SIAM J. SCI. COMPUT.
Vol. 29, No. 4, pp. 1613–1636

MULTIGRID FOR HIGH-DIMENSIONAL ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS ON NON-EQUIDISTANT GRIDS*

H. BIN ZUBAIR†, C. W. OOSTERLEE‡, AND R. WIENANDS§

Abstract. This work presents techniques, theory, and numbers for multigrid in a general d-dimensional setting. The main focus of this paper is the multigrid convergence for high-dimensional partial differential equations on non-equidistant grids such as may be encountered in a sparse-grid solution. As a model problem we have chosen the anisotropic stationary diffusion equation on a rectangular hypercube. We present some techniques for building the general d-dimensional adaptations of the multigrid components and propose grid-coarsening strategies to handle anisotropies that are induced due to discretization on a non-equidistant grid. Apart from the practical formulas and techniques, we present—in detail—the smoothing analysis of the point ω-red-black Jacobi method for a general multidimensional case. We show how relaxation parameters may be evaluated efficiently and used for better convergence. This analysis incorporates *full* and *partial doubling* and *quadrupling* coarsening strategies as well as the second- and the fourth-order finite difference operators. Finally we present some results derived from numerical experiments based on the test problem.

Key words. multigrid, high-dimensional PDEs, anisotropic diffusion equation, coarsening strategies, point-smoothing methods, relaxation parameters, Fourier-smoothing analysis

AMS subject classifications. 65N55, 65F10, 65Y20

DOI. 10.1137/060665695

1. Introduction.

Multidimensional partial differential equations have diverse applications in various fields of applied sciences, including financial engineering [8], molecular biology [3], and quantum dynamics [1, 21]. There are quite a few fast and efficient solution techniques for partial differential equations (henceforth PDEs) of which multigrid ranks among the best. Multigrid is a well-known iterative procedure for the solution of large and sparse linear systems that arise from various kinds of PDE discretizations. The existing literature on the multigrid treatment of various problems, however, rarely explores issues that arise out of growth in the dimensionality of the problem. The implications of dimensionality growth include deterioration of the multigrid convergence rate, impractical storage requirements, and huge amounts of CPU time for single grid solution methods. Our main emphasis in this paper lies on the first challenge. We abbreviate *multigrid for d-dimensional PDEs* as *d-multigrid*. In this paper d represents both *abstract dimensionality* and *dimensions*. So, e.g., $3d$ is to be interpreted as 3-dimensional.

The major hinderance in the numerical solution of multidimensional PDEs is the so-called *curse of dimensionality*, which implies that with the growth in dimensions we have an exponential growth in the number of grid points. This increases the

*Received by the editors July 21, 2006; accepted for publication (in revised form) March 27, 2007; published electronically July 18, 2007. This research has been partially supported by the Dutch government through the national program BSIK: knowledge and research capacity, in the ICT project BRICKS (http://www.bsik-bricks.nl), theme MSV1, and partially by the Government of Pakistan through The HEC-Pakistan's research grant, contract Ref: 1-3/PM-OVER/Neth/SPMU/2004.

http://www.siam.org/journals/sisc/29-4/66569.html

†Delft University of Technology, Faculty EEMCS, Delft Institute of Applied Mathematics, 07.030, Mekelweg 4, 2628 CD, Delft, The Netherlands (h.binzubair@tudelft.nl).

‡CWI—Center for Mathematics and Computer Science, Amsterdam, The Netherlands (c.w.oosterlee@cwi.nl).

§Mathematical Institute, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany (wienands@math.uni-koeln.de).

computational cost of many good algorithms. Although we do not address this issue in particular, we would like to stress that a way to handle dimensionality is the sparse-grid method [9, 21]. One of the characteristics of these *sparse grids* is that these grids are essentially non-equidistant, and therefore efficient solution methods for this type of grid are quite important.

A multigrid treatment of high-dimensional PDEs based on hyperplane relaxation has been proposed by Reisinger and Wittum in [8]. We present a multigrid treatment based on point relaxation and partial coarsening schemes. We demonstrate how the multigrid convergence factor can be brought down for higher d by the suggested grid-coarsening strategies and a proper choice of the relaxation parameters in the smoothing process.

The strategy that we suggest in this paper for good multigrid convergence is that, keeping the point-smoothing method, we coarsen the grid simultaneously along all of the dimensions where the errors are strongly coupled. We call this strategy *simultaneous partial coarsening*. This way we relax the anisotropy at each successive grid level, until the problem is isotropic to the point where full coarsening is feasible. Full coarsening from this stage onwards brings the grid to the coarsest possible level, where we solve exactly. These strategies therefore consist of two main phases: the *partial coarsening phase* and the *full coarsening phase*. We show that multigrid based on *quadrupling* ($h \to 4h$) transfers during the partial coarsening phase of the scheme gives a multigrid speed boost if optimal relaxation parameters are used in the point-smoothing process. We would like to point out that partial coarsening schemes have also been put forward by Larsson, Lien, and Yee identifying the conditions for partial coarsening through local Fourier-smoothing analysis (LFA) [7]. Moreover, the utility of adaptive grid-coarsening in multigrid has also been demonstrated by Elias, Stubley, and Raithby and Horton and Vandewalle in [4] and [5], respectively.

Foremost (in section 2) we point out that the Black–Scholes pricing problem PDE can be reduced to a standard d-dimensional heat equation, indicating the need for fast solution methods for high-dimensional PDEs of Poisson type. The discretization and subsequent implementation of a d-dimensional PDE is somewhat involved, and in section 3 we show how this can be done with Kronecker products. Section 4 deals with d-multigrid, point-smoothing, coarsening strategies and concludes with a computational complexity analysis. Next we provide the LFA for the ω-RB Jacobi method and show how it can be extended to d dimensions. This is section 5. We also point out here how we incorporate partial doubling and quadrupling in this analysis. This section concludes with a tabular presentation (up to $6d$) of some optimal relaxation parameters. Finally in the last section we present quite a few versatile numerical experiments and demonstrate the excellent multigrid convergence that we get.

2. Multi-d equations in finance. The application on which we focus in this section is the pricing of multiasset options through a Black–Scholes model which is a high-dimensional parabolic PDE, reducible to the heat equation.

A generalized d-asset Black–Scholes equation reads

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i\sigma_j S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} + r\sum_{i=1}^{d} S_i \frac{\partial V}{\partial S_i} - rV = 0, \tag{2.1}$$

$$(0 < S_1, \ldots, S_d < \infty, \quad 0 \leqslant t < T).$$

V stands for the option price; S_i are the d underlying asset prices; t is the current time; ρ_{ij} are the correlation coefficients between the Wiener processes modeling the

movement of the price of the ith and the jth assets; σ_i is the volatility of the ith asset price; r is the risk-free interest rate. For pricing options on a basket of d assets, it has been shown [6, 20] that (2.1) under the following transform of the asset price S_i:

$$y_i = \frac{1}{\sigma_i}\left(r - \frac{{\sigma_i}^2}{2}\right)\tau + \frac{1}{\sigma_i}\ln S_i, \quad i = 1, 2, \ldots, d, \tag{2.2}$$
$$\tau = T - t$$

(T is a constant and represents the maturity time of the option) yields the equation

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sum_{i=1}^{d}\frac{\partial^2 V}{\partial {x_i}^2}, \qquad -\infty < x_i < \infty, \quad 0 < \tau \leqslant T, \tag{2.3}$$

which we identify as the heat-conduction equation in d dimensions. When discretized by the implicit Crank–Nicolson scheme, we get a discrete s.p.d. operator. It is worthwhile to mention that the time discretization adds to the *positivity* of the main diagonal of the discrete operator, and so the stationary diffusion equation serves well as a limiting *worst* case, in the study of convergence behavior.

3. The discretization. We first recall that discrete operators can be implemented in two different ways. One is the stencil method, and the other is the matrix method. The stencil method saves storage but is inherently difficult to implement due to the visual constraints—imposed by high dimensionality—on the problem. Therefore, to circumvent the complicated implementation issues we use matrices (in sparse storage formats), and here we present some matrix generation formulas based on Kronecker tensor products.

3.1. The continuous problem and its discretization scheme. For analysis and experimentation we choose the d-dimensional stationary diffusion equation, with Dirichlet boundary conditions, to serve as our model problem. In what follows $\mathbf{x}$ is a d-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, and $\{a_i, b_i, \varepsilon_i\} \in \mathbb{R}$, with $\varepsilon_i > 0$. The continuous problem reads:

$$-Lu(\mathbf{x}) = -\sum_{i=1}^{d}\varepsilon_i\frac{\partial}{\partial x_i}u(\mathbf{x}) = f^{\Omega}(\mathbf{x}), \quad \mathbf{x} \in \Omega = \prod_{i=1}^{d}(a_i, b_i) \subset \mathbb{R}^d, \tag{3.1}$$
$$u(\mathbf{x}) = f^{\Gamma}(\mathbf{x}), \qquad \mathbf{x} \in \Gamma = \partial\Omega \quad (x_i \in \{a_i, b_i\}).$$

Subsequently, the discrete counterpart reads

$$-L_h u_h(\mathbf{x}) = f_h^{\Omega}(\mathbf{x}), \quad \mathbf{x} \in \Omega_h = \prod_{i=1}^{d}(a_i, b_i) \subset \mathbb{R}^d, \tag{3.2}$$
$$u_h(\mathbf{x}) = f_h^{\Gamma}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_h = \partial\Omega_h \quad (x_i \in \{a_i, b_i\}).$$

The discretization of the Laplacian L_h is chosen to be either $O(h^2)$ accurate, giving a $(2d + 1)$-point stencil, or else $O(h^4)$ accurate, with a $(4d + 1)$-point long stencil for all *interior* points. The number of cells in the discretization grid along the ith dimension—represented by N_i—*need not* be equal to the number of cells along (say)

the jth dimension. So with $h_i = (b_i - a_i)/N_i$—the mesh size along the ith dimension—the $1d$ variants of these multidimensional stencils are

$$\left(\frac{\partial^2}{\partial x_i^2}\right)_h \triangleq \frac{1}{h_i^2}[1 \quad -\underline{2} \quad 1] + O(h^2), \tag{3.3}$$

$$\left(\frac{\partial^2}{\partial x_i^2}\right)_h \triangleq \frac{1}{12h_i^2}[-1 \quad 16 \quad -\underline{30} \quad 16 \quad -1] + O(h^4). \tag{3.4}$$

It has to be noted that the $O(h^4)$ long stencil uses the so-called *ghost points* (points outside the discretization grid) when applied to points *near* the boundary. To alleviate this problem we have the option of employing a different stencil, having shorter connections at the boundary. Thus we can either employ the simple $O(h^2)$ operator at the boundaries or else use a different scheme with one-sided differencing. In this work, we use the second-order stencil, for points near the boundaries. The discretization given by (3.2) leads to the following matrix equation:

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{b}_h. \tag{3.5}$$

3.2. The discretization matrix $\mathbf{A_h}$ and implementation in arbitrary dimensions. As we have Dirichlet boundary conditions, we eliminate boundary points from the matrix $\mathbf{A}_h$. This scheme results in a total of M unknowns—$(M \times M)$ being the order of the discretization matrix $\mathbf{A_h}$—with

$$M = \prod_{i=1}^{d} (N_i - 1).$$

We represent the discretization grid $\mathbf{G}$ by

$$\mathbf{G} = [N_1, N_2, \ldots, N_d]. \tag{3.6}$$

The discrete matrix $\mathbf{A}_h$ in (3.5) can be built by the following tensor product formula:

$$\mathbf{A}_h = \sum_{i=1}^{d} \left\{ \bigotimes_{j=i}^{d-1} \mathbf{I}_{(d+i-j)} \otimes \mathbf{L}_i \otimes \bigotimes_{j=1}^{i-1} \mathbf{I}_{(i-j)} \right\}. \tag{3.7}$$

$\otimes$ is the Kronecker tensor product of matrices. Likewise $\bigotimes$ is the cumulative Kronecker tensor product. For example,

$$\bigotimes_{i=1}^{3} \mathbf{P}_i = \mathbf{P}_1 \otimes \mathbf{P}_2 \otimes \mathbf{P}_3.$$

Kronecker tensor products are noncommutative and associative operations (see [10]). The order is determined by the subscripts here, and the associative hierarchy does not matter.

