

Research Article

A MULTIGRID METHOD FOR NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

By

¹I. O. Isah, ²A. Ndanusa, ³R. Muhammad, ⁴K. A. Al-Mustapha

¹(Department of Mathematical Sciences, Prince Abubakar Audu University, Anyigba, Nigeria)

^{2,3}(Department of Mathematics, Federal University of Technology, Minna, Nigeria)

⁴(Department of Mathematical Sciences, Baze University, Abuja, Nigeria)

Corresponding author's email: isah.io@ksu.edu.ng

ABSTRACT

Techniques and analyses of multigrid method for solving elliptic partial differential equations (PDEs) in two dimensions are presented. The focal point of this paper is the applicability of the parametric reaccelerated overrelaxation (PROR) iterative method as a smoother in multigrid solution of elliptic PDEs. The two-dimensional Poisson equation on a unit square domain with Dirichlet boundary conditions is adopted as the model PDE. We present some practical formulae and techniques for building the various multigrid components using Kronecker tensor product of matrices. In addition, we carry out smoothing analysis of the PROR method using Local Fourier Analysis (LFA) and show how optimal relaxation parameters and smoothing factors can be obtained from analytic formulae derived to ensure better convergence. This analysis combines full standard coarsening strategy (doubling) and second order finite difference scheme. The result of PROR smoothing factors in comparison with those of other widely used smoothers is also presented. Results obtained from numerical experiment are displayed and compared with theoretical results.

Keywords: Multigrid, elliptic PDEs, Poisson equation, coarsening strategy, point-smoothing, smoothing factor, local Fourier analysis.

AMS subject classification: 65F10, 65F15, 65M06, 65N55.

I Introduction

Many real world applications are modeled as partial differential equations (PDEs) on rectangular domains. For instance, the set of Maxwell's electromagnetic equations combined with Ohm's law is used as the basic model for describing the propagation of electromagnetic waves in earth's crust, the Schrodinger equation for scattering applications in quantum mechanics can be transformed in some cases to the Helmholtz equation, a prototype of elliptic PDEs [1]. According to [2], multi-dimensional PDEs have diverse applications in many applied science fields such as financial engineering [3], molecular

biology [4], and quantum dynamics [5, 6]. In addition, the Poisson equation in particular, occurs frequently in electromagnetism, fluid dynamics among others.

It is common and often easier to express PDEs as systems of linear equations, a procedure known as discretization. Although, it may be possible to use direct methods such as matrix inversion, Gaussian elimination, and the likes, to exactly solve these systems of equations, they tend to take far too long to be practicable for very large and sparse linear systems. In general, it is a difficult if not impossible task to solve many PDEs exactly via direct methods. Except in simple cases, numerical method is best, if not the only, way of obtaining a solution satisfying certain boundary conditions [7].

The ubiquity of elliptic PDEs in the field of applied sciences, coupled with the difficulty in the use of direct methods for solving large systems obtained from discretizing PDEs, has led to the development of many suitable indirect (iterative) techniques for a wide range of specific applications and problems. [8] suggested that iterative methods are preferred and such techniques remain the leading approach for solving very large linear systems.

Classical iterative methods such as the Jacobi [9], Gauss Seidel [10], Successive overrelaxation (SOR) [11], accelerated overrelaxation (AOR) [12] methods have all been deployed in the solution of PDEs. Other recent variants of SOR and AOR include preconditioned SOR [13], reaccelerated overrelaxation (ROR) [14], and parametric reaccelerated overrelaxation (PROR) [15] methods. The classical iterative methods are however seldom used as stand-alone methods for solving very large linear systems today due to their slow convergence properties. However, when used as smoothers in a multigrid method, the convergence rate of the whole solution process is sped up rapidly.

Multigrid methods are a class of iterative techniques called multi-resolution methods which are very efficient for problems exhibiting multiple scales of behavior. Multigrid methods typically solve the system of equations on a “coarse” grid, and “refine” the solution to the desired accuracy instead of trying to solve the discrete system at full resolution as obtained in classical iterative methods [16]. Many multigrid methods and applications where multigrid have been applied abound in literature; these include multigrid methods for solution of two dimensional (2-D) Poisson problem [17, 18], multigrid smoothing factors for red-black Gauss–Seidel relaxation applied to a class of elliptic operators [19], red-black SOR smoothing in multigrid [20], multigrid solution of three dimensional (3-D) Poisson equation [21], a book on various multigrid treatments and applications [22], multigrid method for higher dimensional PDEs[2], multigrid method based on L-shaped coarsening for PDEs on stretched grids [23], multigrid solution of Poisson equation on irregular voxelized domains with mixed boundary conditions [24], high order compact difference scheme (HOC) and a multigrid solution of 2-D elliptic problems with variable coefficients [25], multigrid method in distributed control problems modeled by Stokes equations [26], fourth-order compact difference scheme in multigrid solution of 2-D elliptic boundary value problems [27], multigrid method with eighth-order compact finite difference scheme for Helmholtz equation [28], multigrid method for solving optimal control problems governed by stochastic PDEs [29], application of tweed and wireframe relaxation methods as smoothers in multigrid solution of elliptic PDEs on stretched, structured grids [30], multigrid strategies for the solution of elliptic PDEs discretized by the Hybrid High-Order method (HHO) [31], and many others.

Although numerous multigrid methods have been developed and/or applied to solving a wide range of problems, most of the existing multigrid methods for solving elliptic partial differential equations often employed Jacobi, Gauss-Seidel, SOR iterative methods and some of their variants as the relaxation schemes (smoothers). The AOR method and its variants have hardly been explored as smoothers in multigrid methods. Thus, the focus of this paper is the use of a recent version of the AOR iterative method as a smoother in multigrid solution of the 2-D Poisson equation chosen as the model problem for elliptic PDEs.

In section 2, we present the discretization of the continuous problem (model PDE) and show how this can be done with Kronecker tensor products. Section 3 gives smoothing and coarsening strategies and grid transfer strategies. Next, we provide the LFA for the PROR method and present some results of smoothing factors and optimal relaxation parameters in section 4. In section 5, we present and discuss the results of numerical experiments and reveal the multigrid convergence result that we get. Lastly, a summary of the work is presented in the form of conclusion in the last section.

