An adaptive multigrid on block-structured grids

O B Feodoritova, V T Zhukov
Keldysh Institute of Applied Mathematics of RAS, Miusskaya sq., 4, Moscow, 125047, Russia
E-mail: feodor@kiam.ru, zhukov@kiam.ru

Abstract. An implementation of the multigrid method on conformal block-structured grids is proposed. The algorithm is intended to solve a boundary-value problem for elliptic PDEs. Each block is discretized using a structured hexahedron grid. The set of multilevel grids are arranged in hierarchical levels. In each block the discrete finite volume scheme is constructed on the fine level grid. The coarse level equations are formed by the Galerkin procedure. The presence of the irregular block connections leads to irregular stencils in the nodes which adjoin of several blocks. Instead of a special irregular node treatment we propose to find the solution in these nodes in additive manner. Such an opportunity is provided by the use of the explicit iterative procedures for both the smoothers and the coarsest grid solver. We also present an adaptive technique which adjusts these procedures for achieving the prescribed multigrid convergence rate. The proposed approach enhances the potential of the multigrid method in ultra-parallel computing.

1. Introduction

The presented multigrid algorithm is intended to solve a boundary-value problem for elliptic PDEs on a multiblock grid. The multiblock techniques have been developed for decades to generate high quality grids. Multigrid/multiblock frameworks are widely used in many areas of science and engineering (cosmology application, CFD computations). We assume domain decomposition is made and in each block a structured hexahedron grid is constructed. The multiblock grid is based on node-to-node matching across the interface of neighboring blocks. A block face might adjoin a few blocks. Such conductivities simplify the decomposition procedure and reduce the total amount of blocks.

We build the multigrid algorithm on a such multiblock grid. In section 2, we give a presentation of the single domain algorithm, which is a version of Fedorenko’s multigrid method [1], see [2–5]. In section 3, we briefly outline the iteration algorithm for the coarsest grid problem. In section 4, the adaptive variant of this algorithm is given. In sections 5 – 8 we introduce adaptive smoothing technique to adjust smoothers for achieving the prescribed multigrid convergence rate and for improving the robustness of the multigrid method even in the presence of a strong anisotropy in the problem. To achieve this goal one need to use a sufficiently large degree in the polynomial smoothing operators. Such approach has attracted due to high efficiency of polynomial smoothers in parallel computing [6–8]. We build smoothers based on Chebyshev polynomials, which are implemented by explicit iterations. In Section 9, we show how the explicit iterative algorithms provide a simple mechanism for multigrid implementation on conformal block-structured grids.
2. General multigrid scheme

For simplicity we consider application of the multigrid method for solving the Poisson equation

\[ Lu = f \]  

in a domain \( \Omega \subset \mathbb{R}^3 \) with homogeneous Dirichlet boundary condition, here \( L \) is the Laplacian operator. All constructions are valid for a general linear elliptic self-adjoint positive semi-definite operator. The space \( U_h \) of grid functions given on a conformal block-structured grid \( \Omega_h \) is determined in the usual way with the \( L^2(\Omega_h) \) inner product and the corresponding norm. On the grid space \( U_h \) we define a discrete operator \( L_h \) approximating the operator \( L \) with the second order of accuracy. To be more specific, it can be assumed that the grid functions are given at the grid nodes and \( L_h \) arises from 7-pt or 27-pt finite volume discretization of the problem (1). The operator \( L_h \) is self-adjoint and its eigenvalues \( \lambda \) are positive and belong to the real interval \( [\lambda_{\text{min}}; \lambda_{\text{max}}] \). The discrete approximation of (1) is the linear system of equations

\[ L_h u_h = f_h. \]  

The approximation (2) is written only on the original grid. The discrete equations on a coarse grid are formed from the fine grid equations. This provides sufficient freedom and convenience in the setting of the auxiliary grids – they are formed from the original grid by removing any specified grid lines.

