

On the Robustness of the xy -Zebra-Gauss-Seidel Smoother on an Anisotropic Diffusion Problem

Michely Laís de Oliveira^{1,*}, Marcio Augusto Villela Pinto², Simone de Fátima Tomazzoni Gonçalves² and Grazielli Vassoler Rutz³

Abstract: Studies of problems involving physical anisotropy are applied in sciences and engineering, for instance, when the thermal conductivity depends on the direction. In this study, the multigrid method was used in order to accelerate the convergence of the iterative methods used to solve this type of problem. The asymptotic convergence factor of the multigrid was determined empirically (computer aided) and also by employing local Fourier analysis (LFA). The mathematical model studied was the 2D anisotropic diffusion equation, in which $\varepsilon > 0$ was the coefficient of a nisotropy. The equation was discretized by the Finite Difference Method (FDM) and Central Differencing Scheme (CDS). Correction Scheme (CS), pointwise Gauss-Seidel smoothers (Lexicographic and Red-Black ordering), and line Gauss-Seidel smoothers (Lexicographic and Zebra ordering) in x and y directions were used for building the multigrid. The best asymptotic convergence factor was obtained by the Gauss-Seidel method in the direction x for $0 < \varepsilon \ll 1$ and in the direction y for $\varepsilon \gg 1$. In this sense, an xy -zebra-GS smoother was proposed, which proved to be efficient and robust for the different anisotropy coefficients. Moreover, the convergence factors calculated empirically and by LFA are in agreement.

Keywords: Physical anisotropy, diffusion problem, finite difference method, multigrid, local Fourier analysis, Gauss-Seidel zebra.

1 Introduction

Numerical simulations on engineering application problems stand out in the research and development of technology. Computational Fluid Dynamics (CFD) is the area of scientific computing that studies numerical methods in order to simulate problems involving fluids in motion with or without heat exchange.

¹ Graduate Program in Numerical Methods in Engineering, Federal University of Paraná, 81531-980, Curitiba-PR, Brazil.

² Department of Mechanical Engineering, Federal University of Paraná, 81531-980, Curitiba-PR, Brazil.

³ Federal Institute of Education, Science and Technology of Santa Catarina, 89813-000, Chapecó-SC, Brazil.

* Corresponding Author: Michely Laís de Oliveira. Email: michelylais@ufpr.br.

In CFD, mathematical models can be discretized by the Finite Difference Method (FDM) [Ferziger and Peric (2002)], which aims to approximate each term of the mathematical model by means of algebraic expressions at each point of the grid, thus generating a linear system of algebraic equations, such as:

$$AT = b, \tag{1}$$

where A is the matrix of the coefficients of size $N \times N$, T is the variable of interest, and b is the vector of independent terms, also known as source term. In order to obtain good results in CFD, it is necessary to reduce the discretization error, which implies in the use of very refined grids thus leading to increased computational costs. Another factor that increases CPU time (t_{CPU}) is that the discretization of these problems results in systems of algebraic equations in which the matrix of the coefficients have a high degree of sparsity [Burden, Faires and Burden (2016)].

In order to solve Eq. (1), direct or iterative methods, hereinafter called solvers, can be employed. As the system of algebraic equations generated is sparse and large, direct methods are not feasible due to their high computational cost [Burden, Faires and Burden (2016)]. In this case, iterative methods are chosen. Such methods start from an initial guess after successive approximations and converge to a solution under certain hypotheses [Burden, Faires and Burden (2016)].

It is known that only high frequency components (oscillatory modes) of the error are reduced promptly in iterative solvers, leaving low frequency components (smooth modes) unaffected [Briggs, Henson and McCormick (2000); Trottenberg, Oosterlee and Schuller (2001); Wesseling (2004)]. Solvers that have this smoothing property are called smoothers.

The multigrid is an efficient method to accelerate the convergence rate of the smoother [Briggs, Henson and McCormick (2000); Trottenberg, Oosterlee and Schuller (2001); Wesseling (2004); Hackbusch (2003)]. It was developed with the purpose of helping iterative methods eliminate smooth components of the error. According to Trottenberg et al. [Trottenberg, Oosterlee and Schuller (2001)], smooth components in a given grid become more oscillatory in the perspective of a coarser grid. The multigrid consists of a set of grids with different degrees of refinement, which are visited during the iterative process, thus effectively reducing the entire frequency spectrum of the error.

Anisotropic problems [Wienands and Joppich (2005)] are common in engineering. In such cases, the efficiency of the multigrid method is reduced [Briggs, Henson and McCormick (2000); Trottenberg, Oosterlee and Schuller (2001)]. Physical (or coefficient) anisotropy occurs, for instance, when the differential equation has constant coefficients in the partial derivatives, but distinct coefficients in the coordinate directions. An example of physical anisotropy occurs in studies involving a material in which the thermal conductivity depends on the direction.

Several studies found in the literature concern the multigrid in anisotropic problems. Wienands et al. [Wienands and Joppich (2005)] calculated the convergence factor of the

multigrid using local Fourier analysis (LFA) in different problems, including those with anisotropic diffusion. As LFA provides good inferred estimates for convergence rates and allows the prediction of multigrid performance, those authors performed the analysis with different smoothers as well as different restriction and prolongation operators. An introduction to LFA can also be found in Trottenberg et al. [Trottenberg, Oosterlee and Schuller (2001); Wesseling (2004); Wienands and Joppich (2005)].

Lew et al. [Lew, Wolters, Dierkes et al. (2009)] compared the efficiency of Jacobi preconditioners, Incomplete Cholesky and Algebraic Multigrid for the conjugated gradients method (AMG-CG) with the purpose of solving iteratively an anisotropic problem applied to electroencephalography. In this study, the results of the AMG-CG method reached one order of magnitude below that of the other standard preconditioners in what concerns to the accuracy to achieve the same level of discretization error.

Gee et al. [Gee, Hu and Tumnaro (2009)] studied anisotropic problems and introduced a proposal for the prolongation operator with the use of the algebraic multigrid method. The authors employed a new solver called Aggregation Smoother, which was used to imitate the semi-coarsening only in directions of strong coupling of the anisotropy coefficient. Their proposal proved efficient for the studied problems.

Oliveira et al. [Oliveira, Pinto and Marchi (2012)] presented a study on geometric anisotropy involving several levels of grid refinement and aspect ratios Q (the ratio between spacings in x and y). The authors also studied some multigrid parameters, such as: Smoothers, restriction, number of levels, and number of pre- and post-smoothing sweeps, besides several coarsening algorithms. From this study, the authors proposed the partial semi-coarsening method, which presented the best performance among those studied. The partial semi-coarsening method was 50 times faster than full-coarsening method when $Q = 16$.

