

## MULTIGRID PRECONDITIONERS FOR BI-CGSTAB FOR THE SPARSE-GRID SOLUTION OF HIGH-DIMENSIONAL ANISOTROPIC DIFFUSION EQUATION

H. BIN ZUBAIR\*, C.C.W. LEENTVAAR\*\* AND C.W. OOSTERLEE\*\*\*

**ABSTRACT.** Robust and efficient solution techniques are developed for *high-dimensional* parabolic partial differential equations (PDEs). Presented is a solver based on the Krylov subspace method Bi-CGSTAB preconditioned with  $d$ -multigrid. Developing *the perfect multigrid method*, as a stand-alone solver for a single problem discretized on a particular grid, often requires a lot of optimal tuning and expert insight; on the other hand Krylov-subspace based methods are robust but much less efficient unless used in combination with a very suitable preconditioner. The preconditioner that we employ is  $d$ -multigrid. We aim for a robust combination of the two so that it results in a solver that converges well for a wide class of discrete problems arising from discretization on various anisotropic grids. This is exactly what we encounter during the sparse grid solution of a high-dimensional problem. Different multigrid components are discussed and presented with operator construction formulae (in abstract  $d$  dimensions). We also present convergence diagrams for various multigrid (solvers and preconditioners) that we develop in this work, and explain their applicability.

*Key words* : Multigrid preconditioning, high dimensional PDE, anisotropic diffusion equation, point-smoothing methods, coarsening strategies, preconditioned Bi-CGSTAB, sparse-grids.

*AMS SUBJECT*: 65M55,65Y20, 91B28.

### 1. INTRODUCTION

Multidimensional PDEs constitute the mathematical model of a number of applied problems stemming from financial engineering [18, 10], quantum mechanics [2, 19] and molecular life-sciences [6], to mention just a few. In the context of their numerical approximation the big limiting factors on the

---

\* Corresponding author. Numerical Analysis Group, Delft Institute of Applied Mathematics. Delft University of Technology, The Netherlands E-mail: h.binzubair@gmail.com, \*\* E-mail: c.c.w.leentvaar@ewi.tudelft.nl, \*\*\* E-mail: c.w.oosterlee@tudelft.nl.

computing speed include, the requirement of successive solutions in time, and the *high-spatial-dimensionality* of the problem which renders an exponential computational complexity on regular tensor product grids. Traditionally, this exponential growth in the number of discrete unknowns is known as the *curse of dimensionality* [1], and it has continually eluded and marred full-grid based solution techniques. The *sparse-grid* solution method [4, 8, 20] relieves this so-called curse to some extent; it is based on discretizing the problem on many grids, each with lesser number of nodes (sparse grids), solving these subproblems and combining the solutions by interpolating them to the region of interest. This way we obtain a *mimic* of the solution on the original *dense* grid. Different combination techniques can be employed; see [8] for the technique that we use.

Efficiency of the sparse grid solution method, depends on the efficient solution of the underlying subproblems, each of which has the same spatial dimensionality as the original problem and which therefore requires an efficient and robust solution approach. The method that we develop and implement in this work, is  $d$ -multigrid preconditioned Bi-CGSTAB. In what follows  $d$ -multigrid refers to multigrid for an arbitrary number of space dimensions, which in turn is abbreviated by the letter  $d$ .

Bi-CGSTAB [16] is a well known iterative solver. It is generally accepted that all iterative solvers based on Krylov subspaces require preconditioning for faster convergence. Preconditioning is a process aimed at clustering the scattered eigenvalues of the coefficient matrix. It is important to point out that the performance of stand-alone multigrid is dependent significantly on the choice of optimal parameters and components, especially for high-dimensional problems; this is not quite the case when multigrid is used as a preconditioner. By choosing multigrid as a Bi-CGSTAB preconditioner we do not need to search for and harness optimal multigrid attributes (which are usually problem and discretization specific) and still reach near optimal efficiency. Important theoretical and experimental insights into multigrid preconditioning of Krylov subspace solvers can be had from earlier work in this context [11, 17, 7].

In Section 2 we define our model problem, viz; the  $d$ -dimensional diffusion equation, which forms the prototype of many applied parabolic problems and thus underscores the need of an efficient solver for discrete diffusion systems. We give the discretization of the continuous model problem with second order finite differences along with implementation in  $d$ -dimensions through Kronecker-tensor products. Next, in Section 3 we present the sparse grid technique along with the computation of accuracy bounds. Section 4 deals with the preconditioner and its components which include point smoothing and grid transfer strategies. These strategies are based on the idea of repeated partial coarsening in the direction(s) of strong coupling. In Section 5 we present experimental results based on the full-grid solution method. This includes results for  $d$ -multigrid employed both as a stand-alone solver as well

as a preconditioner and demonstrates the utility of the preconditioner. Finally we draw some conclusions from this work in Section 6.

