On Multigrid for High-Dimensional Anisotropic Partial Differential Equations

H. bin Zubair*
Numerical Analysis Group, Delft Institute of Applied Mathematics
Delft University of Technology, The Netherlands

Abstract

Robust and efficient solution techniques are developed for *high-dimensional* elliptic partial differential equations (PDEs). High dimensional equations are often treated by the sparse-grid technique which gives rise to a number of smaller sized problems discretized on non-equidistant grids. Such discretization induces a discrete anisotropy in the system. We develop a general and robust multigrid method for high dimensional anisotropic equations with general anisotropies as well as discrete grid-induced anisotropies. Presented in the paper are d-dimensional multigrid components and operators based on Kronecker-tensor products of low dimensional operators. We show how optimal relaxation parameters can be computed for the model d-dimensional problem. Multigrid efficiency in d-dimensions depends on optimal relaxation and ideal coarse-grid correction to quite some extent, and (as opposed to the model problem realm) there exist applications where these optimal attributes may not be accessible. In such a situation, we employ our d-multigrid method as a Bi-CGSTAB preconditioner instead of a stand-alone solver, and demonstrate that this is a close substitution of optimality in the multigrid process. The resulting solver converges well for a wide class of discrete high dimensional problems.

Keywords: d-Multigrid, Multigrid preconditioning, high dimensional PDEs, point-smoothing methods, coarsening strategies, sparse-grids.

AMS Subject Classification: 65M55,65Y20, 91B28

1 Introduction

Multidimensional PDEs constitute the mathematical model of a number of applied problems stemming from financial engineering [8, 10], quantum mechanics [1, 15] and molecular life-sciences [4], to mention just a few. In the context of their numerical approximation the big limiting factors on the 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. Tradionally, this exponential growth in the number of discrete unknowns is known as the *curse of dimensionality*, and it has continually elluded and marred full-grid based solution techniques. The *sparse-grid* solution method [6, 16] 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.

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. 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 [13] 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 [9, 5].

In Section 2 we define our model problem, viz; the d-dimensional stationary diffusion equation, which forms the worst-case prototype of discrete parabolic problems (within one time-step) and thus underscores the need of an efficient solver for discrete elliptic equations. 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

*Correspondence email: h.binzubair@tudelft.nl

d-multigrid 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 convergence results for d-multigrid employed both as a stand-alone solver as well as a preconditioner. In Section 6 we demonstrate the utility of d-multigrid by employing it in the sparse-grid solution process. Finally we draw some conclusions from this work in Section 7.

2 The Model Problem and Discretization

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

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

The discretization of the model problem (1) is $O(\Sigma_{i=1}^{d}(h_i^2))$ finite difference $(2d+1)$ stencil, h_i is the mesh size along the i^{th} space dimension. For k being the size of the time-step with $d = 2$ this yields the following discrete representation in stencil notation:

$$\begin{bmatrix} & -\frac{c_2}{h_2^2} & \\ -\frac{c_1}{h_1^2} & \{\frac{2c_1}{h_1^2} + \frac{2c_2}{h_2^2}\} & -\frac{c_1}{h_1^2} \\ & -\frac{c_2}{h_2^2} & \end{bmatrix} u_{\boldsymbol{h}}(x_1, x_2) = f_{\boldsymbol{h}}^{\Omega}(x_1, x_2) \tag{2}$$

We have Dirichlet boundary conditions prescribed at all boundaries of the *hyper* domain. The discretization grid is given by $\boldsymbol{N} = [N_1, N_2, \cdots, N_d]$, N_i represents the number of divisions along the ith dimension. Inclusion of the boundary grid coordinates in the grid-point enumeration scheme (the non-eliminated boundary scheme) results in $M = \prod_{i=1}^{d}(N_i+1)$ grid points in the finest grid. We represent the index of a grid-point by a d-tuple $(j_d, j_{(d-1)}, \cdots, j_1)$, 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:

$$\boldsymbol{L}_h u_{\boldsymbol{h}}(\boldsymbol{x}) = f_{\boldsymbol{h}}^{\Omega}(\boldsymbol{x}) \tag{3}$$

