - f, \bar{v}\rangle_1}{\langle A\bar{v}, \bar{v}\rangle_1} \tag{4.8h}$$

$$x^{i+1} := \bar{x} - \hat{t}\bar{v}, \qquad x^{i+1} \in \mathbb{R}^n. \tag{4.8i}$$

Remark 4.1. Setting $\hat{t} = 1$ in (4.8h) the algorithm (4.8) is equivalent to the algorithm (2.13). The condition (4.1) guarantees that the algorithm (4.8) reduces the error not worse than (2.13) supposing the input of x^i is the same.

5. Construction of transfer operators

In this section we will describe the construction of transfer operators r, p. Matrix representations of these operators are mutually transposed it is therefore sufficient to describe the construction of the prolongation operator p. As was mentioned in Introduction the generation of p will pass two phases:

1. generation of $\hat{p}$: $\mathbb{R}^m \to \mathbb{R}^n$ using the unknowns aggregation technique
2. "smoothing" $p := M\hat{p}$.

The idea of the unknowns aggregation is following: We will form a disjoint covering of the index set $\{1,\dots,n\}$, i.e. the system of index sets $\{C_i\}_{i=1}^m$ satisfying two conditions:

$$C_i \cap C_j = \emptyset \quad \text{for } i \neq j, \tag{5.1}$$

$$\bigcup_{i=1}^{m} C_i = \{1,\dots,n\}. \tag{5.2}$$

The number m is unknown in advance. Then the operator $\hat{p}\colon \mathbb{R}^m \to \mathbb{R}^n$ is defined by

$$(\hat{p}x)_i = x_j, \quad i \in C_j, x \in \mathbb{R}^m. \tag{5.3}$$

Conditions (5.1), (5.2) easily show that for every index $i \in \{1,\dots,n\}$ there exists the only index $j \in \{1,\dots,m\}$ such that $i \in C_j$. Verbally the definition of $\hat{p}\colon x \to y$ can be expressed in the following way: the j-th component of the vector $x \in \mathbb{R}^m$ will be mapped onto all components of the vector $y \in \mathbb{R}^n$ indices of which are in C_j. It is easy to see that $\hat{p}$ is injective. Unknowns will be called aggregated if their indices are in the same C_j. As the aggregated unknowns are represented in $\mathbb{R}^m$ by one unknown it is suitable to aggregate the unknowns with "close affinity". Therefore only the unknowns modelling the same physical quantities and being "strongly bound by a single equation" will be aggregated. Let $t_i \in \mathbb{N}$, $i = 1,2,\dots,n$, denote the physical interpretation of the i-th variable (for example, 1 denotes temperature, 2 compression). Let $\Theta \in (0,1)$. We define a neighbourhood of the i-th variable

$$N_i = \{j\colon (|a_{ij}| \geqslant \Theta \max_{\substack{k=1,\dots,n \\ k\neq i}} |a_{ik}|) \wedge (t_j = t_i)\} \cup \{i\}. \tag{5.4}$$

We suppose that for each C_j, $j = 1,\dots,m$ there exists $i \in \{1,2\dots,n\}$ such that

$$C_j \subset N_i \quad \text{for some } i \in \{1,2,\dots,n\}. \tag{5.5}$$

The algorithm for the generation of the system $\{C_j\}$: (The result is the disjoint covering $\{C_j\}$ and the number $m = |\{C_j\}|$.)

$$\begin{aligned}
&R := \{1,2,\dots,n\}, j := 0,\\
&\text{for } i := 1,2,\dots,n\\
&\quad \text{if } N_i \subset R \text{ then}\\
&\qquad j := j+1, C_j := N_i, R := R \setminus C_j, \bar{t}_j := t_i,\\
&\text{for } i := 1,2,\dots,n\\
&\quad \text{if } i \in R \text{ then}\\
&\qquad j := j+1, C_j := N_i \cap R, R := R \setminus C_j, \bar{t}_j := t_i,\\
&m := j.
\end{aligned} \tag{5.6}$$

The sequence $\{\bar{t}_j\}_{j=1}^m$ defines the physical interpretation of the variables on a "coarse level". The first loop of the algorithm (5.5) generally does not establish the covering. This is the goal of the second loop. The system of index sets $\{C_j\}_{j=1}^m$ generated by the algorithm (5.6) satisfies (5.1), (5.2) and (5.5). Now we are able to construct the operator $\boldsymbol{p}$. (5.3) implies that the auxiliary operator $\hat{\boldsymbol{p}}$ can be represented by an $n \times m$ matrix with elements

$$\hat{p}_{ij} = \begin{cases} 1 & \text{for } i \in C_j \\ 0 & \text{for } i \notin C_j \end{cases}, \quad i = 1,2,\ldots,n,\ j = 1,2,\ldots,m. \tag{5.7}$$