## 2. THE MODEL PROBLEM AND DISCRETIZATION

In what follows  $\mathbf{x}$  is a  $d$ -tuple  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ . For  $\{a_i, b_i, c_i, \tau_1, \tau_2\} \in \mathbb{R}$  and  $c_i > 0$  the model problem reads:

$$\frac{\partial}{\partial t} u(\mathbf{x}, t) = \sum_{i=1}^d c_i \frac{\partial^2}{\partial x_i^2} u(\mathbf{x}, t); \quad \mathbf{x} \in \left( \Omega = \prod_{i=1}^d [a_i, b_i] \right) \subset \mathbb{R}^d; \quad t \in [\tau_1, \tau_2]; \quad (1)$$

$$u(\mathbf{x}, t) = f^\Gamma(\mathbf{x}, t); \quad \mathbf{x} \in \Gamma = \partial\Omega; \quad x_i \in \{a_i, b_i\}.$$

The spatial discretization of the model problem (1) is  $O(\sum_{i=1}^d (h_i^2))$  finite difference  $(2d + 1)$  stencil,  $h_i$  is the mesh size along the  $i^{\text{th}}$  space dimension. We choose the implicit (second order) Crank-Nicolson time stepping scheme for the temporal discretization. For  $k$  being the size of the time-step with  $d = 2$  this yields the following iterative representation in stencil notation:

$$\begin{bmatrix} -\frac{c_1}{2h_1^2} & \left\{ \frac{c_1}{h_1^2} + \frac{c_2}{h_2^2} + \frac{1}{k} \right\} & -\frac{c_1}{2h_1^2} \\ & -\frac{c_2}{2h_2^2} & \end{bmatrix} u_{\mathbf{h},k}(x_1, x_2, t + k) = \begin{bmatrix} \frac{c_1}{2h_1^2} & \left\{ -\frac{c_1}{h_1^2} - \frac{c_2}{h_2^2} + \frac{1}{k} \right\} & \frac{c_1}{2h_1^2} \\ & \frac{c_2}{2h_2^2} & \end{bmatrix} u_{\mathbf{h},k}(x_1, x_2, t) \quad (2)$$

We have Dirichlet boundary conditions prescribed at all boundaries of the spatial *hyper* domain. Inclusion of the boundary grid coordinates in the grid-point enumeration scheme (the non-eliminated boundary scheme) results in  $M$  grid points. The spatial discretization grid is given by  $\mathbf{N} = [N_1, N_2, \dots, N_d]$ ,  $N_i$  represents the number of divisions along the  $i^{\text{th}}$  dimension, and the total number of points in the (finest) grid is  $M = \prod_{i=1}^d (N_i + 1)$ . We represent the index of a grid-point by a  $d$ -tuple  $(j_d, j_{(d-1)}, \dots, j_1)$ , which is the  $d$ -dimensional extension of the 2 dimensional lexicographic enumeration scheme. Written in terms of matrix operators, the  $d$ -dimensional representation of (2) is:

$$(\mathbf{L}_h + \mathbf{D}_k)u_{\mathbf{h},k}(\mathbf{x}, t + k) = (-\mathbf{L}_h + \mathbf{D}_k)u_{\mathbf{h},k}(\mathbf{x}, t) \quad (3)$$

The matrix-operators have the order  $(M \times M)$ .  $\mathbf{D}_k$  is a diagonal matrix containing  $\frac{1}{k}$  on the main diagonal, and  $\mathbf{L}_h$  is the iteration matrix obtained in

the following way:

$$\mathbf{L}_h = \mathbf{X}_h + \mathbf{B}_h \quad (4)$$

$$\mathbf{X}_h = \sum_{i=1}^d \left\{ \bigotimes_{j=i}^{d-1} \mathbf{I}_{(d+i-j)} \otimes \mathbf{X}_i \otimes \bigotimes_{j=1}^{i-1} \mathbf{I}_{(i-j)} \right\} \quad (5)$$

The operator  $\mathbf{L}_h$  is the spatial operator consisting of the *interior point operator*  $\mathbf{X}_h$  and the *boundary point operator*  $\mathbf{B}_h$ .  $\mathbf{X}_i$  is the difference operator matrix (of order  $N_i + 1$ ) obtained by substituting 0 for the boundary grid-points and applying the following Crank-Nicolson difference stencil to all the interior grid-points along the  $i^{\text{th}}$  dimension:

$$\frac{c_i}{h_i^2} \begin{bmatrix} -\frac{1}{2} & \underline{1} & -\frac{1}{2} \end{bmatrix} \quad (6)$$

$\mathbf{I}_i$  is a diagonal matrix of order  $(N_i + 1)$  containing 1 on the main diagonal, except at the first and the last positions (boundary positions) where it is 0.

Finally, we represent the Crank Nicolson iteration matrix as  $\mathbf{A} = \mathbf{L}_h + \mathbf{D}_k$ . Different grid realizations of this matrix are used at every grid level during the multigrid process. Kronecker-tensor-products are employed in this work for defining the  $d$ -dimensional operators. They are non-commutative and associative operations (see [13]). In the formulae presented above  $\otimes$  is the Kronecker-tensor-product of matrices and  $\bigotimes$  is the cumulative Kronecker-tensor-product in the same sense as the cumulative sum  $\Sigma$ , or the cumulative product  $\Pi$ . The commutative order is determined by the subscripts and the associative hierarchy is immaterial. This completes the discussion of the discretization issues that arise from the arbitrary spatial dimensionality of the model problem.