In this section, we briefly recall the basic details of the multigrid algorithm [1–5] that uses geometrically refined grids. Every multigrid iteration step consists in the transition from a fine grid to the next level up to the coarsest grid and back \( (V \text{- cycle}) \). It is convenient to describe the method, when there are only two grids: fine \( h \) and coarse \( H \). The fine grid functions are denoted in small letters, the coarse grid functions are denoted in capital letters. Let \( r = L_h v - f_h \) be a smooth residual of the fine grid system (2). The prolongation operator \( I \) interpolates a coarse grid function \( W \) on the fine grid function: \( I : W \rightarrow IW \). We require to find such a coarse grid function \( W \), that \( L_h (v - IW) = f_h \). This system is solved in the weak Galerkin form: \( (L_h (v - IW), IV) = (f_h, IV) \) for all \( IV \). Using the equality \( (L_h (v - IW), IV) = (I^* r - I^* L_h IW, V) \) we obtain the coarse grid equations for \( W \):

\[ L_H W = R, \]  

where \( L_H = I^* L_h I, R = I^* r, \) and \( I^* \) is the restriction operator, conjugated to the prolongation operator \( I \). Both operators \( I, I^* \) are computed within each block and only \( I^* \) needs to be computed on the block interfaces. These operators are described in [2–5], where along with the trilinear interpolation operator \( I \), the problem-dependent operators are presented. Such operators provide robustness of the multigrid algorithm for equations with discontinuous coefficients.

The given two-grid algorithm can be easily generalized to an arbitrary number of grid levels using recursion. We implicitly restrict the number of grid levels by the condition \( N_c \gg n \), where \( N_c \) is the size of the coarsest grid system, \( n \) is the number of parallel processes. The efficiency of the algorithm depends on the multigrid triad - intergrid transfer operators, coarsest grid solvers and the smoothing operators. As a restriction operator, we always construct an operator conjugated to a prolongation operator using local calculations, i.e. without using matrix transposition.

Smoothing operators play a key role in the multigrid algorithm; the design and technique of automatic adaptation of the smearers during multigrid iterations is presented in Sections 5–8.
3. Coarsest grid solver
Firstly we introduce the algorithm for solving the coarsest grid problem, i.e. a system of linear equations
\[ A \cdot u = g \]  
(4)
with a residual \( g \) on the right-hand side. Here and below we do not use a subscript to point out the coarsest level. We do not assume that the coarsest grid system is small. On the contrary, we assume that this system is large enough in relation to the number of processors. One can apply an iterative scheme to computed a sufficiently good approximate solution. We propose to solve this system by the Chebyshev iterative method [9] with the optimal set of the parameters \( \omega_j \) reordered for stability:
\[ u^j = u^{j-1} - \omega_j \cdot (A \cdot u^{j-1} - g), \quad j = 1, \ldots, p, \]  
(5)
where \( u^j \) is an iterative solution, \( u^0 \) is an initial guess, usually, \( u^0 \equiv 0 \). A number iterations \( p \) is determined by the condition of achieving the prescribed accuracy \( \varepsilon \) according the criteria \(|r_p| < \varepsilon |r_0|\), where \( r_0 \) and \( r_p = A \cdot u^p - g \) are the initial and final residuals respectively. An estimation of \( p \) is (see [9]):
\[ p = p(\varepsilon, \eta) \approx \frac{\ln(\varepsilon^{-1} + \sqrt{\varepsilon^{-2} - 1})}{\ln \rho}, \quad \rho = \frac{1 + \sqrt{\eta}}{1 - \sqrt{\eta}}, \quad \eta = \frac{\lambda_{\min}}{\lambda_{\max}}, \]  
(6)
where \( \lambda_{\min} \) and \( \lambda_{\max} \) are the minimal and maximal eigenvalues of the coarsest operator \( A \). This iteration process is defined by the Chebyshev polynomial \( F_p \) of degree \( p \), that deviates least from zero on the interval \([\lambda_{\min}; \lambda_{\max}]\) and is normalized by the condition \( F_p(0) = 1 \).

The polynomial \( F_p(\lambda) \) is computed by the standard polynomial
\[ T_p(x) = \cos(p \arccos x), \quad |x| \leq 1; \quad T_p(x) = \cosh(p \text{Arccosh} x), \quad |x| > 1 \]  
(7)
by means of the linear mapping \([-1; 1]\) on \([\lambda_{\min}; \lambda_{\max}]\) which maps the point \( x = 1 \) in the point \( t = \lambda_{\min} \). Notice the point \( \lambda_0 = 0 \) corresponds to \( x_0 = x(\lambda_0) > 1 \), where \( x = x(\lambda) \) is the inverse mapping. The operator \( F_p(A) \) maps an initial guess \( u^0 \) in the new approximate \( u_p = F_p(A) \cdot u_0 + \left[I - F_p(A)\right] \cdot A^{-1} g \). This implies \( r_p = F_p(A) \cdot r_0 \). Applying to both sides of this equality the operator \( A^{-1} \), we get that the error \( z_p = u_p - u \) is \( z_p = F_p(A) \cdot z_0 \). This means that the residual and error components after the completion of the iterations are reduced to \( \varepsilon^{-1} > 1 \) times uniformly over the spectrum \([\lambda_{\min}; \lambda_{\max}]\). One can see the function \( F_p(\lambda) \) takes the values \( \pm \varepsilon \) at the extremum points on the interval \([\lambda_{\min}; \lambda_{\max}]\) and grows rapidly outside this interval according to (7). For the algorithm (5) we need to set the parameters \( \lambda_{\min}, \lambda_{\max} \), usually unknown. As \( \lambda_{\max} \) one can take a fairly good estimate according to the Gershgorin theorem, and as \( \lambda_{\min} \) usually one might use an empirical estimate. We propose an effective adaptive estimate of \( \lambda_{\min} \), which is presented in Section 4.