Peherstorfer et al. [Peherstorfer and Bungartz (2012)] employed the multigrid method to solve problems with geometric anisotropy. Considering the semi-coarsening in space and time with Q -cycle, which was proposed in the work, they were able to obtain efficient results. For instance, in one simulation, a Q -cycle was equivalent to ten V -cycles.

Gmeiner et al. [Gmeiner, Gradl, Gaspar et al. (2013)] performed LFA for multigrid methods on tetrahedral grids and four-color smoother was presented as the most efficient.

Dedner et al. [Dedner, Muller and Scheichl (2014)] studied the multigrid method preconditioned to solve anisotropic problems in geophysical models. As the convergence rates obtained were close to 0.1, they concluded that the multigrid method was efficient to solve the pressure correction equation found in numerical weather prediction models.

Vassoler-Rutz et al. [Vassoler-Rutz and Pinto (2016)] analyzed the effect of physical anisotropy on the multigrid method in a diffusion anisotropy problem. They calculated the asymptotic convergence factor and smoothing factor via LFA and presented a complexity analysis. The authors used Red-Black Gauss-Seidel (GS-RB) and lexicographic Gauss-Seidel (GS-LEX) smoothers and concluded that for strong anisotropies, the complexity order of the multigrid becomes poor in both smoothers analyzed.

Pinto et al. [Pinto, Rodrigo, Gaspar et al. (2016)] showed the robustness of the ILU smoother (incomplete LU decomposition) in some problems, including an anisotropic diffusion problem in triangular grids.

Recent LFA studies on transient problems were carried out by Franco et al. [Franco, Gaspar, Pinto et al. (2018)], who employed the analysis in the study of space-time anisotropy for the 1D and 2D Fourier equation and poroelasticity problems by Franco et al. [Franco, Rodrigo, Gaspar et al. (2018)].

In this work, we analyzed the asymptotic convergence factor of the multigrid, which was determined both empirically and by LFA. The mathematical model considered was the two-dimensional anisotropic diffusion equation. In order to discretize equations, the FDM was used with Central Differencing Scheme (CDS). The algebraic equation systems resulting from the discretization were solved using the smoothers GS-LEX, GS-RB, *x*-line-GS, *y*-line-GS, *x*-zebra-GS and *y*-zebra-GS [Wienands and Joppich (2005)]. Full-Weighting (FW) and bilinear interpolation were used as restriction and prolongation operators, respectively [Trottenberg, Oosterlee and Schuller (2001)].

This article is organized as follows: Section 2 presents the mathematical model for the anisotropic diffusion problem and the detailing of the numerical model, as well as the discretization process of the equation and the boundary conditions employed. Section 3 presents solving methods for systems of linear equations (solvers). The multigrid method is defined in Section 4. The LFA and the details for calculating the asymptotic convergence factor for each smoother studied are included in Section 5. Section 6 contains the numerical results and analyses. The conclusion of this work is presented in Section 7.

2 Mathematical and numerical models

The mathematical model considered in this study refers to the two-dimensional anisotropic diffusion problem in the domain $\Omega = [0, 1] \times [0, 1]$ given by the following equation [Briggs, Henson and McCormick (2000)]:

$$\begin{cases} -T_{xx} - \varepsilon T_{yy} = S, 0 \leq x, y \leq 1 \\ T(0, y) = T(x, 0) = T(x, 1) = T(1, y) = 0 \end{cases} \quad (2)$$

where T is the dependent variable that represents the temperature; T_{xx} and T_{yy} are the second order derivatives of T with respect to x and y , respectively; S is the source term; and the anisotropy coefficient is $\varepsilon > 0$.

For the boundary conditions, the source term S and the analytical solution T are given by:

$$S(x, y) = 2 \left[(1 - 6x^2) y^2 (1 - y^2) + \varepsilon (1 - 6y^2) x^2 (1 - x^2) \right], \quad (3)$$

$$T(x, y) = (x^2 - x^4) (y^4 - y^2). \quad (4)$$

Eq. (2) was discretized by FDM based on the second-order CDS, which resulted in:

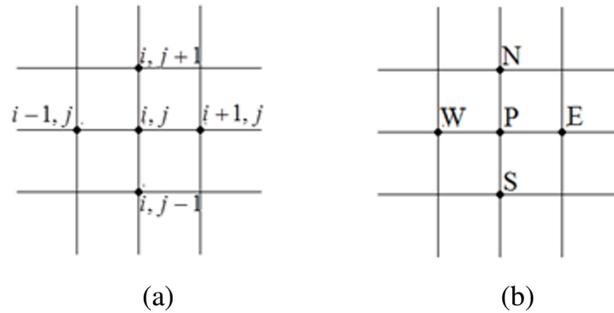


Figure 1: Five points in a uniform two-dimensional grid

$$a_P T_P + a_W T_W + a_N T_N + a_E T_E + a_S T_S = b_P, \tag{5}$$

where the subindices represent the location of the points in the grid. The points P (central), W (west), E (east), N (north) and S (south) shown in Fig. 1(b) correspond to the points (i, j) , $(i - 1, j)$, $(i + 1, j)$, $(i, j + 1)$, $(i, j - 1)$ in Fig. 1(a), respectively.

Considering $h_x=1/(N_x - 1)$ and $h_y=1/(N_y - 1)$, where N_x and N_y denote the number of points in the coordinate directions x and y , respectively, including the boundaries, we have that:

$$a_P = \left(\frac{2}{h_x^2} + \frac{2\varepsilon}{h_y^2} \right), a_W = a_E = -\frac{1}{h_x^2}, a_N = a_S = -\frac{\varepsilon}{h_y^2}, b_P = S_P, \tag{6}$$

for the inner points ($P = 2, \dots, N - 1$) and $a_P = 1, a_W = a_S = a_N = a_E = 0, b_P = S_P$ in all boundaries.

3 Iterative methods and their orderings

Iterative methods are considered efficient in solving sparse and large systems [Burden, Faires and Burden (2016)]. These methods are based on an initial guess for the solution, from which a sequence of approximations is built. Under certain conditions, the estimate converges to the exact solution of the system. In this work, the Gauss-Seidel method and some of its variants were used to solve the system of equations given by Eq. (5).

The Gauss-Seidel method can be classified as a pointwise solver and also as a multi-block solver (line by line, for instance). The difference between each solver is related to the way the unknowns are updated throughout the iterative process.

In pointwise methods, each variable is updated individually. Fig. 2 presents two classic orderings for the Gauss-Seidel pointwise method [Wienands and Joppich (2005)]: (a) Lexicographic ordering, and (b) Red-black ordering. Both orderings were used for comparison purposes.