The matrix-operators have the order $(M \times M)$. $\boldsymbol{L}_h$ is the matrix operator obtained in the following way:

$$\boldsymbol{L}_h = \boldsymbol{X}_h + \boldsymbol{B}_h \tag{4}$$

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

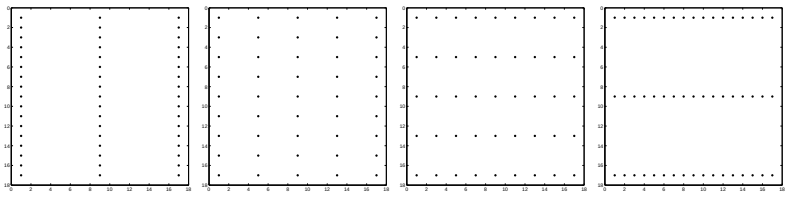
The operator $\boldsymbol{L}_h$ is the spatial operator consisting of the *interior point operator* $\boldsymbol{X}_h$ and the *boundary point operator* $\boldsymbol{B}_h$. $\boldsymbol{X}_i$ is the difference operator matrix (of order $N_i + 1$) obtained by substituting 0 for the boundary grid-points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

Kronecker-tensor-products are employed in this work for defining the d-dimensional operators. They are non-commutative and associative operations (see [11]). 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 Σ, or the cumulative product Π. 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.

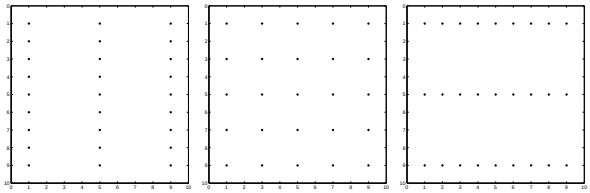
3 The Sparse Grid Method *(A brief overview)*

Consider the task of the numerical approximation of a parabolic d-dimensional problem discretized with $N = 2^n$ points per coordinate. 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 $2^{n.d}$. 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 [16] is a technique that splits the full grid problem of N^d points up into layers of subgrids. Each sub-grid 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 $n_i,\ i = 1, \ldots, d$, 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:

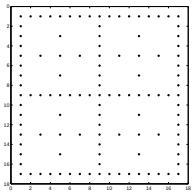
$$|\mathcal{I}_d| = \sum_{i=1}^{d} n_i \tag{6}$$



(a) 16×2 (b) 8×4 (c) 4×8 (d) 2×16



(e) 8×2 (f) 4×4 (g) 2×8



(h) Combined

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

Therefore, the multi-index $\mathcal{I}_d$ of a full grid with $N = 2^n$ points per coordinate reads $\mathcal{I}_d = \{n, n, \ldots, n\}$, 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 [6] The combined sparse grid solution u_n^c corresponding to a full grid solution u_n^f reads

$$u_n^c = \sum_{k=n}^{n+d-1} (-1)^{k+1} \binom{d-1}{k-n} \sum_{|\mathcal{I}_d|=k} u_{\mathcal{I}_d}^f, \tag{7}$$

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 (7), the number of points in the full grid with $n_i = n$ reads $N_f = (2^n)^d$.

From equation (7) it follows that the number of problems to be solved in the sparse grid technique reads

$$Z_{n,d} = \sum_{k=n}^{n+d-1} \binom{k-1}{d-1} = \frac{n}{d} \binom{n+d-1}{d-1} - \frac{n-d}{d} \binom{n-1}{d-1} \tag{8}$$

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

$$N_{|I_d|=m} = 2^m. \tag{9}$$

Combining (8) and (9) results in the total number of points employed in the sparse grid technique

$$N_n^c = \sum_{k=n}^{n+d-1} N_{|I_d|=k} \binom{k-1}{d-1} = \sum_{k=n}^{n+d-1} \binom{k-1}{d-1} 2^k \tag{10}$$