4. Adaptive Chebyshev method for solving the coarsest grid problem
For successful application of the Chebyshev method, one need to know the bounds of the spectrum \([\lambda_{\min}; \lambda_{\max}]\). By \( \lambda_{\min} \) and \( \lambda_{\max} \) we mean the exact (usually unknown) lower and upper bounds of the spectrum, by \( \lambda_{\min}^* \) and \( \lambda_{\max}^* \) are their approximate estimates. It is known that if \( 0 < \lambda_{\min}^* \leq \lambda_{\min} \leq \lambda_{\max} \leq \lambda_{\max}^* \), then the Chebyshev method converges with \( \lambda_{\min}^* \) and \( \lambda_{\max}^* \) bounds. If these estimations are inexact the convergence rate decreases. When applying the adaptation procedure, we will assume the fulfillment of the necessary and sufficient condition \( \lambda_{\max} \leq \lambda_{\max}^* + \lambda_{\min}^* \) of convergence of iterations with inexact estimations of the spectrum bounds [9].
As \( \lambda_{\text{max}}^* \) we take the upper bound for \( \lambda_{\text{max}} \), obtained by the Gershgorin theorem of circles. As initial guess \( \lambda_{\text{min}}^* \) we take a rather rough one, and will refine it in the adaptation cycle or in the external iterative process. We need to set \( \lambda_{\text{min}}^* \geq \lambda_{\text{min}} \). To fulfill this condition, we take Rayleigh-Ritz relation \( \lambda_{\text{min}}^* = (Av, v)/(v, v) \) with any non-zero \( v \), for instance with the right hand side of the system (4).

The procedure for refining the estimate \( \lambda_{\text{min}}^* \) is based on the assumption that during the application of the iteration operator to the initial residual the eigenfunction corresponding to the lower exact bound of the spectrum becomes the leading mode in the resulting residual.

The adaptation algorithm is as follows. Suppose we need to solve the system (4) with a given tolerance \( \varepsilon_{\text{tot}} \). We set a lower tolerance \( \varepsilon_1 < \varepsilon_{\text{tot}} \). To be specific we take \( \varepsilon_{\text{tot}} = 10^{-m} \) with \( m \geq 1 \) and \( \varepsilon_1 = 10^{-i} \) with \( i < m \). After performing one step of the Chebyshev algorithm for the given input data \( \lambda_{\text{min}}^*, \lambda_{\text{max}}^*, \varepsilon_1 \) we have

\[
\delta = \frac{\|r_p\|}{\|r_0\|}, \quad r_p = F_p(A)r_0.
\]

Here \( \delta \) is the real accuracy achieved in the current adaptation cycle, \( r_0 \) and \( r_p \) are the initial and final residuals. If \( \delta > \varepsilon_1 \), then the maximal eigenvalue of the operator \( F_p(A) \) on the interval \([\lambda_{\text{min}}, \lambda_{\text{max}}]\) is reached when \( \lambda = \lambda_{\text{min}} \), i.e. \( \lambda_{\text{max}}(F_p) = F_p(\lambda_{\text{min}}) \). Therefore, this value can be approximately found as a result of performing one step of the known power method of finding the largest eigenvalue of the operator \( F_p(A) \). To approximate \( \lambda_{\text{max}}(F_p) \), we take the resulting ratio \( \delta \) of the residual norms (8). Using \( \lambda_{\text{max}}(F_p) \), we find a new approximation \( \lambda_{\text{new}}^* \) to \( \lambda_{\text{min}}^* \) (see below ) and set \( \lambda_{\text{min}}^* = \lambda_{\text{new}}^* \). If the accuracy \( \varepsilon_{\text{tot}} \) is not achieved, then we proceed to a new adaptation cycle with the data \( \lambda_{\text{min}}^*, \lambda_{\text{max}}^*, \varepsilon_1 \).