### 3. THE SPARSE GRID METHOD (*A brief overview*)

Consider the task of the numerical approximation of a parabolic  $d$  dimensional problem discretized with  $N = 2^n$  points per spatial coordinate and  $m$  time steps. The grid thus formed is termed as a *full-grid* and a full-grid based solution process involves (at the minimal), vectors of the size  $2^{n \cdot d}$ . For 6 space dimensions and only 32 divisions along each dimension, the storage cost is around 9 gigabytes per vector, and grows worse for increasing  $d$ . The sparse grid approach, developed by Zenger and co-workers [4, 20] is a technique that splits the full grid problem of  $N^d$  points up into layers of subgrids. Each subgrid represents a coarsening in several coordinates up to a minimal required number of points. In the so-called *sparse grid combination technique*, the partial solutions that are computed on these grids, are combined a-posteriori by interpolation to a certain point or region. A multi-index  $\mathcal{I}_d$  belonging to a  $d$ -dimensional grid is a collection of numbers  $n_i$ ,  $i = 1, \dots, d$ , which represents a  $d$ -dimensional grid with  $N_i$  grid points in coordinate  $i$ , with  $N_i = 2^{n_i}$ .

The sum of a multi-index  $|\mathcal{I}_d|$  is defined by:

$$|\mathcal{I}_d| = \sum_{i=1}^d n_i \quad (7)$$

Therefore, the multi-index  $\mathcal{I}_d$  of a full grid with  $N = 2^n$  points per coordinate reads  $\mathcal{I}_d = \{n, n, \dots, n\}$ , with  $|\mathcal{I}_d|$  being the layer number.

The full grid solution will be denoted by  $u_n^f$ ; the sparse grid solution after the combination will be denoted by  $u_n^c$  and the exact solution by  $u_E$ . Now, we can define [8] The combined sparse grid solution  $u_n^c$  corresponding to a full grid solution  $u_n^f$  reads

$$u_n^c = \sum_{k=n}^{n+d-1} (-1)^{k+1} \binom{d-1}{k-n} \sum_{|\mathcal{I}_d|=k} u_{\mathcal{I}_d}^f, \quad (8)$$

with  $u_{\mathcal{I}_d}^f$  being the solution of the problem on a grid with multi-index  $\mathcal{I}_d$  such that  $|\mathcal{I}_d|$  equals  $k$ . For sufficiently smooth functions, the sparse grid solution (for most practical purposes) can be used instead of the full grid solution. For a simple 2 dimensional case, the subgrids (as constructed by the sparse-grid scheme) are depicted in Figure 1, Diagrams (a) - (g). Note that the *shape of the stretch* in all these grids is different, which implies that in each of these subproblems we have a different grid induced anisotropy. If the subgrids are simply combined without any interpolation, which means that all the evaluated points in every sub-grid are added with the binomial coefficients (8), the number of points in the full grid with  $n_i = n$  reads  $N_f = (2^n)^d$ .

From equation (8) it follows that the number of problems to be solved in the sparse grid technique reads

$$Z_{n,d} = \sum_{k=n}^{n+d-1} \binom{k-1}{d-1} = \frac{n}{d} \binom{n+d-1}{d-1} - \frac{n-d}{d} \binom{n-1}{d-1} \quad (9)$$

Furthermore the number of points employed in a grid with  $|\mathcal{I}_d| = m$  reads

$$N_{|\mathcal{I}_d|=m} = 2^m. \quad (10)$$

Combining (9) and (10) results in the total number of points employed in the sparse grid technique

$$N_n^c = \sum_{k=n}^{n+d-1} N_{|\mathcal{I}_d|=k} \binom{k-1}{d-1} = \sum_{k=n}^{n+d-1} \binom{k-1}{d-1} 2^k \quad (11)$$

It is known that the error of the discrete solution from a second order finite difference discretization of the 2D Laplacian can be split [9] as

$$u_n^f - u_E = C_1(x_1, h_1)h_1^2 + C_1(x_2, h_2)h_2^2 + D(x_1, h_1, x_2, h_2)h_1^2h_2^2 \quad (12)$$

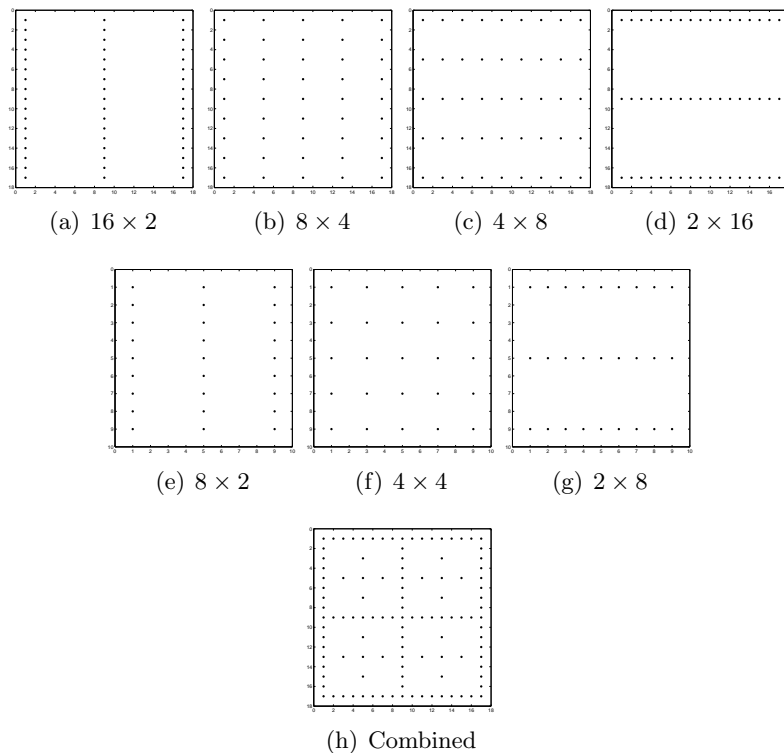


FIGURE 1. Construction of a 2D sparse grid; (a)–(d): grids on layer 5, (e)–(g): grids on layer 4; (h) combined sparse grid solution