The iterative operator of the damped Jacobi method represented by a matrix

$$\boldsymbol{M} = \boldsymbol{I} - \omega \boldsymbol{D}^{-1}\boldsymbol{A}, \quad \omega \in (0,1), \tag{5.8}$$

where $\boldsymbol{D}$ is the diagonal part of $\boldsymbol{A}$ will be used as the smoothing operator. The advantage of the choice (5.8) is evident: The elements m_{ij} of $\boldsymbol{M}$ can be very easily expressed by

$$m_{ij} = \begin{cases} 1-\omega & \text{for } i = j \\ -\omega a_{ij}/a_{ii} & \text{for } i \neq j. \end{cases}, \quad i,j = 1,2,\ldots,n. \tag{5.9}$$

Now we define

$$\boldsymbol{p} = \boldsymbol{M}\hat{\boldsymbol{p}}. \tag{5.10}$$

In some cases it is suitable to use the "simplification" of $\boldsymbol{M}$ to define $\boldsymbol{p}$. If we use the "simplification" $\boldsymbol{M}_s$ described below instead of $\boldsymbol{M}$ the increase of the number of nonzero elements in the rows of $\boldsymbol{A}_l$ during the setup phase (3.1) will be substantially slower. Let $\Theta' \in (0,1)$. Let us define a matrix $\boldsymbol{M}_s = \{m_{ij}^s\}_{i,j=1}^n$ by

$$m_{ij}^s = \begin{cases} m_{ij}, & \text{if } |a_{ij}| \geqslant \Theta' \max\limits_{\substack{k=1,\ldots,n \\ k \neq i}} |a_{ik}|, \\ 0 & \text{elsewhere.} \end{cases} \tag{5.11}$$

Analogously we define

$$\boldsymbol{p} = \boldsymbol{M}_s\hat{\boldsymbol{p}}. \tag{5.12}$$

Practice has shown the good choice of the parametres Θ, Θ' in (5.4) and (5.11) for the level $l \geqslant 1$ is

$$\Theta = \Theta' = 0.1 \cdot (0.3)^{l-1} \tag{5.13}$$

6. Convergence of the two-level method

This section will deal with convergence of the algorithms (2.13) and (4.8) for the transfer operators defined in the previous sections. Let us define subspaces of $\mathbb{R}^n$

$$\hat{T} = \mathrm{Ker}(\hat{\boldsymbol{r}}\boldsymbol{A}), \tag{6.1}$$

$$T = \mathrm{Ker}(\boldsymbol{r}\boldsymbol{A}), \tag{6.2}$$

$$\hat{S} = \mathrm{Im}(\hat{\boldsymbol{p}}), \tag{6.3}$$

$$S = \mathrm{Im}(\boldsymbol{p}). \tag{6.4}$$

Let X be a subspace of $\mathbb{R}^n$. Let $X^{\perp}$ denote the A-orthogonal complement of X, i.e.

$$X^{\perp} = \{x \in \mathbb{R}^n : (x, y)_1 = 0 \ \text{ for every } y \in x\}. \tag{6.5}$$

The next lemma follows easily from the above definitions.

Lemma 6.1.

$$\hat{T} = \hat{S}^{\perp}, \tag{6.6}$$

$$T = S^{\perp}. \tag{6.7}$$

Further the iterative method (1.3) is supposed to be given by

$$S(x) = (\boldsymbol{I} - \omega \boldsymbol{D}^{-1}\boldsymbol{A})x + \omega \boldsymbol{D}^{-1} f, \quad \omega \in (0, 1). \tag{6.8}$$

It is not difficult to see this iterative method fulfils (1.4) and therefore

$$S(\hat{x}) = \hat{x}, \tag{6.9}$$

where $\hat{x} = \boldsymbol{A}^{-1} f$. Let us note the iteration operator $I - \omega \boldsymbol{D}^{-1}\boldsymbol{A}$ of this method was used to construct the transfer operators (see (5.8)). Our dealing with the convergence will be restricted to the case of the prolongation operator defined by (1.11), i.e. the case of the "simplified" matrix $\boldsymbol{M}_s$ defined by (5.11) will not be taken into account. Further we will suppose that $\boldsymbol{M}$ is nonsingular. As $\hat{\boldsymbol{p}}$ is injective, $\boldsymbol{p} = \boldsymbol{M}\hat{\boldsymbol{p}}$ is injective as well and $\boldsymbol{A}_2 = \boldsymbol{r}\boldsymbol{A}\boldsymbol{p}$ is positive definite. Now we will analyze the convergence of the algorithm (2.13).