It is known that the error of the discrete solution from a second order finite difference discretization of the 2D Laplacian can be split [7] as

$$u_n^f - u_E = C_1(x_1, h_1)h_1^2 + C_1(x_2, h_2)h_2^2 + D(x_1, h_1, x_2, h_2)h_1^2 h_2^2 \tag{11}$$

With the combination technique as in Definition 3 and the splitting in (11), the dimension-dependent absolute error (for the Laplacian), reads, [3]:

$$\epsilon_n = |u_n^c - u_E| = \mathbf{O}(h_n^2 \left(\log_2 h_n^{-1}\right)^{d-1}), \tag{12}$$

4 *d*-Multigrid

In this section we explore d-multigrid 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 [12].

These techniques can be extended to arbitrary higher d. A relaxation method based on *hyperplane* relaxation has been proposed in [10], 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 [2]. 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 Point-based Relaxation and Optimal Parameters

Weighted (point-based) Red-Black Gauss-Seidel (ω-GSRB) is a standard method used for relaxation in the context of multigrid [12]. It has excellent smoothing in the context of elliptic equations. The first step in this process is the partitioning of the grid $\boldsymbol{G}$ into the *red* part ($\boldsymbol{G}_R$), and the *black* part ($\boldsymbol{G}_B$). Once this partition has been obtained, ω-GSRB for a d-dimensional setting is no different from its 2-dimensional counterpart. Let the index set containing the grid-points in $\boldsymbol{G}_R$ be represented by $\mathcal{I}_R$ and that containing the grid-points in $\boldsymbol{G}_B$ be represented by $\mathcal{I}_B$. In the standard literature [12], the term *Red* refers to odd and *Black* to even points. The distinction of even and odd for a grid-point $(j_d, j_{(d-1)}, \cdots, j_1)$ is based on the following rule:

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

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.

For the model problem, it is a possiblity to determine and use the optimal relaxation parameters ω_{opt} [14]. For an introduction into Local Fourier Smoothing Analysis, see [12]. In [2] we provide the theoretical details for the analysis of ω-GSRB, which enables us to approximate the optimal relaxation parameters for the d-dimensional case. In Table 1, these numbers are provided (upto 8 dimensions) only for the isotropic case and a particularly interesting anisotropic case, both for doubling and quadrupling, determined for 2 smoothing sweeps (one pre and one post). For space considerations the definition of the smoothing factor $\mu(\omega)$ is omitted here and [2] referred for the same.

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 ϵ_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 certain 5 dimensional problem is $\boldsymbol{N} = [\ 128\ \ 4\ \ 16\ \ 16\ \ 64\ ]$. $c_i = c$ for $i = \{1, 2, \cdots, d\}$ and

Table 1: $\mu(1)$, ω_{opt} and associated $\mu(\omega_{opt})$ for 2 relaxation sweeps, 1 pre and 1 post. Top : Isotropy (full transfer). Bottom: Grid induced anisotropy $\boldsymbol{N} = 128 \times 32^{(d-1)}$ partial transfer along 1 dimension.

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
7	0.73	1.3114	0.39	0.92	1.5536	0.58
8	0.76	1.3405	0.42	0.93	1.5761	0.60

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	0.9023	0.20	0.59	1.1986	0.32
3	0.23	0.9581	0.22	0.65	1.2410	0.35
4	0.24	1.0043	0.23	0.69	1.2761	0.37
5	0.31	1.0445	0.25	0.73	1.3059	0.39
6	0.37	1.0803	0.26	0.76	1.3318	0.41
7	0.41	1.1132	0.28	0.78	1.3544	0.43
8	0.46	1.1433	0.29	0.80	1.3746	0.44

$\Omega = [a, b]^d$. Then the grid is coarsened in the following way;

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

(13)

4.3 Coarse-grid Discretization and The Transfer Operators

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 but as of yet it seems unknown as to 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.