An additional reason to use this version of the power method is the important property of optimal Chebyshev polynomials [10]: outside the optimality segment, Chebyshev polynomials have the opposite extreme property, namely, they deviate from zero as much as possible compared to all polynomials satisfying a given normalization.

The procedure for finding a new approximation \( \lambda_{\text{min}}^* \) is simple. It is clear that the condition \( F_p(\lambda) = \delta \) is attained for \( 0 < \lambda < \lambda_{\text{min}}^* \), which corresponds to the interval \( x > 1 \) for the polynomial \( T_p(x) \), i.e. on the hyperbolic branch. On this branch \( T_p(x) \) is a strictly monotone function, so the root of the equation \( F_p(\lambda) - \delta = 0 \) on the interval \( 0 < \lambda < \lambda_{\text{min}}^* \) exists and unique, it is the new value of \( \lambda_{\text{min}}^* \).

If on the \( k \)-th adaptation cycle \( \delta = \delta_k \leq \varepsilon_{\text{tot}} \), then we perform next adaptation cycle, optionally either with the old data \( \lambda_{\text{min}}^*, \lambda_{\text{max}}^*, \varepsilon_1 \), or we change the tolerance by taking a new value \( \varepsilon_1 = \varepsilon_{\text{tot}}/(\delta_1 \times \cdots \times \delta_k) \). The condition \( \delta_k \leq \varepsilon_{\text{tot}} \) means that the estimates of the bounds of the spectrum are quite satisfactory, therefore the tolerance \( \varepsilon_{\text{tot}} \) can be achieved in the next adaptation cycle. During multigrid iterations this adaptation procedure might be implemented on each multigrid iteration in one step without repeating.

5. Smoothing

A smoothing (relaxation) operator \( S_p \) transforms an approximate solution \( v \) of the system \( A_h u_h = g_h \) on a current grid level to a "smoothed" approximation \( v_{\text{new}} \), i.e. only the oscillating error modes are damped. Two smoothing operators \( S_p = S_p(A_h) \) are presented here (see [3 - 5]). The first one is the Chebyshev polynomial of degree \( p \), while the second one is a rational function called the LI-M. Both of these operators are self-adjoint and their eigenvalues are the values \( S_p(\lambda) \) on the spectrum of \( A_h \). These operators serve to reduce the high-frequency components of an initial error \( e^0 \), \( e^1 = S_p \cdot e^0 \). The function \( S_p(\lambda) \) must be small on the high-frequency subinterval \([\lambda_{\text{min}}; \lambda_{\text{max}}]\). We distinguish high- and low-frequency components of grid functions on \( \Omega_h \), this separation is done with respect to the coarse grid: high-frequency components are not represented on the coarse grid. More strictly, there is a decomposition of the space \( U_h \) into a direct sum of two invariant subspaces associated to two groups of eigenvalues corresponding
to the low-and high-frequency spectral subintervals \([\lambda_{\min}^c; \lambda_{\min}^c]\) and \([\lambda_{\max}^c; \lambda_{\max}^c]\) respectively. The quality of the smoothing procedures depends on the choice of \(\lambda_{\min}^c\) and the construction of a smoothing operator \(S_p\).

How to define optimally the bound \(\lambda_{\min}^c\) and the smoothing operator \(S_p = S_p(A_h)\) are the key questions in multigrid practice. Usually we do not know \(\lambda_{\min}^c\) in advance therefore we take \(\lambda_{\min}^c = \lambda_{\max}^c / 6\) as an initial guess and correct automatically this bound during multigrid iterations. We verified such an adaptation procedure in [4, 5] and here describe the adaptation algorithm with improvement.

When a smoothing operator is applied, the parameters of the method (degree \(p\) of the polynomial and the corresponding set of Chebyshev parameters) are chosen to suppress the high-frequency residual components with the prescribed accuracy \(\varepsilon_{\text{smooth}}\) (usually \(\varepsilon_{\text{smooth}} = 0.5\)). We check the reduction of the norm of the entire residual that provides the desired reduction of the high-frequency residual components. We can estimate the smoothing factor of the operator \(S_p\) in the high-frequency space as follows: \(\rho = \max |S_p(\lambda)| \leq 1\), where maximum is taken over the interval \([\lambda_{\min}^c; \lambda_{\max}^c]\).