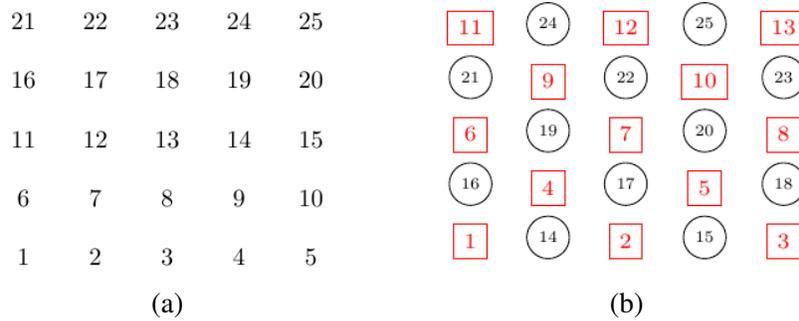


Figure 2: (a) Lexicographic; (b) Red-black orderings

In multi-block methods, each block is updated one at a time. Fig. 3 shows two orderings for the Gauss-Seidel multi-block method: (a) x -line-GS and (b) y -line-GS. Fig. 4 depicts (a) x -zebra-GS and (b) y -zebra-GS.

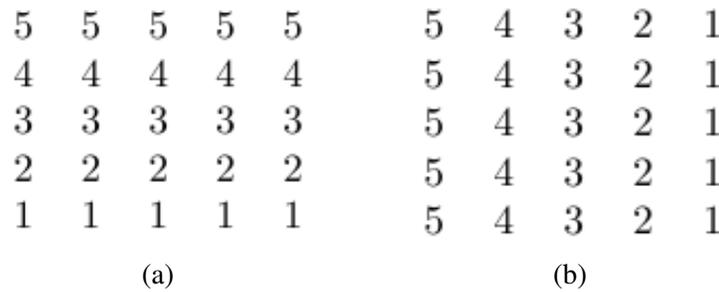


Figure 3: Line ordering in the directions x (a) and y (b)

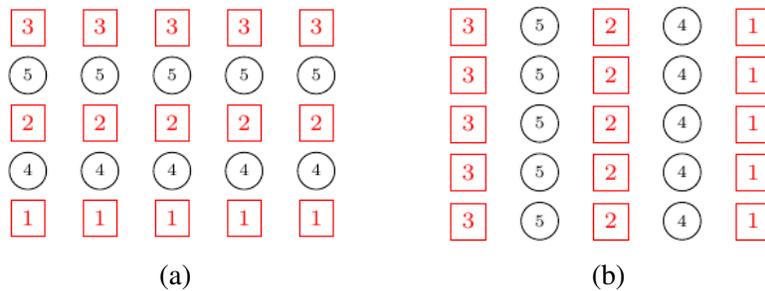


Figure 4: Zebra ordering in the directions x (a) and y (b)

4 Multigrid method

In order to efficiently reduce the discretization error, very refined grids are used, thus generating large linear systems. Therefore, the multigrid is an alternative method to accelerate the convergence rate of the problems [Briggs, Henson and McCormick (2000)].

The basic principles of the multigrid method are its smoothing and fine grid correction properties. Such technique involves transferring information of one problem through several grids, from the finest to the coarsest, so that all error frequencies are smoothed out. The transfer of information between grids is carried out by the restriction and prolongation operators. The way in which these grids are visited is called the multigrid cycle, such as the V-cycle shown in Fig. 5. Several cycles can be executed successively until a stop criterion verified at the end of each cycle is achieved.

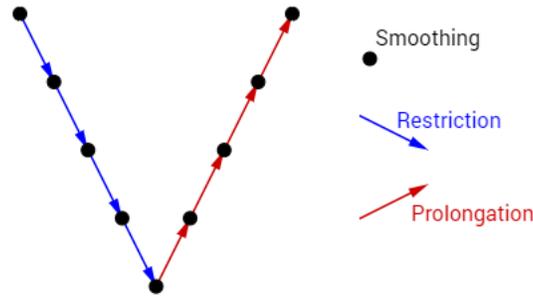


Figure 5: Diagram of a V-cycle applied to a problem with 5 grid levels

Eq. (1) was solved by the multigrid method, by employing CS scheme, V-cycle and zero initial guess.

Considering uniform grids, the grid coarsening ratio is defined as $r = p/q$, where q measures the spacing between the points of the fine grid (Ω^h) and p measures the spacing between the points of the immediately coarser grid, denoted by Ω^H . In this study, $r = 2$ was used (standard coarsening) [Wesseling and Oosterlee (2001)], thus $H = 2h$. For study purposes, the linear system obtained from the discretization was solved by the different solvers (smoothers) with good smoothing properties described in Section 3.

5 Local Fourier Analysis (LFA)

LFA was carried out in order to determine the convergence factor for the two-grid algorithm [Trottenberg, Oosterlee and Schuller (2001)], which estimates the behavior of the asymptotic convergence of the multigrid.

The asymptotic convergence factor $\rho(M_h^{2h}) = \rho_{2g}$ is the spectral radius of the matrix M_h^{2h} , where $M_h^{2h} := S_h^{\nu_2} K_h^{2h} S_h^{\nu_1}$ is the operator of two grids. Taking S_h as the smoothing operator, ν_1 and ν_2 represent the number of pre- and post-smoothing iterations, respectively. The correction operator of the coarse grid is given by $K_h^{2h} = I_h - I_{2h}^h (L_{2h})^{-1} I_h^{2h} L_h$, where I_{2h}^h, I_h^{2h} are the prolongation and restriction transfer operators, respectively, L_{2h}, L_h are the discrete Laplacian operators, and I_h , the identity operator in the fine grid. In this study, LFA was used to determine the asymptotic convergence factor of the multigrid, considering the smoothers GS-LEX, GS-RB, *x*-line-GS, *y*-line-GS, *x*-zebra-GS and *y*-zebra-GS.

The I_h^{2h} restriction operator is characterized by

$$I_h^{2h} w_h(x) = \sum_{\kappa \in V} \hat{t}_\kappa e^{i\theta^\alpha \kappa}, \quad (\kappa = (\kappa_1, \kappa_2) \in \mathbb{Z}^2),$$

where, $t_\kappa \in \mathbb{R}$ are stencil coefficients (more details in Trottenberg et al. [Trottenberg, Oosterlee and Schuller (2001); Wienands and Joppich (2005)]).