With the combination technique as in Definition 3 and the splitting in (12), the dimension-dependent absolute error (for the Laplacian), reads, [5]:

$$\epsilon_n = |u_n^c - u_E| = \mathbf{O}(h_n^2 (\log_2 h_n^{-1})^{d-1}), \quad (13)$$

#### 4. THE $d$ -MULTIGRID PRECONDITIONER

We employ multigrid as a preconditioner for Bi-CGSTAB due to the robustness that the resulting solver offers. In this section we explore the preconditioner and browse through its various components. *Coarse grid correction* and *error smoothing* are two essential components of any multigrid algorithm. For anisotropic PDEs it is well known that full coarsening along with point smoothing does not work, as it does not sufficiently smooth the errors that have to be approximated on the coarse grid. To address anisotropy, the standard way in 2-dimensions is either to coarsen along only 1 dimension (the one where error components are strongly coupled) or else to do a full coarsening but to resort to *line-relaxation* along the strongly coupled dimension [15].

These techniques can be extended to arbitrary higher  $d$ . A relaxation method based on *hyperplane* relaxation has been proposed in [12], which is analogous to *line-relaxation* in 2 dimensions. Contrary to that approach, we proposed a method based on *point smoothing* and *partial coarsening* schemes in [3]. There we treat discrete anisotropies induced by non-equidistant grids. The proposal is to employ partial coarsening so that coarsening only takes place along those directions that have a strong coupling. We extend this technique to the general situation, where anisotropies may stem from two different sources, viz; the mesh size and the presence of constant anisotropic coefficients in the continuous problem.

**4.1. The Relaxation Method.** Point-wise Red-Black Gauss-Seidel (GSRB) is a standard method used for relaxation in the context of multigrid [14, 15]. We employ this in our multigrid preconditioner due to its excellent smoothing properties for elliptic equations. The first step in this process is the partitioning of the grid  $\mathbf{G}$  into the *red* part ( $\mathbf{G}_R$ ), and the *black* part ( $\mathbf{G}_B$ ). Once this partition has been obtained, GSRB for a  $d$ -dimensional setting is no different from its 2-dimensional counterpart. Let the index set containing the grid-points in  $\mathbf{G}_R$  be represented by  $\mathcal{I}_R$  and that containing the grid-points in  $\mathbf{G}_B$  be represented by  $\mathcal{I}_B$ . In the standard literature [14, 15], the term *Red* refers to odd and *Black* to even points. The distinction of even and odd for a grid-point  $(j_d, j_{(d-1)}, \dots, j_1)$  is based on the following rule:

$$\text{Even if: } \text{mod}\left(\sum_{i=1}^d j_i, 2\right) = 0; \quad \text{Odd if: } \text{mod}\left(\sum_{i=1}^d j_i, 2\right) = 1;$$

From an implementational aspect, it is more convenient to adjust this definition, fixing *Red* as the category of the first unknown in the grid, which toggles between even and odd with the increase in  $d$  (for Dirichlet type of boundaries). Thus, we assert that we carry out a *Red-Black Gauss-Seidel* for all  $d$ , even if from the point of view of the above definition, we do an *even – odd* in 2-dimensions, *odd – even* in 3, and so on.

**4.2. Coarsening Strategies to Handle Anisotropies.** We use two grid coarsening strategies based on partial grid transfers to handle the anisotropies in the discretized system. Anisotropies can appear in the discrete system either through the presence of constant anisotropic coefficients in the continuous problem or due to unequal mesh sizes along different dimensions of the domain. These causative factors lead to a single coefficient  $\epsilon_i$  for each dimension, viz  $\epsilon_i = \frac{c_i}{h_i^2}$  with  $h_i = \frac{b_i - a_i}{N_i}$ . See (1). Fourier Analysis suggests that all coefficients within a factor  $< \pm 1.3$  of the maximum coefficient) be considered equivalent for the purpose of partial doubling, i.e. the grid may be halved all along such dimensions *together*. However, as we experiment both with *doubling* ( $h \rightarrow 2h$ ) as well as with *quadrupling* ( $h \rightarrow 4h$ ) partial transfers; we would like to point out that for partial quadrupling the rule is much more strict.

There we pick the maximum coefficient and only do a quadrupling transfer along the dimensions having a coefficient equal to this max. Whenever we require full coarsening, we always resort to full doubling as full quadrupling hampers multigrid convergence.

The application problem under the sparse grid solution method gives -in particular- subproblems where anisotropies originate from the unequal mesh-sizes along the dimensions. For clarity, we illustrate both the doubling and the quadrupling based coarsening strategies through the following simple example; where the anisotropy is only grid-based.