6. Chebyshev polynomial smoother

We introduce a smoother which is based on the Chebyshev polynomial \(F_p(\lambda)\) that deviates least from zero on the high-frequency interval \([\lambda_{\min}^c; \lambda_{\max}^c]\) and is normalized by the condition \(F_p(0) = 1\). The polynomial \(F_p\) is expressed in terms of the standard polynomial \(T_p(x)\) (6). On the entire spectrum, the inequality \(|F_p(\lambda)| < 1\) holds, i.e. this smoothing procedure is a converging iterative method. We define \(p = p(\varepsilon, \eta)\) by the formula (6) with a given value \(\eta = \lambda_{\min}^c / \lambda_{\max}^c\) and a prescribed tolerance \(\varepsilon = \varepsilon_{\text{smooth}}\). We wait that the error components on the interval \([\lambda_{\min}^c; \lambda_{\max}^c]\) are decreased by \(\varepsilon^{-1} > 1\) times uniformly over this spectral range and the smoother error reduction function is \(\rho_{\text{Cheb}}(\lambda) = F_p(\lambda)\) which takes the values \(\pm \varepsilon\) at the extremal points of the polynomial on \([\lambda_{\min}^c; \lambda_{\max}^c]\).

The operator \(F_p(\lambda)\) looks like the ideal smoother. In practice, the situation differs from the ideal one and usually we do not know the value \(\lambda_{\min}^c\). The Chebyshev smoothing operator turns out to be sensitive to the inaccuracy of specifying \(\lambda_{\min}^c\). For this reason, we examine other smoothing procedure, LI-M, which constructed using the Chebyshev polynomial but it presents a rational function of discrete operator.

7. LI-M smoother

For given \(\eta = \lambda_{\min}^c / \lambda_{\max}^c\) and a prescribed tolerance \(\varepsilon = \varepsilon_{\text{smooth}}\) we define \(p = p(\varepsilon, \eta)\) by the formula

\[
p = \left[\frac{\pi}{4} \sqrt{\left(\varepsilon^{-1} - 1\right)/\eta + 1}\right] \tag{9}\]

where \([\circ]\) is the least integer greater than or equal to \(x\). To reduce the error by \(\varepsilon^{-1} > 1\) times on the high-frequency interval \([\lambda_{\min}^c; \lambda_{\max}^c]\), we introduce the LI-M smoother

\[
S_{\nu} = (I - G_p^2) \cdot (I + \tau A_h)^{-1}. \tag{10}\]

The implementation of this operator is similar to the algorithm (5). Here the subscript \(\nu = 2p-1\) is connected with order \(p\) of the Chebyshev polynomial of the first kind \(G_p(\lambda)\) that deviates least from zero on the interval \([\lambda_0; \lambda_{\max}^c]\), where

\[
\lambda_0 = \lambda_{\max}^c \frac{z_1 - 1}{z_1 + 1} \in [-1/\tau; 0], \quad \tau = \frac{1}{\lambda_{\max}^c} \left(16p^2/\pi^2 - 1\right), \quad z_1 = \cos \frac{\pi}{2p}, \tag{11}\]

with the normalization \(G_p(-1/\tau) = 1\). On the interval \([0; \lambda_{\max}^c]\), we have \(|G_p(\lambda)| \leq 1\) and \(|S_{\nu}(\lambda)| \leq 1\), \(G_p(0) = 0\).
The operator (10) is studied in detail in [5]. The polynomial $G_p(\lambda)$ can express in terms of the optimal Chebyshev polynomial $T_p(x)$ (6):

$$G_p(\lambda) = \frac{H_p(\lambda)}{H_p(-1/\tau)}, \quad H_p(\lambda) = \prod_{m=1}^{m=p} (a_m - \lambda) \equiv T_p(z_1 - (z_1 + 1) \cdot \lambda / \lambda_{\text{max}}^*).$$

The zeros $a_m$ of the polynomial $H_p$ present the iteration parameters of the method and they are expressed by the zeros $z_m$ of the Chebyshev polynomial $T_p$:

$$a_m = \frac{z_1 - \beta_m}{1 + z_1}, \quad m = 1, \ldots, p. \quad (12)$$

The transition from a solution $v$ to a "smoothed" solution $v_{\text{new}}$ is executed in $\nu = 2p - 1$ explicit steps:

$$y^m = \frac{1}{1 + \tau \cdot b_m} \{v + \tau \cdot b_m \cdot y^{m-1} - \tau \cdot (A_h \cdot y^{m-1} - f_h)\}, \quad m = 1, \ldots, 2p - 1$$