The I_h^{2h} full-weighting operator was evaluated by:

$$I_h^{2h}(\theta^\alpha) = \frac{1}{4}(1 + \cos \bar{\theta}_1)(1 + \cos \bar{\theta}_2)$$

where $\theta = (\theta_1, \theta_2) \in F^{low} = [\pi/2, \pi/2]^2$, in which F^{low} represents the low-frequency components, considering

$$\bar{\theta}_i := \begin{cases} \theta_i + \pi & \text{se } \theta_i < 0 \\ \theta_i - \pi & \text{se } \theta_i \geq 0 \end{cases}.$$

These components are illustrated in Fig. 6 [Trottenberg, Oosterlee and Schuller (2001)]. The blank region represents the low frequencies, with components represented by (\circ), and the hatched region represents the high frequencies, with the components represented by (\bullet).

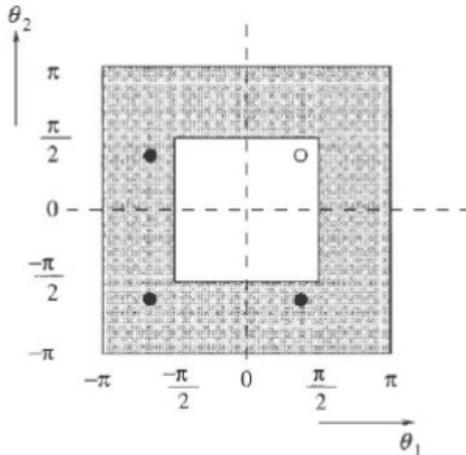


Figure 6: Low and high frequencies [Trottenberg, Oosterlee and Schuller (2001)]

The I_{2h}^h bilinear operator was employed, $I_{2h}^h = I_h^{2hT}$.

The operator L_h is represented by the matrix 4×4 :

$$L_h = \begin{bmatrix} \tilde{L}_h(\theta^{(0,0)}) & & & \\ & \tilde{L}_h(\theta^{(1,1)}) & & \\ & & \tilde{L}_h(\theta^{(1,0)}) & \\ & & & \tilde{L}_h(\theta^{(0,1)}) \end{bmatrix},$$

where \tilde{L}_h are eigenvalues, evaluated by

$$\tilde{L}_h(\theta) = \sum_{\kappa} s_{\kappa} e^{i\theta^{\alpha} \kappa}, \tag{7}$$

where, $s_{\kappa} \in \mathbb{R}$ are stencil coefficients (more details in Trottenberg et al. [Trottenberg, Oosterlee and Schuller (2001); Wienands and Joppich (2005)]).

The discrete Laplace operators L_h and L_{2h} are represented, respectively, by:

$$L_h = \frac{1}{h^2} \begin{bmatrix} L_1 & & & \\ & L_2 & & \\ & & L_3 & \\ & & & L_4 \end{bmatrix},$$

$$L_1 = (2 + 2\varepsilon) - 2(\varepsilon \cos(\theta_1) + \cos(\theta_2)),$$

$$L_2 = (2 + 2\varepsilon) - 2(\varepsilon \cos(\bar{\theta}_1) + \cos(\bar{\theta}_2)),$$

$$L_3 = (2 + 2\varepsilon) - 2(\varepsilon \cos(\bar{\theta}_1) + \cos(\theta_2)),$$

$$L_4 = (2 + 2\varepsilon) - 2(\varepsilon \cos(\theta_1) + \cos(\bar{\theta}_2)) \quad \text{and}$$

$$L_{2h} = \frac{(2 + 2\varepsilon) - 2(\varepsilon \cos(2\theta_1) + \cos(2\theta_2))}{2h^2}.$$

The smoothing operator S_h is given by:

$$S_h = \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) & 0 & 0 & 0 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) & 0 & 0 \\ 0 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) & 0 \\ 0 & 0 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) \end{bmatrix}, \tag{8}$$

where

$$\tilde{S}_h(\theta) := -\frac{\tilde{L}_h^-(\theta)}{\tilde{L}_h^+(\theta)} = \frac{L_h^- e^{i\theta \cdot x/h}}{L_h^+ e^{i\theta \cdot x/h}}. \tag{9}$$

Considering that $L_h = L_h^+ + L_h^-$, where, L_h^- represents approximation before the relaxation step and L_h^+ after the step.

For the GS-LEX smoother, $\tilde{S}(x, y) = (\varepsilon e^{ix} + e^{iy}) / (-2 - 2\varepsilon + \varepsilon e^{-ix} + e^{-iy})$.

For the GS-RB smoother, the smoothing operator S_h^{RB} is given by the product $S_h^{RB} = S_h^{BLACK} S_h^{RED}$, where

$$S_h^{RED} = \frac{1}{2} \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) + 1 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) - 1 & 0 & 0 \\ \tilde{S}(\theta_1, \theta_2) - 1 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & 0 & 0 \\ 0 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & \tilde{S}(\theta_1, \bar{\theta}_2) - 1 \\ 0 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & \tilde{S}(\theta_1, \bar{\theta}_2) + 1 \end{bmatrix},$$

$$S_h^{BLACK} = \frac{1}{2} \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) + 1 & -\tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & 0 & 0 \\ -\tilde{S}(\theta_1, \theta_2) + 1 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & 0 & 0 \\ 0 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & -\tilde{S}(\theta_1, \bar{\theta}_2) + 1 \\ 0 & 0 & -\tilde{S}(\bar{\theta}_1, \theta_2) + 1 & \tilde{S}(\theta_1, \bar{\theta}_2) + 1 \end{bmatrix},$$

with $\tilde{S}(x, y) = (1 - \omega(1 + \varepsilon - \varepsilon \cos x - \cos y))/4$.

The GS-line smoother uses the operator given by Eq. (8), where $\tilde{S}(x, y) = (e^{iy})/(-2 - 2\varepsilon + \varepsilon e^{-ix} + e^{-iy} + \varepsilon e^{ix})$ for the direction x and $\tilde{S}(x, y) = (e^{ix})/(-2 - 2\varepsilon + \varepsilon e^{-ix} + e^{-iy} + \varepsilon e^{iy})$ for the direction y [Wienands and Joppich (2005)].