Lemma 6.2. *The following equalities are valid:*

$$e(\tilde{x}) = \boldsymbol{M}^{\nu_1} e(x^i), \tag{6.10a}$$

$$d = \boldsymbol{A} e(\tilde{x}), \tag{6.10b}$$

$$v = (\boldsymbol{r}\boldsymbol{A}\boldsymbol{p})^{-1} \boldsymbol{r}\boldsymbol{A} e(\tilde{x}), \tag{6.10c}$$

$$e(\tilde{\tilde{x}}) = [\boldsymbol{I} - \boldsymbol{p}(\boldsymbol{r}\boldsymbol{A}\boldsymbol{p})^{-1} \boldsymbol{r}\boldsymbol{A}] e(\tilde{x}), \tag{6.10d}$$

$$e(x^{i+1}) = \boldsymbol{M}^{\nu_2} e(\tilde{\tilde{x}}). \tag{6.10e}$$

Proof is trivial. □

Lemma 6.3. *Let us set*

$$\boldsymbol{Q}_S = \boldsymbol{p}(\boldsymbol{rAp})^{-1}\boldsymbol{rA}, \tag{6.11}$$

$$\boldsymbol{Q}_T = \boldsymbol{I} - \boldsymbol{p}(\boldsymbol{rAp})^{-1}\boldsymbol{rA}. \tag{6.12}$$

Then

$$\mathrm{Im}(\boldsymbol{Q}_S) = S, \tag{6.13}$$

$$\mathrm{Im}(\boldsymbol{Q}_T) = T, \tag{6.14}$$

$$\boldsymbol{Q}_S + \boldsymbol{Q}_T = \boldsymbol{I}, \tag{6.15}$$

$$\mathrm{Ker}(\boldsymbol{Q}_S) = T, \tag{6.16}$$

$$\mathrm{Ker}(\boldsymbol{Q}_T) = S, \tag{6.17}$$

i.e. $\boldsymbol{Q}_S$ *is an* A*-orthogonal projection of* $\mathbb{R}^n$ *into* S *and* $\boldsymbol{Q}_T$ *is an* A*-orthogonal projection of* $\mathbb{R}^n$ *into* T.

Proof is evident. □

Lemma 6.4. $\boldsymbol{M}$ *is a selfadjoint operator with respect to the scalar product* $(\cdot,\cdot)_1$.

Proof.

$$\begin{aligned}
(\boldsymbol{M}x, y)_1 &= \langle \boldsymbol{AM}x, y\rangle_1 \\
&= \langle \boldsymbol{A}(\boldsymbol{I} - \omega \boldsymbol{D}^{-1}\boldsymbol{A})x, y\rangle_1 \\
&= \langle (\boldsymbol{A} - \omega \boldsymbol{AD}^{-1}\boldsymbol{A})x, y\rangle_1 \\
&= \langle (\boldsymbol{I} - \omega \boldsymbol{AD}^{-1})\boldsymbol{A}x, y\rangle_1 \\
&= \langle \boldsymbol{M}^T\boldsymbol{A}x, y\rangle_1 \\
&= \langle \boldsymbol{A}x, \boldsymbol{M}y\rangle_1 = (x, \boldsymbol{M}y)_1, \quad x, y \in \mathbb{R}^n.
\end{aligned}$$

□

Let X_1, X_2 be subspaces of $\mathbb{R}^n$. Let us consider an operator $\boldsymbol{B}\colon X_1 \to X_2$ and let us define an operator norm

$$\|\boldsymbol{B}\|_{X_1 \to X_2} = \sup_{x \in X_1 \setminus \{0\}} \frac{\|\boldsymbol{B}x\|_1}{\|x\|_1}. \tag{6.18}$$

Lemma 6.5. *The following estimate is valid for the algorithm* (2.13):

$$\frac{\|e(x^{i+1})\|_1}{\|e(x^i)\|_1} \leqslant \|\boldsymbol{M}^{\nu_1}\|_{T \to \mathbb{R}^n} \cdot \|\boldsymbol{M}^{\nu_2}\|_{T \to \mathbb{R}^n}. \tag{6.19}$$

Proof. It is easy to see that

$$e(x^{i+1}) = M^{\nu_2} Q_T M^{\nu_1} e(x_i). \tag{6.20}$$

Hence we must estimate the operator norm $\|M^{\nu_2} Q_T M^{\nu_1}\|_{\mathbb{R}^n \to \mathbb{R}^n}$. As $\mathrm{Im}(Q_T) = T$,

$$\|M^{\nu_2} Q_T M^{\nu_1}\|_{\mathbb{R}^n \to \mathbb{R}^n} \leqslant \|Q_T M^{\nu_1}\|_{\mathbb{R}^n \to \mathbb{R}^n} \|M^{\nu_2}\|_{T \to \mathbb{R}^n}. \tag{6.21}$$