II The Continuous Problem and Its Discretization

2.1 The Model PDE

Consider the standard 2-D Poisson PDE:

$$-\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)u(x, y) = f^\Omega(x, y), x, y \in \Omega = \prod_{i=1}^2 (a_i, b_i) \quad (1)$$

with Dirichlet boundary conditions:

$$u(x, y) = g(x, y), x \in (a_1, a_2), y \in (b_1, b_2) \quad (2)$$

According to [23], the 2-D Poisson equation in (1) provides a consistent model upon which new numerical solvers can be tested without digging too deeply into the complications of many application problems. Therefore, it serves well as a model problem in the study of convergence behaviour of elliptic partial differential equation, hence its adoption as the model PDE in the design of the multigrid method in this thesis.

In order to simulate Dirichlet boundary conditions (2), let u be a test function (whose boundary values are defined as g) given as follows:

$$u(x, y) = \frac{\sin(c_1\pi x) + \sin(c_2\pi y)}{c_3 + x + y}, \quad c_1, c_2, c_3 \in R \quad (3)$$

To simulate the right hand side, the negative Laplacian in (1) is computed from the prescribed test function to give $f(x, y)$ as follows:

Differentiating (3) partially with respect to x , we obtain:

$$\frac{\partial u}{\partial x} = \frac{(c_1\pi) \cos(c_1\pi x)(c_3 + x + y) - (\sin(c_1\pi x) + \sin(c_2\pi y))}{(c_3 + x + y)^2} \quad (4)$$

Also, the second derivative of (3) with respect to x with further simplification yields:

$$\frac{\partial^2 u}{\partial x^2} = \frac{(c_1\pi)^2 \sin(c_1\pi x)}{(c_3 + x + y)} + \frac{2(c_1\pi) \cos(c_1\pi x) - 2u}{(c_3 + x + y)^2} \quad (5)$$

Similarly, $\frac{\partial^2 u}{\partial y^2}$ is obtained by differentiating (3) twice with respect to y leading to:

$$\frac{\partial^2 u}{\partial y^2} = \frac{-(c_1\pi)^2 \sin(c_1\pi y)}{(c_3 + x + y)} + \frac{-2(c_1\pi) \cos(c_1\pi y) + 2u}{(c_3 + x + y)^2} \quad (6)$$

Then, the source function, $f(x, y)$ becomes:

$$f = \frac{(c_1\pi)^2 \sin(c_1\pi y) + (c_1\pi)^2 \sin(c_1\pi y)}{(c_3 + x + y)} + \frac{2(c_1\pi) \cos(c_1\pi y) + 2(c_1\pi) \cos(c_1\pi y) - 2u}{(c_3 + x + y)^2} \quad (7)$$

to complete the prescription of the model problem,[see 23].

The continuous problem prescribed above is represented in operator notation as:

$$Lu(x, y) = f^\Omega(x, y) \quad (8)$$

where L is the Laplacian.

Thus, we obtain the discrete equivalent of (8) as:

$$L_h u_h(x, y) = f_h^\Omega(x, y) \quad (9)$$

The discretization of the Laplacian, L_h is taken to be $O(h^2)$ accurate, leading to a 5-point stencil in two dimensions. The number of cells in the discretization grid along the i th dimension (given by N_i) may not necessarily be equal to the number of cells along j th dimension. Thus, the grid size along i th dimension is given by:

$$h_i = (b_i - a_i)/N_i \quad (10)$$

and the one-dimensional (1-D) variant of this stencil is:

$$\frac{\partial^2 u}{\partial x^2} \Big|_{x=x_i} \equiv \frac{1}{h_i^2} [1 \quad -2 \quad 1] u_i + O(h^2) \quad (11)$$

The discrete equation given by (9) can be represented in matrix notation as:

$$A_h u_h = b_h \quad (12)$$

2.2 The Discretization Matrix, A_h

The coefficient matrix, A_h is an $(M \times M)$ matrix with M —unknowns where M is the order of A_h given by:

$$M = \prod_{i=1}^2 (N_i - 1) \quad (13)$$

Let the discretization grid be G , that is:

$$G = [N_1, N_2] \quad (14)$$

Then, applying the d —dimensional tensor product formula proposed in [2], matrix A_h is constructed using the tensor product formula:

$$A_h = \sum_{i=1}^2 \{ \prod_{j=1}^i I_{(2+i-j)} \otimes L_i \otimes \prod_{j=1}^{i-1} I_{(i-j)} \} \quad (15)$$

where \otimes is the Kronecker tensor product of matrices and \prod is the cumulative Kronecker tensor product. For instance,

Also, in (15), I_m ($m \in \{1,2\}$) is the identity matrix of order $N_m - 1$ and L_i is the one dimensional discrete Laplacian operator in the i th dimension. L_i is obtained using the 1-D stencil in (11). The stencil is applied to each point, including the boundaries, to obtain L_i . The left and right boundary vectors are however sorted and integrated into the right hand side, b_h of the linear system (12) as earlier stated (eliminated boundary). The following matrix is obtained for L_i in (15) for $G = \{5,5\}$ to illustrate how matrix is A_h is constructed:

$$L_i = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \quad (16)$$

Similarly, the 1-D Laplacian operator is used to construct the Laplacian for the j th dimension and replaced back in to (15) to obtain the full 2-D discretization matrix, A_h .

2.3 Right Hand Side, b_h

The column vector, b_h is built in a similar manner using Kronecker tensor products following [2]. It consists of the source function f_h^S and the boundary function f_h^B .

Importantly, we define a consistent listing of grid points. Let \mathfrak{I} be an index matrix which denotes the set of indices in the computational grid, where each row (a 2-tuple) represents the index of a grid point and counts in descending order, from right to left, for the ascending order of the dimensions. Then, we have grid coordinates as a pair (i_2, i_1) . That is:

$$\mathfrak{I} = [\tau_2 \tau_1] \quad (17)$$

where each τ_k is a column vector of length M .