In (3.7), $\mathbf{I}_k$ ($k \in \{1, 2, \ldots, d\}$) is the identity matrix of order $(N_k - 1)$, and $\mathbf{L}_i$ is the one-dimensional discrete-Laplacian matrix, constructed through (3.3) and (3.4) as illustrated by the following example. Suppose that $\mathbf{G} = [8, 6]$ (see (3.6)) is the grid that we have for a certain $2d$ problem; then we construct $\mathbf{L}_i$ by writing down the discrete stencil in (3.4) for each point, including the boundaries. Then we isolate the left and the right boundary vectors (as shown below) and incorporate them in the

right-hand side $\mathbf{b}_h$. For example, $\mathbf{L}_1$ is the following matrix (without the left and the right boundary vectors) according to the choice $O(h^4)$ of the computational accuracy:

$$\frac{\varepsilon_1}{12h_1^2}\begin{bmatrix} & & & & & & & & \\ 12 & -24 & 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 16 & -30 & 16 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 16 & -30 & 16 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 16 & -30 & 16 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 16 & -30 & 16 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 16 & -30 & 16 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & -24 & 12 \\ & \multicolumn{8}{c}{\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\mathbf{L}_1}} & \end{bmatrix}.$$

$1d$ discrete Laplacian matrices constructed in this way for each grid dimension are to be substituted in (3.7) for building up the discrete d-dimensional operator matrix $\mathbf{A}_h$. (The $O(h^2)$ discretization is handled similarly.)

3.3. The right-hand side $\mathbf{b}_h$. The right-hand side $\mathbf{b}_h$ of (3.5) consists of the source function f_h^Ω and the boundary function f_h^Γ.

It is important to define a consistent *grid-point enumeration* in high dimensions. In our enumeration scheme we represent the entire set of indices by an *index matrix* $\mathcal{I}$, where each row (a d-tuple) represents the index of a grid point and counts in descending order, i.e., from right to left, for the ascending order of the dimensions. This formulation of the index set is the natural extension of the so-called *lexicographic* order in $2d$. As an illustration of this lexicographic order, we have grid coordinates as a pair (i_2, i_1) in $2d$ and as a triple (i_3, i_2, i_1) in $3d$, and thus, generally in d-dimensions, we have the index of a grid point as $(i_d, i_{(d-1)}, \ldots, i_1)$; piling them up in lexicographic order, we get the entire set as

$$\mathcal{I} = [\iota_d \quad \iota_{(d-1)} \ldots \iota_i \ldots \iota_2 \quad \iota_1], \tag{3.8}$$

where each ι_k is a column vector of length M. Also consider the following definitions, which we require to build the index set for the interior and the boundary points:

$$\begin{aligned} &\eta_i; \quad \ni \quad \eta_i = [1, 2, \ldots, (N_i - 1)]^T \quad (i = 1, 2, \ldots, d) \quad \text{(see (3.6))}, \\ &\mathbf{1}_i; \quad \ni \quad \mathbf{1}_i = \underbrace{[1, 1, \ldots, 1]}_{\text{Total } N_i}{}^T. \end{aligned} \tag{3.9}$$

We now formulate the columns of $\mathcal{I}$ as follows:

$$\iota_i = \bigotimes_{j=i}^{d-1} \mathbf{1}_{(d+i-j)} \otimes \eta_i \otimes \bigotimes_{j=1}^{i-1} \mathbf{1}_{(i-j)}. \tag{3.10}$$

At this stage the vector of source function values can be computed as the development of the index set $\mathcal{I}$ is complete. Thus, computing the source function for each row of $\mathcal{I}$ and denoting it by $\mathcal{S}$, we have $\mathcal{S} = f_{\mathcal{I}}^\Omega$.

Now for computing the contribution of boundaries in $\mathbf{b}_h$, recall that we isolated two column vectors, namely, the left and the right boundary-coefficient vectors from the $1d$ Laplacian operators in each dimension. Considering the case of the ith dimension, if we denote these by l_i and r_i, respectively, then we can define the ith

d-dimensional left and right boundary-coefficient vectors, viz., $\mathcal{L}_i$ and $\mathcal{R}_i$ as follows:

$$\mathcal{L}_i = \bigotimes_{j=i}^{d-1} \mathbf{1}_{(d+i-j)} \otimes l_i \otimes \bigotimes_{j=1}^{i-1} \mathbf{1}_{(i-j)}, \tag{3.11}$$
$$\mathcal{R}_i = \bigotimes_{j=i}^{d-1} \mathbf{1}_{(d+i-j)} \otimes r_i \otimes \bigotimes_{j=1}^{i-1} \mathbf{1}_{(i-j)}.$$

The contribution of the boundary values in $\mathbf{b}_h$ has two parts, i.e., values from the left boundary and values from the right boundary. We denote the two by $\mathcal{B}_L$ and $\mathcal{B}_R$, respectively. $\mathcal{B}_L$ is the cumulative sum of the d left boundaries and likewise for the right. At this point a word about the *boundary-index set* is in order. If in (3.8) any ι_i is replaced by a vector of *left boundary value*, we get a left boundary index set, and if we replace it by a vector of *right boundary value*, we get a right boundary index set. If

$$\mathcal{I}_{L_i} = [\iota_d \quad \iota_{(d-1)} \dots \iota_(i-1) \quad \overline{a_i} \quad \iota_(i+1) \dots \iota_2 \quad \iota_1],$$
$$\mathcal{I}_{R_i} = [\iota_d \quad \iota_{(d-1)} \dots \iota_(i-1) \quad \overline{b_i} \quad \iota_(i+1) \dots \iota_2 \quad \iota_1],$$

then

$$\mathcal{B}_L = \sum_{i=1}^{d} (\mathcal{L}_i \diamond f^{\Gamma}_{\mathcal{I}_{L_i}}), \tag{3.12}$$
$$\mathcal{B}_R = \sum_{i=1}^{d} (\mathcal{R}_i \diamond f^{\Gamma}_{\mathcal{I}_{R_i}}).$$

$\diamond$ represents componentwise multiplication of the operand column vectors.

Thus we have from the right-hand side of (3.5)

$$\mathbf{b}_h = \mathcal{S} + \mathcal{B}_L + \mathcal{B}_R, \tag{3.13}$$

and the discretization is complete.

4. d-multigrid based on point smoothing. The core of this work is the coarsening strategies proposed here, which are based on a mixture of doubling ($h \to 2h$) and quadrupling ($h \to 4h$) grid transfers and which—coupled with point-based relaxation—yield very efficient multigrid methods for problems on non-equidistant grids. The aim is to close upon a robust method that applies for general grid-aligned anisotropies in d dimensions.

Like basic multigrid for two- and three-dimensional problems, d-multigrid also consists of the essential components, the smoothing method and the coarse grid correction. The well-known algorithm of multigrid as presented in [14] does not change for the higher-dimensional case; however, the components have to be generalized to match this new situation. General multigrid algorithms are presented in the literature [11, 12, 13, 14, 15, 17].

For anisotropic problems it is a choice to keep the point-smoothing method and to coarsen only along a subset of the dimensions, precisely, along those that are strongly coupled. This ensures that coarsening takes place only where the errors are smooth. For *nearly* isotropic problems the best strategy is to combine the point-smoothing method with doubling-based full coarsening and to use the optimal relaxation parameters obtained for d dimensions.

4.1. The relaxation method. Of the many available point-smoothing-based relaxation methods, we choose the ω-red-black Jacobi method due to its excellent smoothing effect for problems of the Poisson type. The red-black Jacobi method is equivalent to the Gauss–Seidel red-black method for the $O(h^2)$ $(2d+1)$-point discretization stencil and in that it is commonly abbreviated as GS-RB in the literature. In this section we assume familiarity with standard GS-RB for $2d$ problems; see [11, 14]. ω-RB Jacobi consists of two partial steps, each an ω-Jacobi sweep; the first one *applying to and updating* only the *red* (odd) points and the second one applying to and updating only the *black* (even) points in the grid.

From an implementational point of view, this red-black smoothing procedure which is based on partial steps depends upon a partitioning process by which the grid $\boldsymbol{G}$ can be dissected into the *red* part ($\boldsymbol{G}_R$) and the *black* part ($\boldsymbol{G}_B$). The grid-point enumeration that we employ in our implementation scheme is such that the points are arranged linearly (in a column vector) and counted out in lexicographical order for a d-dimensional grid. The unequal number of cells along different dimensions of the grid makes this partitioning process somewhat nontrivial. In Appendix A we present a way to bring about this segregation of odd and even points from a purely implementational aspect.

We also employ optimal relaxation parameters ω_{opt} in the relaxation process. It is well known that $\omega = 1$ serves as a good choice for the $2d$ isotropic case. In the case of anisotropy and higher dimensions the error-smoothing effect of the relaxation method can be enhanced by the use of optimal relaxation parameters [19]. This implies that a search for ω_{opt} pays off. We employ a d-dimensional LFA for this purpose; see section 5.

4.2. Coarsening strategies to handle anisotropies. We present two *grid-adaptive* coarsening procedures as our test cases, both of which have shown excellent convergence results. We call them *strategy 1* and *strategy 2*. Both of the strategies are similar in the sense that they consist of two distinct coarsening phases: the partial coarsening phase and the full coarsening phase. The difference is in the partial coarsening phase, where the *transfer type* in strategy 1 is doubling (h $\rightarrow$ 2h) and in strategy 2 is quadrupling (h $\rightarrow$ 4h). In the full coarsening phase both of the strategies are identical and consist of doubling transfers only.