Using Lemma lemma 6.3 we get

$$\begin{aligned} \|Q_T M^{\nu_1} x\|_1^2 &= (Q_T M^{\nu_1} x, Q_T M^{\nu_1} x)_1 \\ &= (Q_T M^{\nu_1} x, M^{\nu_1} x)_1 \\ &= (M^{\nu_1} Q_T M^{\nu_1} x, x)_1 \\ &\leqslant \|M^{\nu_1} Q_T M^{\nu_1} x\|_1 \cdot \|x\|_1 \\ &\leqslant \|M^{\nu_1}\|_{T \to \mathbb{R}^n} \cdot \|Q_T M^{\nu_1} x\|_1 \cdot \|x\|_1, \end{aligned}$$

which implies

$$\|Q_T M^{\nu_1} x\|_1 \leqslant \|M^{\nu_1}\|_{T \to \mathbb{R}^n} \cdot \|x\|_1 \tag{6.22}$$

and hence

$$\|Q_T M^{\nu_1}\|_{\mathbb{R}^n \to \mathbb{R}^n} \leqslant \|M^{\nu}\|_{T \to \mathbb{R}^n}. \tag{6.23}$$

Substituting (6.23) into (6.21) we get

$$\|M^{\nu_2} Q_T M^{\nu_1}\|_{\mathbb{R}^n \to \mathbb{R}^n} \leqslant \|M^{\nu_1}\|_{T \to \mathbb{R}^n} \cdot \|M^{\nu_2}\|_{T \to \mathbb{R}^n}. \tag{6.24}$$

□

Lemma 6.6. *For M the following inequalities are valid:*

$$\|M\|_{T \to \mathbb{R}^n} \leqslant \|M\|_{\hat{T} \to \mathbb{R}^n}, \tag{6.25}$$

$$\|M^2\|_{T \to \mathbb{R}^n} \leqslant \|M\|^2_{\hat{T} \to \mathbb{R}^n}. \tag{6.26}$$

Proof. For every $x \in T$ we have due to (6.2)

$$0 = \hat{r} A x = \hat{r} M^T A x = \hat{r} A M x, \tag{6.27}$$

$\hat{r}$ denotes the adjoint operator to $\hat{p}$, therefore by (6.1)

$$M x \in \hat{T}. \tag{6.28}$$

As

$$\|Mx\|_1^2 = (Mx, Mx)_1 = (M^2x, x)_1 \leqslant \|M^2x\|_1 \cdot \|x\|_1, \tag{6.29}$$

$$\frac{\|M^2x\|_1}{\|Mx\|_1} \geqslant \frac{\|Mx\|_1}{\|x\|_1} \tag{6.30}$$

is fulfilled for $x \in T \setminus \{0\}$. Using (6.28), (6.30) we get (6.25). It is clear that

$$\|M^2\|_{T \to \mathbb{R}^n} \leqslant \|M\|_{T \to \mathbb{R}^n} \cdot \|M\|_{\hat{T} \to \mathbb{R}^n} \tag{6.31}$$

is valid. (6.26) is the consequence of (6.25) and (6.31). □

Lemma 6.7. *Let $\nu_1 = \nu_2 = 2$. Then for the algorithm* (2.13) *the estimate*

$$\frac{\|e(x^{i+1})\|_1}{\|e(x^i)\|_1} \leqslant \|M\|_{\hat{T} \to \mathbb{R}^n}^4 \tag{6.32}$$

holds.

Proof is the immediate consequence of Lemmas 6.5 and Lemma 6.6. □

Remark 6.1. If $\nu_1 \geqslant 2$, $\nu_2 \geqslant 2$ and $\varrho(M) \leqslant 1$, the estimate (6.32) is valid as well.

The operator norm $\|M\|_{\hat{T} \to \mathbb{R}^n}$ can be estimated using Brandt's technique with help of Céa's trick known from the theory of the finite element method. The following well-known lemma is proved e.g. in [1], Theorem 6.4.

Lemma 6.8. *Let $\boldsymbol{D}$ be positive definite, let $C > 0$ be such that for every $e \in \mathbb{R}^n$ there is a $v \in \mathbb{R}^m$ such that*

$$C\|\boldsymbol{D}^{1/2}(e - \hat{\boldsymbol{p}}v)\|_1 \leqslant \|e\|_1. \tag{6.33}$$

Then

$$\|M\|_{\hat{T} \to \mathbb{R}^n}^2 \leqslant 1 - C\omega\big[2 - \omega\varrho(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2})\big]. \tag{6.34}$$

Another estimate can be found in [3], Lemma 7.4.
(6.32) is valid also for the modified algorithm (4.8)—see Remark 4.1.