Also, we consider the following definitions required for building the index set for the interior and the boundary points:

$$\left. \begin{aligned} \gamma_i : \gamma_i &= [1, 2, \dots, (N_i - 1)]^T, i = (1,2) \\ \mathbf{1}_i : \mathbf{1}_i &= [1, 1, \dots, 1]^T \end{aligned} \right\} \quad (18)$$

Now, the column of \mathfrak{I} are formed using the Kronecker tensor product:

$$\tau_i = \prod_{j=1}^i \mathbf{1}_{(2+i-j)} \otimes \gamma_i \otimes \prod_{j=1}^{i-1} \mathbf{1}_{(i-j)} \quad (19)$$

to complete the construction of the index set. The vector of the source function for each row of \mathfrak{I} denoted by S , can then be computed. This is represented as:

$$S = f_{\mathfrak{I}}^S \quad (20)$$

In addition, to complete the specification of b_h , contribution of boundaries to b_h needs to be computed. We recall that two column vectors; the left and the right boundary-coefficient vectors were isolated from the 1-D Laplacian operators in each

dimension, Now, considering the i th dimension, let the left and the right boundary-coefficient vectors be l_i and r_i , respectively, then the i th 2-D left and right boundary-coefficient vectors respectively denoted by \mathcal{L}_i and \mathcal{R}_i are given by the Kronecker tensor formulae:

$$\left. \begin{aligned} \mathcal{L}_i &= \tau_{j=i} \otimes 1_{(2+i-j)} \otimes l_i \otimes \prod_{j=1}^{i-1} I_{(i-j)} \\ \mathcal{R}_i &= \tau_{j=i} \otimes 1_{(2+i-j)} \otimes r_i \otimes \prod_{j=1}^{i-1} I_{(i-j)} \end{aligned} \right\} \quad (21)$$

Also, let the contribution from the left boundary and the right boundary in b_n be represented as \mathcal{B}_L and \mathcal{B}_R , respectively, where \mathcal{B}_L is the cumulative sum of the two left boundaries while \mathcal{B}_R is the sum of the two right boundaries. Worthy of note is that if any τ_i in (3.17) is replaced by a vector of left boundary value, a left boundary index set is obtained, and if it is replaced by a vector of right boundary value, we get a right boundary index set. That is, if:

$$\left. \begin{aligned} \mathfrak{L}_i &= [\tau_2 \tau_1 \dots \tau_{(i-1)} \bar{a}_i \tau_{(i+1)} \dots \tau_2 \tau_1] \\ \mathfrak{R}_i &= [\tau_2 \tau_1 \dots \tau_{(i-1)} \bar{b}_i \tau_{(i+1)} \dots \tau_2 \tau_1] \end{aligned} \right\} \quad (22)$$

then:

$$\left. \begin{aligned} \mathcal{B}_L &= \sum_{i=1}^2 (\mathcal{L}_i \diamond f_{\mathfrak{L}_i}^T) \\ \mathcal{B}_R &= \sum_{i=1}^2 (\mathcal{R}_i \diamond f_{\mathfrak{R}_i}^T) \end{aligned} \right\} \quad (23)$$

where \diamond is the component-by-component multiplication of the operand column vectors. Thus we obtain the right side of discrete problem as:

$$h_n = S + \mathcal{B}_L + \mathcal{B}_R \quad (24)$$

to obtain the complete discretization of the model PDE

III Design of the Method

3.1 Multigrid Based on Point Relaxation

Multigrid methods consists of essential components such as grid coarsening and smoothing strategies, grid transfer schemes, the smoothing method and the coarse grid correction. The famous algorithm of multigrid is presented in [22] and has been used by many other authors in literature.

The technique of relaxation (smoothing) in multigrid is mainly classified into point relaxation and block relaxation methods. Although block relaxation methods (line and plane smoothing) are mostly used for anisotropic PDEs, they are quite expensive in terms of CPU-time because each relaxation step consists of solving a small linear system exactly (or to a satisfactory level of accuracy). Fortunately, point smoothing works well in most cases, particularly for standard coarsening. More so, in most situations, block smoothing can be substituted with point-relaxation methods with some modifications. Thus point relaxation approach is adopted as the smoothing strategy in the multigrid method discussed in this research.

3.2 The Relaxation Scheme

The core of the multigrid method proposed in this research is the relaxation scheme deployed as a smoother. Other components are essentially some simple and basic choices existing in literature. As will be shown in section 4, the smoothing analysis of the PROR method in [15], which we now denote by α, r, ω –PROR, proves its admirable smoothing effect for problems of the Poisson type and is thus chosen as the smoother for our proposed multigrid method. Also, although we do not provide the smoothing analysis of the PROR method in the red-black setting (abbreviated as α, r, ω –RBPROR), it will be considered in the implementation. The α, r, ω –RBPROR method consists of two partial steps, each step being an α, r, ω –PROR sweep. The first step applies to and updates only the red (odd) points while the second one applies to and updates only the black (even) points in the grid. From an implementation angle, this red-black smoothing procedure depends upon a partitioning process by which the grid G can be separated into the red part (G_R) and the black part (G_B). The grid-point listing employed in the implementation scheme is such that the points are arranged in a column vector and listed out in lexicographic order. We also make use of optimal relaxation parameters (α_{opt}, r_{opt} and ω_{opt}) in the relaxation (smoothing) process. The error-smoothing effect of the relaxation method can be enhanced by the use of optimal relaxation and acceleration parameters [see, 20].

The α, r, ω –PROR method is given as follows:

$$x^{(n+1)} = [(1 + \alpha)I - \omega L]^{-1} \{ [(1 + \alpha - r + r\omega)I + (r - \omega - r\omega)L + (r - r\omega)U] x^{(n)} + (r - r\omega)b \} \quad (25)$$

with iteration matrix:

$$L_{\alpha, r, \omega} = [(1 + \alpha)I - \omega L]^{-1} \{ [(1 + \alpha - r + r\omega)I + (r - \omega - r\omega)L + (r - r\omega)U] \} \quad (26)$$

where α is a fixed parameter and r and ω are acceleration and relaxation parameters respectively.