In strategy 1 we first identify the dimension(s) having the strongest coupling. This is indicated by the magnitude of the *coupling factor* $\tilde{c}_i$ defined as

$$\tilde{c}_i = \varepsilon_i \times \left(\frac{N_i}{b_i - a_i} \right)^2 ;$$

see (3.1) and (3.6). The larger the coupling factor $\tilde{c}_i$, the stronger the coupling. Fourier analysis suggests that all dimensions having a coupling factor $\tilde{c}_i$ within a range of 1.3 times the largest coupling factor identified can be doubled simultaneously. This decision is made (at each successive grid level) and identifies all of those dimensions along which doubling will take place. Employing this strategy recursively makes the discrete problem isotropic inasmuch as all coupling factors are within this range; onwards from here full doubling takes over. Once the coarsest possible grid is reached, an exact solution takes place. In the last section we evaluate this strategy for the $O(h^2)$ $(2d+1)$-point stencil and for the $O(h^4)$ $(4d+1)$-point long stencil.

In strategy 2 the threshold value is 1.0, which means that we quadruple only along the dimension(s) which are identified as having the strongest coupling (having the largest $\tilde{c}_i$). This ensures that quadrupling (in comparison with doubling) takes

place along fewer dimensions. This strategy gives good convergence when employed in conjunction with optimal relaxation parameters and is cheaper than strategy 1 because of quadrupling in the partial phase. Moreover, we suggest that a strategy based on quadrupling in the full coarsening phase should not be employed in a general multidimensional case, as full quadrupling always loses against full doubling and hence is quite apt to hamper multigrid convergence. See, for an example, section 6, Figure 3.

We take the grid size along each dimension always as a power of 2. When the anisotropy stems only from discretization on non-equidistant grids, as encountered in the sparse-grid solution, the sequence of coarse grids generated by the two strategies are as follows.

Example 1. Suppose that a particular discretization grid for a certain $5d$ problem is $\mathbf{G} = [\,32\ \ 8\ \ 8\ \ 128\ \ 32\,]$. $\varepsilon_i = 1\ \forall\ \ i$ and $\Omega = (0,1)^5$. Then the sequence of grids that we get is the following:

(4.1)

Strategy 1	*Strategy* 2
$\Omega_6 = [\ 32\ \ 8\ \ 8\ \ 128\ \ 32\ ]$	
$\Omega_5 = [\ 32\ \ 8\ \ 8\ \ 64\ \ 32\ ]$	$\Omega_4 = [\ 32\ \ 8\ \ 8\ \ 128\ \ 32\ ]$
$\Omega_4 = [\ 32\ \ 8\ \ 8\ \ 32\ \ 32\ ]$	$\Omega_3 = [\ 32\ \ 8\ \ 8\ \ 32\ \ 32\ ]$
$\Omega_3 = [\ 16\ \ 8\ \ 8\ \ 16\ \ 16\ ]$	$\Omega_2 = [\ 8\ \ 8\ \ 8\ \ 8\ \ 8\ ]$
$\Omega_2 = [\ 8\ \ 8\ \ 8\ \ 8\ \ 8\ ]$	$\Omega_1 = [\ 4\ \ 4\ \ 4\ \ 4\ \ 4\ ]$
$\Omega_1 = [\ 4\ \ 4\ \ 4\ \ 4\ \ 4\ ]$	$\Omega_0 = [\ 2\ \ 2\ \ 2\ \ 2\ \ 2\ ]$
$\Omega_0 = [\ 2\ \ 2\ \ 2\ \ 2\ \ 2\ ]$	

Results for this particular experiment are available in section 6, Figure 4.

4.3. Coarse-grid discretization. An important component in the coarse-grid correction process is the choice of the coarse-grid operator L_H. In this paper we use the coarse-grid analog of the discrete operator on the fine grid. Once the next coarser grid is decided, we discretize the Laplacian using the same discrete stencils as presented in section 3.

A particularly good choice of the coarse-grid operator for the $O(h^4)$ accuracy is to employ the $O(h^4)$ long stencil only along the noncoarsened dimensions of the grid and to discretize with the $O(h^2)$ stencil on the coarse grids along the dimensions where partial coarsening takes place. This has the marked advantage of saving CPU time as now the coarse-grid operator has increased sparsity. Moreover, on very coarse grids this is advantageous, because at grid points adjacent to boundary points the long stencil cannot be applied since it has entries which lie outside the discrete domain, whereas the $O(h^2)$ discretization can be applied throughout the domain. The overall accuracy remains fourth-order as we have the fourth-order accuracy on the finest grid. This coarse-grid discretization scheme fits very nicely with the numerical experiments, and 2-*grid and* 3-*grid analysis* (not shown here) confirm this.

We do not use the Galerkin operator because of its disadvantage of being usually more dense than the simple coarse-grid analog of the fine-grid operator (unless special transfer operators are employed to generate the coarse-grid operators). In high-d dimensions this disadvantage becomes more serious and impractical.

4.4. The transfer operators. We employ the d-dimensional analogs of the full-weighting (FW) restriction operator and of the bilinear interpolation operator in two dimensions for the intergrid transfers of the grid functions. In this section we present a tensor formulation to generate the restriction and prolongation operator matrices.

For completeness we first mention [14] that a $2d$ FW restriction operator

$$I_h^{2h} \triangleq \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}_h^{2h}$$

is the Kronecker tensor product of the following x_1 and x_2 directional 1-dimensional FW operators:

$$(I_h^{2h})_{x_1} \triangleq \frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix}, \quad (I_h^{2h})_{x_2} \triangleq \frac{1}{4}\begin{bmatrix}1\\ 2\\ 1\end{bmatrix}.$$

A formula based on Kronecker tensor products for building up a FW restriction operator matrix $\mathbf{R}$ reads

$$\mathbf{R} = \prod_{i=1}^{d} (\mathbf{R}_i)^{k_i}, \tag{4.2}$$

$$(\mathbf{R}_i)^{k_i} = \prod_{l=0}^{k_i-1}\left[\bigotimes_{j=i}^{d-1}\mathbf{I}_{N_{(d+i-j)}} \otimes \mathbf{O}_{\left[N_i/2^{(k_i-l-1)}\right]} \otimes \bigotimes_{j=1}^{i-1}\mathbf{I}_{\left[N_{(i-j)}/2^{k_{(i-j)}}\right]}\right].$$

We now define the quantities involved in (4.2) for the dummy subscript a.

$\mathbf{I}_a$ is the identity matrix of order $(a-1)\times(a-1)$.

$\mathbf{O}_a$ is the $1d$ FW restriction operator matrix, order $= \left(\frac{a}{2}-1\right)\times(a-1)$.

$\mathbf{G} = [N_1, N_2, \ldots, N_d]$, as in (3.6).

$\mathbf{T} = [k_1, k_2, \ldots, k_d]$ is the coarsening request; k_i is the count of $(h \to 2h)$ transfers along the ith dimension. We say that *quadrupling* takes place along the ith dimension if $k_i = 2$. For reasons of space it is not possible to verify (in this manuscript) (4.2) for any realistic example; however, it is trivial to verify the same with a matrix manipulation software package.

Once the FW restriction operator matrix in d dimensions is set, the prolongation (d-linear interpolation) operator matrix can be obtained by the following relation:

$$\mathbf{P} = 2^{(\sum_{i=1}^{d} k_i)}(\mathbf{R}^T). \tag{4.3}$$

A generalized restriction operator (4.2) gives us the freedom to experiment with different types of coarsening strategies depending on the grid. Note that the FW restriction operator given by (4.2) provides the required matrix for any number of coarsenings along any number of dimensions for an abstract d-dimensional problem.

4.5. Computational work for *d*-multigrid. The practical feasibility of a d-multigrid method also has to take into account an estimate of the computational work that it involves. Following the notation from [11, 14], we call this work estimate W_l, assuming that a particular multigrid method is based on a hierarchy of grids $(\Omega_l, \Omega_{l-1}, \ldots, \Omega_1, \Omega_0)$, where Ω_0 is the coarsest grid. The *computational work* W_l per multigrid cycle on Ω_l is modeled by the recursion:

$$W_1 = W_1^0 + W_0, \qquad W_k = W_k^{k-1} + \gamma_{k-1}W_{k-1} \quad (k = 1, 2, \ldots, l). \tag{4.4}$$

W_k^{k-1} is the work estimate for a single 2-grid cycle (h_k, h_{k-1}) excluding the work needed to solve the defect equation on $\Omega_{h_{k-1}}$, and W_0 is the work required to solve

exactly on the coarsest grid. It is reasonable to assume that the multigrid components (relaxation, computation of defect, and the transfer of grid functions) applied to a single unknown require a number of arithmetic operations which—independent of k—is bounded by a small constant C. With M_k the total number of unknowns at grid level k, we have

$$W_k^{k-1} \leqslant CM_k \qquad (k = 1, 2, \ldots, l).$$

For a fixed cycle index γ (the number of times the coarse grids are cycled) and with the work on the coarsest grid neglected, this leads to

$$W_l \leqslant C\left[M_l + \gamma M_{l-1} + \gamma^2 M_{l-2} + \cdots + \gamma^{l-1} M_1\right]. \tag{4.5}$$

Before we proceed further, we consider it important to point out at this stage that the coarsening strategies (algorithms) that we use do not employ the same transfers at all grid levels. Depending on the anisotropy of the system, strategy 1 employs $(h \to 2h)$ transfers (along the strongly coupled dimensions) so that after application of this strategy at each level the grid comes closer to isotropy. When an acceptable isotropy is achieved strategy 1 resorts to standard full coarsening until the grid is brought to the coarsest possible level which the particular discretization allows. Strategy 2 is *hybrid* in the sense that it is composed of a mixture of standard and quadrupling transfers; moreover, it is *adaptive* just like strategy 1. In strategy 2 we employ full $(h \to 2h)$ transfers when isotropy is achieved, because quadrupling along all dimensions did not prove good even with the use of optimal relaxation parameters. Note that, for the type of grid-induced anisotropy encountered in the sparse-grid solution method, the number of dimensions along which the grid is coarsened increases (or remains constant in the case of an equidistant grid) at each successive level. This gives the following:

$$\begin{aligned} W_l &\leqslant \frac{2^{j\cdot s}}{(2^{j\cdot s} - \gamma)} C M_l, \\ W_l &\leqslant \frac{\tau}{(\tau - \gamma)} C M_l \qquad \text{for} \quad \gamma < \tau = 2^{j\cdot s}. \end{aligned} \tag{4.6}$$

Here j represents the number of dimensions coarsened, $s = 1$ for $(h \to 2h)$ transfers, and $s = 2$ for quadrupling. The worst case would be when the grid is highly stretched along a single dimension, which implies that coarsening takes place only along one dimension, so $j = 1$. With quadrupling, i.e., $s = 2$, this still renders $\tau = 4$, which implies that the complexity of the method is still $O(M_l)$ for $\gamma = 1, 2, 3$. This feature makes quadrupling particularly attractive in higher dimensions for non-equidistant grids. Strategy 1 also gives an $O(M_l)$ algorithm for $\gamma = 1, 2, 3$ as long as $j \geq 2$, whereas for $j = 1$ one has to apply a V-cycle.