Example: The grid for a 5 dimensional problem is  $\mathbf{N} = [128 \ 4 \ 16 \ 16 \ 64]$ .  $c_i = c$  for  $i = \{1, 2, \dots, d\}$  and  $\Omega = [a, b]^d$ . Then the grid is coarsened in the following way;

Strategy 1, doubling (h $\rightarrow$ 2h)	Strategy 2, quadrupling (h $\rightarrow$ 4h)
$\Omega_6 = [ \ 128 \ 4 \ 16 \ 16 \ 64 \ ]$	$\Omega_4 = [ \ 128 \ 4 \ 16 \ 16 \ 64 \ ]$
$\Omega_5 = [ \ 64 \ 4 \ 16 \ 16 \ 64 \ ]$	$\Omega_3 = [ \ 64 \ 4 \ 16 \ 16 \ 64 \ ]$
$\Omega_4 = [ \ 32 \ 4 \ 16 \ 16 \ 32 \ ]$	$\Omega_2 = [ \ 16 \ 4 \ 16 \ 16 \ 16 \ ]$
$\Omega_3 = [ \ 16 \ 4 \ 16 \ 16 \ 16 \ ]$	$\Omega_1 = [ \ 4 \ 4 \ 4 \ 4 \ 4 \ ]$
$\Omega_2 = [ \ 8 \ 4 \ 8 \ 8 \ 8 \ ]$	$\Omega_0 = [ \ 2 \ 2 \ 2 \ 2 \ 2 \ ]$
$\Omega_1 = [ \ 4 \ 4 \ 4 \ 4 \ 4 \ ]$	
$\Omega_0 = [ \ 2 \ 2 \ 2 \ 2 \ 2 \ ]$	

(14)

**4.3. Coarse-grid Discretization.** An important component in the coarse grid correction process is the choice of the coarse-grid operator  $L_H$ . We use the coarse-grid analog of the discrete operator on the fine-grid. Once the next coarser-grid is decided we build the operator using the same scheme as in Section 2.

Another option is to use the Galerkin operator. Some especial transfer operators (in 1 and 2 dimensions) can be employed to generate a relatively sparse Galerkin operator [14] but as of yet it is unknown how this kind of transfer might be extended to abstract  $d$ -dimensions. A significant disadvantage of employing the Galerkin operator (constructed with the usual transfer operators) is its being much more dense than the coarse-grid analog of the fine-grid operator. This issue becomes more serious with increasing  $d$ .

**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 [15] that a  $2d$  FW restriction operator is the Kronecker tensor product of the following

$x_1$  and  $x_2$  directional 1-dimensional FW operators:

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

The following formula -based on Kronecker tensor products- gives a FW restriction operator matrix  $\mathbf{R}$  (for the non-eliminated boundary scheme). It unifies doubling and quadrupling transfers into the following compact form:

$$\mathbf{R} = \prod_{i=1}^d (\mathbf{R}_i)^{t_i}, \quad (15)$$

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

We now define the quantities involved in (15) for the dummy subscript  $a$ .  $\mathbf{I}_a$  is a diagonal matrix of order  $(a+1) \times (a+1)$ . The diagonal entry is 1, except at the first and the last positions where it is 0.  $\mathbf{O}_a$  is the 1d FW restriction operator matrix, order =  $(\frac{a}{2} + 1) \times (a+1)$ , obtained by applying the 1d transfer stencil at the interior points and taking 0 at the boundaries.  $\mathbf{N} = [N_1, N_2, \dots, N_d]$ , is the grid description.  $\mathbf{T} = [t_1, t_2, \dots, t_d]$  is the coarsening request,  $t_i$  is the count of ( $h \rightarrow 2h$ ) transfers along the  $i^{\text{th}}$  dimension. *We say that quadrupling takes place along the  $i^{\text{th}}$  dimension if  $t_i = 2$ .* It is trivial to verify this formula 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:

$$\mathbf{P} = 2^{\sum_{i=1}^d t_i} (\mathbf{R}^T). \quad (16)$$

$\mathbf{R}^T$  is the transpose of the restriction operator matrix  $\mathbf{R}$ . With this general transfer operator we can experiment with different transfers based on doubling and quadrupling, depending on the anisotropy of the discrete system. This FW restriction operator provides the required matrix for any number of coarsenings along any number of dimensions for an abstract  $d$ -dimensional problem. The stage is all set now to experiment with the preconditioner and check out its utility and efficiency.

## 5. NUMERICAL EXPERIMENTS

In this work *full-grid solution* refers to a solution on a regular tensor-product grid (where the sparse grid technique is not used). As indicated in Section 3 the sparse-grid solution technique gives rise to a number of sub-problems each leading to a discrete anisotropic  $d$ -dimensional linear system. It is the efficient solution of these smaller sized full-grid problems that is responsible for the overall efficiency of the sparse grid method. As described in the previous section, we have quite a strong and robust multigrid preconditioner. In this