3.3 Coarsening Strategy

For the coarse grid H , standard full coarsening strategy is applied to the 2-D model problem. It involves doubling the grid size of the fine grid, h in both direction, i.e. $H = 2h$. In two dimensions and standard coarsening, the relation between the coarse and fine grid is given by:

$$\Omega_{2h} = \frac{1}{4} \Omega_h \quad (27)$$

3.4 Discretization of the Coarse Grid

A very vital component in the coarse-grid correction process is the choice of the coarse-grid operator, $I_{H,r}$. According to [1], re-discretizing the differential operator (i.e. the Laplacian, in the case of the model PDE in this work) on the coarse grid is sufficient as the approximation to the discrete operator on the coarse grid in many situations. This means that the direct equivalent of the fine grid operator on the coarse grid, H is used. As soon as the next coarser grid is decided, the Laplacian is discretized using the same discrete stencils as presented in section 2. This is traditionally known as the discretization coarse grid operator (DCG). Thus, in this work, the DCG operator, which is the coarse-grid analog of the discrete operator on the fine grid is employed. Another coarse grid operator in vogue is the Galerkin coarse grid operator (GCG). However, using the

GCG is disadvantageous in that it is usually denser than the DCG operator except special transfer operators are employed to generate the coarse-grid operators which is not the case in this paper.

3.5 Grid Transfer Operators

The full weighting (FW) restriction and the bilinear interpolation operators in two dimensions are used for the inter-grid transfers (restriction and prolongation respectively) of the grid functions. The 2-D FW restriction operator is given by:

$$I_h^{2h} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{2h}^{2h} \quad (28)$$

while the bilinear prolongation is given by the 9-point stencil:

$$I_{2h}^h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{2h}^h \quad (29)$$

The generation of the FW restriction and bilinear prolongation operator matrices via Kronecker tensors is presented next. We note that the 2-D FW restriction (28) is the Kronecker tensor product of the 1-D FW operators in x_1 and x_2 directions given below:

$$\left. \begin{aligned} (I_h^{2h})_{x_1} &= \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \\ (I_h^{2h})_{x_2} &= \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \end{aligned} \right\} (30)$$

We construct the 2-D FW restriction operator matrix, \mathbf{R} via the Kronecker tensor products formulae:

$$\left. \begin{aligned} \mathbf{R} &= \prod_{i=1}^2 (\mathbf{R}_i)^{k_i} \\ (\mathbf{R}_i)^{k_i} &= \prod_{l=0}^{k_i} \left[\prod_{j=1}^l \mathbf{I}_{N_{(2+i-j)}} \otimes \mathbf{O}_{[N_i/2^{(k_i-l-1)}]} \otimes \prod_{j=1}^{l-1} \mathbf{I}_{[N_{(i-j)}/2^{k_i-j}]} \right] \end{aligned} \right\} (31)$$

The quantities in (31) are given as follows:

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

\mathbf{I}_α is the 1-D FW restriction matrix of order $\left(\frac{\alpha}{2} - 1\right) \times (\alpha - 1)$.

$T = [k_1, k_2]$ is the coarsening request; k_i is the count of ($h \rightarrow 2h$) transfers along the i th dimension where α is any dummy subscript.

Next, given the prescription of the 2-D FW restriction operator matrix, the prolongation (bi-linear interpolation) operator matrix can then be obtained with the following relation:

$$\mathbf{P} = 2^{(\sum_{i=1}^2 k_i)} (\mathbf{R}^T) \quad (32)$$

to complete the specification of the grid transfer operators.

Worthy of note is that the restriction operator in (31) is a generalized one which gives us the leverage to experiment with different types of coarsening strategies depending on the grid. Also, it offers the required matrix for any number of coarsening along any number of dimensions for an abstract d-dimensional problem [see 2].

IV Local Fourier Analysis of the α, r, ω –PROR Method

Local Fourier Analysis (LFA) is a tool for investigating the convergence behavior of multigrid methods introduced by Brandt [32] with contributions from other notable authors such as Stuben [33], [34], [35], [36], and [22]. Much details, including the validity of LFA alongside related theorems and proofs, are presented in [35]. In what follows in this section, we focus on the design of a Fourier representation for α, r, ω –PROR Method applied to the discrete operator of the model PDE (Laplacian) given in section 2.1, the definition of the smoothing factor and the estimation of relaxation and acceleration parameters. It is assumed that the reader is familiar with the basics of the LFA tool as can be found in [22, 33, 34, 35, 36]. Also, we adopt the definitions and notations as obtainable in [19, 20, 22, 36] to carry out our analysis in this section.

Consider the Laplacian operator given in (1) as follows:

$$-\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) \quad (32)$$

The continuous partial derivatives in (32) are discretized by second-order finite difference in (11), to give their discrete equivalents, denoted by L_h . Although we stated in section 2.1 that the grid spacing may not necessarily be equal, for the purpose of this analysis, we assume equal grid spacing in both directions, i.e. $h_1 = h_2 = h$. LFA considers only the ‘local nature’ of the discrete operator, i.e. any general discrete operator, nonlinear, with non-constant coefficients, can be linearized locally and can be replaced locally (by freezing the coefficients) by an operator with constant coefficients. Thus for an effective analysis through LFA tool, we must have constant coefficients, and also neglect the effects of the boundaries. More specifically, all arising operators are extended to an infinite grid given by:

$$G_h = \{x = kh = (k_1h_1, k_2h_2), k \in \mathbb{Z}^2\} \quad (33)$$

where h is the grid size.

The fundamental quantities in LFA are the discrete eigen-functions (also known as Fourier components or Fourier modes) of the resulting infinite grid operators. These grid functions are given by:

$$\varphi(\theta, x) = e^{i\theta x/h} = e^{i\theta_1 x_1/h_1} e^{i\theta_2 x_2/h_2} \quad \text{for } x \in G_h \quad (34)$$

where θ varies continuously in \mathbb{R}^2 . We note that:

$$\varphi(\theta, x) \equiv \varphi(\theta', x) \quad \text{for } x \in G_h \quad \text{iff } \theta = \theta' \pmod{2\pi} \quad (35)$$

That is, (3.5) holds if the difference of multiples of 2π is between all the components of the 2-tuple, $(\theta \& \theta')$. Thus it is sufficient to consider these grid functions only for $\theta \in [-\pi, \pi)^2$ which can be written as $-\pi \leq \theta < \pi$, where θ refers to both components θ_1 and θ_2 . For $-\pi \leq \theta < \pi$, these functions, $\varphi(\theta, x)$ are linearly independent on G_h . Also, $\varphi(\theta, x)$ are sometimes written as $\varphi_h(\theta, x)$ for clarity purpose since they are defined on G_h and in that way, depend on h .