The x -zebra-GS smoother, similarly to GS-RB, is given by:

$$S_h^{RED} = \frac{1}{2} \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) + 1 & 0 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) - 1 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & \tilde{S}(\bar{\theta}_1, \theta_2) - 1 & 0 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) - 1 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & 0 \\ \tilde{S}(\theta_1, \theta_2) - 1 & 0 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) + 1 \end{bmatrix},$$

$$S_h^{BLACK} = \frac{1}{2} \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) + 1 & 0 & 0 & -\tilde{S}(\theta_1, \bar{\theta}_2) + 1 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & -\tilde{S}(\bar{\theta}_1, \theta_2) + 1 & 0 \\ 0 & -\tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & 0 \\ -\tilde{S}(\theta_1, \theta_2) + 1 & 0 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) + 1 \end{bmatrix},$$

with, $\tilde{S}(x, y) = \cos y/(1 + \varepsilon - \varepsilon \cos x)$ (Wienands and Joppich (2005)). For the y -zebra-GS, we have:

$$S_h^{RED} = \frac{1}{2} \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) + 1 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) - 1 & 0 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) - 1 \\ \tilde{S}(\theta_1, \theta_2) - 1 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & 0 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) - 1 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) + 1 \end{bmatrix},$$

$$S_h^{BLACK} = \frac{1}{2} \begin{bmatrix} \tilde{S}(\theta_1, \theta_2) + 1 & 0 & -\tilde{S}(\bar{\theta}_1, \theta_2) + 1 & 0 \\ 0 & \tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & 0 & -\tilde{S}(\theta_1, \bar{\theta}_2) + 1 \\ -\tilde{S}(\theta_1, \theta_2) + 1 & 0 & \tilde{S}(\bar{\theta}_1, \theta_2) + 1 & 0 \\ 0 & -\tilde{S}(\bar{\theta}_1, \bar{\theta}_2) + 1 & 0 & \tilde{S}(\theta_1, \bar{\theta}_2) + 1 \end{bmatrix},$$

with $\tilde{S}(x, y) = \varepsilon \cos x/(1 + \varepsilon - \cos y)$ [Wienands and Joppich (2005)].

6 Numerical results

The results presented refer to the Eqs. (2), (3) and (4), with the domain $\Omega = [0, 1] \times [0, 1]$. It is expected that these results will be extended to larger domains.

In order to accelerate the convergence of the iterative methods, the geometric multigrid was used with zero initial guess, V-cycle and number of pre- and post- smoothing iterations $\nu_1 = \nu_2 = 1$ and $\nu = \nu_1 + \nu_2$. Since the problem investigated is linear, the multigrid was employed with the correction scheme, which is recommended for this case [Briggs, Henson and McCormick (2000)]. FW and bilinear interpolation were used as restriction and prolongation operators, respectively.

The infinity norm of the residual was used as stop criterion of the iterative process, nondimensionalized by the initial guess, that is, $\|R^n\|_\infty / \|R^0\|_0 \leq tol$, where R^n is the residual in the iteration n , R^0 is the residual in the initial guess and $tol = 10^{-9}$ is the tolerance adopted. For every size of the problem N , the number of L levels employed was L_{max} , in which L_{max} denotes the highest possible number of levels that can be employed for a certain problem, considering $N = 3 \times 3$ points in the coarsest grid. For instance, if $N = 33 \times 33$ and $r = 2$, the following group of grids is obtained: $33 \times 33, 17 \times 17, 9 \times 9, 5 \times 5, 3 \times 3$, that is, $L_{max} = 5$.

The algorithms from this study concerning to the simulations were computed in Fortran 2003 language, using the Intel 9.1 Visual Fortran compiler with double precision. However, the algorithms concerning to the LFA were computed in MATLAB R2015a language. All simulations were performed in a computer with Intel Core i7 2.6 GHz processor, 8 GB RAM, 64-bit Windows 10, using double precision arithmetic.

The t_{CPU} was measured based on the $CPUtime$ function from Fortran, in which t_{CPU} is defined as the time interval needed for generating the grids, establishing the initial guess, computing the coefficients and solving the linear system represented in Eq. (1) until the stop criterion established is achieved.

6.1 Preliminary results

In order to assess the level of reliability of the numerical solution, tests were performed to verify the codes that are part of the different smoothers studied.

Fig. 7 presents the infinity norm of the numerical error *versus* $N = 5 \times 5$ up to 2049×2049 for all the smoothers studied, with anisotropy factor fixed at $\varepsilon = 10^{-3}$. The results obtained show that, regardless of the smoother used, the errors were essentially the same (the biggest difference was of the order of 4.0×10^{-9}) and decreased with the grid refinement. Fig. 8 depicts the numerical error for the anisotropy coefficients assessed with $N = 5 \times 5$ up to 2049×2049 , with fixed smoother (*x*-zebra-GS). Likewise, regardless of the coefficient of anisotropy, the errors were virtually the same (the biggest difference was of the order of 6.0×10^{-8}) and decreased with the grid refinement.

According to Marchi et al. [Marchi and Silva (2002)], the apparent and effective orders of the numerical error are given, respectively, by:

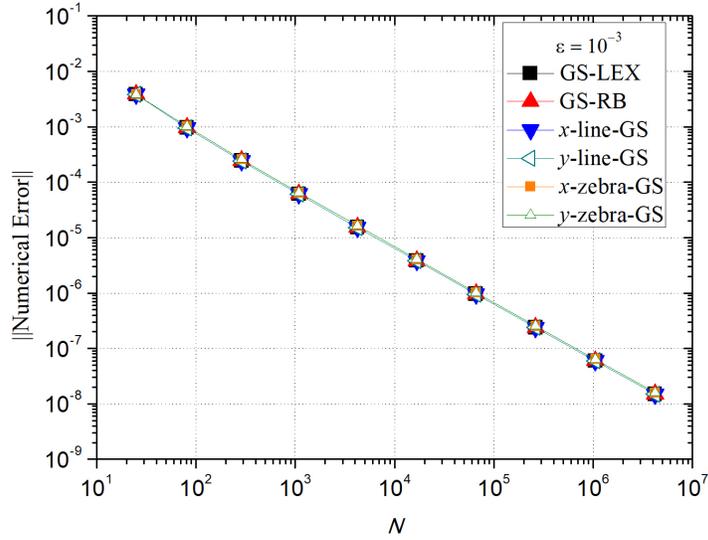


Figure 7: Infinite norm of the numerical error vs. N for the different smoothers, with anisotropy coefficient $\varepsilon = 10^{-3}$

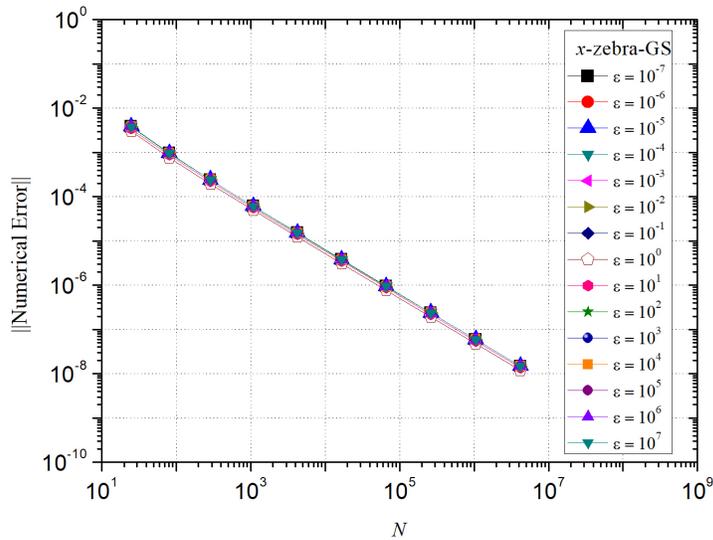


Figure 8: Numerical error vs. N for the x -zebra-GS smoother with several anisotropy coefficients

$$p_U = \frac{\log\left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2}\right)}{\log(q)} \quad \text{and} \quad p_E = \frac{\log\left(\frac{\Phi - \phi_2}{\Phi - \phi_1}\right)}{\log(q)},$$

where ϕ_1, ϕ_2 e ϕ_3 represent three solutions in three distinct grids with sizes h_1, h_2 and h_3 , fine, coarse and super coarse grids, respectively, Φ is the analytical solution and $q = h_2/h_1 = h_3/h_2$ is the refinement ratio.