If the problem is isotropic with an equidistant grid, we essentially have $s = 1$ and $j = d$, which gives the work estimates that appear in Table 4.1.

The $2d$ and $3d$ results are well known [11], and an estimate for a general d (isotropic problem on an equidistant grid) can be obtained by setting $j = d$ in (4.6).

This leads us now to apply the standard multigrid procedure to our model problem as the multigrid components have been adjusted for a general high-dimensional setting, along with a computational complexity estimate. A LFA of the ω-RB Jacobi method follows, through an implementation of which we derive the relaxation parameters for our experiments.

TABLE 4.1
Multigrid work estimates for the isotropic problem on equidistant grids.

γ	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$
1	$\frac{4}{3}\ CM_l$	$\frac{8}{7}\ CM_l$	$\frac{16}{15}\ CM_l$	$\frac{32}{31}\ CM_l$	$\frac{64}{63}\ CM_l$
2	$2\ CM_l$	$\frac{4}{3}\ CM_l$	$\frac{8}{7}\ CM_l$	$\frac{16}{15}\ CM_l$	$\frac{32}{31}\ CM_l$
4	$O(M_l\ \log_2\ M_l)$	$2\ CM_l$	$\frac{4}{3}\ CM_l$	$\frac{8}{7}\ CM_l$	$\frac{16}{15}\ CM_l$

5. Local Fourier smoothing analysis of the ω-RB Jacobi method. LFA [2] is a tool for analyzing the convergence behavior of multigrid methods. The finer details including the validity of LFA along with related theorems and proofs are presented in [15]. In this section we concentrate on the development of a Fourier representation for the ω-RB Jacobi method applied to the model operator from section 3.1, the definition of the *smoothing factor*, the evaluation of relaxation parameters, and extension to a general d-dimensional setting. We assume familiarity with the basics of this tool; see [11, 12, 14, 15, 17] for the same. For the most part in this section we carry on the definitions and notations as present in [14, 17, 18, 19].

Consider the d-dimensional anisotropic diffusion operator

$$-\sum_{i=1}^{d} \varepsilon_i \frac{\partial^2}{\partial x_i^2}, \quad \text{with} \quad \varepsilon_i > 0. \tag{5.1}$$

With respect to the analysis it is more convenient to scale the real positive coefficients ε_i as in [18, 19] and to replace (5.1) by

$$-\sum_{i=1}^{d} c_i \frac{\partial^2}{\partial x_i^2}, \quad \text{with} \quad c_i = \varepsilon_i / \sum_{j=1}^{d} \varepsilon_j \quad \text{and hence} \quad \sum_{i=1}^{d} c_i = 1. \tag{5.2}$$

The partial derivatives are discretized by second- or fourth-order differences (see (3.3), (3.4)) leading to the discrete counterparts of (5.2) denoted by L_h^{2o} and L_h^{4o}, respectively. In contrast to section 3.1 we assume that we are dealing with the same mesh size in each space direction, i.e., $h = h_1 = \cdots = h_d$. Here possible anisotropies caused by different mesh sizes are modeled by varying the coefficients c_i, which is more appropriate for the analysis.

LFA takes into account only the *local* nature of the operator. Thus for an effective analysis through this tool one has to have constant coefficients, and one has to neglect the boundary effects. More precisely, all occurring operators are extended to an infinite grid

$$\mathbf{G}_h := \left\{ \boldsymbol{x} = (x_1, \ldots, x_d)^T = \boldsymbol{\kappa} h = h\,(\kappa_1, \ldots, \kappa_d)^T \ :\ \boldsymbol{\kappa} \in \mathbb{Z} \right\}, \tag{5.3}$$

with mesh size h. The discrete eigenfunctions (*Fourier components*) of the resulting infinite grid operators serve as fundamental quantities in LFA; they are obviously given by

$$\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = e^{i\boldsymbol{\theta}\boldsymbol{x}/h}, \quad \boldsymbol{x} \in \mathbf{G}_h. \tag{5.4}$$

For this analysis we assume that $\boldsymbol{\theta}$ varies continously in $\mathbb{R}^d$. As 2π is the period of φ, we are led to the identity:

$$\varphi(\boldsymbol{\theta}, \boldsymbol{x}) \equiv \varphi(\boldsymbol{\theta}', \boldsymbol{x}) \qquad \text{for } \boldsymbol{x} \in \mathbf{G}_h \quad \text{iff } \boldsymbol{\theta} = \boldsymbol{\theta}' (\text{mod } 2\pi), \tag{5.5}$$

where this difference—of multiples of 2π—is between all of the components of the d-tuples, $(\boldsymbol{\theta} \ \& \ \boldsymbol{\theta}')$; thus it suffices to consider these functions only for $\boldsymbol{\theta} \in \Theta = [-\pi, \pi)^d$.

The corresponding eigenvalues (*Fourier symbols*) of L_h^{2o} and L_h^{4o} read

$$\widetilde{L}_h^{2o}(\boldsymbol{\theta}) = \frac{2}{h^2}\left(1 - \sum_{i=1}^{d} c_i \cos(\theta_i)\right) \quad \text{and}$$

$$\widetilde{L}_h^{4o}(\boldsymbol{\theta}) = \frac{1}{6h^2}\left(15 - \sum_{i=1}^{d} c_i \left(16\cos(\theta_i) - \cos(2\theta_i)\right)\right),$$

respectively.

5.1. High and low Fourier frequencies. If full coarsening ($\boldsymbol{H} = 2\boldsymbol{h}$ or $\boldsymbol{H} = 4\boldsymbol{h}$ in the case of quadrupling) is selected, the Fourier components $\varphi_h(\boldsymbol{\theta}, \boldsymbol{x})$, with $|\boldsymbol{\theta}| := \max\{|\theta_1|, \ldots, |\theta_d|\} \le \pi/2$ (or $\le \pi/4$), are also visible on the coarse grid $\mathbf{G}_H$, whereas components with $|\boldsymbol{\theta}| > \pi/2$ ($> \pi/4$) coincide with certain $\varphi_h(\hat{\boldsymbol{\theta}}, \boldsymbol{x})$, where $|\hat{\boldsymbol{\theta}}| \le \pi/2$ ($\le \pi/4$). This observation leads to the distinction between high and low Fourier frequencies.

DEFINITION 1 (high and low frequencies for full coarsening). *An element θ_j ($j \in \{1, \ldots, d\}$) of a Fourier frequency $\boldsymbol{\theta}$ is called* low *if*

$$-\pi/2 \le \theta_j < \pi/2 \quad (\textit{or } -\pi/4 \le \theta_j < \pi/4 \textit{ in the case of quadrupling}) \quad \textit{for} \quad \boldsymbol{\theta} \in \Theta.$$

Otherwise it is called high. *We speak of a* low Fourier frequency $\boldsymbol{\theta} \in \Theta$, *if all of its elements are low. Otherwise it is named a* high frequency.

The distinction obviously depends on the coarsening as, for different coarsening strategies, different sets of Fourier frequencies are visible on the coarse grid. We define the set of coordinate indices by $\mathcal{J} := \{1, \ldots, d\}$. In the case of partial coarsening (compare with section 4.2), the grid is coarsened only in a subset $\{x_j \mid j \in \mathcal{J}^c \subset \mathcal{J}\}$ of the coordinate directions and remains fixed in the other coordinates x_j, with $j \in \mathcal{J}^f = \mathcal{J} \setminus \mathcal{J}^c$. Especially for two-dimensional problems, this is often called semicoarsening as well. In this case, we have for the coarse grid mesh size $\boldsymbol{H}$ that $H_j = 2h_j$ (or $H_j = 4h_j$ in the case of quadrupling) for $j \in \mathcal{J}^c$ and $H_j = h_j$ for $j \in \mathcal{J}^f$. Then the definition of high and low frequencies has to be adapted accordingly.

DEFINITION 2 (high and low frequencies for partial coarsening). *A Fourier frequency $\boldsymbol{\theta} \in \Theta$ is called* low *if all elements $\theta_j \in \mathcal{I}^c$ are low; compare with Definition* 1. *Otherwise it is called a* high frequency.

5.2. Fourier representation of ω-RB Jacobi. ω-RB Jacobi relaxation consists of two partial steps of ω-Jacobi-type; compare with section 4.1. The iteration matrix of classical damped Jacobi relaxation is given by $\mathbf{S}_h = \mathbf{I}_h - \omega \mathbf{D}_h^{-1} \mathbf{A}_h$, where $\mathbf{I}_h$ denotes the $(M \times M)$-identity matrix and $\mathbf{D}_h$ the diagonal part of $\mathbf{A}_h$. Hence, the Fourier components remain eigenfunctions of ω-Jacobi relaxation. The Fourier symbols of ω-Jacobi relaxation applied to L_h^{2o} and L_h^{4o} read

$$A^{2o}(\boldsymbol{\theta}, \omega) = 1 - \omega \frac{h^2}{2} \widetilde{L}_h(\boldsymbol{\theta}) = 1 - \omega\left(1 - \sum_{i=1}^{d} c_i \cos(\theta_i)\right) \quad \text{and}$$

$$A^{4o}(\boldsymbol{\theta}, \omega) = 1 - \omega \frac{6h^2}{15} \widetilde{L}_h(\boldsymbol{\theta}) = 1 - \omega\left(1 - \frac{1}{15}\sum_{i=1}^{d} c_i \left(16\cos(\theta_i) - \cos(2\theta_i)\right)\right),$$

respectively; see [16, 18, 19].

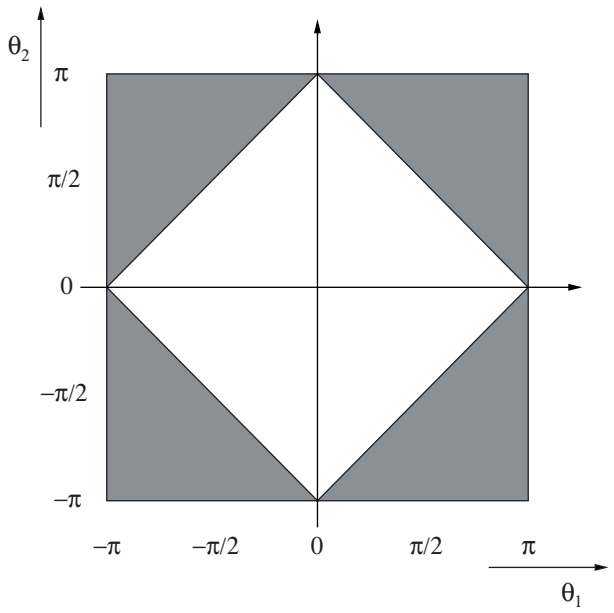

FIG. 1. *Low (white region) and high (shaded region) frequencies for red-black coarsening in two dimensions.*

Note that for pattern relaxations such as ω-RB Jacobi the Fourier components are no longer eigenfunctions. However, the minimal invariant subspaces for ω-RB Jacobi are two-dimensional; see [18]. Each component $\varphi_h(\boldsymbol{\theta}, \boldsymbol{x})$ is coupled only with $\varphi_h(\widehat{\boldsymbol{\theta}}, \boldsymbol{x})$, where

$$\widehat{\theta}_j := \theta_j - \operatorname{sign}(\theta_j)\,\pi, \quad \text{with} \quad j \in \mathcal{J}.$$

Note that ω-RB Jacobi couples those Fourier components that alias on the coarse grid in the case of red-black coarsening. Considering red-black coarsening, the set of low frequencies (compare with Definitions 1 and 2) is given by

$$\Theta_{RB} := \left\{ \boldsymbol{\theta} \in \Theta \,:\, |\boldsymbol{\theta}| := \sum_{i=1}^{d} |\theta_i| \le \frac{d}{2}\pi \right\}. \tag{5.6}$$

For $d = 2$, the set of low frequencies Θ_{RB} is illustrated in Figure 1.