4.1 High and Low Fourier Frequencies

In LFA, two distinct kinds of Fourier components; the high frequency components and the low frequency components are distinguishable with respect to the grids G_h and G_H . In a 2-grid setting, the classification of the frequencies $\theta = (\theta_1, \theta_2)$ as either high or low is dependent upon the coarsening strategy that is employed. However, for the purpose of this paper, it is

sufficient to discuss only the case of standard (full) coarsening. Furthermore, the classification into this two categories of frequency components is based on the fundamental observation that only those frequency components $\varphi(\theta, \cdot)$ on G_H with $-\frac{\pi}{2} \leq \theta < \frac{\pi}{2}$ are distinguishable on G_H . For each $\theta' \in [-\frac{\pi}{2}, \frac{\pi}{2})^2$, three other frequency components $\varphi(\theta, \cdot)$ on G_H with $\theta \in [-\pi, \pi)^2$ coincide on G_H with $\varphi(\theta', \cdot)$ and not visible on G_H . Evidently, the following identity holds:

$$\varphi(\theta, x) \equiv \varphi(\theta', x) \text{ for } x \in G_H \text{ iff } \theta = \theta' \pmod{\pi} \quad (35)$$

leading to the definition below

DEFINITION 1 (high and low frequencies for full coarsening): An element θ_j ($j \in \{1, 2\}$) of a Fourier frequency θ is termed low if $-\frac{\pi}{2} \leq \theta_j < \frac{\pi}{2}$ for $\theta \in [-\pi, \pi)^2$. Otherwise it is called high. We say we have a low Fourier frequency θ , if all of its elements are low. Otherwise it is called a high frequency. Therefore, $\varphi(\theta, \cdot)$ is called a low frequency mode if:

$$\theta \in T^{low} := \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2 \quad (37)$$

and $\varphi(\theta, \cdot)$ is called a high frequency mode if:

$$\theta \in T^{high} := [-\pi, \pi)^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2 \quad (38)$$

The distinction described above clearly depends on the coarsening strategy, as, for different coarsening strategies, different sets of Fourier frequencies are visible on the coarse grid.

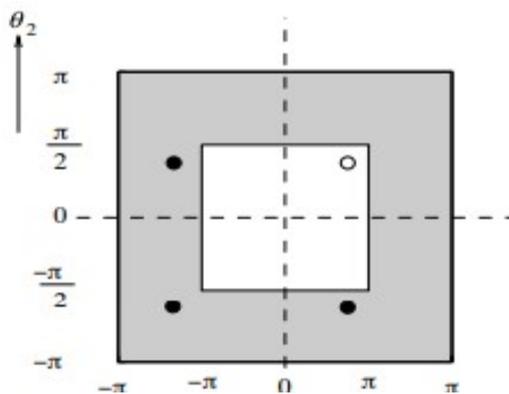


Figure. 1: high and low frequencies in 2-D

4.2 Fourier Representation of α, r, ω –PROR

Lemmas

Lemma 1 (see [22])

For $-\pi \leq \theta < \pi$, all grid functions $\varphi(\theta, \mathbf{x})$ are (formal) eigenfunctions of any discrete operator which can be described by a difference stencil $L_h = [s_k]_h$, ($\mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2$). Then, the following relation holds:

$$L_h \varphi(\theta, \mathbf{x}) = \tilde{L}_h(\theta) \varphi(\theta, \mathbf{x}) \quad (\mathbf{x} \in G_h) \quad (39)$$

where $\tilde{L}_h(\theta)$ is the corresponding eigenvalue (formal) or the symbol of L_h given by:

$$\tilde{L}_h(\theta) = \sum_k s_k e^{i\theta \cdot \mathbf{k}} \quad (40)$$

Lemma 2 (see [22])

Suppose a relaxation method can be written locally as:

$$L_h^+ \bar{w}_h + L_h^- w_h = f_h \quad (41)$$

where w_h corresponds to the old approximation of u_h and \bar{w}_h is the new approximation, then the relaxation is characterized by the splitting:

$$L_h = L_h^+ + L_h^- \quad (42)$$

Under the assumption in (41) and (42), all grid functions, $\varphi(\theta, \cdot)$ with $\tilde{L}_h^+(\theta) \neq 0$ are eigenfunctions of S_h such that:

$$S_h \varphi(\theta, \mathbf{x}) = \tilde{S}_h(\theta) \varphi(\theta, \mathbf{x}) \quad (-\pi \leq \theta < \pi) \quad (43)$$

where $\tilde{S}_h(\theta)$ is the amplification factor given by:

$$\tilde{S}_h(\theta) := -\frac{\tilde{L}_h^-(\theta)}{\tilde{L}_h^+(\theta)} \quad (44)$$

Now, the PROR method (25) can be rewritten in a notation consistent with [22] as follows:

$$[(1 + \alpha)D - rL]u^n - [(1 + \alpha - \omega + r\omega)D + (\omega - r - r\omega)L + (\omega - r\omega)U]u^{n-1} + (\omega - r\omega)\bar{b} \quad (45)$$

where D denotes the diagonal part, L and U are the lower and upper parts of the iteration matrix (26) respectively. Equation (45) further simplifies to:

$$\frac{1}{\omega - r\omega} [(1 + \alpha)D - rL]u^n = \left[\frac{(1 + \alpha - \omega + r\omega)}{\omega - r\omega} D + \frac{(\omega - r - r\omega)}{\omega - r\omega} L + U \right] u^{n-1} + \bar{b} \quad (46)$$

$$\frac{1}{\omega(1-r)} [(1+\alpha)D - rL]u^n = \left[\frac{(1+\alpha - \omega(1-r))}{\omega(1-r)} D + \frac{(\omega(1-r) - r)}{\omega(1-r)} L + U \right] u^{n-1} + \bar{b} \quad (47)$$

Also, the discrete equivalent of the model problem is of the form:

$$L_h u_h = f_h \quad (48)$$

where L_h represent the discrete Laplacian operator given by:

$$-\Delta_h = -\frac{1}{h^2} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \quad (49)$$

and the stencil representation of (49) is as follows:

$$-\Delta_h = \frac{1}{h^2} \begin{bmatrix} & 1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} \quad (50)$$