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 [12] 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:

$$\mathbf{R} = \prod_{i=1}^{d} (\mathbf{R}_i)^{t_i}, \tag{14}$$

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

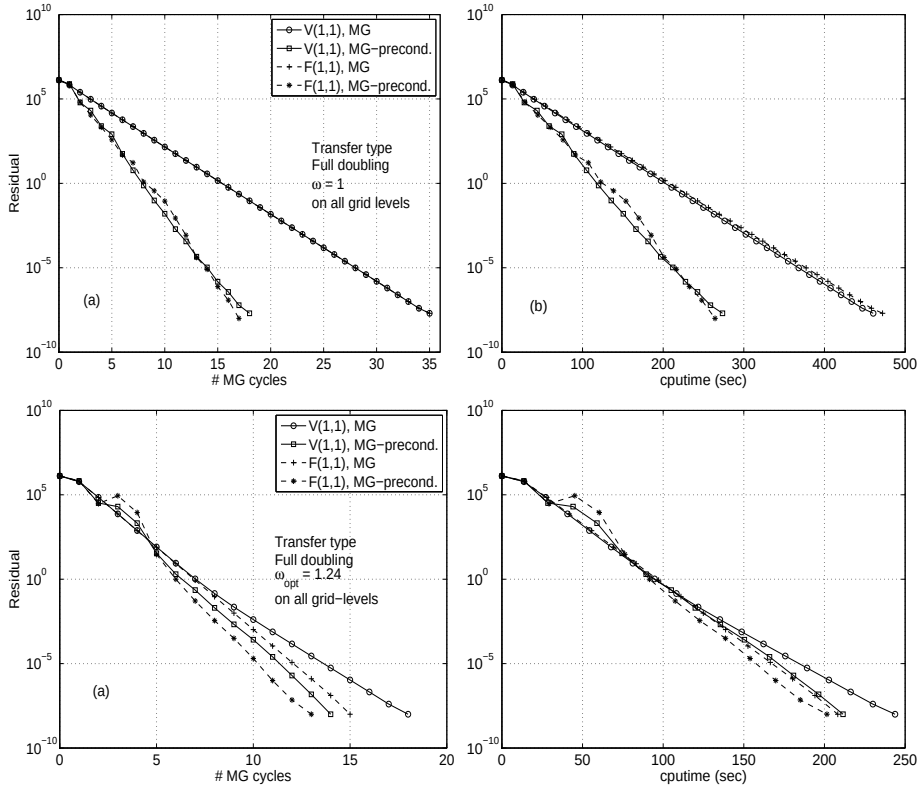


Figure 2: Convergence diagram for a 5 dimensional isotropic problem, 32 divisions along each dimension.

We now define the quantities involved in (14) 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 $= \left(\frac{a}{2}+1\right) \times (a+1)$, obtained by applying the 1d transfer stencil at the interior points and taking 0 at the boundaries. $\boldsymbol{N} = [N_1, N_2, \cdots, N_d]$, is the grid description. $\boldsymbol{T} = [t_1, t_2, \cdots, t_d]$ is the coarsening request, t_i is the count of $(h \rightarrow 2h)$ transfers along the i^{th} dimension. *We say that quadrupling takes place along the i^{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:

$$\mathbf{P} = 2^{\sum_{i=1}^{d} t_i}(\mathbf{R}^T). \tag{15}$$

$\mathbf{R}^T$ 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, *Full-grid Solution*

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 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:

$$u(\boldsymbol{x}) = \frac{\sum_{i=1}^{d} sin(d\pi^2 x_i)}{d\pi + \sum_{i=1}^{d} x_i} \tag{16}$$