We decompose the set of Fourier frequencies Θ into a direct sum of such two-dimensional subspaces that are coupled by ω-RB Jacobi:

$$\Theta = \bigoplus_{\boldsymbol{\theta} \in \Theta_{RB}} \left\{ \boldsymbol{\theta}, \widehat{\boldsymbol{\theta}} \right\}.$$

The Fourier representations of the half-sweep operators which represent the smoothing steps over the red points (R) and the black points (B) w.r.t. the two-dimensional

minimal invariant subspaces are given in [19]. They read

$$\widetilde{S}_h^R(\boldsymbol{\theta},\omega) = \frac{1}{2}\begin{pmatrix} A(\boldsymbol{\theta},\omega)+1 & A(\widehat{\boldsymbol{\theta}},\omega)-1 \\ A(\boldsymbol{\theta},\omega)-1 & A(\widehat{\boldsymbol{\theta}},\omega)+1 \end{pmatrix}$$

$$\text{and} \qquad \widetilde{S}_h^B(\boldsymbol{\theta},\omega) = \frac{1}{2}\begin{pmatrix} A(\boldsymbol{\theta},\omega)+1 & -A(\widehat{\boldsymbol{\theta}},\omega)+1 \\ -A(\boldsymbol{\theta},\omega)+1 & A(\widehat{\boldsymbol{\theta}},\omega)+1 \end{pmatrix},$$

$$\text{with} \quad A(\boldsymbol{\theta},\omega) = A^{2o}(\boldsymbol{\theta},\omega), A^{4o}(\boldsymbol{\theta},\omega) \quad \text{and} \quad A(\widehat{\boldsymbol{\theta}},\omega) = A^{2o}(\widehat{\boldsymbol{\theta}},\omega), A^{4o}(\widehat{\boldsymbol{\theta}},\omega).$$

The transformation matrix $\widetilde{S}_h(\boldsymbol{\theta},\omega)$ for $(\phi_h(\boldsymbol{\theta},\,.\,), \phi_h(\widehat{\boldsymbol{\theta}},\,.\,))^T$ after one relaxation step is then given by

$$\widetilde{S}_h(\boldsymbol{\theta},\omega) = \widetilde{S}_h^B(\boldsymbol{\theta},\omega)\,\widetilde{S}_h^R(\boldsymbol{\theta},\omega). \tag{5.7}$$

The four components of the (2×2)-matrix $\widetilde{S}_h(\boldsymbol{\theta},\omega) = (s_{ij})_{i,j=1,2}$ read

$$\begin{aligned} s_{11} &= \frac{1}{4}\left[(A(\boldsymbol{\theta},\omega)+1)^2 + (A(\widehat{\boldsymbol{\theta}},\omega)-1)(1-A(\boldsymbol{\theta},\omega))\right], \\ s_{12} &= \frac{1}{4}\left[(A(\boldsymbol{\theta},\omega)+1)(1-A(\widehat{\boldsymbol{\theta}},\omega)) + A(\widehat{\boldsymbol{\theta}},\omega)^2 - 1\right], \\ s_{21} &= \frac{1}{4}\left[(A(\widehat{\boldsymbol{\theta}},\omega)+1)(1-A(\boldsymbol{\theta},\omega)) + A(\boldsymbol{\theta},\omega)^2 - 1\right], \\ s_{22} &= \frac{1}{4}\left[(A(\widehat{\boldsymbol{\theta}},\omega)+1)^2 + (A(\widehat{\boldsymbol{\theta}},\omega)-1)(1-A(\boldsymbol{\theta},\omega))\right]. \end{aligned} \tag{5.8}$$

5.3. Smoothing factor. In order to measure the smoothing properties of pattern relaxation methods, we adopt the general definition of the smoothing factor from [11]. Here the "real" coarse grid correction for a two-grid method is replaced by an "ideal" coarse grid correction operator Q_h^H which annihilates the low-frequency error components and leaves the high-frequency components unchanged. Q_h^H is a projection operator onto the space of high-frequency components. The Fourier representation of the ideal coarse grid correction operator w.r.t. the subspaces $\{\phi_{\boldsymbol{h}}(\boldsymbol{\theta},\,.\,), \phi_{\boldsymbol{h}}(\widehat{\boldsymbol{\theta}},\,.\,)\}$ $(\boldsymbol{\theta}\in\Theta_{RB})$ is then given by

$$\widetilde{Q}_h^H(\boldsymbol{\theta}) = \begin{pmatrix} q(\boldsymbol{\theta}) & 0 \\ 0 & q(\widehat{\boldsymbol{\theta}}) \end{pmatrix}, \tag{5.9}$$

where $q=0$ for a low argument and 1 otherwise; compare with Definitions 1 and 2.

The smoothing factor μ of the ω-RB Jacobi operator is defined as the worst factor by which the high-frequency errors are reduced per iteration step. So with ν denoting the number of relaxation sweeps, and ρ the matrix spectral radius, we have

$$\mu(\omega) := \sup_{\boldsymbol{\theta}\in\Theta_{RB}} \left\{ \sqrt[\nu]{\rho\left(\widetilde{Q}_h^H(\boldsymbol{\theta})\widetilde{S}_h^\nu(\boldsymbol{\theta},\omega)\right)} \right\}. \tag{5.10}$$

The smoothing factor can easily be calculated by a computer program which allows for a numerical determination of optimal relaxation parameters for different coarsening

TABLE 5.1
$\mu(1)$, ω_{opt}, *and associated* $\mu(\omega_{opt})$ *for* $\nu = 1$ *relaxation sweep.*

Left: *Doubling along all dimensions, equidistant grid, isotropy,* $O(h^2)$.
Right: *Quadrupling along all dimensions, equidistant grid, isotropy,* $O(h^2)$.

Doubling ($h \to 2h$)					Quadrupling ($h \to 4h$)			
d	$\mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	ω_{ub}	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	ω_{ub}
2	0.25	1.049	0.16	1.072	0.73	1.315	0.31	1.316
3	0.44	1.133	0.23	1.144	0.81	1.398	0.40	1.393
4	0.56	1.195	0.28	1.202	0.86	1.454	0.45	1.455
5	0.64	1.243	0.31	1.250	0.89	1.496	0.50	1.502
6	0.69	1.283	0.35	1.285	0.90	1.528	0.53	1.519

Left: *Doubling along 1 dimension, non-equidistant grid,* $128 \times 32^{(d-1)}$, $O(h^2)$.
Right: *Quadrupling along 1 dimension, non-equidistant grid,* $128 \times 32^{(d-1)}$, $O(h^2)$.

Doubling ($h \to 2h$)					Quadrupling ($h \to 4h$)			
d	$\mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	ω_{ub}	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	ω_{ub}
2	0.125	0.946	0.08	1.033	0.52	1.182	0.22	1.181
3	0.125	0.963	0.10	1.033	0.55	1.190	0.25	1.197
4	0.125	0.980	0.11	1.033	0.57	1.199	0.27	1.207
5	0.125	0.997	0.122	1.033	0.59	1.209	0.29	1.219
6	0.15	1.013	0.13	1.04	0.60	1.219	0.30	1.225

strategies and both types of discretization under consideration. We would like to emphasize that the optimal relaxation parameter depends on the number of relaxation steps ν as can be seen from (5.10).

Note that there are explicit (but very lengthy) analytical formulas for the optimal relaxation parameters ω_{opt} in the case of full coarsening applied to the second-order discretization [16, 19]. Moreover, there is a close to optimal upper bound ω_{ub} [19] for the optimal relaxation parameters which is given by the following handy expression:

$$\omega_{opt} < \omega_{ub} = \frac{2}{1 + \sqrt{1 - \mu(\omega = 1)}}.$$

For partial coarsening and especially for the fourth-order discretization, it seems to be very difficult to derive analytic expressions for ω_{opt}. However, for the second-order discretization it turned out that ω_{ub} is a satisfactory approximation for ω_{opt} even in the case of partial coarsening and quadrupling (but not necessarily an upper bound anymore); see Table 5.1, which presents ω_{opt} (optimized for $\nu = 1$) and ω_{ub} for equidistant and non-equidistant grids. This is a nice generalization of the results from [19]. For the fourth-order discretization this is no longer true, and we have to stay with the numerically derived values.

From the values in Table 5.2 (optimized for $\nu = 2$), we see that the *smoothness enhancement effect* of using optimal relaxation parameters is more prominent and pronouced with strategy 2. With strategy 1 this enhancement becomes prominent in the case of *nearly isotropic problems* (with grids that are equidistant along $(d - 2)$ or more dimensions). With anisotropic problems where anisotropy is induced by discretization on grids highly elongated along a single dimension (and dealt with strategy 1), the choice $\omega = 1$ is more suitable—first—because the optimal values themselves are very close to 1 and therefore do not bring about a substantial enhancement—and second—the cost of relaxation itself is cut down, which saves CPU time.

TABLE 5.2
$\mu(1)$, ω_{opt}, *and associated* $\mu(\omega_{opt})$ *for* $\nu = 2$ *relaxation sweeps.*

Left: *Doubling along all dimensions, equidistant grid, isotropy,* $O(h^2)$.
Right: *Quadrupling along all dimensions, equidistant grid, isotropy,* $O(h^2)$.

Doubling ($h \to 2h$)				Quadrupling ($h \to 4h$)		
d	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$
2	0.25	1.0107	0.23	0.73	1.3062	0.39
3	0.44	1.1136	0.28	0.81	1.3928	0.46
4	0.56	1.1832	0.31	0.86	1.4507	0.50
5	0.64	1.2356	0.35	0.89	1.4934	0.53
6	0.69	1.2771	0.37	0.90	1.5266	0.56

Left: *Doubling along 1 dimension, non-equidistant grid,* $128 \times 32^{(d-1)}$, $O(h^2)$.
Right: *Quadrupling along 1 dimension, non-equidistant grid,* $128 \times 32^{(d-1)}$, $O(h^2)$.

Doubling ($h \to 2h$)				Quadrupling ($h \to 4h$)		
d	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$
2	0.23	0.8490	0.18	0.52	1.1598	0.30
3	0.23	0.8682	0.19	0.55	1.1735	0.31
4	0.23	0.8858	0.19	0.57	1.1864	0.32
5	0.23	0.9022	0.20	0.59	1.1986	0.32
6	0.23	0.9174	0.20	0.60	1.2100	0.33

Left: *Doubling along all dimensions, equidistant grid, isotropy,* $O(h^4)$.
Right: *Quadrupling along all dimensions, equidistant grid, isotropy,* $O(h^4)$.