If the PROR relaxation method conforms with (41) and (42), then we obtain:

$$L_h^+ = \frac{1+\alpha}{\omega(1-r)} D - \frac{r}{\omega(1-r)} L \quad (51)$$

and in view of (50), L_h^+ in stencil form is given as:

$$L_h^+ = \frac{1}{h^2} \begin{bmatrix} & 0 & \\ -\frac{r}{\omega(1-r)} & 4\left(\frac{1+\alpha}{\omega(1-r)}\right) & 0 \\ & -\frac{r}{\omega(1-r)} & \end{bmatrix} \quad (52)$$

Also, from (47):

$$L_h^- = - \left[\frac{(1+\alpha - \omega(1-r))}{\omega(1-r)} D + \frac{(\omega(1-r) - r)}{\omega(1-r)} L + U \right] \quad (53)$$

$$= \left(1 - \frac{1+\alpha}{\omega(1-r)} \right) D - \left(1 - \frac{r}{\omega(1-r)} \right) L - U \quad (54)$$

Again, using (50), we obtain the stencil form of L_h^- as:

$$L_h^- = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ \left(1 - \frac{r}{\omega(1-r)} \right) & 4\left(1 - \frac{1+\alpha}{\omega(1-r)} \right) & -1 \\ & \left(1 - \frac{r}{\omega(1-r)} \right) & \end{bmatrix} \quad (55)$$

Applying *Lemma 1*, we have:

$$\hat{L}_h^+(\theta, \alpha, r, \omega) = L_h^+ e^{i\theta \cdot \bar{x}/h} = \frac{1}{h^2} \begin{bmatrix} 0 & 4\left(\frac{1+\alpha}{\omega(1-r)}\right) & 0 \\ -\frac{r}{\omega(1-r)} & & \\ & -\frac{r}{\omega(1-r)} & \end{bmatrix} \quad (56)$$

$$= [s_k]_h \cdot e^{i\theta \cdot \bar{x}/h} \quad (57)$$

where:

$$\hat{L}_h^+(\theta, \alpha, r, \omega) = \sum_k s_k e^{i\theta \cdot k}, \quad k = \bar{x}/h = (k_1, k_2), k \in \mathbb{Z}^2 \quad (58)$$

Thus, we obtain:

$$\hat{L}_h^+(\theta, \alpha, r, \omega) = \frac{1}{h^2} \left[4\left(\frac{1+\alpha}{\omega(1-r)}\right) - \frac{r}{\omega(1-r)} e^{-i\theta_1} - \frac{r}{\omega(1-r)} e^{-i\theta_2} \right] \quad (59)$$

Also, similar manipulation in favour of L_h^- leads to:

$$\hat{L}_h^-(\theta, \alpha, r, \omega) = \frac{1}{h^2} \left[4\left(1 - \frac{1+\alpha}{\omega(1-r)}\right) - \left(1 - \frac{r}{\omega(1-r)}\right) e^{-i\theta_1} - \left(1 - \frac{r}{\omega(1-r)}\right) e^{-i\theta_2} - e^{i\theta_1} - e^{i\theta_2} \right] \quad (60)$$

Then by Lemma 2:

$$\hat{S}_h(\theta, \alpha, r, \omega) = -\frac{\hat{L}_h^-(\theta, \alpha, r, \omega)}{\hat{L}_h^+(\theta, \alpha, r, \omega)} = \frac{\left[4\left(1 - \frac{1+\alpha}{\omega(1-r)}\right) - \left(1 - \frac{r}{\omega(1-r)}\right) e^{-i\theta_1} - \left(1 - \frac{r}{\omega(1-r)}\right) e^{-i\theta_2} - e^{i\theta_1} - e^{i\theta_2} \right]}{\left[4\left(\frac{1+\alpha}{\omega(1-r)}\right) - \frac{r}{\omega(1-r)} e^{-i\theta_1} - \frac{r}{\omega(1-r)} e^{-i\theta_2} \right]} \quad (61)$$

4.3 Smoothing Factor

The smoothing factor of an iterative method's operator (iteration matrix) is the worst factor by which the high frequency errors are reduced per iteration step. In order to measure the smoothing properties of PROR point relaxation method, we adopt the definition of the smoothing factor from [22] as obtained in Lemma 2. Thus, the smoothing factor, μ , associated with the PROR iterative method is given by:

$$\mu[\hat{S}_h(\theta, \alpha, r, \omega)] = \sup \{ |\hat{S}_h(\theta, \alpha, r, \omega)| : \theta \in T^{high} \} \quad (62)$$

$$= \sup \left\{ \left| \frac{\left[4\left(1 - \frac{1+\alpha}{\omega(1-r)}\right) - \left(1 - \frac{r}{\omega(1-r)}\right) e^{-i\theta_1} - \left(1 - \frac{r}{\omega(1-r)}\right) e^{-i\theta_2} - e^{i\theta_1} - e^{i\theta_2} \right]}{\left[4\left(\frac{1+\alpha}{\omega(1-r)}\right) - \frac{r}{\omega(1-r)} e^{-i\theta_1} - \frac{r}{\omega(1-r)} e^{-i\theta_2} \right]} \right| \right\} \quad (63)$$

The smoothing factor can easily be calculated by a computer program which allows for a numerical determination of optimal relaxation and acceleration parameters. Thus, the optimal values of the parameters r and ω for a fixed value of α as well as the smoothing factor, μ in (63) are obtained over the high frequency Fourier domain, θ for different values of h with the aid

of a programme written in MATLAB software and presented alongside those of some other well-known smoothers in the table that follows. We note however that explicit analytical formulae that can be used for determining the optimal relaxation parameters in the case of full coarsening applied to the second-order discretization may be possible [see 20, 37].

Table 1: Smoothing Factors of $\alpha, r, \omega - \text{PROR}$ Scheme on Various Equidistant Grids.

$\alpha, r, \omega - \text{PROR} (\alpha_{\text{opt}} = 4.00)$			
G	r_{opt}	ω_{opt}	$\mu(\alpha_{\text{opt}}, r_{\text{opt}}, \omega_{\text{opt}})$
[64, 64]	4.00	-1.69	0.4487
[128, 128]	4.00	-1.69	0.4489
[256, 256]	4.00	-1.69	0.4489

In Table 1, the smoothing factors of $\alpha, r, \omega - \text{PROR}$ smoother obtained using optimal relaxation and acceleration parameters, standard full coarsening (doubling) and second order discretization on various equidistant grids are displayed. The table revealed the effect of using α and optimal parameters in $\alpha, r, \omega - \text{PROR}$ relaxation scheme to enhance smoothness.