where $y^0 = v$, $\tau$ and the iteration parameters $\{b_1, \ldots, b_{2p-1}\} \equiv \{a_1, \ldots, a_p, a_2, \ldots, a_p\}$ are defined by (11) and (12). The result is a smoothed approximation to the solution: $v_{\text{new}} = y^{2p-1}$. The spectrum of the smoothing operator (10) is

$$\rho_{\text{LIM}}(\lambda) = \frac{1 - G_p^2(\lambda)}{1 + \tau \lambda}.$$ 

By the construction of $G_p$, we have the following inequality for the damping factor:

$$0 \leq \rho_{\text{LIM}}(\lambda) \leq \frac{1}{1 + \tau \lambda}. \quad (13)$$

8. Adaptive smoothers

During multigrid iterations at some grid level we consider a pre-smoothing stage. After smoothing step for the given bounds $\lambda_{\text{min}}^c$ and $\lambda_{\text{max}}^c$ we obtain the relation $\delta = \|r_p\|/\|r_0\| < 1$ for the initial and final residuals. The prescribed smoothing accuracy $\varepsilon$ might not be achieved, i.e. $\varepsilon < \delta < 1$. Then we refine the bound $\lambda_{\text{min}}^c$ by the following formulas.

For the Chebyshev smoother, inverting formulas (5), we find a new bound $\lambda_{\text{min}}^c$:

$$\lambda_{\text{min}}^c = \left(\frac{\rho_1 - 1}{\rho_1 + 1}\right)^2 \cdot \lambda_{\text{max}}^c, \quad \rho_1 = \left(\delta^{-1} + \sqrt{\delta^{-2} - 1}\right)^{1/p}. \quad (14)$$

At next smoothing step, we use this new value $\lambda_{\text{min}}^c$ to construct the smoothing process according to the formulas (5).

For the LI-M smoother, according to the inequality (13), we can assume that the sought bound $\lambda_{\text{min}}^c$ corresponds to the largest eigenvalue of the operator $(I + \tau A_h)^{-1}$. Then $\delta \approx (1 + \tau \lambda_{\text{min}}^c)^{-1}$ and we can find a new left bound of the high-frequency spectrum corresponding to a decrease in the residual in $\delta^{-1}$ times:

$$\lambda_{\text{min}}^c = \frac{\pi^2}{16 p^2} \left(\delta^{-1} - 1\right) \lambda_{\text{max}}. \quad (15)$$

It is easy to verify that this new $\lambda_{\text{min}}^c$ leads to increasing the degree $p$:

$$p_{\text{new}} = p \left[(\varepsilon^{-1} - 1)/(\delta^{-1} - 1) + 1\right]^{1/2}. \quad (16)$$
After computing the new parameters ($\lambda \text{new}$ or $p = p_{\text{new}}$) at all the grid levels we continue the multigrid iterations with these new values, repeating if necessary (i.e. if $\varepsilon < \delta < 1$) adaptation procedure. If $\delta > 1$, then we expand the smoothing interval taking $\lambda \text{new} := 0.25\lambda \text{min}$.

Receiving the refined value, we continue multigrid iterations with this new value, repeating the adaptation procedure. As a rule, in calculations after several (3 - 4) multigrid iterations, the smoother parameters are stabilized and the specified smoothing tolerance $\varepsilon = \varepsilon_{\text{smooth}}$ is achieved, see [4, 5], and the asymptotic multigrid convergence rate becomes $\varepsilon^2$. This technique allows us to achieve automatically the prescribed convergence rate of the multigrid solver. The computations confirm that the adaptive technique is a useful tool in designing efficient multigrid algorithms and provides robustness of the multigrid method even in the presence of a strong anisotropy in the problems, see [4, 5].