Doubling ($h \to 2h$)				Quadrupling ($h \to 4h$)		
d	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$
2	0.28	1.0260	0.25	0.76	1.3110	0.40
3	0.46	1.1108	0.29	0.84	1.3782	0.47
4	0.57	1.1683	0.33	0.87	1.4238	0.52
5	0.65	1.2128	0.36	0.90	1.4579	0.55
6	0.70	1.2492	0.38	0.91	1.4847	0.58

Left: *Doubling along 1 dimension, non-equidistant grid,* $128 \times 32^{(d-1)}$, $O(h^4)$.
Right: *Quadrupling along 1 dimension, non-equidistant grid,* $128 \times 32^{(d-1)}$, $O(h^4)$.

Doubling ($h \to 2h$)				Quadrupling ($h \to 4h$)		
d	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$	$\mu(\omega) = \mu(1)$	ω_{opt}	$\mu(\omega_{opt})$
2	0.24	0.8859	0.20	0.59	1.1900	0.35
3	0.26	0.8957	0.21	0.61	1.1942	0.37
4	0.25	0.9122	0.22	0.62	1.2046	0.38
5	0.25	0.9268	0.22	0.64	1.2146	0.38
6	0.25	0.9399	0.23	0.65	1.2243	0.39

6. Numerical experiments. We now present the results of some of our numerical experiments. The emphasis is on grid-induced anisotropies. The PDE is discretized on various equidistant and non-equidistant grids, and the spectral radius ρ of the multigrid iteration operator M_h is presented, which represents the asymptotic convergence factor. During the course of an experiment the only quantity available to estimate this factor is the defect d_h^i after the ith multigrid cycle. The numerical results—presented in Tables 6.1–6.4—depict a close match between the theoretical smoothing factors μ (computed for coarsening based on the finest grid) and an emperical estimate of the multigrid convergence factor $\rho(M_h)$, which we denote by q^m and define as

$$q^m := \frac{\| d_h^m \|}{\| d_h^{m-1} \|},$$

where m represents the number of iterations or multigrid cycles that the discrete problem takes to converge to the numerical solution.

All of the experiments employ one pre- and one postsmoothing, and so the smoothing factor is displayed as a square for a correspondence with q^m; compare with (5.10). The optimal relaxation parameters ω_{opt} that we employ are computed for $\nu = 2$. For each order of accuracy and for each dimension, we have chosen two kinds of grid, one equidistant and one non-equidistant (highly stretched in one dimension). q^m is displayed against the number of multigrid cycles that the experiment took to converge to the tolerance value, which for all of the experiments is 10^{-10}, i.e.,

$$\frac{||\mathbf{b}_h - \mathbf{A}_h \mathbf{u}_h^m||}{||\mathbf{b}_h||} \mid \leq tol.$$

This termination criterion is equivalent to the one based on relative residual reduction because our starting soluton in these experiments is always an all zero vector. Moreover this criterion corresponds to a residual reduction by ≈ 7 orders of magnitude. Through the numerical solution of the model problem we approximate the following test function:

$$u(\mathbf{x}) = \frac{\sum_{i=1}^{d} \sin(d\pi^2 x_i)}{d\pi + \sum_{i=1}^{d} x_i}, \tag{6.1}$$

$\varepsilon_i = 1 \ \forall \ i$, and $\Omega = (0,1)^d$ for all of the problems. The values of this function at the boundary are taken as Dirichlet boundary conditions, and its analytically computed Laplacian forms the source function for these experiments. The experiments include the V- and the W-cycles. In some of the experiments the grids used for the $O(h^2)$ operator are different from the ones for the $O(h^4)$ operator. This serves only the purpose of accumulating results for slightly *different-sized* experiments. C_2^2 indicates the use of the second-order stencil along all coarsened and noncoarsened dimensions; likewise for C_4^4. C_2^4 indicates the use of the $O(h^4)$ long stencil along all noncoarsened dimensions and the use of the $O(h^2)$ stencil on the coarse grids along the dimensions where coarsening takes place. This *hybrid* coarse grid discretization gives fourth-order accuracy and converges *faster* than the conventional fourth-order long stencil.

Table 6.1 presents experimental results for equidistant grids. A comparison of the convergence results with and without optimal relaxation parameters indicates the benefits of using them for high-dimensional problems. The cutdown in the multigrid convergence factor as well as in the number of multigrid cycles (required to converge to tol) is quite significant for $d \geq 3$.

Table 6.2 presents experimental results for non-equidistant grids which we have chosen to be highly stretched in only one dimension. Because of this characteristic these experiments are computationally more expensive than any other as coarsening takes place only along the elongated dimension. This is exactly the opposite of the previous case, where coarsening took place along all dimensions, and hence the cutdown in the number of unknowns at each level was optimal. In this table we display the results that we get from strategy 1, which is based purely on $h \rightarrow 2h$ transfers. Optimal relaxation parameters in this case pay off only with V-cycles, with $\omega = 1$ serving as a perfect compromise with W-cycles.

In Table 6.3 we have reworked the experiments of Table 6.2 but with strategy 2 this time. Partial quadrupling (strategy 2) ensures an $O(M_l)$ algorithm even with grids of this type. Note that the computational complexity of the experiments in Table 6.2 is $O(M_l \ \log_2 \ M_l)$, even though the multigrid convergence factor is quite impressive

TABLE 6.1

Results of numerical experiments for $V(1,1)$ and $W(1,1)$ on equidistant grids. The observed convergence rate with the number of iterations, i.e., (q^m/# it.), is presented. For a correspondence comparison the smoothing factors $[\mu(1)]^2$ and $[\mu(\omega_{opt})]^2$ (computed for coarsening based on the finest grid) are also displayed. Results presented include experiments with $\omega = 1$ as well as $\omega = \omega_{opt}$.

$G = 128^2$	V	W	V	W
	$[\mu(1)]^2 = 0.06$		$[\mu(1.011)]^2 = 0.05$	
$O(h^2)C_2^2$	0.10/8	0.06/7	0.09/8	0.05/7
$G = 128^2$	$[\mu(1)]^2 = 0.08$		$[\mu(1.026)]^2 = 0.06$	
$O(h^4)C_4^4$	0.13/9	0.10/8	0.12/9	0.08/7
$O(h^4)C_2^4$	0.10/8	0.07/7	0.09/8	0.05/7
$G = 128^3$	V	W	V	W
	$[\mu(1)]^2 = 0.20$		$[\mu(1.114)]^2 = 0.08$	
$O(h^2)C_2^2$	0.22/11	0.18/10	0.12/9	0.07/7
$G = 64^3$	$[\mu(1)]^2 = 0.21$		$[\mu(1.111)]^2 = 0.08$	
$O(h^4)C_4^4$	0.26/12	0.22/11	0.16/10	0.09/8
$O(h^4)C_2^4$	0.24/12	0.21/11	0.13/9	0.07/7
$G = 64^4$	V	W	V	W
	$[\mu(1)]^2 = 0.32$		$[\mu(1.183)]^2 = 0.10$	
$O(h^2)C_2^2$	0.33/14	0.30/12	0.16/10	0.08/7
$G = 32^4$	$[\mu(1)]^2 = 0.33$		$[\mu(0.168)]^2 = 0.11$	
$O(h^4)C_4^4$	0.39/16	0.34/14	0.20/10	0.11/9
$O(h^4)C_2^4$	0.35/15	0.34/14	0.15/9	0.11/8
$G = 16^5$	V	W	V	W
	$[\mu(1)]^2 = 0.41$		$[\mu(1.236)]^2 = 0.12$	
$O(h^2)C_2^2$	0.38/16	0.38/15	0.18/10	0.09/8
$G = 8^6$	V	W	V	W
	$[\mu(1)]^2 = 0.48$		$[\mu(1.277)]^2 = 0.14$	
$O(h^2)C_2^2$	0.35/15	0.34/15	0.12/9	0.11/9

there. These results show the important role of optimal relaxation parameters in enhancing convergence of the multigrid with quadrupling transfers.

To make the discussion complete we have included some more experiments on a *nearly equidistant grid*. The results are reported in Figure 2. The convergence report is depicted against the iteration scale as well as against the CPU time scale. A comparison of the results of strategies 1 and 2 suggests that, for these kind of grids, a combination of strategy 2 with V-cycles and optimal relaxation parameters works very nicely.

A graphical presentation of the convergence behavior of an isotropic 4-dimensional multigrid experiment appears in Figure 3. The defect reduction is displayed against the iteration and the CPU time scale. To emphasize the inefficiency of full quadrupling with completely equidistant grids, we have included the results against full doubling. Clearly enough, here standard coarsening is the strategy of choice. This is the main reason why we do not keep quadrupling transfers in strategy 2, once grid equidistance has been achieved, but rather resort to doubling from this stage onwards. Quadrupling along all dimensions at the same time hampers the multigrid convergence rate as well as increasing the overall computation time.

Finally, we have performed two general 5-dimensional anisotropic experiments. For the first experiment the anisotropy comes only from discretization on a non-equidistant grid; and the results appear in Figure 4, the sequence of grids that we get

TABLE 6.2

Results of numerical experiments for $V(1,1)$ and $W(1,1)$ with strategy 1 on grids stretched along 1 dimension. The observed convergence rate with the number of iterations, i.e., $(q^m/\#\text{ it.})$, is presented. For a correspondence comparison the smoothing factors $[\mu(1)]^2$ and $[\mu(\omega_{opt})]^2$ (computed for coarsening based on the finest grid) are also displayed. Results presented include experiments with $\omega = 1$ as well as $\omega = \omega_{opt}$.