Table 2: Comparison of Smoothing Factors of Some Existing Smoothers

G	$\alpha, r, \omega - \text{PROR} (\alpha_{\text{opt}} = 4.00)$			SOR		$\omega - \text{Jacobi}$		Gauss-Seidel	
	r_{opt}	ω_{opt}	$\mu(\alpha_{\text{opt}}, r_{\text{opt}}, \omega_{\text{opt}})$	ω_{opt}	$\mu(\omega_{\text{opt}})$	ω_{opt}	$\mu(\omega_{\text{opt}})$	ω_{opt}	$\mu(\omega_{\text{opt}})$
[64, 64]	4.00	-1.69	0.4487	1.0	0.4998	0.80	0.6000	-	0.4998
[128, 128]	4.00	-1.69	0.4489	1.0	0.5000	0.80	0.6000	-	0.5000
[256, 256]	4.00	-1.69	0.4489	1.0	0.5000	0.80	0.6000	-	0.5000

Table 2 shows a comparison of the smoothing factors obtained for $\alpha, r, \omega - \text{PROR}$ relaxation scheme with those of SOR, Jacobi and Gauss-Seidel relaxation schemes. The table showed that the $\alpha, r, \omega - \text{PROR}$ demonstrates better smoothing ability than the other iterative methods.

V. Results and Discussion

We now show the multigrid convergence of the smoother in this work through numerical example. The model PDE (1) is discretized on various equidistant grids using the second order finite difference scheme, and the spectral radius, ρ of the multigrid iteration operator, M_h (also the asymptotic convergence factor) is obtained. The only quantity accessible to estimate this factor in a multigrid experiment is the residual (or defect) after the i th multigrid cycle denoted by r_h^i . Thus, the empirical estimate of the multigrid convergence factor $\rho(M_h)$, denoted by q^m and called the contraction number is given as follows:

$$q^m = \left(\frac{\|r_h^m\|}{\|r_h^{m-1}\|} \right) \quad (64)$$

where m is the number of iterations or multigrid cycles required for the discrete problem to converge to the numerical solution.

In our experiments, we make two different smoothing choices; one pre- and one post-smoothing, and then two pre- and two post-smoothing steps of the relaxation schemes. Multigrid V-cycle is also deployed as the multigrid control cycle, so that the two different smoothing choices are denoted by V(1, 1) and V(2, 2) respectively. For a correspondence with q^m , the smoothing factor is presented as a square and compared with the smoothing factor obtained from the LFA analysis presented in Table 1. The contraction number, q^m is presented against the number of multigrid cycles (iterations) that the experiments took to converge to a tolerance value of 10^{-10} , representing the termination criterion for all of the experiments, That is:

$$\frac{\|b_h - A_h u_h^m\|}{\|b_h\|} \leq \text{tolerance} \quad (65)$$

This termination criterion is similar to the one based on relative residual reduction since our starting solution (initial guess) in these experiments is an all-zero vector every time. We approximate the test function in (3) for $c_1 - c_2 - c_3 - 2\pi$ via the numerical solution of the model problem (1). As prescribed in (2), the values of this test function at the boundary are taken as Dirichlet boundary conditions, and its Laplacian obtained analytically in (7) gives the source function for our numerical experiments. The results are displayed in the succeeding tables.

Table 3: Results of Numerical Experiment for V(1, 1) and V(2, 2) on Equidistant Grids

$\alpha, r, \omega - \text{PROR } (\alpha_{\text{opt}} = 4.00)$					
G	$[\mu(\alpha_{\text{opt}}, r_{\text{opt}}, \omega_{\text{opt}})]^2$	$\rho(M_h) = q^m$			
		V(1,1)	No of Iter	V(2,2)	No of Iter
[64, 64]	0.2013	0.1219	8	0.0433	6
[128, 128]	0.2015	0.1185	8	0.0416	6
[256, 256]	0.2015	0.1153	8	0.0293	5

Table 3 presents results of solving model problem using $\alpha, r, \omega - \text{PROR}$ methods on various equidistant grids. A close look at the convergence factors clearly shows that the PROR gave appreciable convergence results. In addition, it is observed that the reduction in the multigrid convergence factors as well as in the number of multigrid cycles obtained for V(2, 2) compared to V(1, 1) was quite significant, indicating that V(2,2) is a better choice for this problem.

Table 4: Comparison of Results Between $\alpha, r, \omega - \text{PROR}$ and $\alpha, r, \omega - \text{RB PROR}$

G	$\alpha, r, \omega - \text{PROR } (\alpha_{\text{opt}} = 4.00)$				$\alpha, r, \omega - \text{RB PROR } (\alpha_{\text{opt}} = 4.00)$			
	$\rho(M_h) = q^m$		$\rho(M_h) = q^m$		$\rho(M_h) = q^m$		$\rho(M_h) = q^m$	
	V(1,1)	No of Iter	V(2,2)	No of Iter	V(1,1)	No of Iter	V(2,2)	No of Iter
[64, 64]	0.1219	8	0.0433	6	0.0539	6	0.0162	5
[128, 128]	0.1185	8	0.0416	6	0.0517	6	0.0156	5
[256, 256]	0.1153	8	0.0293	5	0.047	6	0.0087	4

In Table 4, we display experimental results of using $\alpha, r, \omega - \text{PROR}$ method in comparison with those of PROR in red-black ordering ($\alpha, r, \omega - \text{RB PROR}$). It is observed that using the PROR method in red-black setting gave considerably better convergence results than the lexicographic $\alpha, r, \omega - \text{PROR}$ method.

VI Conclusion

In this paper, we established how acceptable multigrid convergence can be achieved with the use of a newly introduced smoother when used with optimum parameter values in the smoothing process. Through numerical experiments supported by theoretical analysis by means of LFA, we have demonstrated that the PROR method is a viable choice as the relaxation method in multigrid solution of elliptic PDEs discretized on rectangular grids as results revealed the appreciable multigrid convergence that can be achieved with the PROR smoother presented in this work.