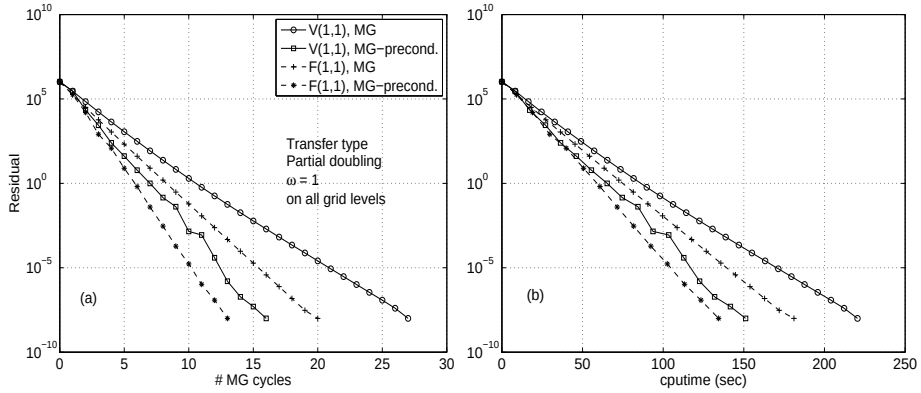


Figure 3: Convergence diagram for a 6 dimensional anisotropic problem, grid stretched along $d/2$ dimensions and given by $\boldsymbol{N} = [32, 8, 32, 8, 32, 8]$. # of unknowns is 26,198,073. Diagram (a) shows a comparison between multigrid as a solver and multigrid as a preconditioner, on the iteration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

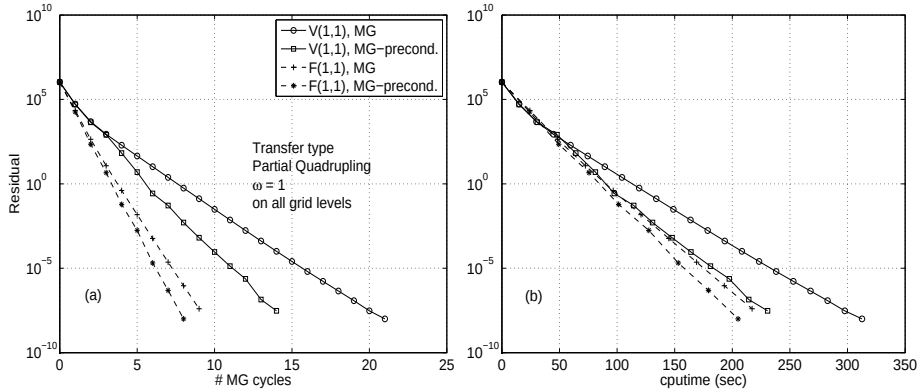


Figure 4: Convergence diagram for a 7 dimensional anisotropic problem, grid stretched along 1 dimension, and given by $\boldsymbol{N} = [8, 8, 8, 64, 8, 8, 8]$, # of unknowns is 34,543,665.

We have conducted a number of numerical experiments -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 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 outperforms the V(1,1) multigrid solver; Figure 2, (here we choose $\omega = 1$ in ω-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 some 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 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).

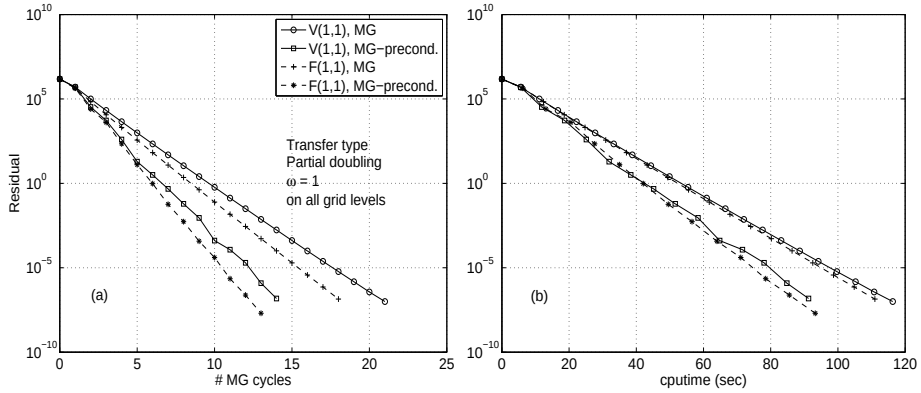


Figure 5: Convergence diagram for a 4 dimensional anisotropic problem, grid stretched along $(d-1)$ dimensions, and given by $\boldsymbol{N} = [128, 128, 128, 8]$, # of unknowns is 19,320,201