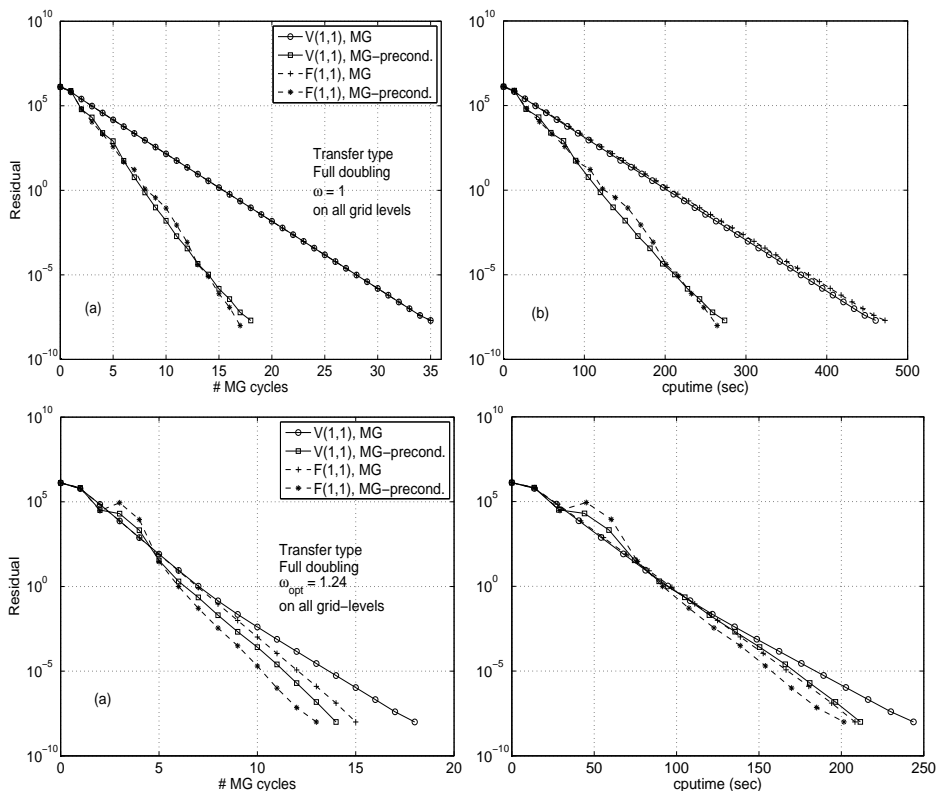


FIGURE 2. Convergence diagram for a 5 dimensional isotropic problem, 32 divisions along each dimension.

section we test its performance as a stand-alone solver versus as a preconditioner for Bi-CGSTAB in a full-grid solution process.

**5.1.  $d$ -Multigrid Performance (*Solver vs Preconditioner*).** A useful numerical insight for time-marching solution processes (our ultimate aim in this work) comes from an insight into the stationary process per time-step. Through the numerical solution of the PDE we approximate the following test function:

$$u(\mathbf{x}) = \frac{\sum_{i=1}^d \sin(d\pi^2 x_i)}{d\pi + \sum_{i=1}^d x_i} \tag{17}$$

We have conducted a number of numerical experiments both isotropic and anisotropic, and have included the convergence graphs for them. These graphs show the residual reduction against iteration and cputime for  $d$ -multigrid used in these two contexts (solver and preconditioner). A word of caution while examining these graphs is just in place. Multigrid is an  $O(M)$  solver (where  $M$  is the number of unknowns on the finest grid) when optimal relaxation and ideal coarse grid correction are available. In such a situation multigrid

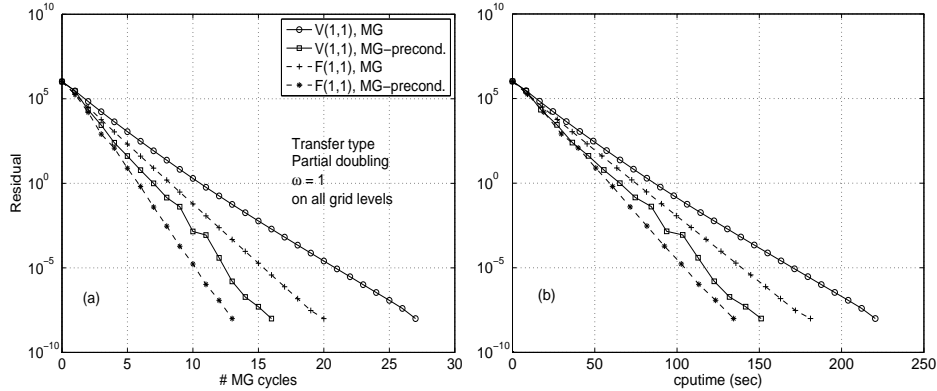


FIGURE 3. Convergence diagram for a 6 dimensional anisotropic problem, grid stretched along  $d/2$  dimensions and given by  $\mathbf{N} = [32, 8, 32, 8, 32, 8]$ . # of unknowns is 26,198,073. Diagram (a) shows a comparison between multigrid as a solver and multigrid as a preconditioner, on the iteration scale, Diagram(b) on the cputime scale.

is extremely efficient. Some of the graphs here show a tough competition between multigrid as a solver against multigrid as a Bi-CGSTAB preconditioner. This happens due to the fact that for the model-problem the employed relaxation method and the coarse grid correction form near-optimal  $d$ -multigrid attributes. Evidently, in any situation where tuning multigrid with optimal attributes is not a choice, multigrid works better as a Krylov-preconditioner than as a stand-alone solver.

First of all, we check out  $d$ -multigrid performance for a 5 dimensional isotropic case, with 32 divisions along all dimensions of the domain. The number of unknowns in the system is 39,135,393. In this case the V(1,1) multigrid (multigrid method based on V cycles with 1 pre and 1 post smoothing steps) preconditioned Bi-CGSTAB far out-performs the V(1,1) multigrid solver; Figure 2, (here we choose  $\omega = 1$  in  $\omega$ -GSRB) Diagrams (a) and (b). However, with  $\omega_{opt} = 1.24$  included in the game (a possibility for the model problem) the comparison is not as bright; Figure 2 Diagrams (c) and (d). This confirms that no great Krylov induced enhancement should be expected when multigrid (as a solver) approaches optimality.