Fig. 9 shows that the apparent (p_U) and effective (p_E) orders for the infinity norm of the numerical error for the x -zebra-GS smoother tend to the asymptotic order, $p_L = 2$ (due to the use of second order CDS) when the grid is refined. Tests carried out with different anisotropy coefficients and other smoothers demonstrated similar behavior to the one depicted in Fig. 9.

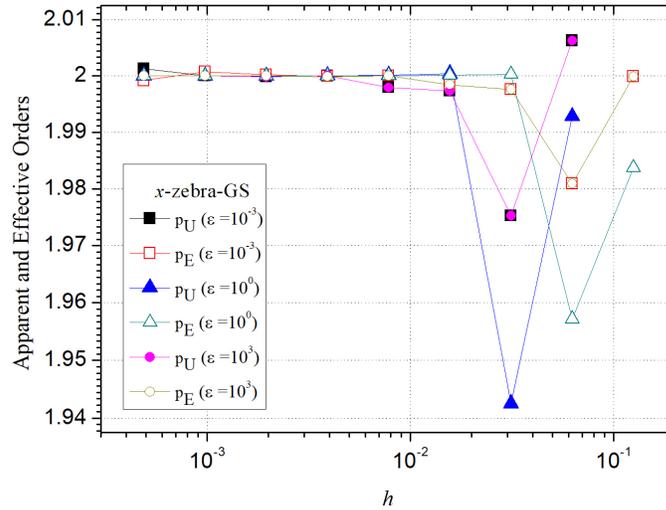


Figure 9: Apparent and effective orders vs. h for the infinity norm of the error, with x -zebra-GS smoother and the anisotropy coefficients $\epsilon = 1, 10^{-3}$ and 10^3

6.2 Asymptotic convergence factor

Fig. 10 shows the convergence factor of two grids (ρ_{2g}) calculated via LFA for each one of the smoothers assessed, for a problem $N = 1025 \times 1025$. This data is presented for different anisotropy coefficients (ϵ). When the problem is isotropic ($\epsilon = 1$), all solvers converge, $0 < \rho_{2g} \leq 0.2$. It is possible to observe that for anisotropic problems, when $\epsilon \gg 1$, only the y -line-GS and y -zebra-GS smoothers present good convergence factors, that is, $\rho_{2g} \approx 0$, however, the GS-LEX, GS-RB, x -line-GS and x -zebra-GS smoothers diverge, that is, $\rho_{2g} \approx 1$. On the other hand, for $\epsilon \ll 1$, x -line-GS and x -zebra-GS present

better convergence factors than the other smoothers studied (GS-LEX, GS-RB, y -line-GS and y -zebra-GS), which confirms that block smoothers are more efficient than pointwise smoothers. Moreover, for $\varepsilon \gg 1$, there is strong coupling in the direction y (see Eq. (2)), therefore, line smoothers in this direction are more efficient. For $\varepsilon \ll 1$, there is strong coupling in the direction x , consequently, line smoothers in this direction are more efficient.

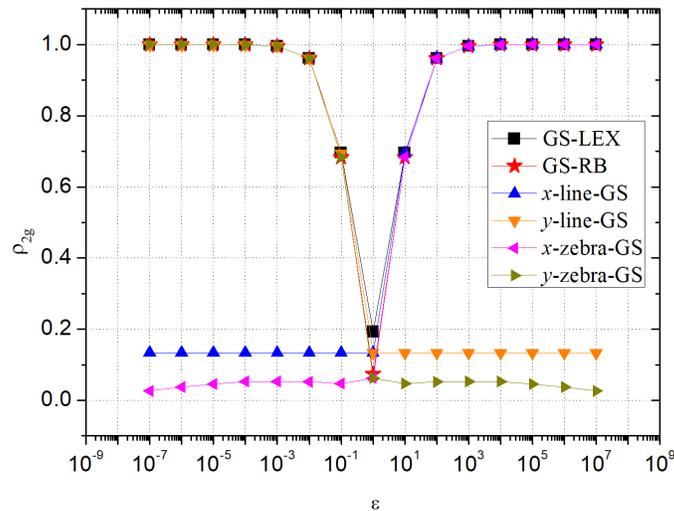


Figure 10: Asymptotic convergence factor via LFA (ρ_{2g}) vs. the anisotropy coefficient (ε)

According to Trottenberg et al. [Trottenberg, Oosterlee and Schuller (2001)], the computational cost (complexity) of the GS-line and GS-zebra smoothers is practically the same. However, one advantage of using the GS-zebra smoother is that its convergence factor for the multigrid is better than that of the GS-line smoother, as shown in Fig. 10.

Considering the presented results and with the aim of developing a robust and efficient algorithm for solving the anisotropic problem in question (Eq. (2)), a variation of the zebra smoother called xy -zebra-GS is proposed. In this case, the x -zebra-GS is employed when $\varepsilon \ll 1$ and the y -zebra-GS is employed when $\varepsilon \gg 1$, as shown by Algorithm 1.

For the proposed smoother, xy -zebra-GS, the asymptotic convergence factor of two grids (ρ_{2g}), estimated by LFA, was compared with the empirical convergence factor (ρ_h), computed for a fine grid, obtained with ten levels of refinement ($N = 1025 \times 1025$) and for different values of ε . Besides showing the robustness of Algorithm 1 in relation to different values of ε , Fig. 11 highlights that the LFA provides extremely accurate predictions (of the order of 0.3×10^{-1}) for the convergence factors, which confirms its importance as a preliminary analysis tool.