Theorem 1. *Let $\nu_1 \geqslant 2$, $\nu_2 \geqslant 2$, $\varrho(M) \leqslant 1$, let $\boldsymbol{D}$ be positive definite. Further let $C > 0$ fulfil the condition* (6.34). *Then for the rate of convergence of the algorithms* (2.13) *and* (4.8) *we have*

$$\frac{\|e(x^{i+1})\|_1}{\|e(x^i)\|_1} \leqslant \left\{1 - C\omega\big[2 - \omega\varrho(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2})\big]\right\}^2. \tag{6.35}$$

Proof follows immediately from Lemmas 6.7, 6.8 and Remarks 4.1 and 6.1. □

Now we will analyze the convergence of the algorithm (4.8). Remark 4.1 guarantees that every estimate of the rate of convergence of the algorithm (2.13) given by $\|e(x^{i+1})\|_1/\|e(x^i)\|_1 < \delta$, $0 \leqslant \delta < 1$, holds for the algorithm (4.8) as well. Our aim is to get the estimate which shows that (4.8) converges faster. The results below are the generalization of the results published in [2].

Lemma 6.9. *For the algorithm* (4.8) *we have*

$$\|e(x^{i+1})\|_1 = \min_{t \in \mathbf{R}} \|M^{\nu_2}[M^{\nu_1} e(x^i) - tpv]\|_1, \tag{6.36}$$

where

$$v = (A_2)^{-1} r A e(\tilde{x}), \quad \tilde{x} = S^{\nu_1}(x^i).$$

Proof is trivial. □

Lemma 6.10. *Let us set*

$$\tilde{\tilde{e}} = e(\tilde{x}) - pv. \tag{6.37}$$

Then

$$\|e(x^{i+1})\|_1 = \min_{\xi \in \mathbf{R}} \|M^{\nu_2}(\tilde{\tilde{e}} - \xi pv)\|_1 \tag{6.38}$$

holds for the algorithm (4.8).

Proof follows using the relation $e(\tilde{x}) = M^{\nu_1} e(x^i)$ and Lemma 6.9 for $t = \xi + 1$. □

Remark 6.2. It is not difficult to see that

$$\tilde{\tilde{e}} = Q_T M^{\nu_1} e(x^i) \tag{6.39}$$

and therefore

$$\tilde{\tilde{e}} \in T. \tag{6.40}$$

Lemma 6.2 implies that $M^{\nu_2} Q_T M^{\nu_1}$ is the iteration operator of the algorithm (2.13), therefore $M^{\nu_2}\tilde{\tilde{e}}$ is the error of the result of the algorithm (2.13) supposing that the same iteration x^i is on the input of both algorithms (2.13) and (4.8).

Definition 6.1. Let $e(\tilde{x}) \in \mathbb{R}^n \setminus (S \cup T)$. Let us define

$$k = \frac{\|\tilde{\tilde{e}}\|_1}{\|e(\tilde{x})\|_1}, \tag{6.41}$$

$$q_T = \frac{\|M^{\nu_2}\tilde{\tilde{e}}\|_1}{\|\tilde{\tilde{e}}\|_1}, \tag{6.42}$$

$$q_S = \frac{\|M^{\nu_2}pv\|_1}{\|pv\|_1}, \tag{6.43}$$

$$r = \frac{\|M^{\nu_2}e(\tilde{x})\|_1}{\|e(\tilde{x})\|_1}, \tag{6.44}$$

$$\bar{q}_T = \|M^{\nu_2}\|_{T\to\mathbb{R}^n}, \tag{6.45}$$

$$\bar{q}_S = \|M^{\nu_2}\|_{S\to\mathbb{R}^n}. \tag{6.46}$$

Remark 6.3. $\tilde{\tilde{e}} = Q_T e(\tilde{x})$, $pv = Q_s e(\tilde{x})$. As $e(\tilde{x}) \notin S$, $\tilde{\tilde{e}}$ is different from zero, analogously $e(\tilde{x}) \notin T$ and therefore $pv \neq 0$. Further $0 \notin \mathbb{R}^n \setminus (S \cup T)$, and hence $e(\tilde{x}) \neq 0$. The numbers k, q_T and q_S are therefore well-defined. The following inequalities are evident:

$$q_T \leqslant \bar{q}_T, \tag{6.47}$$

$$q_S \leqslant \bar{q}_S, \tag{6.48}$$

$$0 < k < 1. \tag{6.49}$$

Lemma 6.11. *Let $e(\tilde{x}) \in \mathbb{R}^n \setminus (S \cup T)$. Then*

$$\|e(x^{i+1})\|_1^2 \leqslant \tag{6.50}$$
$$\leqslant \|M^{\nu_2}Q_T M^{\nu_1}\|^2_{\mathbb{R}^n\to\mathbb{R}^n}\left\{1 - \frac{[r^2 - q_T^2k^2 - q_S^2(1-k^2)]^2}{4q_S^2(1-k^2)q_T^2k^2}\right\}\|e(x^i)\|_1^2$$

holds for the algorithm (4.8)

Proof. It is not difficult to prove that the right-hand side of (6.38) is minimal for

$$\hat{\xi} = \frac{(M^{\nu_2}\tilde{\tilde{e}}, M^{\nu_2}pv)_1}{\|M^{\nu_2}pv\|_1^2}. \tag{6.51}$$