Next we present experiments based on problems with *discrete anisotropies* that result from discretization on a non-equidistant grid, i.e. a grid where the number of divisions is different along different dimensions of the hyper domain. We have selected 3 high dimensional problems, each with a different discrete anisotropy. The problems have been chosen with the aim of harvesting experimental results for grids highly stretched along 1 dimension as well as grids highly stretched along multiple dimensions. The anisotropies are handled

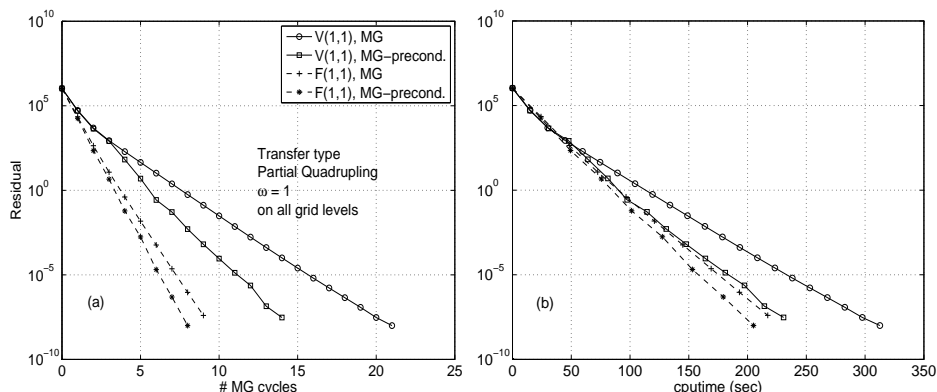


FIGURE 4. Convergence diagram for a 7 dimensional anisotropic problem, grid stretched along 1 dimension, and given by  $\mathbf{N} = [8, 8, 8, 64, 8, 8, 8]$ , # of unknowns is 34,543,665.

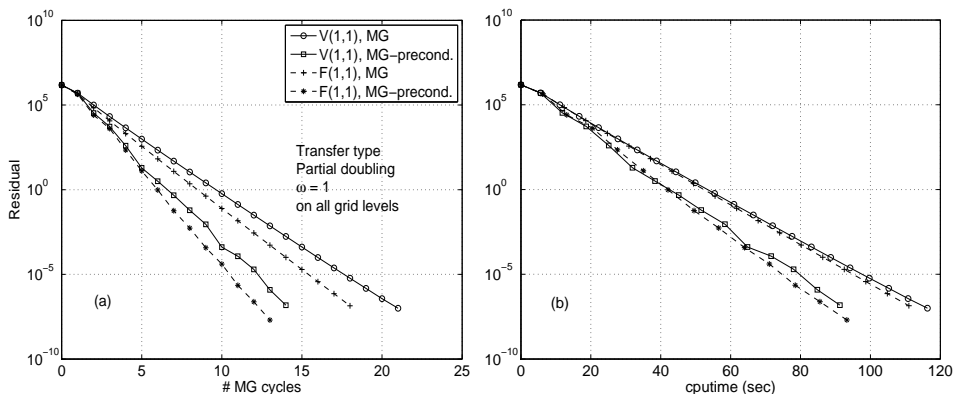


FIGURE 5. Convergence diagram for a 4 dimensional anisotropic problem, grid stretched along  $(d - 1)$  dimensions, and given by  $\mathbf{N} = [128, 128, 128, 8]$ , # of unknowns is 19,320,201

with the partial coarsening schemes as illustrated by the example in Section 4.2. The results appear in Figures 3, 4 and 5. Here, we find that the F(1,1) MG cycles are more suited than V(1,1), both for the stand-alone multigrid solver as well as a preconditioner for Bi-CGSTAB, if the coarsening strategy is based on doubling. Quadrupling suits the situation more when the grid is stretched along only a few dimensions, (preferably  $< d/2$ ) and when optimal relaxation parameters are available. However, with quadrupling, V(2,2) and F(1,1) cycles seem to yield better results than V(1,1).

In [3] we pointed out that for grids stretched along a single dimension, a method based on partial quadrupling as the coarsening strategy has an  $O(M)$  complexity even with  $W$  cycles; achieving this is not possible with methods based on partial doubling along 1 dimension. This fact makes it a strategy of choice for such grids, Figure 6. We have only chosen the cputime scale (for presenting results) because with different transfer operators, the number of cycles are not comparable. Quadrupling -in contrast with doubling- relies on optimal relaxation to quite some extent; in fact, the better the relaxation process the shorter the cputime. This points us to the fact that if optimization in the relaxation process is an impossibility we might be better off with doubling for all kinds of grid-based discrete anisotropies.

The multigrid convergence factors in all these experiments are quite low (around an average of 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.

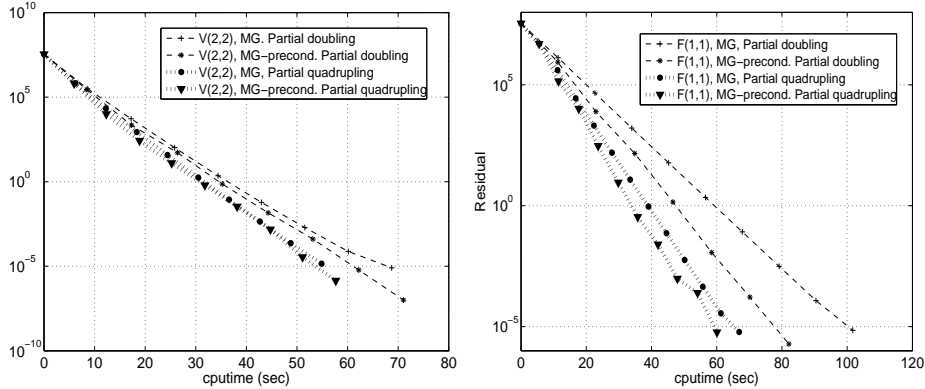


FIGURE 6. Convergence diagrams comparing  $d$ -multigrid based on doubling transfers against quadrupling transfers for V(2,2) -Diagram (a)- and F(1,1) -Diagram(b)- multigrid cycles. This 5 dimensional problem is discretized on  $\mathbf{N} = [8, 8, 2048, 8, 8]$ , # of unknowns is 13,443,489.