Algorithm 1: xy -zebra-GS

```

if  $\varepsilon \geq 1$  then
  | Apply the  $y$ -zebra-GS smoother;
else
  | Apply the  $x$ -zebra-GS smoother;
end
    
```

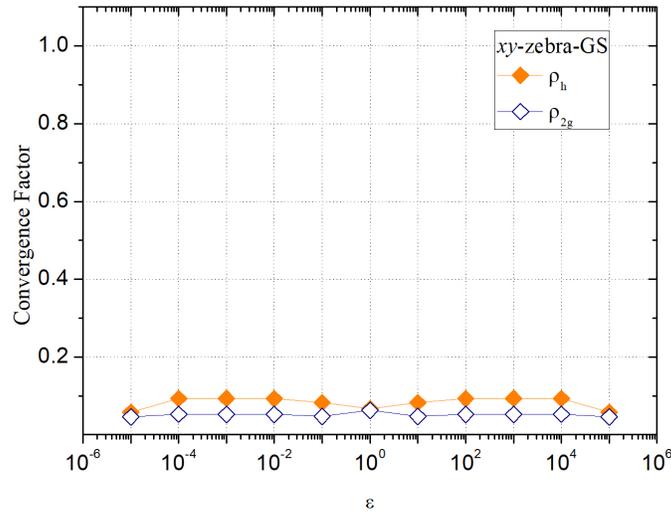


Figure 11: Asymptotic (ρ_{2g}) and empirical (ρ_h) convergence factors for $N = 1025 \times 1025$ vs. the anisotropy coefficient (ε)

The robustness of the proposed smoother, xy -zebra-GS, regarding the variation of N can be observed in Fig. 12, in which the asymptotic convergence factor ρ_{2g} , calculated via LFA, is compared with the ρ_h value, calculated empirically for different values of ε , in several grids (5×5 up to 2049×2049). Noticeably, for different values of ε and N , the convergence factor $\rho_h \approx 0.1$ presents good results.

The smoother given by Algorithm 1 was used in all the following tests due to its efficiency and robustness.

6.3 Number of V-cycles

The number of V-cycles for each size of problem (5×5 up to 2049×2049) is shown in Table 1 for the xy -zebra-GS smoother proposed in this study. Only results for $\varepsilon = 10^a$, $a \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ are presented, since results for $\varepsilon = 10^{-a}$ are quite similar. The number of V-cycles is roughly constant as the value of N increases, which is in agreement with the literature [Trottenberg, Oosterlee and Schuller (2001)]. In addition, it is possible to notice that the proposed smoother needs few V-cycles regardless of the number of points N .

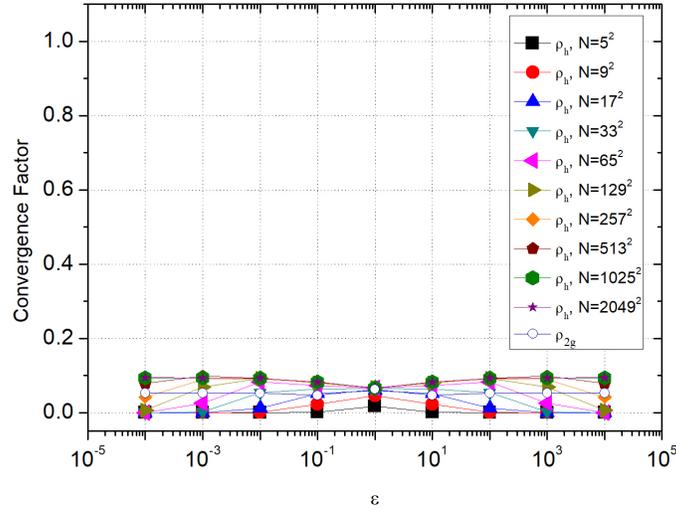


Figure 12: Asymptotic (ρ_{2g}) and empirical (ρ_h) convergence factor obtained via LFA for grids $N = 5 \times 5$ up to $N = 2049 \times 2049$ versus the anisotropy coefficient (ε)

Table 1: V-cycles as function of the number of points N and different values of ε

N	$\varepsilon = 10^0$	$\varepsilon = 10^1$	$\varepsilon = 10^2$	$\varepsilon = 10^3$	$\varepsilon = 10^4$	$\varepsilon = 10^5$	$\varepsilon = 10^6$	$\varepsilon = 10^7$
5×5	6	4	2	2	2	1	1	1
9×9	8	6	3	2	2	2	1	1
17×17	8	7	5	3	2	2	1	1
33×33	9	8	7	4	2	2	2	1
65×65	9	9	8	6	3	2	2	1
129×129	9	9	9	8	5	3	2	2
257×257	9	9	9	9	7	3	2	2
513×513	9	9	10	9	8	5	3	2
1025×1025	9	9	10	9	9	8	4	2
2049×2049	10	9	10	10	9	9	6	2

6.4 Computational cost

In order to assess the computational cost of the xy -zebra-GS smoother, a previous analysis of complexity based on the effect of the number of unknowns on the t_{CPU} was carried out. Consequently, several simulations were carried out considering isotropic and anisotropic problems with $\varepsilon = 10^a$ and $\varepsilon = 10^{-a}$, $a \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.

Fig. 13 depicts a graph of the t_{CPU} curves for some anisotropy coefficients (ε) for several values of N . It was observed that the t_{CPU} curves obtained for $\varepsilon = 10^{-a}$ are similar to the

ones obtained for $\varepsilon = 10^a$, in which $a \in \{0, 1, 2, 3, 4, 5, 6, 7\}$, therefore, only the odd part of the latter are shown in the figure.

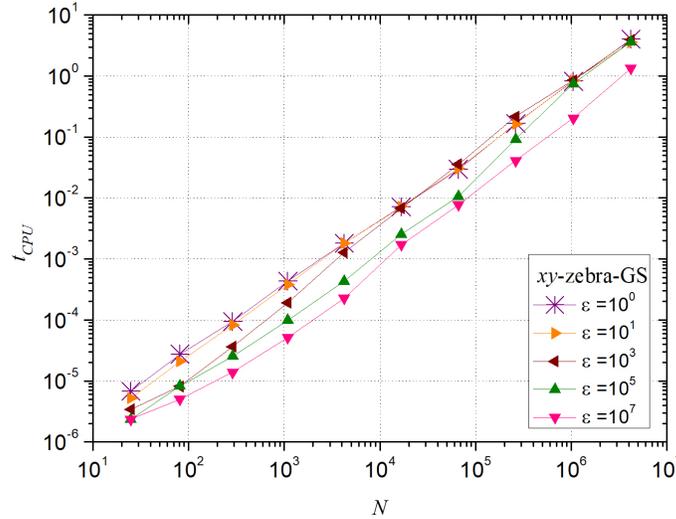


Figure 13: $t_{CPU}(s)$ vs. the number of points (N) for some values of ε

Finally, with the t_{CPU} results obtained using the xy -zebra-GS smoother, a geometrical adjustment of the type $t_{CPU} = cN^p$ was made in order to evaluate the performance of the multigrid, where c is a constant related to the method, p represents the order of complexity of the algorithm, or the gradient of the curve, and N is the size of the problem. Ideally, the multigrid method presents $p = 1$, which means that the t_{CPU} increases proportionally to the number of unknowns N [Wesseling (2004)].