To demonstrate the efficiency of the adaptive technique, we present results of solution of a strong anisotropic model problem. We consider the Poisson equation with pure Neumann boundary conditions in the anisotropic rectangular parallelepiped with edge size $10000 \times 100 \times 1$ on computational grids with $N = 128^3, 256^3, 512^3, 1024^3,$ and $2048^3$ nodes. The results shown in table 1 are obtained on the 64 processor nodes of the K100 computer of Keldysh Institute in the cubic processor topology. Input data for all grids: 5 grid levels, the relative tolerance $\varepsilon_{\text{tot}} = 10^{-7}$, the relative tolerance for the coarsest level $\varepsilon_{\text{tol}} = 10^{-5}$, smoothing tolerance $\varepsilon_{\text{smooth}} = 0.5$. The spectrum splitting boundary is set by the parameter $\eta = \lambda \text{min}/\lambda \text{max}$. In this problem we know an almost accurate estimate $\eta_{\text{exact}} = 5.10^{-5}$. In calculation without adaptation we fix this value for all smoothing stages. In calculation with adaptation for all smoothing stages we take as the initial value $\eta = 1/6$. This value is accurate estimate for isotropic problem but in these computations without adaptation the usage of the value $\eta = 1/6$ leads to stagnation of the multigrid iterations. It follows from table 1 that adaptation reduces the calculation time several times even in comparison with the exact setting of the spectrum boundary. This is explained by the fact that firstly in the multigrid iterations the smoothing overhead is less than when the iterations reach the asymptotic mode. The LI-M adaptive smoother provides similar results in this problem.

| Grid size | No adaptation | Adaptation |
|-----------|---------------|------------|
| $128^3$   | 2.3           | 1          |
| $256^3$   | 23            | 8          |
| $512^3$   | 191           | 42         |
| $1024^3$  | 1500          | 356        |
| $2048^3$  | 14190         | 3517       |

9. Multigrid method on conformal block-structured Cartesian grids

The multiblock techniques have been developed for decades to generate grids with high smoothness, orthogonality and sufficient resolution. To obtain flexibility as well as efficient computation, we use hexahedral multiblock grids which are widely used in many application, see for instance [11]. We assume that a domain decomposition is made with hexahedral multiblock grid. This grid is based on point-to-point matching across the interface of neighboring blocks. In each block the discretization scheme is constructed on the fine level grid by the finite-volume method which leads to the 7-pt or 27-pt stencils in each interior node (in 3D case). The coarse level equations are formed by the Galerkin procedure, see (3). The presence of the irregular connections of the blocks leads to irregular stencils in the nodes which adjoin a number of
several blocks. For application of the multigrid such irregular nodes create certain difficulties, to overcome which instead irregular node treatment we can obtain approximation in these nodes in an additive manner. Such an opportunity is provided by the use of the explicit iterative procedures for the smoothers and the coarsest grid solver. Each explicit iteration is computed separately in each block, and then the solution on the interfaces is adjusted with solution data exchange. This ensures the result exactly corresponds to the work of the explicit iteration on the global block-structured grid. This means that a decomposition of a single domain into several blocks does not change the approximation of even in the vicinity of an interface. The proposed multigrid algorithm for block-structured grids with the explicit iterations, enhances the computational potential of the multigrid method in the development of ultra-parallel computing.

We explain the main features of the proposed method using a simple example: a domain $R$ is divided into three subdomains $R_1, R_2, R_3$ (see fig. 1).

![Figure 1. An example of a 2D block-structured grid](image)

The sought function $u$ is defined in each subdomain $R_i$ on the grid $\tilde{\omega}_i$. The internal nodes of each grid $\omega_i$ are denoted as $\omega_i$. Multigrid nested meshes $\tilde{\omega}_i^l$ are generated for each subdomain $R_i$ by assigning on the original mesh $\omega_i$ some sets of grid lines in each of the two directions, here the superscript $l$ in the notation $\tilde{\omega}_i^l$ means a number of a nested grid, $l = 1, 2, \ldots, N_c$, and $N_c$ is amount of grid levels. We implement the multigrid operators not only at the inner nodes of each subdomain but also on the interfaces $EB, ED, EF$. On the external boundaries we assume the Dirichlet boundary conditions. The entries of the discrete finite volume system at each grid node can be interpreted as a result of summing the coefficients over the cells, the common vertex of which is a given node. If it is an internal node for a subdomain, then the summation is performed over four adjacent cells and leads to the standard 9-pt stencil. For an interface node belonging to $\tilde{\omega}_i^l$ and lying the interfaces $EB, ED, EF$ we take into account only the cells of the grid $\tilde{\omega}_i^l$.

It is important that the discrete stencils of the multigrid operators (restriction, prolongation, smoothers, coarsest grid solver) in each subdomain $R_i$ consist only of their own grid nodes $\tilde{\omega}_i^l$. 
These are formal templates which do not provide an approximation at the interface nodes. After calculating by the procedure presented below we obtain that the real stencil at the interfaces includes nodes of all regions adjacent to it, and provides the necessary approximation.