## 6. CONCLUSIONS

A promising technique to handle high dimensional PDE problems numerically is the sparse grid method; which gives rise to an abundant number of smaller sized problems on equidistant and non-equidistant grids. The non-equidistant grids generate a discrete anisotropy in the system. In the context of developing a  $d$ -multigrid method for these problems we have shown in this paper how these anisotropies can be treated by a suitable combination of partial coarsening strategies and point based relaxation. The partial coarsening

schemes are based on doubling and quadrupling transfers. It turns out that for isotropic systems, an F(1,1) cycle with/without an optimal relaxation parameters is a very nice combination for a multigrid method. For anisotropic problems, a V(1,1) method without optimal parameter(s) proves excellent. A speed-up can be had by employing partial quadrupling as the coarsening strategy for anisotropic problems provided that optimal relaxation parameters are accessible. For partial quadrupling, the V(2,2) and F(1,1) cycles with optimal relaxation parameters seem good.

Although it is well known that multigrid methods are amongst the fastest solvers for elliptic equations, the throughput of a multigrid solver usually depends on how best it could be tuned with optimal attributes which include optimal relaxation and an ideal coarse grid correction. It is often difficult to reach this optimality in a practical situation. To quite some extent this could be substituted by having multigrid as a preconditioner of a suitable Krylov subspace method, such as Bi-CGSTAB. We have shown in this paper how such a  $d$ -multigrid method may be set up and employed as a preconditioner, and supplemented the development with numerical experiments and convergence diagrams. The resulting PDE solver is quite robust and generally applicable to a wide class of discrete parabolic and elliptic problems. Further on, we plan to exploit the parallel features of this solver by automating this parallelism.

**ACKNOWLEDGMENTS:** This research has quite a few sponsors and interest partners. In parts, it is 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; by the Government of Pakistan through The HEC-Pakistan's research grants; by the Textile Institute of Pakistan; and by the Dutch Technology foundation STW. We would like to express our thanks to all these sponsors.

#### REFERENCES

- [1] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, New Jersey, 1961.
- [2] G. Beylkin and J.M. Martin. Algorithms for numerical analysis in high dimensions. *SIAM J. Sci. Comp.*, 26(6): 2133–2159, 2005.
- [3] H. bin Zubair, C.W. Oosterlee, and R. Wienands. Multigrid for high dimensional elliptic partial differential equations on non-equidistant grids. Technical report, Delft University of Technology, 2006.
- [4] H.J. Bungartz and M. Griebel. *Sparse Grids*. *Acta Numerica*, pages 147–269, May 2004.
- [5] H.J. Bungartz, M. Griebel, D. Rösche, and C. Zenger. Pointwise convergence of the combination technique for the Laplace equation. *East-West J. Numer. Math.*, 2:21–45, 1994.
- [6] J. Elf, P. Lötstedt, and P. Sjöberg. Problems of high dimension in molecular biology. In *Proceedings of the 19<sup>th</sup> GAMM-Seminar Leipzig*, pages 21–30, 2003.

- [7] Y.A. Erlangga, C.W. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous helmholtz problems. *SIAM J. Sci. Comput.*, 27: 1471–1492, 2006.
- [8] M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In *Proceedings of the IMACS International Symposium on Iterative Methods in Linear Algebra*, pages 263–281. Elsevier, Amsterdam, 1992.
- [9] P. Hofmann. Asymptotic expansions of the discretization error of boundary value problems of the laplace equation in rectangular domains. *Numerische Mathematik*, 9:302–322, 1967.
- [10] Y. K. Kwok. *Mathematical models of financial derivatives*. Springer Finance. Springer-Verlag, Singapore, second edition, 1998.
- [11] C.W. Oosterlee and T. Washio. An evaluation of parallel multigrid as a solver and as a preconditioner for singularly perturbed problems. *SIAM J. Sci. Comput.*, 19: 87–110, 1998.
- [12] C. Reisinger and G. Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. Technical report, Oxford University, 2006.
- [13] H.W. Steeb. *Kronecker Product of Matrices and Applications*. Wissenschaftsverlag, 1991.
- [14] K. Stüben and U. Trottenberg. *Multigrid Methods: fundamental algorithms, model problem analysis and applications*. Springer Berlin, 1982.
- [15] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.
- [16] H. A. Van Der Vorst. A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13: 631–644, 1992.
- [17] R. Wienands, C.W. Oosterlee, and T. Washio. Fourier analysis of gmres( $m$ ) preconditioned by multigrid. *SIAM J. Sci. Comput.*, 22: 582–603, 2000.
- [18] Z. You-lan, W. Xiaonan, and C. I-Liang. *Derivative Securities and Difference Methods*. Springer Finance. Springer Science+Business Media Inc. USA, 2004.
- [19] H. Yserentant. Sparse grid spaces for the numerical solution of the electronic schrödinger equation. *Numer. Math.*, 101: 381–389, 2005.
- [20] C. Zenger. Sparse grids. In *Proceedings of the 6th GAMM seminar, Notes on numerical fluid mechanics*, volume 31, 1990.