The results of the geometrical adjustment are shown in Tab. 2. Only the values of p and c for the anisotropy coefficients $= 10^a$, $a \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ are depicted. The results for $\varepsilon = 10^{-a}$ are similar to the results for $\varepsilon = 10^a$. These results confirm that the t_{CPU} of the multigrid with the xy -zebra-GS smoother grows roughly linearly with the increase of N , since in every case, p is near one.

7 Conclusion

The aim of this study was to propose an efficient and robust smoother for a two-dimensional diffusion problem with physical anisotropy. In order to reach the goal, it was necessary to perform a study of the convergence factor of the multigrid method for some smoothers by means of LFA. The geometrical multigrid method was employed with Correction Scheme, V-cycle, FW restriction, bilinear interpolation, zero initial guess and $r = 2$. The smoothers assessed were the well-known pointwise smoothers GS-RB and GS-LEX, and the block smoothers x -line-GS, y -line-GS, x -zebra-GS and y -zebra-GS. The xy -zebra-GS smoother was proposed, which is a combination of the x -zebra-GS smoother with the y -zebra-GS smoother. The empirical and asymptotic convergence factor, the number of V-cycles and the computational cost according to the order of complexity of the smoother proposed were assessed. As result of this work, we verified that:

Table 2: Order of complexity p and the coefficient c regarding the coefficient of anisotropy ε

ε	p	c
10^0	1.092	2.183×10^{-7}
10^1	1.055	3.646×10^{-7}
10^2	1.071	3.255×10^{-7}
10^3	1.091	6.017×10^{-7}
10^4	1.076	2.674×10^{-7}
10^5	1.165	6.996×10^{-7}
10^6	1.331	3.685×10^{-7}
10^7	1.151	2.187×10^{-7}

- The convergence factor computed empirically and by means of LFA resulted in very similar values, which confirms the importance and reliability of the LFA;
- Concerning to the smoothers assessed, the convergence factors of the block smoothers are more efficient than those of the point smoothers;
- Among the block smoothers assessed, the zebra-GS presents the best convergence factors;
- Among all the smoothers assessed, the x -zebra-GS and y -zebra-GS have the best convergence factor for $0 < \varepsilon \ll 1$ and $\varepsilon \gg 1$, respectively;
- In relation to the xy -zebra-GS smoother, proposed in this study, the number of V-cycles tends to be constant as N increases;
- The xy -zebra-GS smoother presents good convergence factors and low computational cost regardless of the anisotropy factor, thus being a robust and efficient smoother for the anisotropic problem studied.

Acknowledgement: The authors would like to thank the Graduate Program in Numerical Methods in Engineering (PPGMNE) of the Federal University of Paraná (Brazil). This study was financed partly by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

Briggs, W. L.; Henson, V. E.; McCormick, S. F. (2000): *A Multigrid Tutorial*. Siam, Philadelphia, 2nd edition.

Burden, R. L.; Faires, J. D.; Burden, A. M. (2016): *Numerical Analysis*. Cengage, Boston, 10th edition.

Dedner, A.; Muller, E.; Scheichl, R. (2014): Efficient multigrid preconditioners for anisotropic problems in geophysical modelling. *Numerical Analysis*.

Ferziger, J. H.; Peric, M. (2002): *Computational Methods for Fluid Dynamics*. Springer, Berlin, 3rd edition.

Franco, S. R.; Gaspar, F. J.; Pinto, M. A. V.; Rodrigo, C. (2018): Multigrid method based on a space-time approach with standard coarsening for parabolic problems. *Applied Mathematics and Computation*, vol. 317, pp. 25-34.

Franco, S. R.; Rodrigo, C.; Gaspar, F. J.; Pinto, M. A. V. (2018): A multigrid waveform relaxation method for solving the poroelasticity equations. *Computational and Applied Mathematics*, pp. 1-16.

Gee, W. M.; Hu, J.; Tummara, S. R. (2009): A new smoothed aggregation multigrid for anisotropic problems. *Numerical Linear Algebra with Applications*, vol. 16, no. 1, pp. 19-37.

Gmeiner, B.; Gradl, T.; Gaspar, F.; Rde, U. (2013): Optimization of the multigrid-convergence rate on semi-structured meshes by local fourier analysis. *Computers & Mathematics with Applications*, vol. 65, no. 4, pp. 694-711.

Hackbusch, W. (2003): *Multi-grid Methods and Applications*, volume 4. Springer Science & Business Media, Berlin.

Lew, S.; Wolters, C.; Dierkes, T.; Roer, C.; MacLeod, R. (2009): Accuracy and run-time comparison for different potential approaches and iterative solvers in finite element method based eeg source analysis. *Applied Numerical Mathematics*, vol. 59, no. 8, pp. 1970-1988.

Marchi, C. H.; Silva, A. F. C. (2002): Unidimensional numerical solution error estimation for convergent apparent order. *Numerical Heat Transfer: Part B: Fundamentals*, vol. 42, no. 2, pp. 167-188.

Oliveira, F.; Pinto, M. A. V.; Marchi, C. H. (2012): Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids. *Applied Mathematical Modelling*, vol. 36, no. 10, pp. 4665-4676.

Peherstorfer, B.; Bungartz, H. J. (2012): Semi-coarsening in space and time for the hierarchical transformation multigrid method. *Procedia Computer Science*, vol. 9, pp. 2000-2003.

Pinto, M. A. V.; Rodrigo, C.; Gaspar, F. J.; Oosterlee, C. W. (2016): On the robustness of ilu smoothers on triangular grids. *Applied Numerical Mathematics*, vol. 106, pp. 37-52.

Trottenberg, U.; Oosterlee, C. W.; Schuller, A. (2001): *Multigrid*. Academic Press, San Diego.

Vassoler-Rutz, G.; Pinto, M. A. V. (2016): Local fourier analysis and optimization of the parameters of the multigrid method for the anisotropic diffusion equation. *XXXVII Iberian Latin American Congress on the Computational Methods in Engineering*.

Wesseling, P. (2004): *An Introduction to Multigrid Methods*. R. T. Edwards, Philadelphia.

Wesseling, P.; Oosterlee, C. W. (2001): Geometric multigrid with applications to computational fluid dynamics. *Journal of Computation and Applied Mathematics*, vol. 128, pp. 311-334.

Wienands, R.; Joppich, W. (2005): *Practical Fourier Analysis for Multigrid Methods*. Chapman and Hall/CRC Press, Boca Raton, FL.