References

- [1] bin Zubair H., *Efficient multigrid methods based on improved coarse grid correction techniques*, Doctoral Thesis, Delft University of Technology, Delft, The Netherlands, 2009.
- [2] bin Zubair H., Leentvaar C.C.W., and Oosterlee C.W., Efficient d-multigrid preconditioners for sparse-grid solution of high-dimensional partial differential equations, *International Journal of Computer Mathematics*, 84(8), 2007, 1129–1147.
- [3] Reisinger C., and Wittum G., On multigrid for anisotropic equations and variational inequalities, *Computing and Visualization in Science*, 2004.
- [4] Elf J., Lotstedt P., and Sjoberg P., Problems of high dimension in molecular biology, *Proceedings of the 19th GAMM-Seminar*, Max-Planck-Institute, Leipzig, 2003, 21–30.
- [5] Beylkin G., and Mohlenkamp M.J., Algorithms for numerical analysis in high dimensions, *SIAM Journal of Scientific Computing*, 26, 2005, 2133–2159.
- [6] Yserentant H., Sparse grid spaces for the numerical solution of the electronic Schrodinger equation, *Numer. Math.*, 101, 2005, 381–389.
- [7] Stephenson G., *Mathematical methods for science students (2nd ed.)* (UK: Longman, 1973).
- [8] Ames W. F., *Numerical methods for partial differential equations. (2nd ed.)* (Great Britain: Thomas Nelson & Sons Ltd, 1977) pp 365.
- [9] Jacobi C.G.J., *Gesammelte werke* (Berlin: G. Reimer, 1844) pp 467.
- [10] Seidel L., Abh bayer akad wiss, *Naturwiss. Kl*, 11(3), 1874, 81-108.
- [11] Young D.M., *Iterative Methods for solving partial differential equations of elliptic type*, Doctoral Thesis, Havard University, Cambridge, MA, 1950.
- [12] Hadjidimos A., Accelerated overrelaxation method, *Mathematics of Computation*, 141(32), 1978, 149-157.
- [13] Ndanusa A., and Adeboye, K.R., Preconditioned SOR Iterative Methods for L-matrices, *American Journal of Computational and Applied Mathematics*, 2(6), 2012, 300-305.
- [14] Vatti V.B.K., Rao G.C., and Pai S.S., Reaccelerated overrelaxation (ROR) method. *Bulletin of the International Mathematical Virtual Institute*, 10(2), 2020, 315-324.
- [15] Isah I.O., Ndanusa A., Shehu M.D., and Yusuf A., Parametric reaccelerated overrelaxation (PROR) method for numerical solution of linear systems, *Science World Journal*, 17(1), 2022, 60-64,
- [16] Miller J., and Bancroft J.C., Multigrid Principles, *CREWES Research Report*, 15, 2003, 1-32.
- [17] Taft, R.K., *A comparison of two multi-grid methods for solving the Poisson problem*, Doctoral diss., Department of Atmospheric Science Colorado State University Fort Collins, CO 80523, 1987.
- [18] Gupta M.M., Koutatchou J., and Zhang J., Comparison of second and fourth order discretizations for multigrid Poisson solvers, *Journal of Computational Physics*, 132(2), 1997, 226–232.
- [19] Yavneh I., Multigrid smoothing factors for red-black Gauss–Seidel relaxation applied to a class of elliptic operators, *SIAM Journal of Numerical Analysis*, 32, 1995, 1126–1138.
- [20] Yavneh I., On red-black SOR smoothing in multigrid, *SIAM Journal of Scientific Computing*, 17, 1996, 180–192.
- [21] Zhang J., Fast and high accuracy multigrid solution of the three dimensional Poisson equation, *Journal of Computational Physics*, 143, 1998, 449–461.

- [22] Trottenberg U., Oosterlee C.W., and Schuller A., *Multigrid* (Academic Press, London, 2001).
- [23] bin Zubair H., MacLachlan S. P., and Oosterlee C. W., A geometric multigrid method based on L-shaped coarsening for PDEs on stretched grids, *Numerical Linear Algebra with Applications, Wiley Interscience*, 2009.
- [24] McAdams A., Sifakis E., & Teran J., A parallel multigrid Poisson solver for fluids simulation on large grids, *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2010.
- [25] Lan B., Ge Y., Wang Y., and Zhan Y., High order compact difference scheme and multigrid method for 2D elliptic problems with variable coefficients and interior/boundary layers on non-uniform grids, *Journal of Applied Mathematics and Physics*, (3), 2015, 509-523.
- [26] Butt M.M., and Yuan Y., A full multigrid method for distributed control problems constrained by stokes equations, *Numer. Math. Theor. Meth. Appl*, 10 (3), 2017, 639-655.
- [27] Wang Y., and Ge Y., High-order compact difference scheme and multigrid method for solving the 2D elliptic problems, *Mathematical Problems in Engineering*, 2018, 1-11.
- [28] Ghaffar F., Ullah S., and Badshah N., Multigrid method with eighth-order compact finite difference scheme for Helmholtz equation, Accepted for publication in *Physica Scripta*, 2020.
- [29] Butt M.M., Multigrid method for optimal control problem constrained by stochastic Stokes equations with noise, *Mathematics*, 9, 2021, 1-13.
- [30] Bewley T., Mashayek A., Cavaglieri D., and Luchini P., Tweed and wireframe: accelerated relaxation algorithms for multigrid solution of elliptic PDEs on stretched structured grids, *arXiv:2110.08389v1 [math.NA]*, 2021.
- [31] Di Pietro D.A., Hülsemann F., Matalon P., Mycek P., Rude U., and Ruiz D., Toward robust, fast solutions of elliptic equations on complex domains through HHO discretizations and non-nested multigrid methods, 2021.
- [32] Brandt A., Multi-level adaptive solutions to boundary-value problems, *Math. Comp.*, 31, 1977, 333–390.
- [33] Stuben K., and Trottenberg U., *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications* (Springer, Berlin, 1982).
- [34] Reisinger C., and Wittum G., On the construction of fast solvers for elliptic equations, *Technical report*, von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, 1982.
- [35] Wesseling P., *An introduction to multigrid methods, Pure Appl. Math.* (John Wiley & Sons, New York, 1992).
- [36] Wienands R., and Joppich W., *Practical Fourier analysis for multigrid methods, numerical insights* (Chapman and Hall/CRC, Boca Raton, FL, 2005).
- [37] Wienands R., *Extended local Fourier analysis for multigrid: optimal smoothing, coarse grid correction, and preconditioning*, Ph.D. thesis, University of Cologne, 2001.