$G = 512 \times 32$	V	W	V	W
	$[\mu(1)]^2 = 0.05$		$[\mu(0.835)]^2 = 0.03$	
$O(h^2)C_2^2$	0.06/8	0.003/4	0.06/8	0.03/6
$G = 512 \times 32$	$[\mu(1)]^2 = 0.06$		$[\mu(0.879)]^2 = 0.04$	
$O(h^4)C_4^4$	0.10/8	0.03/6	0.09/8	0.07/7
$O(h^4)C_2^4$	0.10/7	0.02/6	0.05/7	0.04/6
$G = 512 \times 32^2$	V	W	V	W
	$[\mu(1)]^2 = 0.05$		$[\mu(0.835)]^2 = 0.03$	
$O(h^2)C_2^2$	0.20/11	0.005/4	0.12/9	0.03/6
$G = 128 \times 32^2$	$[\mu(1)]^2 = 0.06$		$[\mu(0.978)]^2 = 0.06$	
$O(h^4)C_4^4$	0.24/11	0.04/6	0.17/9	0.05/6
$O(h^4)C_2^4$	0.17/10	0.04/6	0.11/8	0.04/6
$G = 128 \times 8^3$	V	W	V	W
	$[\mu(1)]^2 = 0.05$		$[\mu(0.836)]^2 = 0.03$	
$O(h^2)C_2^2$	0.20/11	0.007/4	0.11/8	0.04/5
$G = 128 \times 32^3$	$[\mu(1)]^2 = 0.06$		$[\mu(1.9121)]^2 = 0.05$	
$O(h^4)C_4^4$	0.33/13	0.04/6	0.21/9	0.06/6
$O(h^4)C_2^4$	0.27/12	0.05/6	0.15/9	0.03/6
$G = 128 \times 8^4$	V	W	V	W
	$[\mu(1)]^2 = 0.05$		$[\mu(0.837)]^2 = 0.03$	
$O(h^2)C_2^2$	0.24/12	0.009/4	0.11/8	0.04/6
$G = 128 \times 8^5$	V	W	V	W
	$[\mu(1)]^2 = 0.05$		$[\mu(0.837)]^2 = 0.03$	
$O(h^2)C_2^2$	0.27/13	0.009/5	0.12/9	0.04/6

with both coarsening strategies are displayed in section 4.2, (4.1). It is quite apparent that the use of optimal relaxation parameters pays off significantly well with V-cycles in combination with strategy 1 ($\omega = 1$ serving as a perfect choice for W-cycles) and with both V- and W-cycles with strategy 2. The second experiment is more general in that it has anisotropic coefficients ε_i in the PDE, it is discretized on the same non-equidistant finest grid, and the domain dimensions are not exactly unit. The results are displayed in Table 6.4. Clearly, V-cycles with optimal weighting in the relaxation process coupled with either coarsening strategy give excellent results. We would like to point out that we expect strategy 1 to be more dependable than strategy 2 if the number of dimensions coarsened on the finest grid is greater than $d/2$.

We have used Matlab as our testing platform; the same experiments implemented in C (in more optimized computer programs) are expected to render smaller CPU times than those presented.

Remark 1. The multigrid convergence factors displayed in the tables are mostly under 0.1, implying that a full multigrid algorithm starting on the coarsest grid is expected to reach an approximate solution up to the discretization accuracy in just one or two cycles.

Remark 2. It is also important to point out that, for very coarse discretization grids (say, 8 points along all dimensions), the asymptotic convergence of the relaxation

TABLE 6.3

Results of numerical experiments for $V(1,1)$ and $W(1,1)$ with strategy 2 on grids stretched along 1 dimension. The observed convergence rate with the number of iterations, i.e., $(q^m/\#\text{ it.})$, is presented. For a correspondence comparison the smoothing factors $[\mu(1)]^2$ and $[\mu(\omega_{opt})]^2$ (computed for coarsening based on the finest grid) are also displayed. Results presented include experiments with $\omega = 1$ as well as $\omega = \omega_{opt}$.

$G = 512 \times 32$	V	W	V	W
	$[\mu(1)]^2 = 0.25$		$[\mu(1.147)]^2 = 0.09$	
$O(h^2)C_2^2$	0.27/13	0.24/11	0.14/10	0.08/9
$G = 512 \times 32$	$[\mu(1)]^2 = 0.32$		$[\mu(1.190)]^2 = 0.12$	
$O(h^4)C_4^4$	0.34/14	0.31/13	0.20/10	0.13/8
$O(h^4)C_2^4$	0.31/13	0.30/13	0.12/9	0.09/7
$G = 512 \times 32^2$	V	W	V	W
	$[\mu(1)]^2 = 0.25$		$[\mu(1.147)]^2 = 0.09$	
$O(h^2)C_2^2$	0.30/13	0.24/11	0.17/13	0.08/10
$G = 128 \times 32^2$	$[\mu(1)]^2 = 0.37$		$[\mu(1.194)]^2 = 0.14$	
$O(h^4)C_4^4$	0.34/14	0.35/14	0.22/10	0.13/9
$O(h^4)C_2^4$	0.34/14	0.35/14	0.16/9	0.12/8
$G = 128 \times 8^3$	V	W	V	W
	$[\mu(1)]^2 = 0.25$		$[\mu(1.148)]^2 = 0.09$	
$O(h^2)C_2^2$	0.32	0.24/11	0.16/13	0.08/10
$G = 128 \times 32^3$	$[\mu(1)]^2 = 0.39$		$[\mu(1.205)]^2 = 0.14$	
$O(h^4)C_4^4$	0.38/16	0.37/15	0.24/11	0.14/9
$O(h^4)C_2^4$	0.38/16	0.36/14	0.22/11	0.12/8
$G = 128 \times 8^4$	V	W	V	W
	$[\mu(1)]^2 = 0.26$		$[\mu(1.149)]^2 = 0.09$	
$O(h^2)C_2^2$	0.35/15	0.24/11	0.17/13	0.08/10
$G = 128 \times 8^5$	V	W	V	W
	$[\mu(1)]^2 = 0.26$		$[\mu(1.150]^2 = 0.09$	
$O(h^2)C_2^2$	0.38/16	0.24/11	0.18/13	0.08/10

TABLE 6.4

A general 5d experiment on a non-equidistant grid $G = [32\ \ 8\ \ 8\ \ 128\ \ 32]$. The domain $\Omega = (0.11, 1.21) \times (0.5, 1.51) \times (0.25, 1.26) \times (0, 1) \times (0, 1.11)$, and constant coefficients $\boldsymbol{c} = [200.33, 10^{-7}, 2, 1.5, 200.49]$.

Strategy 1	V	W	V	W
	$[\mu(1)]^2 = 0.08$		$[\mu(1.031)]^2 = 0.058$	
$O(h^2)C_2^2$	0.13/11	0.07/9	0.08/9	0.058/8
Strategy 2	V	W	V	W
	$[\mu(1)]^2 = 0.55$		$[\mu(1.319)]^2 = 0.16$	
$O(h^2)C_2^2$	0.53/31	0.53/31	0.16/11	0.14/11

method ω-RB Jacobi is also quite satisfactory. To confirm this, we conducted a 6-dimensional asymptotic convergence experiment without a hierarchy of multiple grids and measured the number of iterations and the time taken to reach the tolerance (10^{-10}). Though the use of multigrid cuts down the number of iterations by around a factor of 2.5, the time taken to meet the tolerance level is the same.

7. Conclusion. The central emphasis of this paper is on multigrid techniques for high-dimensional elliptic PDEs. To alleviate the implementation issues we have

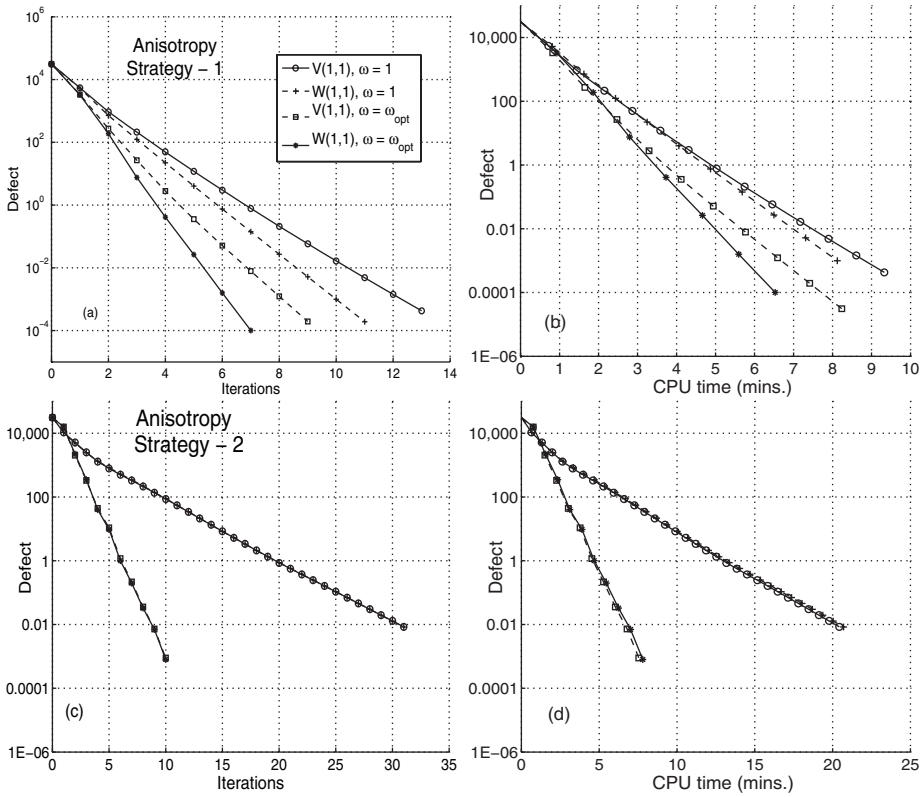

FIG. 2. *Convergence behavior of different multigrid cycles for a* 4*d problem, on a non-equidistant grid* [64 64 64 16] *(anisotropy with* 3,750,705 *unknowns), left side defect vs. iterations, right side defect vs. CPU time.*

shown how difference operator matrices can be put together through tensor methods. This renders testing in abstract higher d dimensions easy. Of course, tailored routines (for specific applications) can be programmed and tuned as per requirement. The main idea has been to demonstrate how the multigrid convergence factor can be reduced efficiently with appropriate coarsening strategies combined with the use of relaxation parameters in the smoothing process. Through numerical results supported by the LFA we have shown that partial quadrupling is a strategy of choice in higher dimensions and ensures a computational complexity of $O(M_l)$ even in the worst case, i.e., coarsening in one dimension only. This we have confirmed through a complexity analysis. Results of the numerical experiments display the excellent multigrid convergence that can be brought about with the presented strategies.

Appendix A. Implementational aspects of ω-RB Jacobi. ω-RB Jacobi can be implemented in several ways depending on the test platform. We have here a method that is suitable for iterations over a solution vector as opposed to explicit updates in a loop. The scheme that we present in this appendix is developed to bring about the odd-even (red-black) partitioning with minimal manipulation of the grid points.