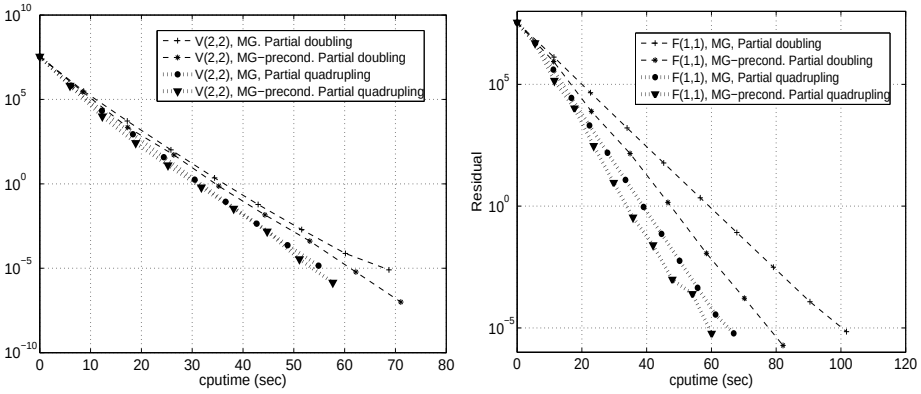


Figure 6: Convergence diagrams comparing d-multigrid based on doubling transfers against quadrupling transfers for V(2,2) -Diagram (a)- and F(1,1) -Diagram(b)- multigrid cycles. This 5 dimensional problem is discretized on $\boldsymbol{N} = [8, 8, 2048, 8, 8]$, # of unknowns is 13,443,489.

In [2] 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.

6 Numerical Experiments, *Sparse-grid Solution*

As the solver works for every kind of grid that might arise within the sparse grid setting (as described in section 4), the aim is now to reach a reasonable number of dimensions. The test function for the sparse grid stationary experiment reads:

$$u(\boldsymbol{x}) = \prod_{i=1}^{d} e^{x_i^2} = \exp\left(\sum_{i=1}^{d} x_i^2\right), \tag{17}$$

with $\Omega = [0,1]^d$, $c_i = 1$. The approximation is done for $2 \leqslant d \leqslant 8$ with a mimic of the grid with 1024 cells per coordinate. In a full grid setting the maximum problem would have a size of $1024^8 = 2^{80}$, i.e., 2^{53} GB of memory, which is - of course - immensely huge. The maximum size of an 8D problem in the sparse grid setting chosen here is $1024 \times 2^7 = 2^{17}$, which is only 1 MB. The solution at the central point in the domain $x_i = 0.5$ is computed here. The error and convergence for all values of d are plotted in Figure 7.

The table and figures show the dependence of the number of dimensions in the convergence according to the

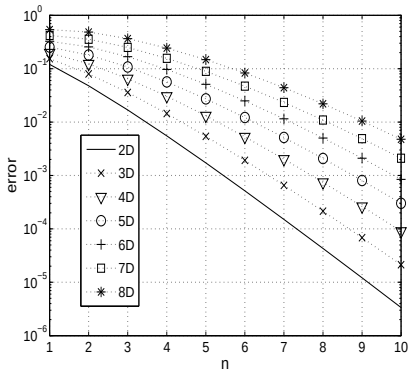


Figure 7: Decay of the error $|u_n^c - u_E|$, with u_E the exact solution (17)