Using (6.38) and (6.51) we get

$$\|e(x^{i+1})\|_1^2 = \|M^{\nu_2}Q_T M^{\nu_1}e(x^i)\|_1^2 - \frac{(M^{\nu_2}\tilde{\tilde{e}}, M^{\nu_2}pv)_1^2}{\|M^{\nu_2}pv\|_1^2}. \tag{6.52}$$

Further

$$\begin{aligned}\|M^{\nu_2}pv\|_1^2 &= q_S^2\|pv\|_1^2\\ &= q_S^2\|Q_S e(\tilde{x})\|_1^2 = q_S^2\|(I-Q_T)e(\tilde{x})\|_1^2\\ &= q_S^2(\|e(\tilde{x})\|_1^2 - \|Q_T e(\tilde{x})\|_1^2)\\ &= q_S^2\Big(1-\frac{\|Q_T e(\tilde{x})\|_1^2}{\|e(\tilde{x})\|_1^2}\Big)\|e(\tilde{x})\|_1^2\\ &= q_S^2\Big(1-\frac{\|\tilde{\tilde{e}}\|_1^2}{\|e(\tilde{x})\|_1^2}\Big)\|e(\tilde{x})\|_1^2\\ &= q_S^2(1-k^2)\|e(\tilde{x})\|_1^2.\end{aligned}$$

We have proved

$$\|M^{\nu_2}pv\|_1^2 = q_S^2(1-k^2)\|e(\tilde{x})\|_1^2. \tag{6.53}$$

Using this inequality we have

$$\begin{aligned}\|M^{\nu_2}e(\tilde{x})\|_1^2 &= \big(M^{\nu_2}(\tilde{\tilde{e}}+pv), M^{\nu_2}(\tilde{\tilde{e}}+pv)\big)_1\\ &= \|M^{\nu_2}\tilde{\tilde{e}}\|_1^2 + 2(M^{\nu_2}\tilde{\tilde{e}}, M^{\nu_2}pv)_1 + \|M^{\nu_2}pv\|_1^2\\ &= q_T^2\|\tilde{\tilde{e}}\|_1^2 + 2(M^{\nu_2}\tilde{\tilde{e}}, M^{\nu_2}pv)_1 + q_S^2(1-k^2)\|e(\tilde{x})\|_1^2\\ &= q_T^2k^2\|e(\tilde{x})\|_1^2 + 2(M^{\nu_2}\tilde{\tilde{e}}, M^{\nu_2}pv)_1\\ &\quad + q_S^2(1-k^2)\|e(\tilde{x})\|_1^2.\end{aligned} \tag{6.54}$$

(6.54) and (6.44) yield

$$(M^{\nu_2}\tilde{\tilde{e}}, M^{\nu_2}pv)_1 = \frac{1}{2}\big[r^2 - q_T^2k^2 - q_S^2(1-k^2)\big]\cdot\|e(\tilde{x})\|_1^2. \tag{6.55}$$

Substituting (6.55) and (6.53) into (6.52) we get

$$\begin{aligned}\|e(x^{i+1})\|^2 = \|M^{\nu_2}Q_{\mathbf{T}}M^{\nu_1}e(x^i)\|_1^2\\ - \frac{\big[r^2 - q_T^2k^2 - q_S^2(1-k^2)\big]^2}{4q_S^2(1-k^2)}\|e(\tilde{x})\|_1^2.\end{aligned} \tag{6.56}$$

Remark 6.2 and Definition 6.1 yield

$$\|M^{\nu_2}Q_{\mathbf{T}}M^{\nu_1}e(x^i)\|_1 = q_Tk\|e(\tilde{x})\|_1 \tag{6.57}$$

and

$$\|M^{\nu_2}Q_{\mathbf{T}}M^{\nu_1}e(x^i)\|_1 \leqslant \|M^{\nu_2}Q_{\mathbf{T}}M^{\nu_1}\|_{\mathbf{R}^n\to\mathbf{R}^n}\|e(x^i)\|_1. \tag{6.58}$$

From (6.57) and (6.58) the inequality

$$q_Tk\|e(\tilde{x})\|_1 \leqslant \|M^{\nu_2}Q_{\mathbf{T}}M^{\nu_1}\|_{\mathbf{R}^n\to\mathbf{R}^n}\|e(x^i)\|_1 \tag{6.59}$$

follows. We get (6.50) using (6.56), (6.57) and (6.59) □

Definition 6.2. For every $\bar{r} \in (0,1)$ let us define

$$\mathcal{A}(\bar{r}) = \left\{ x \in \mathbb{R}^n, x \neq \hat{x}, \frac{\|M^{\nu_1+\nu_2} e(x)\|_1}{\|M^{\nu_1} e(x)\|_1} > \bar{r} \right\}.$$

Remark 6.4. If $\bar{r} < \varrho(M)$, then $\mathcal{A}(\bar{r}) \neq \emptyset$. If $x = x^i$ then $\frac{\|M^{\nu_2+\nu_1} e(x)\|_1}{\|M^{\nu_1} e(x)\|_1} = r$ from (6.44).