We store three vectors in this scheme (two of which have just half the storage requirement as the first one). The first vector is the unmanipulated vector (henceforth

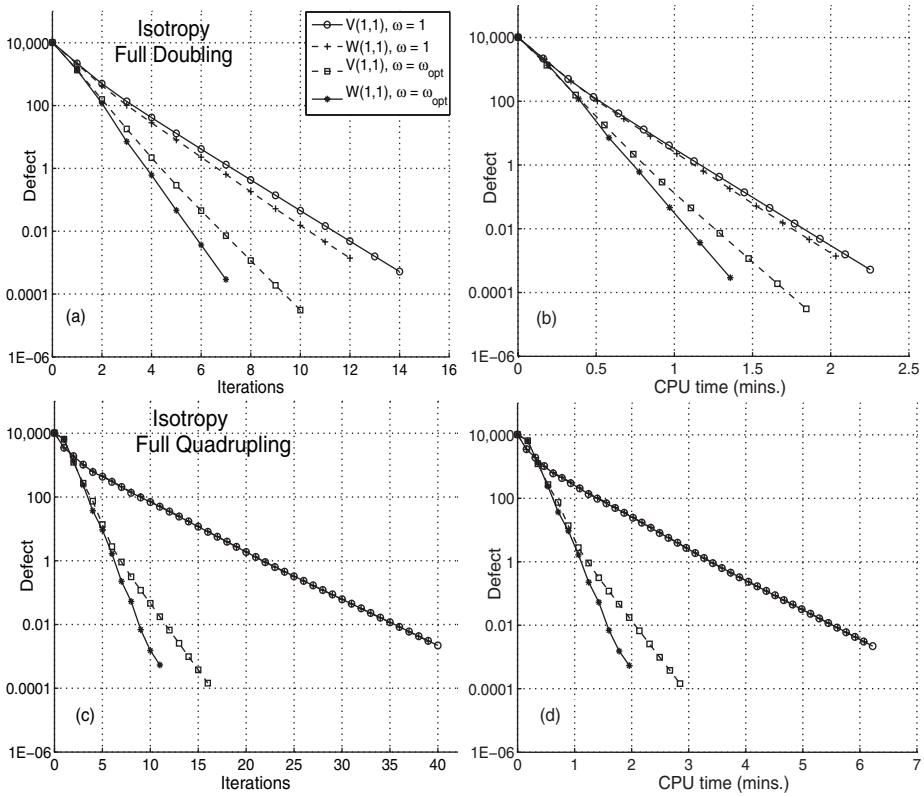

FIG. 3. *Convergence behavior of different multigrid cycles for a 4d problem, on an equidistant grid* 32^4 *(isotropy with* 923,521 *unknowns), left side defect vs. iterations, right side defect vs. CPU time.*

called the *main vector*) containing values for all grid points (red as well as black); the second vector has the values at the black points ejected out and replaced with zeros. Symmetrically, the third has the values at the red positions ejected out and replaced by zeros. This ejection process is actually where our injection operators (henceforth called *ejectors*) fit in.

First we construct the partition of the main vector storing the red and the black parts; then we carry out the first partial ω-Jacobi sweep by updating only the red part. This new red part along with the previously stored black part represents the main vector after the first partial sweep. Carrying out the second partial sweep in exactly the same manner, now for the black part instead, gives one ω-RB Jacobi iteration.

We present two *injection* operators, one for points of each color (even/odd category). We denote these *ejectors* by $\mathcal{E}_R$ and $\mathcal{E}_B$, with

$$\mathcal{E}_R = \left(\bigoplus_{i=1}^{d} \eta_{(d-i+1)} \right) \mod 2,$$

$$\mathcal{E}_B = (\mathcal{E}_R + 1) \mod 2.$$

$\bigoplus$ is the cumulative tensor sum of η_i, which counts the interior points along the ith dimension (see (3.9)); i has the reverse order (from d to 1) to match the lexicographic

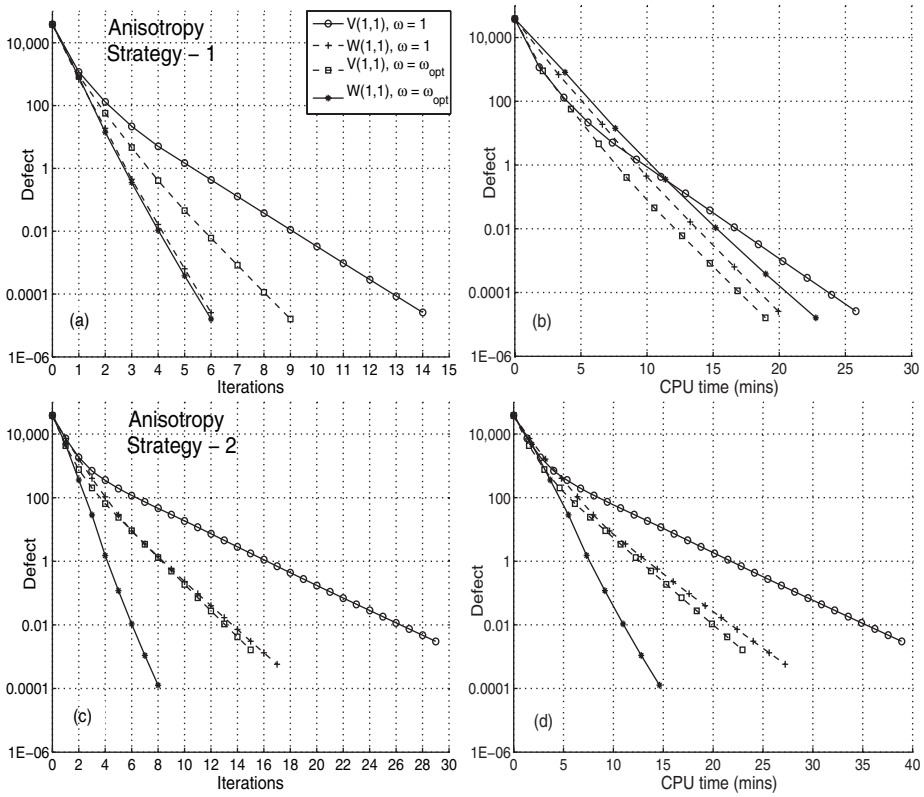

FIG. 4. *Convergence behavior of different multigrid cycles for a* 5*d problem, on a non-equidistant grid* [32 8 8 128 32] *(anisotropy with* 5,980,303 *unknowns), left side defect vs. iterations, right side defect vs. CPU time.*

layout of the grid points. Due to space limitations we can provide only a $2d$ example, although this formulation is true in general for an abstract higher dimension d.

Example 2. Consider a $2d$ grid $\mathbf{G} = [4, 5]$. In all we have 12 interior points, which, when counted in the lexicographic order, appear as follows:

$$\mathbf{u} = [u_{11}\ \ u_{12}\ \ u_{13}\ \ u_{14}\ \ u_{21}\ \ u_{22}\ \ u_{23}\ \ u_{24}\ \ u_{31}\ \ u_{32}\ \ u_{33}\ \ u_{34}]^T.$$

Evidently, $\boldsymbol{G}$, the collection of all of the points, has the following partition:

$$\begin{aligned} \boldsymbol{G}_R &= \{u_{11}, u_{13}, u_{22}, u_{24}, u_{31}, u_{33}\}, \\ \boldsymbol{G}_B &= \{u_{12}, u_{14}, u_{21}, u_{23}, u_{32}, u_{34}\}. \end{aligned}$$

Therefore according to our scheme

$$\begin{aligned} \mathbf{u}_R &= [\ \ u_{11}\ \ 0\ \ u_{13}\ \ 0\ \ 0\ \ u_{22}\ \ 0\ \ u_{24}\ \ u_{31}\ \ 0\ \ u_{33}\ \ 0\ \ ]^T, \\ \mathbf{u}_B &= [\ \ 0\ \ u_{12}\ \ 0\ \ u_{14}\ \ u_{21}\ \ 0\ \ u_{23}\ \ 0\ \ 0\ \ u_{32}\ \ 0\ \ u_{34}\ \ ]^T, \end{aligned}$$

and η_i in this case would be

$$\eta_1 = [1\ \ 2\ \ 3\ \ 4]^T \quad \text{and} \quad \eta_2 = [1\ \ 2\ \ 3]^T,$$

which leads to

$$\begin{aligned} \mathcal{E}_R &= (\eta_2 \oplus \eta_1) \mod 2 \\ &= [2\ 3\ 4\ 5\ 3\ 4\ 5\ 6\ 4\ 5\ 6\ 7]^T \mod 2 \\ &= [0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1]^T \\ \therefore \quad \mathcal{E}_B &= [1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0]^T. \end{aligned}$$

These red and black point ejectors now can be used to partition the grid as described. Once this partition is obtained, ω-RB Jacobi relaxation sweeps are trivial to perform, in that they are no different than their $2d$ counterparts.

REFERENCES

[1] G. Beylkin and M.J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159.

[2] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] J. Elf, P. Lötstedt, and P. Sjöberg, *Problems of high dimension in molecular biology*, in Proceedings of the 19th GAMM-Seminar Leipzig, Max-Planck-Institute, Leipzig, 2003, pp. 21–30.

[4] S.R. Elias, G.D. Stubley, and G.D. Raithby, *An adaptive agglomeration method for additive correction multigrid.*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 887–903.

[5] G. Horton and S. Vandewalle, *A Space-Time Multigrid Method for Parabolic PDEs*, Technical report 6/93, IMMD 3, Universitaet Erlangen, 1993.

[6] Y.K. Kwok, *Mathematical Models of Financial Derivatives*, 2nd ed., Springer Finance, Springer, Singapore, 1998.

[7] J. Larsson, F.S. Lien, and E. Yee, *Conditional semicoarsening multigrid algorithm for the poisson equation on anisotropic grids*, J. Comput. Phys., 208 (2005), pp. 368–383.

[8] C. Reisinger and G. Wittum, *On multigrid for anisotropic equations and variational inequalities*, Comput. Vis. Sci., (2004).

[9] C. Reisinger and G. Wittum, *Efficient Hierarchical Approximation of High-Dimensional Option Pricing Problems*, SIAM J. Sci. Comput., 29 (2007), pp. 440–458.

[10] H.W. Steeb, *Kronecker Product of Matrices and Applications*, Wissenschaftsverlag, 1991.

[11] K. Stüben and U. Trottenberg, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications*, Springer, Berlin, 1982.

[12] C. Reisinger and G. Wittum, *On the Construction of Fast Solvers for Elliptic Equations*, Technical report, von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, 1982.

[13] C.A. Thole and U. Trottenberg, *Basic smoothing procedures for the multigrid treatment of elliptic 3-d operators.*, Appl. Math. Comput., 19 (1986), pp. 333–345.

[14] U. Trottenberg, C.W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, New York, 2001.

[15] P. Wesseling, *An Introduction to Multigrid Methods*, Pure Appl. Math., John Wiley & Sons, New York, 1992.

[16] R. Wienands, *Extended Local Fourier Analysis for Multigrid: Optimal Smoothing, Coarse Grid Correction, and Preconditioning*, Ph.D. thesis, University of Cologne, 2001.

[17] R. Wienands and W. Joppich, *Practical Fourier Analysis for Multigrid Methods*, Numerical Insights, Chapman and Hall/CRC, Boca Raton, FL, 2005.

[18] I. Yavneh, *Multigrid smoothing factors for red-black Gauss–Seidel relaxation applied to a class of elliptic operators*, SIAM J. Numer. Anal., 32 (1995), pp. 1126–1138.

[19] I. Yavneh, *On red-black SOR smoothing in multigrid*, SIAM J. Sci. Comput., 17 (1996), pp. 180–192.

[20] Z. You-lan, W. Xiaonan, and C. I-Liang, *Derivative Securities and Difference Methods*, Springer Finance, Springer Science+Business Media Inc. USA, New York, 2004.

[21] H. Yserentant, *Sparse grid spaces for the numerical solution of the electronic Schrödinger equation*, Numer. Math., 101 (2005), pp. 381–389.