Table 2: Comparison of the total number of pure multigrid and multigrid preconditioned Bi-CGSTAB iterations (maximum, minimum and average number), for all the subgrids on the finest layer of a d-dimensional problem, for increasing d, # grids is the number of subgrids solved

d	Max	Min	Average	# grids	Max	Min	Average	# grids
	Bi-CGSTAB with MG				Pure Multigrid			
2	12.0	4.0	9.400	10	13.0	5.0	10.000	10
3	14.0	6.0	10.691	55	20.0	6.0	12.436	55
4	14.0	6.0	10.982	220	21.0	7.0	13.155	220
5	16.0	6.0	10.999	715	24.0	8.0	13.221	715
6	16.0	6.0	10.928	2002	22.0	8.0	13.185	2002
7	16.0	6.0	10.791	5005	21.0	8.0	12.997	5005
8	14.0	6.0	10.488	11440	20.0	8.0	12.762	11440

theoretical convergence ratio in equation (12). Although the theoretical convergence of the sparse grid method is low when d is high at small numbers of n_{max} (the largest number of cells in one direction), the convergence in this test experiment is reasonable. A possible reason may be the smoothness of the analytic solution.

Table 2 presents a comparison of the total number of multigrid cycles when multigrid is employed both as a solver as well as a preconditioner for Bi-CGSTAB for solving all sparse grid subproblems on a layer of a d-dimensional problem, with d increasing. Presented are the maximum, the minimum and the average numbers of iterations, as well as the total number of subproblems solved on which these numbers are based. The stopping criterion is the residual being smaller than 10^{-14}, which is severe but gives a good insight in the comparison. For both solvers we choose smoothing relaxation parameter $\omega = 1$. We see that the average number of multigrid cycles -when multigrid is used as a precondioner instead of being used as a solver- does not reduce significantly for low values of d. However the cycle difference in the two usages do become significant for higher d because then the total number of subproblems also increase binomially. As the number of subproblems to be solved is more than 5000 for the 7D problem a gain in average of about 2 multigrid iterations is still interesting.

The time for the highest level in the case $d = 8$ over 10^5 seconds which is 28 hours. However, the number of subproblems is 24300, so the average computational time per grid is only 5 seconds. If the sparse grid method is parallelized over 10 machines, the time would be 2.8 hours in total, because the sparse grid is only a combination technique of subproblems. Parallelization is a future task in our project.

7 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 been partially funded by HEC Pakistan and partially by the Delft University of Technology. Special thanks are due to C.W. Oosterlee and C.C.W. Leentvaar for their valuable insights and suggestions.

References

[1] G. Beylkin and J.M. Martin. Algorithms for numerical analysis in high dimensions. *SIAM J. Sci. Comp.*, 26(6): 2133–2159, 2005.

[2] 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.

[3] H.J. Bungartz, M. Griebel, D. Röschke, and C. Zenger. Pointwise convergence of the combination technique for the Laplace equation. *East-West J. Numer. Math.*, 2:21–45, 1994.

[4] J. Elf, P. Lötstedt, and P. Sjöberg. Problems of high dimension in molecular biology. In *Proceedings of the 19th GAMM-Seminar Leipzig*, pages 21–30, 2003.

[5] 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.

[6] 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.

[7] 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.

[8] Y. K. Kwok. *Mathematical models of financial derivatives*. Springer Finance. Springer-Verlag, Singapore, second edition, 1998.

[9] 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.

[10] C. Reisinger and G. Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. Technical report, Oxford University, 2006.

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

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

[13] 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.

[14] I. Yavneh. Multigrid smoothing factors for red-black gauss–seidel relaxation applied to a class of elliptic operators. *SIAM J. Num. Anal.*, 32: 1126–1138, 1995.

[15] H. Yserentant. Sparse grid spaces for the numerical solution of the electronic schrödinger equation. *Numer. Math.*, 101: 381–389, 2005.

[16] C. Zenger. Sparse grids. In *Proceedings of the 6th GAMM seminar, Notes on numerical fluid mechanics*, volume 31, 1990.