Lemma 6.12. *Let $\bar{r} \in (0,1)$ such that*

$$\bar{q}_T < \bar{r}, \quad \bar{q}_S < \bar{r}. \tag{6.60}$$

Then for every $x^i \in \mathcal{A}(\bar{r})$

$$\frac{\|e(x^{i+1})\|_1^2}{\|e(x^i)\|_1^2} \leqslant \|M^{\nu_2} Q_T M^{\nu_1}\|^2_{\mathbb{R}^n \to \mathbb{R}^n} \left\{ 1 - \inf_{k \in (0,1)} \frac{\left[\bar{r}^2 - \bar{q}_T^2 k^2 - \bar{q}_S^2 (1-k^2)\right]^2}{4\bar{q}_S^2 (1-k^2) \bar{q}_T^2 k^2} \right\} \tag{6.61}$$

holds for the algorithm (4.8).

Remark 6.5. The assumption $x^i \in \mathcal{A}(\bar{r})$ means that the error of the approximation x^i after pre-smoothing (i.e. $M^{\nu_1} e(x^i)$) is smooth. Lemma 6.12 states that the nearer to 1 $\bar{r}$ is (i.e. x^i smoother) the more effective the iteration is. Smoothness of $M^{\nu_1} e(x^i)$ can be improved by increasing the number of pre-smoothing steps ν_1.

Proof. (6.60) implies $e(\tilde{x}) \in \mathbb{R}^n \setminus (S \cup T)$ and (6.61) is therefore the consequence of Lemma 6.11 taking into account that $q_T < \bar{q}_T$ and $q_S < \bar{q}_S$, $r \geqslant \bar{r}$ (see Remark 6.4). □

Remark 6.6. As $M^{\nu_2} Q_T M^{\nu_1}$ is the iteration operator of the algorithm (2.13),

$$\|M^{\nu_2} Q_T M^{\nu_1}\|^2_{\mathbb{R}^n \to \mathbb{R}^n} = \sup \frac{\|e(x^{i+1})\|_1}{\|e(x^i)\|_1}.$$

Using Lemma 6.5 and (6.61) we get

$$\frac{\|e(x^{i+1})\|_1^2}{\|e(x^i)\|_1^2} \leqslant \|M^{\nu_1}\|^2_{T \to \mathbb{R}^n} \cdot \|M^{\nu_2}\|^2_{T \to \mathbb{R}^n} \left\{ 1 - \inf_{k \in (0,1)} \frac{\left[\bar{r}^2 - \bar{q}_T^2 k^2 - \bar{q}_S^2 (1-k^2)\right]^2}{4\bar{q}_S^2 (1-k^2) \bar{q}_T^2 k^2} \right\}. \tag{6.62}$$

Theorem 2. *Let $\nu_1 \geqslant 2$, $\nu_2 \geqslant 2$, $\varrho(M) \leqslant 1$. Let us set*

$$\hat{q}_T = \|M\|_{\hat{T} \to \mathbb{R}^n}. \tag{6.63}$$

Let $\bar{r} \in (0,1)$ *be such that* $\hat{q}_T^2 < \bar{r}$, $\bar{q}_s < \bar{r}$. *Then for every* $x^i \in \mathcal{A}(\bar{r})$ *the estimate*

$$\frac{\|e(x^{i+1})\|_1^2}{\|e(x^i)\|_1^2} \leqslant \hat{q}_T^8 \Big(1 - \inf_{k \in (0,1)} \frac{\left[\bar{r}^2 - \hat{q}_T^4 k^2 - \bar{q}_S^2 (1-k^2)\right]^2}{4\bar{q}_S^2 (1-k^2)\hat{q}_T^2}\Big)$$

is valid for the algorithm (4.8).

Proof. Using Lemma 6.6 and the assumption $\nu_1, \nu_2 > 2$ we get

$$\|\boldsymbol{M}^{\nu_1}\|_{T \to \mathbf{R}^n} \leqslant \hat{q}_T^2,$$
$$\bar{q}_T = \|\boldsymbol{M}^{\nu_2}\|_{T \to \mathbf{R}^n} \leqslant \hat{q}_T.$$

Now the statement follows immediately from Lemma 6.12. □

Remark 6.7. $\hat{q}_T = \|\boldsymbol{M}\|_{\hat{T} \to \mathbf{R}^n}$ can be estimated using Lemma 6.8, i.e. under the assumptions $\boldsymbol{D}$ is positive definite and (6.33) we have

$$\hat{q}_T^2 \leqslant 1 - C\omega[2 - \omega\varrho(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})].$$

The operator norm $\bar{q}_S$ is estimated in [2] for a model example.