We explain how the multiblock multigrid algorithm works using our smoother as the example. On the smoothing stage for the entire domain $R$ on each iteration (which is the explicit iteration), we calculate the function $u$ in each block $R_i$ independently. Therefore two values $u$ are assigned to each node belonging to one of the interfaces $EB$, $ED$, $EF$, and three values $u$ are assigned to the vertex $E$ in this particular example (fig. 1). We call such node as a multiple node. The interface nodes have the multiplicity $q = 2$, the multiplicity of the inner vertices can be arbitrary, in this case the multiplicity of the vertex $E$ is $q = 3$. At multiple nodes of the region $R$, after each iteration we recalculation of values as follows. Let the multiplicity of a node be $q$, and $u_i$, $i = 1, 2, ..., q$ are the values obtained at a current iteration. For the grid $\omega_l$ we denote by $s_i$ the volume of the cell associated with the considered node. We define:

$$u = \frac{1}{q} \sum_{i=1}^{q} s_i u_i,$$

$$s = \sum_{i=1}^{q} s_i.$$

We correct the iterative approximation of $u_i$, $i = 1, 2, ..., q$ taking $u_i = u$. The total result is algebraically identical to the calculation the finite-volume approximation written on irregular stencils. Note that for each node with $q = 2, 3, 4, 5, 6$ the real stencil includes 9, 7, 9, 11, 13 nodes respectively. The rest of the multigrid operators ($I$, $I^*$ and the coarsest grid solver) are implemented in a similar way without any special efforts. Thus, instead of the direct treatment of the irregular grid nodes we propose a simple computational procedure convenient for both data description and for parallel implementation of the multigrid.

10. Conclusion

In this paper we reviewed practical parallel multigrid technique which is appropriate for implementation on conformal block-structured grids with potential scalability on ultra-parallel computers with large amount of processors. We proposed an adaptive technique based on analysis of convergence history which provides efficient design of smoothers with optimal smoothing factors. This technique allows us to achieve automatically the prescribed convergence rate of the multigrid solver. Here this technique is applied for block-structured Cartesian grids but we believe that adaptive approach will be useful for unstructured grids and for algebraic multigrid as well. The field of application of the adaptive multigrid method is ultra-parallel technologies, multiple solution of elliptic equations with different right-hand sides, which is typical for a multigrid method, preconditioning the Krylov subspace solver by Chebyshev iterations, solving implicit schemes for nonstationary equations with coefficients that are independent or weakly dependent on time.

References

[1] Fedorenko R P 1962 A relaxation method for solving elliptic difference equations Comput. Math. Math. Phys. 1 (4) pp 1092–1096
[2] Zhukov V T, Novikova N D and Feodoritova O B 2014 Parallel multigrid method for solving elliptic equations Math. Models and Computer Simulations 6 (4) pp 425–434
[3] Zhukov V T, Novikova N D and Feodoritova O B 2015 Multigrid method for elliptic equations with anisotropic discontinuous coefficients Comput. Math. Math. Phys. 55 (7) pp 1150–1163
[4] Zhukov V T, Novikova N D and Feodoritova O B 2015 On the Solution of Evolution Equations Based on Multigrid and Explicit Iterative Methods Comput. Math. Math. Phys. 55(8) pp 1276–1289
[5] Zhukov V T, Novikova N D and Feodoritova O B 2015 Multigrid method for anisotropic diffusion equations based on adaptive Chebyshev smoothers Math. Models and Computer Simulations 7 (2) pp 117–127
[6] Brannick J, Hu X, Rodrigo C and Zikatanov L 2015 Local Fourier Analysis of Multigrid Methods with Polynomial Smoothers and Aggressive coarsening Numer. Mathematics: Theory, Methods and Applications 8 (1) pp 1–21
[7] Baker A H, Falgout R D, Kolev T V U. and Yang M 2011 Multigrid smoothers for ultraparallel computing
SIAM J. on Scientific Computing 33 (5) pp 2864–2887
[8] Vanek P and Brezina M 2013 Nearly optimal convergence result for multigrid with aggressive coarsening and
polynomial smoothing Applications of Mathematics 58 (4) pp 369–388
[9] Samarskii A N and Nikolayev E S 1989 Solution Methods for Grid Equations Vol 1 Direct Methods, Vol 2
Iterative Methods. Birkhauser Verlag, Basel Boston Berlin
[10] Suetin P K Classical orthogonal polynomials 1978 Nauka Moscow (In Russian)
[11] Epstein B, Rubin T and Seror 2003 Accurate Multiblock Navier-Stokes Solver for Complex Aerodynamic
Configurations AIAA Journal 41 (4) pp 582–594