Remark 6.8. If the assumptions $\hat{q}_T^2 < \bar{r}$, $\bar{q}_S < \bar{r}$, $x^i \in \mathcal{A}(\bar{r})$ are not fulfilled the rate of convergence can be estimated via Theorem 1.

Remark 6.9. Due to Theorem 2 it is desirable for $\bar{r} \in (0,1)$ to be as large as possible. This aim can be reached by choosing ν_1 sufficiently large. Practice has shown the optimal value of ν_1 is approximately 4–8, that is about twice the number of smoothing iterations recommended for the algorithms of the type (2.13).

7. Numerical experiments

In this section results of experiments carried out using the algorithm (4.8) are contained. The algorithm was tested on the following boundary value problem discretized by the finite differences method on the regular square qrid:

$$-\frac{\partial}{\partial x}\left(\varepsilon(x,y)\frac{\partial u}{\partial x}\right) - \frac{\partial^2 u}{\partial y^2} = f \quad \text{on } \Omega = (0,1) \times (0,1),$$
$$u = 0 \quad \text{on } \partial\Omega.$$

Experiments were carried out for the following two cases:

(7.1) $\qquad \varepsilon(x,y) = \text{const},$

(7.2) $\qquad \varepsilon(x,y) = 100^{x+y-1}.$

Assigned parameters of the method:

$$\begin{aligned} \nu_1 &= 7 \\ \nu_2 &= 2 \\ \omega &= 0.63 \quad \text{(see (5.8))} \\ \Theta &= 0.1 \quad \text{(see (5.4))} \\ \gamma &= 2 \quad \text{(see (3.2e))} \\ n &= 2500 \end{aligned}$$

Notation in the table:

[ϱ –] $\left(\frac{\|e(x^3)\|_1}{\|e(x^0)\|_1}\right)^{1/3}$

[τ_i –] time of one iteration (in the iterative phase)

[τ_p –] time of the setup phase divided by t_i

[C_g –] the so called geometrical complexity i.e. the total number of equations over all levels divided by the number of equations of the finest level

[C_a –] the so called algebraic complexity, i.e. the total number of nonzero coefficients over all levels divided by the number of nonzero coefficients of the finest level.

problem	ε	τ_i	τ_p	C_g	C_a	ϱ
(7.1)	10^{-4}	37	1.8	1.57	1.93	$4.19 \cdot 10^{-3}$
	10^{-3}	25	2.1	1.50	1.84	$4.12 \cdot 10^{-3}$
	10^{-2}	40	1.3	1.52	2.08	$3.82 \cdot 10^{-3}$
	10^{-1}	26	2.0	1.43	1.76	$4.00 \cdot 10^{-3}$
	1	34	2.2	1.41	2.16	$7.00 \cdot 10^{-3}$
	10	27	1.6	1.43	1.75	$4.04 \cdot 10^{-3}$
	100	40	1.2	1.52	2.11	$3.87 \cdot 10^{-3}$
	1000	25	1.7	1.50	1.84	$3.93 \cdot 10^{-3}$
	10000	33	1.2	1.57	1.93	$4.09 \cdot 10^{-3}$
(7.2)		34	1.1	1.55	1.92	$3.32 \cdot 10^{-3}$

References

[1] *R. Blaheta*: Iterative Methods for Numerical Solving of the Boundary Value Problems of Elasticity. Thesis, Ostrava, 1989. (In Czech.)

[2] *S. Míka, P. Vaněk*: Modification of the Two-level Algorithm with Overcorrection. Appl. Math. *37* (1992), no. 1.

[3] *S. Míka, P. Vaněk*: The Acceleration of Two-level Algorithm by Aggregation in Smoothing Process. Appl. Math. *37* (1992), no. 5.

[4] *P. Vaněk*: Acceleration of a Two-level Algorithm by Smoothing Transfer Operators. Appl. Math. *37* (1992), no. 4..

[5] *W. Hackbusch*: Multi-Grid Methods and Applications. Springer-Verlag, 1985.

[6] *J. Mandel*: Adaptive Iterative Solvers in Finite Elements. To appear.

[7] *O. Axelsson, V.A. Barker*: Finite Element Solution of Boundary Value Problems. Academic Press, 1984.
[8] *S.F. McCormick*: Multi-Grid Methods. SIAM (1987).
[9] *P. Leitl*: private communication. Nynice, 1993.
[10] *J. Mandel*: Balancing Domain Decomposition. Communications in Numerical Methods in Engineering *9* (1993).

Author's address: *Petr Vaněk*, Dept. of Math., ZČU, Americká 42, 306 14 Plzeň, Czech Republic, e-mail: **pvanek@kma.zcu.cz**.